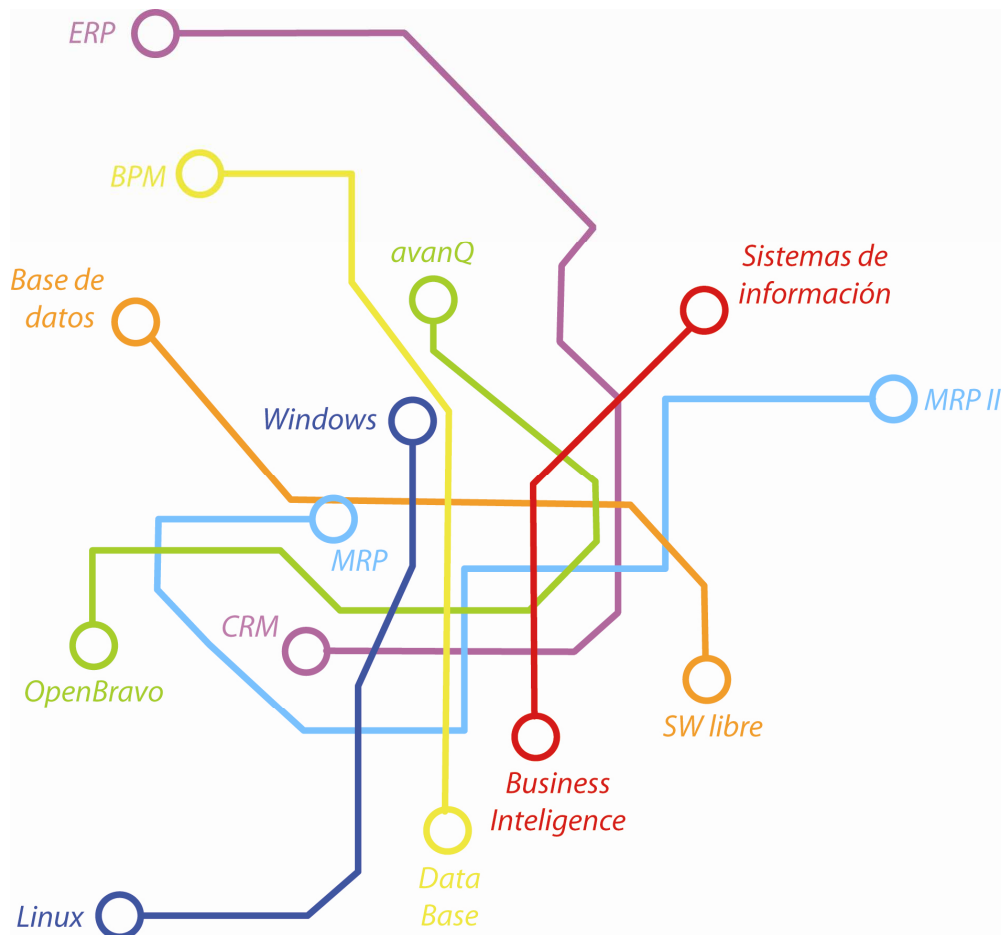


Proyecto Final de carrera

II-A-DOEEFC- 74

Estudio comparativo de paquetes

ERP en el ámbito del SW libre



Índice

ÍNDICE

1.- Objetivos del proyecto	pág. 6
2.- Introducción	pág. 8
3.- Software Libre	pág. 11
3.1.- ¿Qué es el Software Libre?	pág. 13
3.2.- Ventajas del Software Libre	pág. 14
3.3.- Historia	pág. 17
3.3.1.- Breve historia de Linux	pág. 18
3.3.2.- La aparición de las distribuciones de GNU/Linux	pág. 18
3.4.- Licencias	pag. 19
3.4.1. - Licencias de Software Libre	pág. 20
3.4.2. - Open Source	pág. 22
4.- Sistemas de Información	pág. 23
4.1.- Introducción	pág. 25
4.2.- La transición hacia la sociedad de la información	pág. 27
4.3.- Características	pág. 28
4.4.- Definición	pág. 30
4.5.- Estructura de un S.I	pág. 34
4.6.- Integración frente a independencia	pág. 36
4.7.- Funciones básicas de tratamiento de la información	pág. 37
4.8.- Actividades de un sistema de información	pág. 40
4.9.- Tipos y Usos de los Sistemas de Información	pág. 42
4.10.- Construcción de un sistema de información	pág. 48
4.11.- Evolución de los sistemas de información	pág. 51
5.- Enterprise Resources Planning (ERP)	pág. 55
5.1.- Introducción	pág. 58
5.2.- Definición	pág. 58
5.3.- Estructura de un ERP	pág. 60
5.3.1.- El sistema básico de un ERP	pág. 60
5.3.2.- Módulos de Aprovisionamiento	pág. 62
5.3.3.- Módulo de producción	pág. 63
5.3.4.- Módulos de ventas	pág. 63
5.3.5.- Módulo de finanzas	pág. 64
5.3.6.- Módulo de recursos humanos	pág. 64
5.3.7.- Módulo de gestión de medios técnicos y mantenimiento	pág. 65
5.4.- Características generales de un ERP	pág. 65
5.4.1.- Capacidad de parametrización	pág. 65
5.4.2.- Adaptación a la estructura de la empresa	pág. 65
5.4.3.- Interfaz de usuario avanzada y flexible	pág. 66
5.4.4.- Integración con otras aplicaciones	pág. 66
5.4.5.- Capacidad de acceso a información	pág. 66
5.4.6.- Otras características	pág. 66
5.5.- El ERP como cadena de valor extendida	pág. 66
5.6.- Evolución histórica de los sistemas ERP	pág. 69
5.6.1.- Antecedentes del software de gestión	pág. 69
5.6.2.- Primera etapa: la gestión informatizada de las listas de materiales	pág. 70
5.6.3.- La gestión de necesidades de material: el MRP	pág. 71
5.6.4.- El MRP a ciclo cerrado: la gestión de cargas y capacidades	pág. 73
5.6.5.- El MRP II: la gestión de recursos de fabricación	pág. 74
5.6.6.- ERP: planificación de recursos de empresa	pág. 75
5.6.7.- SCM: la gestión de la cadena de suministros	pág. 76
5.6.8.- Los retos actuales: CRM Y PLM	pág. 77
5.7.- Situación de los ERPs en la empresa	pág. 78
5.8.- Proceso de implantación de un ERP	pág. 80
5.8.1.- Introducción	pág. 80
5.8.2.- Aspectos a considerar	pág. 80

5.8.2.1.- Expectativas generadas ante la implantación de un ERP	pág. 80
5.8.2.2.- Costes asociados a la implantación de un ERP	pág. 81
5.8.2.3.- Componentes de una implantación de un ERP	pág. 82
5.8.2.4.- Fases de una implantación	pág. 84
5.8.2.5.- Ventajas e Inconvenientes de su implantación	pág. 86
5.8.3.- Enfoque metodológico	pág. 87
5.8.3.1.- Análisis de la situación actual	pág. 88
5.8.3.2.- Análisis de Requisitos	pág. 88
5.8.3.3.- Identificación Alternativas	pág. 88
5.8.3.4.- Selección Alternativa	pág. 89
5.8.3.5.- Planificación Implantación	pág. 89
5.8.4.- Elección del ERP: Parámetros	pág. 89
6.- CRM (Customer Relationship Management)	pág. 92
6.1.- Introducción	pág. 94
6.2.- Antecedentes de la Gestión de las Relaciones con el Cliente (CRM)	pág. 94
6.3.- Delimitación del concepto de CRM	pág. 98
6.4.- Principales factores para la implantación exitosa de CMR en las organizaciones	pág. 106
6.5.- Principales problemas y barreras para la implantación exitosa de CRM	pág. 110
6.6.- Modelo de Implantación de CRM en la organización	pág. 113
6.7.- Conclusión	pág. 118
7.- Criterios comparativos de paquetes ERP	pág. 119
7.1.- Introducción	pág. 121
7.2.- Cobertura funcional	pág. 122
7.3.- Flexibilidad	pág. 123
7.3.1.- Personalización	pág. 123
7.3.2.- Actualizaciones flexibles	pág. 124
7.3.3.- Internacionalización	pág. 124
7.3.4.- Facilidad de uso	pág. 125
7.3.5.- Arquitectura	pág. 125
7.3.6.- Escalabilidad	pág. 127
7.3.7.- Seguridad	pág. 127
7.3.8.- Interfaces	pág. 127
7.3.9.- Independencia del sistema operativo	pág. 128
7.3.10.- Independencia del sistema de bases de datos	pág. 128
7.3.11.- Lenguaje de programación	pág. 128
7.4.- Soporte	pág. 129
7.4.1.- Infraestructura de Soporte	pág. 129
7.4.2.- Formación	pág. 129
7.4.3.- Documentación	pág. 129
7.5.- Continuidad	pág. 129
7.5.1.- Estructura del proyecto	pág. 131
7.5.2.- Actividad de la comunidad	pág. 131
7.5.3.- Transparencia	pág. 131
7.5.4.- Frecuencia de las actualizaciones	pág. 132
7.5.5.- Otros efectos acordados	pág. 132
7.6.- Madurez	pág. 133
7.6.1.- Estado del desarrollo	pág. 133
7.6.2.- Lugar de referencia	pág. 133
8.- Características ERPs analizados en profundidad	pág. 134
8.1.- Introducción	pág. 137
8.2.- Tablas comparativas	pág. 137
8.3.- Características ERPs analizados	pág. 142
8.3.1.- Abanq	pág. 142
8.3.1.1.- Flexibilidad	pág. 142
8.3.1.2.- Soporte	pág. 146
8.3.1.3.- Continuidad	pág. 147

8.3.1.4.- Madurez	pág. 148
8.3.2.- Adempiere	pág. 149
8.3.2.1.- Flexibilidad	pág. 150
8.3.2.2.- Soporte	pág. 153
8.3.2.3.- Continuidad	pág. 154
8.3.2.4.- Madurez	pág. 155
8.3.3.- Compiere	pág. 156
8.3.3.1.- Flexibilidad	pág. 156
8.3.3.2.- Soporte	pág. 159
8.3.3.3.- Continuidad	pág. 160
8.3.3.4.- Madurez	pág. 161
8.3.4.- Oasis erp	pág. 163
8.3.4.1.- Flexibilidad	pág. 164
8.3.4.2.- Soporte	pág. 165
8.3.4.3.- Continuidad	pág. 166
8.3.4.4.- Madurez	pág. 167
8.3.5.- Openbravo	pág. 168
8.3.5.1.- Flexibilidad	pág. 168
8.3.5.2.- Soporte	pág. 172
8.3.5.3.- Continuidad	pág. 173
8.3.5.4.- Madurez	pág. 174
8.3.6.- Openerp	pág. 175
8.3.6.1.- Flexibilidad	pág. 175
8.3.6.2.- Soporte	pág. 179
8.3.6.3.- Continuidad	pág. 180
8.3.6.4.- Madurez	pág. 181
8.3.7.- OpenXpertya	pág. 183
8.3.7.1.- Flexibilidad	pág. 183
8.3.7.2.- Soporte	pág. 187
8.3.7.3.- Continuidad	pág. 188
8.3.7.4.- Madurez	pág. 189
8.4.- Conclusiones a partir de los puntos anteriores	pág. 190
8.4.1.- Flexibilidad	pág. 190
8.4.2.- Soporte	pág. 191
8.4.3.- Continuidad	pág. 192
8.4.4.- Madurez	pág. 193
8.4.5.- Conclusión	pág. 193
8.5.- Instalación Openbravo	pág. 194
8.5.1.- Requisitos de intalación	pág. 194
8.5.2.- Entorno	pág. 195
8.5.2.1.- Componentes	pág. 195
8.5.2.2.- Módulos	pág. 196
8.6.- Instalación Abanq	pág. 197
8.6.1.- Requisitos de intalación	pág. 197
8.6.2.- Entorno	pág. 198
8.6.2.1.- Módulos	pág. 198
8.6.2.2.- Área de facturación	pág. 198
8.6.2.3.- Área financiera	pág. 199
8.7.- Conclusiones instalación	pág. 200
9.- Business Process Management (BPM)	pág. 202
9.1.- Introducción	pág. 204
9.2.- Definición	pág. 204
9.3.- El BPM clave en las organizaciones	pág. 206
9.4.- Alcance del BPM	pág. 207
9.5.- Arquitectura Empresarial – Modelos de Negocio	pág. 208
9.6.- Automatización y orquestación de procesos, organización y sistemas	pág. 210
9.7.- Beneficios	pág. 212
9.8.- Monitorización de procesos y recursos empresariales	pág. 213

9.9.- Diez prácticas recomendadas de BPM	pág. 213
9.10.- Los diez escollos a evitar en BPM	pág. 215
9.11.- Conclusiones	pág. 216
10.- Business Intelligence	pág. 217
10.1.- Introducción	pág. 219
10.2.- Definición	pág. 219
10.3.- Componentes de Business Intelligence	pág. 221
10.3.1.- Fuentes de información	pág. 222
10.3.2.- Calidad de los datos	pág. 224
10.3.3.- Proceso de extracción, transformación y carga (ETL)	pág. 226
10.3.4.- Herramientas ETL	pág. 231
10.3.5.- <i>Datawarehouse</i> o almacén de datos	pág. 232
10.3.6.- Gestión del <i>datawarehouse</i>	pág. 240
10.3.7.- Herramientas de <i>Business Intelligence</i>	pág. 240
10.3.8.- Visualización	pág. 244
10.3.9.- ¿Quiénes son los usuarios de las herramientas de <i>Business Intelligence</i> ?	pág. 244
10.4.- Necesidad y fases de planificación de un proyecto BI	pág. 245
10.5.- Elementos clave para el éxito o fracaso de un proyecto BI	pág. 252
10.6.- Usos de Business Intelligence y nuevas tendencias	pág. 252
10.7.- Conclusiones	pág. 256
11.- Conclusion	pág. 258
12.- Anexo paquetes ERP	pág. 262
13.- Bibliografía	pág. 413

Objetivos del proyecto

1.- Objetivos del Proyecto

- La introducción al lector en el término de software libre, sus ventajas, desventajas y características, así como los tipos de licencias actuales, teniendo en cuenta en cada punto a su competidor, el software propietario.
- Introducir el concepto de información, así como los sistemas de información, sus características relevantes y su evolución, para el correcto uso de los mismos.
- Introducir el concepto de planificación de recursos empresariales ERP, su estructura, sus características fundamentales, así como su evolución y situación actual de manera que el lector se sitúe en el contexto adecuado para entender la base del proyecto.
- Mostrar el concepto de CRM o gestión de las relaciones con los clientes como herramienta imprescindible en la actualidad para las empresas, explicando su evolución, modelo, características y los factores adecuados para una implantación exitosa.
- Mostrar los criterios comparativos en profundidad que nos servirán para analizar los ERPs posteriormente.
- Estudiar a través de los criterios comparativos ya citados, una serie de ERPs de software libre de manera que el lector logre tener una perspectiva objetiva a la hora de elegir su herramienta ERP.
- Introducir el termino BPM, como herramienta de mejora de procesos de negocio, su arquitectura así como factores clave para su correcta implantación.
- Conocer las herramientas de análisis de información de Business Intelligence, sus características y su composición, así como las fases y elementos clave para su correcta implantación.
- Finalmente se anexarán distintos manuales proporcionados por las herramientas analizadas de manera que el lector pueda ampliar sus conocimientos sobre los mismos.

Introducción

2.- Introducción

En la actualidad las herramientas de software libre han adquirido una importancia relevante en las empresas, llegando a competir en muchos sectores con las herramientas propietarias, su uso depende en gran parte del conocimiento de las mismas.

Los términos más coste mejor calidad, han quedado obsoletos, la existencia de herramientas de software libre ayuda a que empresas con menor presupuesto logren una adecuación de sus sistemas a sus competidores.

La necesidad que presentan las empresas de tener que trabajar con determinado software para realizar los diferentes tipos de trabajos y procesos en sus departamentos, ha dado lugar a que surja en el mercado una serie de programas llamados ERP (Enterprise Resource Planning o Sistema de Planificación de Recursos Empresariales). Lejos de crear para la empresa una solución integral, completa y ajustada en todos los aspectos, este tipo de programas requieren recursos para su adquisición e implantación.

La utilización estratégica de la información en las empresas es una necesidad cada vez mayor, por lo que contar con mecanismos que permitan orientar la información hacia el control y la consecución de los objetivos fijados se hace cada vez más indispensable. Por ello debe tenerse en cuenta que la elección de sistemas ERP puede ser un elemento importante en la búsqueda de ventajas competitivas y en la supervivencia a corto y a medio plazo. El sistema ERP tiene como objetivo principal satisfacer las diferentes necesidades de información de la empresa, tanto interna como externa, para lograr una mejor eficiencia en la gestión de la misma, con datos más precisos, más rápidos de obtener, fiables y de fácil comprensión, permitiendo a los directivos y analistas de información tomar decisiones para realizar las acciones pertinentes y definir las estrategias a implementar en el futuro.

Los ERP ofrecen muchas ventajas y ahorros significativos a las empresas, pero es muy importante poder medir correctamente estos ahorros ya que, en la mayoría de los casos, la implantación de este tipo de software tiene un coste muy elevado y si no se tiene un plan bien definido de los objetivos a conseguir ni de cuáles son los recursos de que dispone la empresa para implantarlo, las consecuencias posteriores son dramáticas. Es muy importante que la empresa se adapte a los cambios y avances tecnológicos lo antes posibles para ser competitiva y lograr un crecimiento constante, con lo que la buena elección de un ERP es indispensable y fundamental en este avance.

Las herramientas ERP de software libre son soluciones empresariales de menor coste que alcanzan objetivos similares a las herramientas ERP de

software propietario, la buena elección de la misma podrá ser un arma de diferenciación entre empresas del mismo nivel.

La evolución de los sistemas de información así como su tecnología, provee a las empresas de un mayor número de herramientas para el tratamiento de la información.

Vivir en una sociedad en la que la información es poder, poseerla de calidad y en la mayor brevedad posible, aunque sea en minúsculas fracciones de tiempo antes que tu competidor, puede llevar a la empresa que la posea a adquirir una ventaja relevantes e inimaginables para el ciudadano de a pie. Pero poseer la información adecuada en la actualidad carece de sentido sino se posee herramientas adecuadas para su análisis y posterior toma de decisiones. En los últimos años han surgido conceptos como Customer Relationship Management (CRM), Business Process Management (BPM) o Business Intelligence (BI), desconocidos para muchos que no son más que la reafirmación de la importancia de la información en la sociedad actual, la búsqueda de la optimización de procesos, las relaciones óptimas con los clientes, así como el análisis de la información importante y perfectamente estructurada dotan de mayor competitividad al mercado ofreciendo soluciones para mejorar día a día.

Finalmente, la elección de una adecuada herramienta de gestión global que cumpla todas las necesidades de una empresa con un coste mínimo es la base de este proyecto, saber elegir, qué queremos ya no es suficiente, también debemos saber, para qué lo queremos y cuanto tiempo nos será útil.

Software libre

3.- Software Libre

3.1. - ¿Qué es el Software Libre?

3.2. - Ventajas del Software Libre

3.3. - Historia

3.3.1. - Breve historia de Linux

3.3.2. - La aparición de las distribuciones de GNU/Linux

3.4. - Licencias

3.4.1. - Licencias de Software Libre

3.4.2. - Open Source

3.1. - ¿Qué es el Software Libre?

Software Libre (en inglés *free software*) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El Software Libre puede ser configurado, mejorado y utilizado sin tener que pagar derechos de autor por ello. Esto significa que por el código no debemos pagar, aunque si podríamos hacerlo por la contratación de servicios derivados, como por ejemplo instalación, configuración, soporte, auditoría, formación o cambios sobre la aplicación original.

El Software Libre por tanto es una cuestión de libertad y no de precio. Para que un software sea considerado libre es necesario que cumpla los cuatros principios definidos por Richard M. Stallman en su libro: “Software libre para una sociedad libre” (disponible de forma gratuita en castellano y en inglés):

1. La libertad de ejecutar el programa sea cual sea el propósito.
2. La libertad para modificar el programa para ajustarlo a tus necesidades. (Para que se trate de una libertad efectiva en la práctica, deberás tener acceso al código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil.)
3. La libertad de redistribuir copias, ya sea de forma gratuita, ya sea a cambio del pago de un precio.
4. La libertad de distribuir versiones modificadas del programa, de tal forma que la comunidad pueda aprovechar las mejora introducidas.

Dado que nos referimos a la libertad y no al precio, no existe contradicción alguna entre la venta de copias y el software libre. De hecho, la libertad para vender copias es crucial: las colecciones de software libre a la venta en formato de CD-ROM son muy importantes para la comunidad y venderlas es una forma de recaudar fondos para el desarrollo de software libre. Por lo tanto, cualquier programa que no podamos incluir en estas colecciones no podrá calificarse de software libre.

Dada la ambigüedad del calificativo «libre», llevamos mucho tiempo buscando alternativas, pero nadie ha encontrado ninguna satisfactoria. La lengua inglesa es de las más rica en lo que a palabras y matices se refiere, pero carece de un término simple e inequívoco para “libre” en el sentido de libertad (“unfettered” [sin cadenas]) sería el calificativo que más se ajusta al significado. Alternativas como “liberado”, “libertad” o “abierto” no significan lo mismo o presentan otros inconvenientes.

Actualmente hay disponibles miles de programas de Software Libre que pueden cubrir las necesidades de casi cualquier empresa, y en general con la misma calidad o superior que el software no libre, también llamado software propietario o privativo. Pero debido a ciertas barreras su uso no es tan extendido como debiese.

3.2. -Ventajas del Software Libre

El uso de Software Libre proporciona ventajas significativas para el tejido empresarial de cualquier región, que el software propietario no puede ofrecer:

- **Es más económico:** El bajo o nulo coste de los productos libres permiten proporcionar a las PYMES servicios y ampliar sus infraestructuras sin que se vean mermados sus intentos de crecimiento por no poder hacer frente al pago de grandes cantidades en licencias. Mediante el uso de Software Libre, las empresas, en su mayoría PYMES que disponen de escaso recursos humanos y con poca inversión en I+D, podrán beneficiarse de aplicaciones de alta calidad a bajo coste, disponiendo de las mismas facilidades que las grandes empresas del sector y aumentando de esta forma su posición competitiva.
- **Software adaptado:** El acceso al código fuente del programa proporciona la posibilidad de ajustar una aplicación a las necesidades concretas de cualquier persona, colectivo o empresa. Por ejemplo, para labores de localización del software, traduciéndolo a cualquier idioma, adaptándolo al modelo de negocio de la empresa o añadiendo funcionalidad extra no contemplada en la aplicación original.
- **Cultura de colaboración y modelo científico:** La cultura de colaboración sigue el modelo científico de desarrollo y puede generar resultados brillantes. El desarrollo de Software Libre se basa en un trabajo cooperativo entre personas comunicadas por Internet que deciden poner sus conocimientos a disposición del público. Este modelo es similar al modelo científico-tradicional, en el que la innovación y el conocimiento pertenecen a la humanidad, no a la empresa.
- **Independencia del proveedor:** El Software Libre al no depender de un proveedor único permite que cualquier empresa pueda proporcionar

servicios de soporte sobre una aplicación, de esta manera si un proveedor desaparece, siempre se podrá continuar mejorando dicho programa.

- **Fomento de la industria local:** Este es uno de los grandes beneficios del Software Libre, ya que las empresas TIC locales pueden ampliar su modelo de negocio con productos libres, sin depender de proveedores foráneos. La mayor parte del software propietario que se utiliza en España procede de empresas foráneas, con lo que el dinero invertido en software favorece a otros países. Sin embargo, al utilizar Software Libre es posible recurrir a empresas locales para obtener servicios sobre un programa concreto. Fomentando de esta manera la industria local y el empleo.
- **Mejores prestaciones con el mismo hardware:** Por lo general los requisitos de procesamiento y memoria del Software Libre son menores que en las aplicaciones propietarias y optimizan los recursos del ordenador. Esto permite no tener que renovar el parque informático de una empresa cada pocos años o recuperar equipos informáticos obsoletos ya retirados para realizar algunas acciones determinadas.
- **Libertad de uso y redistribución:** Las licencias de Software Libre existentes permiten la instalación del software tantas veces y en tantas máquinas como el usuario desee sin tener que pagar nada por ello.
- **Aumento de la productividad:** El acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero. El secretismo tecnológico es uno de los grandes frenos y desequilibrios existentes para el desarrollo en el modelo de propiedad intelectual.
- **Soporte y compatibilidad a largo plazo:** Este punto, más que una ventaja del Software Libre es una desventaja del software propietario, y la elección de Software Libre evita este problema. Al vendedor, una vez alcanzado el máximo número de ventas que puede realizar de un producto, no le interesa que sus clientes continúen con él y optan por sacar un nuevo producto. Y para obligar al usuario a que deje de utilizar la versión anterior acaban por no dar soporte ni solucionar fallos que puedan surgir, y en ciertos casos por producir formatos de ficheros incompatibles entre versiones diferentes del mismo programa. Véase diferentes versiones de Windows que dejan de ser

soportadas por Microsoft o software de grabación que no admite nuevos modelos de grabadoras ópticas sin una actualización, aún cuando la grabadora nueva emplee el mismo mecanismo de grabación que la antigua.

- **Formatos estándar:** Los formatos estándar permiten una interoperatividad más alta entre sistemas, evitando incompatibilidades. Los estándares *de facto* son válidos en ocasiones para lograr una alta interoperatividad si se omite el hecho que estos exigen el pago de royalties a terceros y que por razones de mercado no interesa que se perpetúen demasiado tiempo.
- **Mayor estabilidad y seguridad:** Los sistemas GNU/Linux cuentan con una mayor estabilidad de trabajo, no siendo necesario reiniciar el computador con frecuencia debido a la pérdida de rendimiento. Pueden funcionar de forma continuada un gran número de horas. Así mismo, la seguridad en sistemas operativos GNU/Linux es mucho más alta que en otro tipo de sistemas, desde el control de usuarios y la ejecución de aplicaciones hasta los problemas inexistentes de virus. Estas características son las que hacen que el Software Libre esté presente en la mayoría de servidores de Internet y de las grandes empresas. El acceso al código fuente permite además que tanto *hackers* como empresas de seguridad de todo el mundo puedan auditar los programas, por lo que la existencia de puertas traseras es ilógica ya que pondría en evidencia el producto y la comunidad que lo genera.
- **Corrección más rápida y eficiente de fallos:** El funcionamiento e interés conjunto de la comunidad ha demostrado solucionar más rápidamente los fallos de seguridad en el Software Libre, algo que en el software propietario es más difícil y costoso. En ocasiones cuando se notifica a las empresas propietarias del software algún problema en su software, éstas niegan inicialmente la existencia de dichos fallos por cuestiones de imagen y cuando finalmente admiten la existencia de esos Bugs, tardan semanas o meses hasta proporcionar los parches de seguridad.
- **Métodos simples y unificados de gestión de software:** Actualmente la mayoría de distribuciones de Linux incorporan algún sistema que unifican el método de instalación de programas, librerías, etc. Esto simplifica hasta el grado de marcar o desmarcar una casilla la gestión del software, y permiten el acceso a miles de aplicaciones de forma segura y gratuita. Este sistema de acceso y gestión del software se hace prácticamente utópico si se extrapola al mercado propietario.

- Sistema en expansión: Las ventajas especialmente económicas que el Software Libre aporta a muchas empresas y las aportaciones de la comunidad han permitido un constante crecimiento del Software Libre, hasta superar en ocasiones, como en el del Software para Internet, al mercado propietario. El Software Libre ya no es una promesa, es una realidad y se utiliza en sistemas de producción de algunas de las empresas tecnológicas más importantes como Telefónica, IBM, SUN Microsystems, Google, Sony, Hewlett-Packard, Oracle o incluso la NASA. Podemos augurar sin lugar a dudas un futuro crecimiento de su empleo y una consolidación bien merecida.

3.3. - Historia

Durante los años 60 y 70 era muy habitual que los programadores y desarrolladores compartieran entre sí sus programas sin ninguna restricción, hasta que a finales de los años 70 comenzaron a surgir los acuerdos de licencia, pero no fue hasta la década de los 80 cuando aparecieron los primeros sistemas operativos privativos que forzaban a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones del software.

Fue el 27 de septiembre de 1983 cuando Richard Stallman anunció públicamente el proyecto GNU (GNU's Not Unix) con el objetivo de crear un sistema operativo completamente libre compatible con Unix: el sistema GNU. Al anuncio original, siguieron otros ensayos como el "Manifiesto GNU", donde refleja las motivaciones para iniciar el proyecto, entre las que destaca "volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras".

En 1985 surgió la Free Software Foundation (FSF), fundada de nuevo por Richard Stallman, con el propósito de difundir el movimiento del "Software Libre". Fue entonces cuando Stallman definió el concepto de "Free Software" o "Software Libre" y el concepto de "copyleft", que restringe la apropiación del software y otorga la libertad a los usuarios.

GNU se encaminó principalmente al desarrollo de un sistema operativo gratuito, compatible con UNIX, que pudiese modificarse según las necesidades de cada usuario. Después de algunos años se disponía de lo básico para un sistema operativo: intérprete de lenguajes y editor de texto, herramientas para el trabajo en red y un compilador; aunque aún faltaba el Kernel o Núcleo para hacer funcionar todo el sistema.

El Kernel del sistema operativo GNU, surgió en 1990 cuando el universitario finlandés Linus Torvalds decidió ampliar el sistema operativo Minix, al que llamó Linux, desarrollado por el profesor Andrew S. Tanenbaum con fines educativos. Gracias al desarrollo de Linux, Stallman y

sus colaboradores encontraron la solución que necesitaban para GNU, el Kernel, a partir de aquí nace GNU/Linux que es la unión de GNU y de Linux.

Actualmente los sistemas GNU/Linux son una solución real utilizada por multitud de empresas, administraciones y usuarios de todo el mundo. GNU/Linux ofrece un sistema estable, potente y seguro junto a una gran cantidad de Software Libre que crece y se mejora día a día por millones de personas.

3.3.1. Breve historia de Linux

La historia de Linux empieza en Finlandia (1991), cuando el estudiante de la Universidad de Helsinki, Linus B. Torvalds, se planteó aprovechar mejor los recursos de su ordenador (un PC con procesador Intel 386) y se instaló en él una versión reducida del sistema operativo Unix (<http://www.unix-systems.org>) llamada Minix.

Sin embargo, debido a las limitaciones del Minix, Linus decidió reescribir algunas partes del sistema, añadiéndole mayor funcionalidad. Posteriormente, decidió difundir el código fuente por Internet, de manera gratuita y con el nombre de Linux (contracción de Linus y Unix). El anuncio inicial de Linux tuvo lugar en agosto de 1991. Era la versión 0.01.

La primera versión "oficial", la 0.02, se hizo pública el 5 de octubre de 1991, para la que se incorporaron algunos programas GNU <http://www.gnu.org/home.es.html> como el shell bash o el compilador GCC.

La versión estable de Linux fue la 1.0 y apareció en marzo de 1994. Gracias al uso de Internet, Linux ha tenido un crecimiento espectacular en los últimos tiempos, siendo un proyecto con cada vez más colaboradores que mejoran día a día el sistema.

Hay que hacer hincapié también en que el término Linux se refiere al núcleo del sistema (parte que interactúa con el hardware de la máquina). Cuando se habla de todo el conjunto que forma el núcleo, y todos los demás proyectos GNU (shells, compiladores, escritorios y las distintas aplicaciones en general), se debe hablar ya del sistema operativo GNU/Linux.

A lo largo de la historia de GNU/Linux han surgido muchas variantes suyas, conocidas como distribuciones. Una distribución GNU/Linux es una variante de ese sistema operativo que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, educativas o empresariales.

3.3.2. La aparición de las distribuciones de GNU/Linux

En 1992 apareció la primera distribución Linux, conocida como MCC Interim. A mediados de 1992 la distribución de Linux más popular era SLS Linux (Softlanding Linus System). Slackware apareció en 1993 como resultado de los cambios y limpieza que realizó Patrick Volkering a la distribución Linux SLS. A partir de Slackware.

Otra distribución que se basó en Slackware es la conocida distribución SUSE Linux, en 2004 esta distribución fue comprada por la multinacional americana Novell, después en 2005 fue liberada para que fuera la comunidad la que desarrollara esta distribución, que pasó llamarse openSUSE.

En el año 1993 Ian Murdik fundó el proyecto Debian junto al manifiesto base para la creación de la distribución Debian. Esta es una de las comunidades de Software Libre más prestigiosas y reconocidas del mundo. A partir de Debian han surgido muchas otras distribuciones como son Corel, Skolelinux o Knoppix. También ha sido la distribución elegida por Mark Shuttleworth y su empresa Canonical Ltd. Ubuntu, que nació en 2004 con el objetivo de acercar a todos los usuarios los sistemas GNU/Linux. Actualmente es una de las distribuciones más populares por su facilidad de uso y sus actualizaciones continuas.

Debido al éxito alcanzando por Ubuntu han surgido multitud de distribuciones derivadas de ésta como es el caso de Molinux (<http://molinux.info>), que es la distribución desarrollada por la Junta de Comunidades de Castilla-La Mancha para acercar las Tecnologías de la Información a la sociedad castellano-manchega. Entre las ventajas de esta distribución regional se encuentran: software completamente en español, versiones actualizadas semestralmente y soporte gratuito a través de teléfono, correo electrónico o foros web.

Podemos encontrar el árbol genealógico de GNU/Linux en: http://www.linux-es.org/files/distribuciones_en_el_tiempo.png

3.4. - Licencias

Como ya se ha comentado, fue en la década de los 80 cuando comenzó a aparecer software sujeto a licencias que limitaba las libertades de los usuarios.

Una licencia es, desde el punto de vista del Derecho, un contrato mediante el cual una persona recibe de otra el derecho de uso de varios de sus bienes, normalmente de carácter no tangible o intelectual, a cambio del pago de una cantidad determinada por el uso de los mismos.

Al adquirir una licencia software, ya sea pagando o gratuitamente, podemos encontrar dos roles principales que median la transacción como se muestra a continuación (Figura 3.1).

Proveedor	
¿Qué es?	Persona física o jurídica que concede algún derecho (como podría ser el derecho de uso), a un usuario o empresa. En el caso del Software Libre además se conceden los derechos de modificación y redistribución del software.
¿Quién puede ser?	<ul style="list-style-type: none"> ● Autor o autores del software. ● Persona física o jurídica que tenga derechos de explotación o distribución.
Usuario	
¿Qué es?	Es la persona física o jurídica que adquiere algún derecho o derechos sobre un software por medio de la licencia (pagando o no por ella), según los términos y condiciones que se establecen en la misma (casi siempre impuestos por el proveedor del software).
¿Qué obligaciones tiene?	<ul style="list-style-type: none"> ● Pagar el precio de la licencia (cuando sea de pago) ● Respetar las limitaciones de uso que le impone la licencia de software, software cuya propiedad no le pertenece.

Figura 3.1: Roles de la adquisición de una licencia software. Fuente: Guía Molinux para Pymes.

Sin embargo hay importantes diferencias en cuanto a los derechos y limitaciones que obtenemos a la hora de adquirir una licencia software libre o propietario.

Cuando el usuario adquiere una licencia de software propietario, aparte de abonar un precio por ella, verá que sus derechos como usuario están bastante restringidos:

- Ejecutar el programa.
- Aprovechar sus aplicaciones.
- Hacer una copia de seguridad del mismo.

Pero, si se adquiere una licencia de software libre, las libertades del usuario son mucho más amplias, pudiendo:

- Usar el software libremente sin ningún tipo de restricción.
- Estudiar como funciona y modificarlo según tus necesidades.
- Redistribuirlo con o sin modificaciones, ya sea de manera gratuita o cobrando.

3.4.1. Licencias de Software Libre

Una licencia es aquella autorización formal con carácter contractual que el autor de un producto da a los usuarios de ese bien. Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciataria. Pero para que una licencia pueda ser considerada de software libre ha de cumplir una serie de condiciones que vienen dadas en la definición de software libre por la Fundación de Software Libre, en inglés Free Software Foundation (FSF), y que son:

- Libertad para usar el programa con cualquier propósito
- Libertad para estudiar cómo funciona el programa y para modificarlo
- Libertad para mejorar el programa
- Libertad para redistribuir tanto copias del programa como las propias modificaciones

Las libertades del software están garantizadas por una serie de condiciones que se plasman en una licencia. En el siguiente enlace (<http://www.fsf.org/>) se puede encontrar un listado con algunas de la licencias de software más conocidas y reconocidas por la FSF y el proyecto GNU. También puede consultarse un listado de licencias reconocidas por la Open Source Initiative (OSI) en su página Web, que salvo excepciones en ambos movimientos coinciden.

Una de las características del software libre es la libertad para hacer obras derivadas por parte de terceros, siendo éstas legalmente obras nuevas. Las licencias de software libre se pueden clasificar en dos grandes grupos según la licencia con la que se pueda redistribuir las obras derivadas:

- Por un lado están las licencias robustas o también conocidas como licencias con copyleft que obligan a que las obras derivadas mantenga los términos de la licencia original. Ejemplo de esta licencia es la Licencia Publica General, GNU GPL en la que el autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecerán siempre libres. Esto hace que no sea imposible crear un producto con partes no licenciadas bajo la GPL u otra licencia compatible.
- En el otro lado se encuentran las licencias permisivas o sin copyleft, las cuales no restringen el tipo de licencia de las obras derivadas, pudiendo distribuirse incluso bajo una licencia no libre, ejemplo de estas licencias son la BSD o Apache.

En algunas ocasiones el titular de los derechos de autor (copyright) de un software puede publicarlo al mismo tiempo bajo diferentes licencias dual. Este tipo de licenciamiento se conoce como Dual. Por ejemplo, podría

publicarse un software bajo licencia libre y también una versión modificada bajo otro tipo de licencia. Esta técnica ha sido usada en ocasiones como modelo de negocio por empresas que desarrollan Software Libre, como por ejemplo MySQL, que aunque se distribuye con licencia GPL permite distribuirse en productos no libres a través de una licencia comercial; esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión libre.

3.4.2. Open Source

En 1998 nace el término Open Source fruto de una reunión entre Eric S. Raymon, Bruce Perens, Am Ockman, Todd Anderson, Chris Peterson, John Hall y Larry Augustin, entre otros. Entre sus objetivos se encontraba evitar la confusión del término Free Software, ya que en inglés, free tiene el significado de libre y de gratis.

La diferencia principal entre el Software Libre (Free Software) y el Open Source (Código Abierto) son principalmente filosóficas, de hecho ambos reconocer casi las mismas licencias. Los principales ideales del movimiento Open Source son:

- Apostar por la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho.
- Darle mayor relevancia a los beneficios prácticos de compartir el código fuente.
- Interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto.
- Evitar la ambigüedad del termino inglés free (gratis o libre) en “Free Software”.

Mientras que en el Software Libre el principio fundamental es la libertad para los usuarios y la comunidad.

Sistemas de Información

4.-Sistema de Información

4.- Sistemas de Información

- 4.1.- Introducción
- 4.2.- La transición hacia la sociedad de la información
- 4.3.- Características
- 4.4.- Definición
- 4.5.- Estructura de un S.I.
- 4.6.- Integración frente a Independencia
- 4.7.- Funciones básicas de tratamiento de la información
- 4.8.- Actividades de un sistema de información
- 4.9.- Tipos y Usos de los Sistemas de Información
- 4.10.- Construcción de un sistema de información
- 4.11.- Evolución de los sistemas de información



4.1.-Introducción:

En los últimos años se han incorporado a nuestro entorno numerosos avances tecnológicos que han inundado hogares y oficinas. Ninguno de ellos ha tenido una influencia tan grande como el ordenador. Las tecnologías de información han irrumpido en la sociedad de manera explosiva, lo que permite hablar de auténtica revolución tecnológica. La velocidad con que los ordenadores han cobrado protagonismo se ha debido a numerosos factores de tipo económico y sociológico. No cabe duda de que la disminución del precio, el tamaño y peso de los equipos, junto con el increíble aumento de las prestaciones y la continua aparición de nuevos y más completos programas que amplían el campo de aplicación de estas herramientas, han conducido a la tremenda proliferación de ordenadores a la que asistimos. Además, las tecnologías de la información presentan un gran potencial de crecimiento que no permite prever un punto de madurez del producto en los próximos años.

Las transformaciones sociales producidas por los sistemas de información han sido de tal magnitud, que hoy por hoy es impensable una vuelta atrás o el abandono de su uso. Son demasiadas las actividades cotidianas que tienen que ver con un ordenador: desde las transacciones bancarias a las comunicaciones telefónicas, el supermercado, el juego, la enseñanza..., demasiadas las aportaciones a la “sociedad del bienestar” para predecir un retroceso. En la empresa, la preocupación permanente por la mejora de la administración, las finanzas y la producción han conducido a la rápida adopción de sistemas automáticos capaces de facilitar las tareas mecánicas y rutinarias, evitar errores y mejorar el control de la cartera de clientes con el incremento consiguiente de la calidad que percibe el mercado.

Durante las tres últimas décadas hemos asistido a una segunda revolución tecnológica a causa de la integración de los ordenadores y los sistemas de información en la estrategia empresarial, factor básico de nuevas ventajas competitivas en manos de los directivos y arma poderosísima para obtener nuevas oportunidades de negocio. Los directivos han ido comprendiendo el papel estratégico que los sistemas de información pueden desempeñar en la organización. Con ellos se pueden levantar barreras para la entrada de nuevos competidores en nuestro mercado, se consiguen ventajas competitivas basadas en un mejor servicio, en la diferenciación, en la agilidad..., se fideliza al cliente se integra al proveedor, todo lo cual disminuye los costes asociados a nuestra “cadena de valor”.

Era previsible que en la década de los noventa la significación de los sistemas de información fuese mayor si cabe. Sin embargo, muchos

directivos de hoy empiezan a plantearse si realmente los sistemas de información no son más que un servicio indiferenciado susceptible de ser subcontratado al igual que los servicios de seguridad o de telecomunicaciones. Esta idea, que encaja perfectamente en el modelo de “corporación virtual”, se ve reforzada por la posibilidad de ahorros en torno al 30 por 100 respecto al gasto de un departamento de informática propio. Pero, pese a que muchas empresas norteamericanas han optado por esta nueva fórmula, existe un riesgo real en el hecho de dejar en manos de un tercero una herramienta que por estratégica, no puede separarse de la esencia misma de la organización. En verdad, no se puede considerar a los proveedores externos de servicios informáticos (o de cualesquiera otro) socios estratégicos, porque obviamente las metas y objetivos empresariales no coinciden nunca; por otra parte, la contratación de proveedores externos puede resultar, si no se negocia bien, más cara a largo plazo que el mantenimiento de las capacidades propias.

Es un hecho que los departamentos de informática integrados en la organización pueden resultar muy competitivos si se gestionan correctamente, lo que es perfectamente aplicable tanto a las grandes corporaciones como a las PYME. Es cierto que se puede reducir los costes de la mayor parte de los departamentos de informática ligando la política de sistemas de información a la estrategia de la empresa mediante la adecuada adaptación de los recursos humanos y materiales a las esencia del negocio y recurriendo a las herramientas más potentes, desde el benchmarking a los procesos de reingeniería, análisis de valor y programas de calidad total. La evolución imparable hacia la descentralización de la ejecución y la toma de decisiones conlleva la distribución de la capacidad (de la potencia) informática y la disolución de los monolíticos departamentos de informática (auténticas fortalezas de poder en algunas organizaciones) que dan paso, poco a poco, a la aparición de un nuevo concepto: el centro de información corporativo.

Siguiendo esta tendencia, se ha podido constatar en los últimos años el ascenso de los organigramas de los responsables de sistemas de información (en inglés Chief Information Officer o CIO). Así, cada vez más, se considera al director de informática un gestor. Cada vez menos, un técnico. Su formación suele incluir algún master en administración de empresas más que cursos de especialización en sistemas operativos más o menos complejos. Se desmitifica su función y se asume que el éxito de su trabajo depende de su capacidad de integrar de manera coherente las decisiones y planes sobre sistemas de información en la estrategia empresarial.

Por otra parte, es habitual oír hablar de que esta o aquella empresa ha obtenido ventajas competitivas y estratégicas mediante un adecuado uso de las tecnologías de información. Sin embargo, lo cierto es que casi nunca es posible liderar ese tipo de cambios en un sector y lo que se impone es la necesidad de “no perder el tren”, es decir, saber ser un buen seguidor de los

líderes del mercado. Se trata de no caer en la desventaja competitiva más que de ser capaz de generar una ventaja relativa. Innovar puede ser a veces tan peligroso como no reaccionar a tiempo y correctamente a las nuevas condiciones del entorno. De hecho, muchos de los programas de innovación han acabado en sonoros fracasos. Saber reaccionar a las mejoras introducidas por otros no sólo abarata los desarrollos, sino que asegura la existencia de una meta alcanzable en tiempo y costo. En manos del directivo está elegir una u otra opción, para lo cual necesitará adquirir una visión global y empresarial de los sistemas de información.

Para alcanzar un objetivo estratégico hacen falta tres requisitos: tener una visión de lo que se quiere, conocer aproximadamente las herramientas y recursos necesarios para su obtención y dar los primeros pasos.

4.2.- La transición hacia la sociedad de la información

Una gran parte de las ocupaciones profesionales está vinculada en la actualidad a la creación, procesamiento y distribución de la información. Como se ha comentado en el punto anterior nos enfrentamos a un cambio social sin precedentes, provocado por un rápido aumento en la eficiencia de la microelectrónica, la reducción de costes en el tratamiento de la información y por la convergencia de áreas como las telecomunicaciones y la informática.

Estas tendencias tecnológicas tienen aplicaciones y repercusiones en prácticamente todos los campos de actividad social, como la industria, las finanzas o el comercio, y en el modo de vida social.

En este medio ambiente profundamente sujeto a cambios imprevisibles y extremadamente dependiente de la información, las organizaciones van adquiriendo conocimientos y experiencias que les ayudan a obtener mayor rentabilidad de sus recursos de información, así como a conseguir aumentos de la productividad de su información.

Este proceso de aprendizaje se puede describir a través de varias etapas:

- Primera Etapa: En la que el principal objetivo de las organizaciones es controlar la información, es decir, desarrollar y aplicar procesos y procedimientos para organizar mejor los documentos o papeles que generan. Es la hora de la “gestión de los impresos”
- Segunda Etapa: En la cual, las empresas, envueltas ya en la carrera de las tecnologías de la información, las aplican progresivamente y por separado al proceso de datos, la toma de decisiones, las comunicaciones de la organización o la automatización de oficinas.

- Tercera Etapa: En esta, las organizaciones empiezan a adquirir conciencia de que la información es un activo tan importante como los recursos humanos, los medios de producción o financieros. Con este sentimiento se dan cuenta de la necesidad de crear una nueva función que resuelva los problemas de información de la empresa y gestione para ella los recursos “ad hoc” de que ésta dispone, lo mismo que se gestionan y resuelven recursos y problemas de personal, de producción o de financiación.
- Cuarta Etapa: Las organizaciones adoptan una estrategia todavía más activa en el uso de la información, intentando encontrar en ella el medio de saber lo que hace y planea la competencia, a través del desarrollo de una inteligencia empresarial.
- Quinta Etapa: En la cual, las organizaciones integran la información en la estrategia corporativa, reafirmando su valor como recurso, y utilizándola, junto con sus tecnologías, para concebir nuevas formas de diseño, fabricación y venta de sus productos tradicionales, y haciéndola instrumento de diversificación mediante la concepción de productos específicos derivados de la información que poseen.

A lo largo de este proceso de aprendizaje se asiste a una cristalización progresiva de la función *información* en la organización, orientada a desarrollar una base de información efectiva para guiar y orientar las decisiones y elevar la eficacia en la gestión.

La información pasa a ser para la organización, un recurso tan importante como los medios humanos o los recursos financieros. Se trata de un activo estratégico crítico que debe ser gestionado mediante una función específica cuyo objetivo será poner en operación mecanismos que permitan a la organización adquirir o producir los datos o información que necesite para el cumplimiento de sus objetivos y planes corporativos, departamentales o individuales con la suficiente calidad y precisión, en el momento adecuado y con el mismo coste.

4.3.- Características

Si tuviéramos que resumir con una sola frase el principal cometido de un Sistema de información dentro de una organización, podríamos afirmar que éste se encarga de entregar la información oportuna y precisa, con la presentación y el formato adecuados, a la persona que la necesita dentro de la organización para tomar una decisión o realizar alguna operación y justo en el momento en que esta persona necesita disponer de dicha información.

Hoy en día, la información debería ser considerada como uno de los más valiosos recursos de una organización y el Sistema de Información es el

encargado de que ésta sea gestionada siguiendo criterios de eficacia y eficiencia.

En primer lugar, se debería hacer la distinción entre datos e información, término que en ocasiones se pueden llegar a confundir.

Los datos reflejan hechos recogidos en la organización y que están todavía sin procesar, mientras que la información se obtiene una vez que estos hechos se procesan, agregan y presentan de la manera adecuada para que puedan ser útiles a alguien dentro de la organización, por lo que de este modo estos datos organizados y procesados presentan un mayor valor que en su estado original.

Los datos quedan perfectamente identificados por elementos simbólicos (letras y números), que reflejan valores o resultados de mediciones.

Sin embargo, la información son “datos dotados de relevancia y propósito”, como señala Peter Drucker (“Gestión del Conocimiento”; Ediciones Deusto; 2003), que permiten reducir la incertidumbre de quien los recibe.

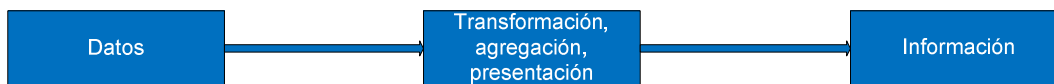


Figura 4.1.- El proceso de transformación de los datos en información. Fuente: “Gestión del Conocimiento”; Ediciones Deusto; 2003.

La información será útil para la organización en la medida en que facilite la toma de decisiones (figura 4.1) y, para ello, ha de cumplir una serie de requisitos, entre los que caber citar:

- **Exactitud:** la información ha de ser precisa y libre de errores
- **Compleitud:** La información debe contener todos aquellos hechos que pudieran ser importantes.
- **Económica:** el coste en que se debe incurrir para obtener la información debería ser menor que el beneficio proporcionado por ésta a la organización.
- **Confianza:** para dar crédito a la información obtenida, se ha de garantizar tanto la calidad de los datos utilizados, como la de las fuentes de información.

- **Relevancia:** la información ha de ser útil para la toma de decisiones. En este sentido, conviene evitar todos aquellos hechos que sean superfluos o que no aporten ningún valor.
- **Nivel de detalle:** la información debería presentar el nivel de detalle indicado a la decisión que se destina. Se debe proporcionar con la presentación y el formato adecuados, para que resulte sencilla y fácil de manejar.
- **Oportunidad:** se debe entregar la información a la persona que corresponde y en el momento en que ésta la necesita para poder tomar una decisión.
- **Verificabilidad:** la información ha de poder ser contrastada y comprobada en todo momento.

Por otra parte, no debemos olvidar que el exceso de información también puede ser causa de problemas, suponiendo un obstáculo en vez de una ayuda para la toma de decisiones.

Asimismo, cada función y nivel organizativo en la empresa tiene diferentes necesidades de información, afectando a los formatos, origen, periodicidades, nivel de agregación y otras características. A medida que se asciende en el escalafón organizativo de la empresa, observaremos cómo la información requerida aumenta en nivel de agregación (menor nivel de detalle), incorpora información del entorno y hace un mayor énfasis en el medio y largo plazo, a diferencia de la información puramente operativa, que normalmente se refiere a los hechos ocurridos dentro de la propia empresa y en un corto plazo.

En definitiva, la información y el conocimiento que acumulan las organizaciones debieran ser considerados como un recurso más, al mismo nivel que el capital, los bienes e instalaciones o el personal. En consecuencia, es necesario gestionarlo y explotarlo adecuadamente, para que pueda contribuir a la consecución de las metas y objetivos fijados por la organización.

4.4.-Definición

La información es indefectiblemente un concepto polisémico que engrosa el *corpus conceptual* de varias disciplinas aparentemente tan distantes como la comunicación, la pedagogía, la psicología, la cibernética, entre otras.

La interdisciplinariedad es además, una particularidad de la Ciencia de la Información, y es, en este contexto, que se insertan los sistemas de información.

Borko (Borko H. Information Science: What is it? American Documentation), al intentar definir la Ciencia de la Información, parte del criterio de que es una “ciencia interdisciplinar”, punto de convergencia de varios campos del conocimiento, como por ejemplo, la lógica matemática, la lingüística, la psicología, la bibliotecología, la administración y las técnicas computacionales, entre otras.

La escuela soviética, identificada con los preceptos de *Mijailov y Guliarevskii* (Mijailov A, Guliarevskii RS. Curso introductorio sobre informática-documentación. La Habana), también reconocía la integración de las ciencias como un fenómeno inherente a su especialización y parte constituyente de las leyes del desarrollo científico.

Saracevic (Saracevic T. Interdisciplinary nature of Information Science, 1995) afirmaba que el problema básico para la comprensión de la información y de la comunicación, de sus manifestaciones y efectos sobre el ser humano, de sus principios y sus aplicaciones para hacer accesible el conocimiento acumulado, particularmente con el uso de las tecnologías, es que no puede resolverse desde una sola disciplina.

A pesar de reconocer que la Ciencia de la Información es por naturaleza interdisciplinar, y en consonancia, sus enfoques deben partir permanentemente desde la perspectiva holística (ver cada parte como un todo), algo que se encuentra, al menos, durante muchos años a la hora de encarar su estudio, son una serie de distintos sistemas como “sistemas de almacenamiento y recuperación de la información”, “sistema de clasificación decimal”, “sistema de información documental”, “sistemas de archivos”, “sistemas de bibliotecas públicas”, entre otros.

Esto se corrobora si se considera que el tratamiento sobre “sistemas de información”, que provienen del campo de estudio que nos compete, según la literatura revisada, aparecen en fuentes que datan de la década de los años 80 - la primera que encontramos data de 1983 y se halla en la versión original del Glosario de la ALA (La Asociación Americana de Bibliotecarios, por sus siglas en inglés) de Bibliotecología y Ciencia de Información; entonces, puede deducirse que surgen de manera tardía con respecto a otras disciplinas.

La ALA identifica como un sistema de información aquel “*sistema completo diseñado para la generación, colección, organización, almacenamiento, recuperación y difusión de la información en una institución, organización u otra área institucional definida*” (ALA. Glosario de la ALA de Bibliotecología y Ciencias de la Información. México: Ediciones Díaz de Santos).

Muñoz Cruz (Muñoz Cruz V. Gestión y planificación de sistemas y servicios de información) señala que “un *sistema de información es un conjunto de elementos o componentes relacionados con la información que*

interaccionan entre ellos para lograr un objetivo: facilitar y recuperar información”.

Las definiciones anteriores coinciden en el carácter funcionalista que se otorga a los sistemas de información. Tanto la ALA como *Muñoz Cruz*, los reducen básicamente a la “recuperación” y “difusión” de información.

Buckland (Buckland M. Information and information systems. New York: Greenwood Press) introduce una perspectiva cognitiva al entender que los sistemas de información “*facilitan el proceso de aprendizaje, estimulan la curiosidad, suprimen la memorización de hechos y datos que pueden perjudicar el desarrollo del pensamiento crítico y la autoestima*”. Revela la intencionalidad implícita de obtener información para que el sistema pueda denominarse como tal, basándose en la necesidad de conocimiento.

A propósito de la intencionalidad que propone *Buckland*, es vital este indicador en el proceso no sólo de obtención de información sino de adquisición de conocimientos. Sólo cuando el sujeto tiene la intención de conocer los objetos y sujetos es cuando pueden considerarse fuente de información.

Capurro (Capurro R. Epistemología y Ciencia de la Información) afirma que “*está dirigido a sustentar la producción, recolección, organización, interpretación, almacenamiento, recuperación, diseminación, transformación y uso de los conocimientos y debe concebirse en el marco de un grupo social concreto y para áreas determinadas. Sólo tiene sentido hablar de un conocimiento como informativo en relación a un presupuesto conocido y compartido con otros con respecto al cual la información puede tener el carácter de ser nueva y relevante para un grupo o para un individuo*”, introduciendo un enfoque social a la definición de sistemas de información

Baiget (BAIGET, T.: Análisis, diseño y gestión de sistemas de información) aporta el enfoque técnico al definir el sistema de información como “*la entidad constituida por partes que interaccionan entre ellas de una forma dinámica, coordinadas para conseguir objetivos comunes*”. Por “*dinámica*” hace referencia a no ser necesariamente lineal o proporcional, y adaptada a cada situación momentánea.

Ponjuán (Ponjuán Dante G. Los sistemas de Información. En: Los sistemas de información: principios y aplicaciones. La Habana: Félix Varela; 2004) es del criterio de que el objetivo concreto particular de los sistemas de información se traduce en responder a la satisfacción de necesidades de una organización o de un individuo o grupo de individuos. Por tanto, permanentemente se intenta comprobar su grado de eficiencia, introduciendo abiertamente el enfoque de gestión.

En esencia, y sobre la base de los apuntes de *Codina* (La investigación en sistemas de información. En: Tramullas Saz J (ed). Actas del seminario Tendencias de investigación en Documentación.,1996), puede definirse un *sistema de información como el conjunto de los elementos y procesos que intervienen dinámicamente en la explotación de información cognitiva concebida en el marco de un grupo social concreto y para áreas determinadas, cuyo propósito es facilitarles el acceso al conocimiento y apoyarlos en la toma correcta de decisiones.*

Son sistemas altamente complejos que, en su dinámica, tienden a superarse a sí mismo, porque no sólo manipulan, analizan e interrogan para recuperar información, sino que, también son capaces de generar informaciones evaluadas para el apoyo a la toma de decisiones, incluso sobre productos muy sofisticados que surgen con los nuevos enfoques de gestión. Ahora, deben superarse, una vez más, para adelantarse en la difícil propuesta de organizar y potenciar los activos cognitivos totales.

Al idearse como la implicación e interrelación de todos los flujos de información de la organización, transitan hacia lo abierto, en la formación de organizaciones que clasifican como sistemas de información en pos del conocimiento.

Bueno Campos (Economía de la empresa: Análisis de las decisiones empresariales, ed. Piramide, 1996) señala que los sistemas de información constan de los siguientes elementos:

- La información: conjunto de datos estructurados según los mensajes a comunicar.
- Los beneficiarios de la información: los miembros de la organización y agentes relacionados con ella.
- Los elementos soporte: Proceso de tratamiento de información, sistemas de análisis de datos, procedimientos de comunicación o difusores de información y soportes de información.

Particularmente interesante su propuesta, al considerar los beneficiarios de la información como parte integrante del sistema.

Los sistemas de información son un entramado de sistemas y subsistemas donde los flujos de información se bifurcan para responder a beneficiarios que, en algunos momentos, se hallan en el contexto del ambiente, mientras que, en otros, son proveedores o procesadores de dicha información, para asumir así diferentes funciones en el sistema.

Es por ello, que resulta muy difícil pensar que algún sistema de información pueda por sí solo contener todos los recursos de información que necesita y actuar de manera independiente para responder a entes o entidades, cuyas necesidades pueden modificarse abruptamente de acuerdo con contextos determinados.

Es, por tanto imperioso cambiar los modelos mentales tradicionales donde los sistemas de información se han visto durante mucho tiempo como sistemas aislados, cerrados mediante “murallas” espacio-temporales; enmarcados en una organización; para decididamente transitar hacia lo abierto, en la conformación de organizaciones que clasifican como sistemas de información en pos del conocimiento.

Para el análisis y diseño de un sistema de información, se consideran diferentes tipos de modelos. El profesor *López Yépez* (El desarrollo de los sistemas de información y documentación. Cuadernos E.U.B.D. Complutense 1991.), a partir de varios criterios, distingue tres modelos de sistemas de información que son básicos para comprender, diagnosticar y ver con amplitud un escenario posible de sistemas de información:

- *Modelo A*: Sistema que contempla desde una perspectiva general, individual y con subsistemas. Su estudio sirve para el desarrollo del resto de los modelos. Se compone básicamente de los procesos de “adquisición de los datos, transmisión, proceso que incluye el almacenamiento y recuperación de la información, utilización y transferencia, este último como sinónimo de comunicación o diseminación”.
- *Modelo B*: Sistema basado en el enfoque de gestión de información, que parte de consideraciones como que “la información es un bien económico, la información es el nervio de la organización y la organización es en sí un sistema de información”.
- *Modelo C*: Es el resultado de la conjunción de redes y centros de información, enmarcado en las políticas nacionales y territoriales de información. El autor especifica que el sistema actúa como una red bajo el principio de coordinación de centros en que, por delegación, se invisten de determinada responsabilidad en la recolección y difusión de fuentes. Es, en este modelo, donde intervienen las políticas, estrategias, y todo un entramado regulatorio que se relaciona con el mundo de las decisiones.

Los tres modelos aportan elementos vitales para la visión de un nuevo modelo, llamémosle *Modelo D*, donde confluyen los procesos dinámicos de la gestión de información a la que se le agregaría el modelado de los ambientes de colaboración para la gestión del conocimiento.

En definitiva, Entenderemos como sistema de información (S.I.) el conjunto de procedimientos, manuales y automatizados, y de funciones dirigidas a la recogida, elaboración, evaluación, almacenamiento, recuperación, condensación y distribución de información, dentro de una organización, orientadas a promover el flujo de las mismas desde el punto en el que generan hasta el destinatario final de las mismas.

4.5. – Estructura de un S.I.

Un S.I. completo para una gran organización es un instrumento enormemente complejo que está constituido por un gran número de partes, o subsistemas, que interaccionan unos con otros en grado diferente y cuya estructuración tiene simultáneamente una dimensión vertical y horizontal.

Estructura vertical:

En su dimensión vertical el S.I. posee distintos niveles jerárquicos:

- a) Nivel operacional. En el que se manejan los procedimientos de rutina relacionados con las distintas actividades de la organización. En este nivel tiene lugar el grueso del tratamiento de datos y el sistema mantiene vínculos estrechos con los procesos físicos realizados por la organización.

Así los subsistemas operacionales recogen datos de los sucesos del mundo real. Los datos entran en el S.I. y se almacenan en una Base de Datos que se actualiza para reflejar las consecuencias del suceso. Habitualmente el resultado del proceso se materializa en un documento de trabajo específico.

- b) Nivel táctico. Trata de las tomas de decisiones a plazo relativamente corto basadas en información elaborada a partir de datos transaccionales o procedentes de fuentes externas formalizadas.

Las decisiones tomadas a nivel táctico se implementan generalmente a través de la parte operacional del S.I. mediante un procedimiento automatizado en un S.I. integrado o a través de medios más informales en otros casos.

- c) Nivel estratégico. Trata las decisiones más amplias, a mayor plazo, apoyadas menos en información formal procedente de datos transaccionales y que dependen en gran medida de fuentes de información externa. El límite entre los componentes tácticos y estratégicos es difuso. En cualquier caso las decisiones a nivel estratégico son más generales y menos susceptibles de formalización.

Estructura horizontal:

En su estructura horizontal, y dentro de cada nivel, las funciones se subdividen en aplicaciones o procedimientos. Así por ejemplo el nivel operativo de una empresa de fabricación incluiría subsistemas de entrada de pedidos, control de inventario, etc...

Estos subsistemas pueden estar directamente conectados unos con otros aportando un alto grado de integración o por el contrario pueden estar concebidos bajo un enfoque separado o autónomo que contempla cada

aplicación o procedimiento de manera separada e independiente de los restantes procedimientos automatizados de la organización.

4.6 – Integración frente a Independencia

Como hemos mencionado anteriormente, una cuestión fundamental en el diseño de un sistema es el equilibrio entre integración e independencia. Un sistema integrado, M.I.S. (Management Information System) según algunos autores, es aquél que tiene un alto grado de coordinación con entrada y salidas rígidamente establecidas, teniendo en cuenta los efectos de un subsistema sobre los otros y en el que los recursos son ampliamente compartidos.

Las principales ventajas derivadas de un enfoque integrado son las siguientes:

- a) Mayor eficiencia conjunta y una interrelación más efectiva de actividades entre subsistemas.
- b) Incorporación de hábitos para compartir ampliamente los recursos obteniendo beneficios potenciales, debidos a economías de escala y especialización.
- c) Posibilidad de abordar las decisiones desde la perspectiva del sistema común conjunto en vez de sobre una base subóptima que utilice información y objetivos locales.

Como contrapartida, el coste fundamental de la integración es la complejidad y riesgo añadidos. En la práctica, intentar diseñar un sistema completamente integrado puede resultar una tarea demasiado compleja para ser realizada con éxito.

Además la integración estricta tiene también el riesgo de ser más vulnerable a incertidumbres. Así la fragmentación tiene su precio en términos de eficiencia, pero sin cierto grado de duplicación y elasticidad, el riesgo de un fallo generalizado puede ser inaceptablemente alto para cualquier organización.

Por su parte, el enfoque independiente, separado o autónomo, anteriormente mencionado, aporta al sistema simplicidad, sensibilidad y consistencia frente a un entorno incierto. Sin embargo, la consecución con éxito de objetivos subóptimos en subsistemas independientes solo puede aproximarse a los objetivos globales del sistema.

Habitualmente los diseñadores de un sistema se enfrentan a difíciles elecciones entre la integración y la independencia.

No se trata de decidir un enfoque integrado o no en sentido estricto, sino de determinar el equilibrio relativo aconsejable entre integración e independencia, el grado en que un sistema dado deba favorecer una u otra.

Cuestiones como el tamaño de la organización su grado de diversificación y el análisis de las interacciones entre sus principales divisiones determinarán el grado de descentralización más adecuado.

Por otra parte, los avances tecnológicos tienen efectos combinados en esta cuestión. Así los procesadores de gran capacidad de los grandes ordenadores y las herramientas de desarrollo potentes hacen viable operar la mayor complejidad de las aplicaciones que propician un acoplamiento estricto y una comparación de datos de gran volumen mientras que los minis y microordenadores de bajo coste y potentes facilitan el cálculo económico distribuido y la asunción por los usuarios de un panel más importante en la atención de sus propias necesidades de información.

4.7. – Funciones básicas de tratamiento de la información

Dentro de la complejidad general del S.I., las funciones realizadas dentro de cada subistema tienden a ser conceptualmente claras. Así los datos entran en el sistema y luego son transmitidos, almacenados, manipulados y presentados. Veamos ahora brevemente los principales aspectos de las funciones básicas de tratamiento de la información dentro del S.I.

Ingreso de datos

- a) Técnicas más apropiadas (operación de teclado manual o reconocimiento óptico de caracteres) a emplear y su coste.
- b) Control de errores a través de procesos de verificación y edición.
- c) Enfoque integrado capturando solamente una vez un elemento dado de datos y a continuación compartirlo con todas las aplicaciones que lo necesitan. Para grandes volúmenes de entradas, los pros y contras antes mencionados en relación con la integración o independencia, casi siempre favorecen una sustancial comparación de datos.
- d) Interactividad como medio para mejorar sustancialmente la eficacia y calidad de las operaciones al estar directamente apoyadas por personal operativo y ser susceptibles de controles de error inmediatos.

Almacenamiento de datos

El S.I. debe mantener grandes ficheros de datos destinados a suministrar la información para el tratamiento de transacciones y para la toma de decisiones. Los principales aspectos a considerar son:

- a) Papel de la Base de Datos en la organización a fin de que se mantenga como una representación suficientemente fiable de la realidad.
- b) Organización de la Base de Datos de forma que se facilite el acceso a partes específicas.
- c) Almacenamiento en línea versus fuera de línea. (online vs. offline). Este aspecto contempla el estudio de procedimientos que minimicen las necesidades de almacenamiento y el tiempo necesario para generar la información útil. Estos objetivos se logran proporcionando al sistema un medio de almacenamiento a largo plazo para datos con una pequeña probabilidad de acceso y sustituyendo algunos datos detallados de transacciones, por datos resumidos que se preparan y conservan mediante almacenamiento en línea.

Cálculo

La forma habitual de cálculo implícita en la mayoría de los S.I., se refiere a cálculos matemáticos, manipulación de datos y ejecución de diversas acciones de acuerdo con los resultados obtenidos. Mediante los cálculos el S.I. transforma los datos brutos en información utilizable por el propio sistema e en forma ajena al mismo.

Como respuesta a la necesidad de cálculo prevista el diseño de un S.I. debe contemplar la necesaria potencia de tratamiento de los equipos soporte. En este sentido es conveniente señalar que asistimos en la actualidad a una demanda insaciable de potencia de cálculo a bajo coste, fundamentalmente provocada por el hecho de que cierto número de tareas de soporte al usuario son extraordinariamente intensivas en cálculo.

Presentación de los resultados

La función de presentación de un S.I. proporciona una conexión esencial, o interfaz, entre el sistema y el usuario. Su finalidad es presentar la información de tal modo que mejore la capacidad del usuario para percibir y actuar sobre los hechos reflejados por la información. La eficacia de este interfaz se está convirtiendo en algo de la mayor importancia a medida que nos desplazamos rápidamente al uso casi universal de sistemas interactivos de respuesta rápida, que dependen fuertemente de una relación estrecha entre usuario y máquina.

Un diseñador dispone actualmente de una gran variedad de opciones para presentar la información. La tendencia apunta a mayor resolución multiplicidad de colores y dispositivos multifunción, tanto para las salidas impresas como para las presentaciones en pantalla. A pesar de los avances tecnológicos experimentados, el verdadero problema sigue estando en el diseño del Interfaz humano, de modo que el sistema proporcione el modo más eficaz de presentación de los resultados a los usuarios.

Comunicaciones

Los sistemas de Información actuales se diferencian muy notablemente de los del pasado en su creciente apoyo en las comunicaciones. Los avances experimentados en los Sistemas de Información están estrechamente relacionados con los avances realizados en el mundo de las telecomunicaciones.

Así hemos asistido a sistemas que dependían muy poco o nada de las telecomunicaciones y donde los datos eran comunicados mediante transporte físico de medios de almacenamiento. Más tarde pasamos al uso extendido de terminales de entrada de tareas a distancia que no incorporaban ninguna capacidad de procesamiento.

Ahora asistimos a la implantación de sistemas informáticos distribuidos en los que los ordenadores a través de la organización están conectados por medio de una red de telecomunicaciones. Cada ordenador remoto sobre la red tiene, generalmente, capacidades de cálculo autónomo significativas para servir a las necesidades especializadas de sus usuarios locales proporcionando también acceso a recursos mantenidos en otras localizaciones, eventualmente en un ordenador central con una potencia de procesamiento considerable y gran capacidad de almacenamiento en línea sobre disco.

Un sistema distribuido de este tipo es el vehículo por el que un Sistema de Información alcanza todas las partes de la organización. A nivel operativo, los terminales de trabajo individuales soportan la entrada de datos y los procesos de transacciones. Para la toma de decisiones a nivel táctico y estratégico, proporcionan cálculo autónomo y acceso a información selectiva almacenada en la Base de Datos personal del usuario.

Los recursos necesarios no disponibles sobre un terminal de trabajo personal (estación de trabajo) pueden ser alcanzados a través de la red a nivel departamental si la comparación es rápida, conveniente y eficiente o a nivel corporativo si se trata de recursos demasiado costosos para estar duplicados a nivel departamental.

Los diseñadores de la red de telecomunicaciones deberán seleccionar una combinación específica de topología de red, anchos de banda, protocolos de comunicaciones, equipo terminal y proveedores de comunicaciones. Su objetivo será un diseño que cumpla con las necesidades particulares de la organización a un coste aceptable. Los factores a considerar son el número y distribución geográfica de las estaciones de trabajo y de los ordenadores conectados a la red, el volumen esperado de tráfico entre los nodos, las demoras permitidas por colas en conexión de comunicaciones compartidas y los requisitos de fiabilidad y seguridad.

Hay que señalar por último que una de las grandes ventajas de una arquitectura distribuida es que no fuerza a un equilibrio de alternativas: cada tarea individual puede ser analizada con el fin de determinar si debería estar distribuida o no. En general, las aplicaciones grandes y complejas, para las que el equilibrio entre ventajas y desventajas tiende a favorecer la centralización, se mantendrán sobre el ordenador central; las aplicaciones de complejidad media tenderán a desplazarse a miniordenadores distribuidos a nivel departamental y las aplicaciones sencillas individuales se centrarán en los microordenadores (estaciones de trabajo).

El desarrollo de un sistema distribuido de este tipo es un reto importante para las organizaciones. Los usuarios deben desempeñar el papel dominante en la definición de los requisitos y la asunción de responsabilidades de desarrollar un entorno agradable para que los usuarios realicen sus tareas. Este entorno requiere una gestión efectiva de las Bases de Datos compartidas y de las disponibilidades de comunicaciones, un soporte técnico de alta calidad para los usuarios y un conjunto accesible de estándares que permita a los usuarios desarrollar aplicaciones con el resto de sistemas.

4.8.- Actividades de un sistema de información

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

El equipo computacional: el hardware necesario para que el sistema de información pueda operar.

El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáners, la voz, los monitores sensibles al tacto, el teclado y el ratón, entre otras.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfase automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interfase automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes.

A continuación se muestran las diferentes actividades que puede realizar un Sistema de Información de Control de Clientes:

Actividades que realiza un Sistema de Información:

Entradas:

- Datos generales del cliente: nombre, dirección, tipo de cliente, etc.
- Políticas de créditos: límite de crédito, plazo de pago, etc.
- Facturas (interfase automático).
- Pagos, depuraciones, etc.

Proceso:

- Cálculo de antigüedad de saldos.
- Cálculo de intereses moratorios.
- Cálculo del saldo de un cliente.

Almacenamiento:

- Movimientos del mes (pagos, depuraciones).
- Catálogo de clientes.
- Facturas.

Salidas:

- Reporte de pagos.
- Estados de cuenta.
- Pólizas contables (interfaz automática)
- Consultas de saldos en pantalla de una terminal.

Las diferentes actividades que realiza un Sistema de Información se pueden observar en el diseño conceptual ilustrado en la en la siguiente figura.

4.9. - Tipos y Usos de los Sistemas de Información

Durante los próximos años, los Sistemas de Información (Figura 4.2) cumplirán tres objetivos básicos dentro de las organizaciones:

1. Automatización de procesos operativos.
2. Proporcionar información que sirva de apoyo al proceso de toma de decisiones.

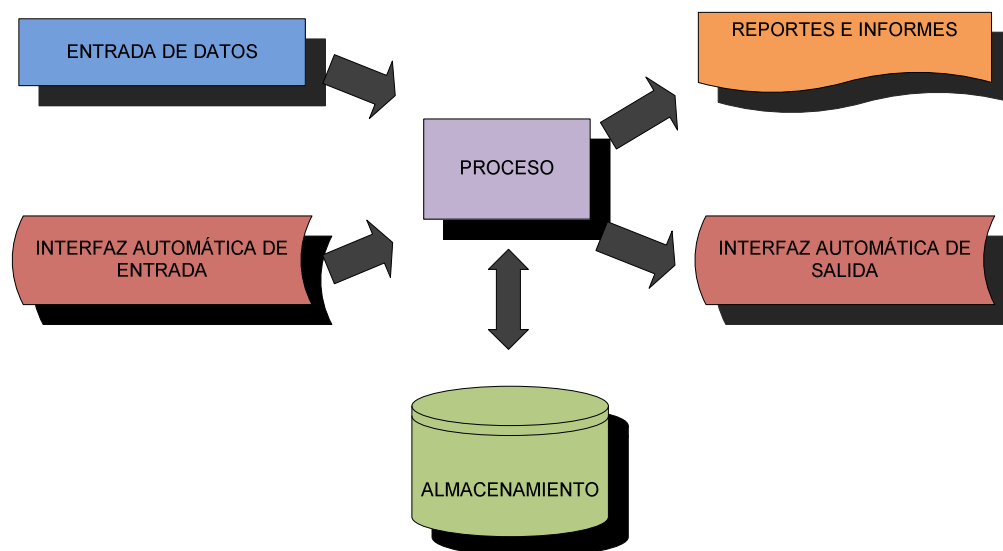


Figura 4.2: Diseño conceptual de un sistema de información. Fuente: Análisis y Diseño de Sistemas de Información. Kendall & Kendall. (2004) Prentice Hall

3. Lograr ventajas competitivas a través de su implantación y uso.

Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información (figura 4.3).

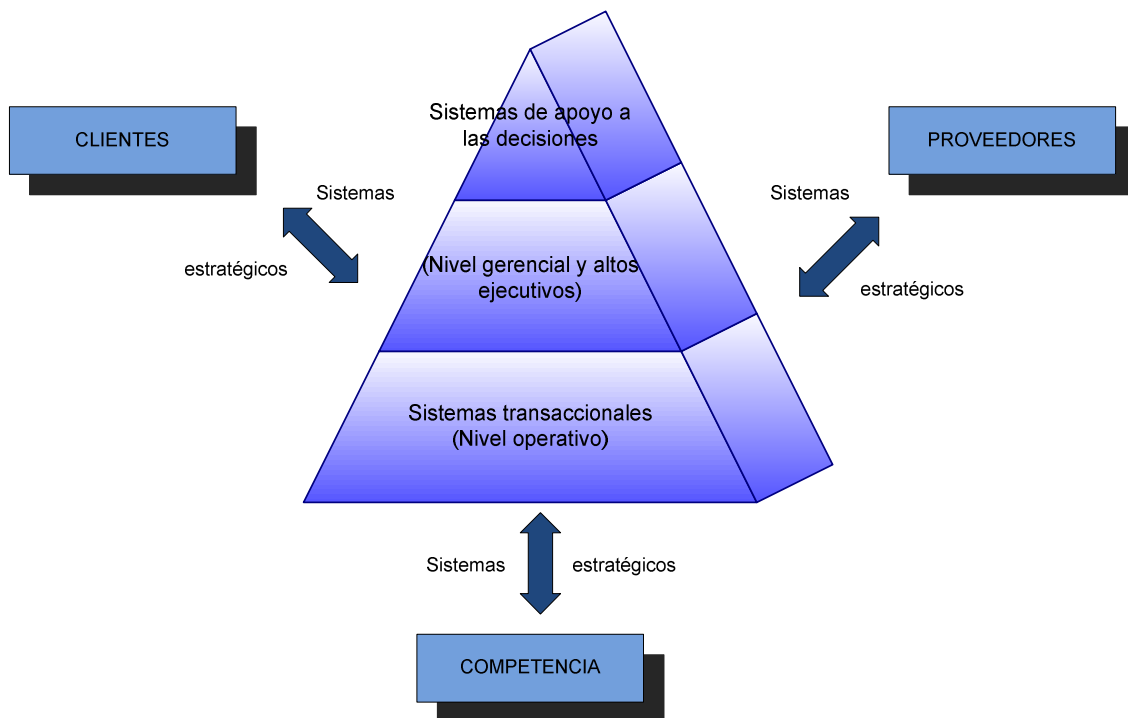


Figura 4.3: Tipos y usos de los Sistemas de Información. . Fuente: Análisis y Diseño de Sistemas de Información. Kendall & Kendall. (2004) Prentice Hall

A continuación se mencionan las principales características de estos tipos de Sistemas de Información.

Sistemas Transaccionales.

Son llamados TPS cuyas siglas corresponden a Transaction Processing System, o sistemas de procesamiento de transacciones.

Un ejemplo es la Corporación Financiera Internacional (CFI), filial del Banco Internacional para la Reconstrucción y el Desarrollo, cuyo sistema de transacciones funciona de la siguiente manera: El CFI busca inversores

interesados en los países más desarrollados y el capital proveído por éstos, es transferido a empresas privadas de países subdesarrollados cuyo capital privado no basta.

Otro ejemplo es el de la industria naviera, el cual por medio de su sistema de transacciones internacionales transportan diferentes tipos de carga de acuerdo a pedidos en diferentes países, siendo uno de los más transportados el petróleo, cuyos pedidos pueden ser ya sea privado o por contrato.

Los barcos transportan el petróleo desde los campos petrolíferos a las refinerías, siguiendo una serie de tratados y convenciones internacionales.

Sus principales características son:

- A través de éstos suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Con frecuencia son el primer tipo de Sistemas de Información que se implanta en las organizaciones. Se empieza apoyando las tareas a nivel operativo de la organización.
- Son intensivos en entrada y salida de información; sus cálculos y procesos suelen ser simples y poco sofisticados.
- Tienen la propiedad de ser recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.
- Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables.

Sistemas de Apoyo de las Decisiones. Las principales características de estos son:

- Suelen introducirse después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, genera poca información como resultado, pero puede realizar muchos cálculos durante su proceso.

- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser Sistemas de Información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivos y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un Sistema de Compra de Materiales que indique cuándo debe hacerse un pedido al proveedor o un Sistema de Simulación de Negocios que apoye la decisión de introducir un nuevo producto al mercado.
- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.

Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios, modelos de inventarios, etc.

Sistemas Estratégicos. Sus principales características son:

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Típicamente su forma de desarrollo es a base de incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.
- Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los Sistema Estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el uso de cajeros automáticos en los bancos en un Sistema Estratégico, ya que brinda ventaja sobre un banco que no posee tal servicio. Si un banco nuevo decide abrir sus puertas al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.

- Apoyan el proceso de innovación de productos y proceso dentro de la empresa debido a que buscan ventajas respecto a los competidores y una forma de hacerlo es innovando o creando productos y procesos.

Un ejemplo de estos Sistemas de Información dentro de la empresa puede ser un sistema MRP (Manufacturing Resource Planning) enfocado a reducir sustancialmente el desperdicio en el proceso productivo, o bien, un Centro de Información que proporcione todo tipo de información; como situación de créditos, embarques, tiempos de entrega, etc. En este contexto los ejemplos anteriores constituyen un Sistema de Información Estratégico si y sólo sí, apoyan o dan forma a la estructura competitiva de la empresa.

Por último, es importante aclarar que algunos autores consideran un cuarto tipo de sistemas de información denominado Sistemas Personales de Información, el cual está enfocado a incrementar la productividad de sus usuarios.

Sistemas de Conocimiento:

KWS, knowledge work system, o sistema de manejo de conocimiento.

Un ejemplo es el de aplicaciones como Photoshop, la cual ayuda a diseñadores gráficos en crear su arte publicitario por medio de poderosas herramientas con las cuales se puede manipular y modificar distintos tipos de gráficos y fotografías.

Sistemas Expertos:

AI, artificial intelligence, o inteligencia artificial.

Un famoso sistema experto es MYCIN, el cual es un sistema experto para la realización de diagnósticos, el cual aconseja a los médicos en la investigación y determinación de diagnósticos en el campo de las enfermedades infecciosas de la sangre. El sistema MYCIN, al ser consultado por el médico, solicita primero datos generales sobre el paciente: nombre, edad, síntomas, etc. Una vez conocida esta información por parte del sistema, el Sistema Experto plantea unas hipótesis. Para verificar la hipótesis el sistema consulta a la base de conocimientos, y también haciendo una serie de preguntas al usuario. Con las respuestas que recibe, el MYCIN verifica o rechaza las hipótesis planteadas.

Sistemas de Apoyo a Grupos:

GDSS, group decision support system, o sistemas de apoyo a decisiones de grupo.

Un sistema GDSS es el Vision Quest, el cual permite realizar junta electrónicas. Entre sus ventajas se encuentra su facilidad de uso.

Cualquiera puede conducir una junta electrónica y el sistema puede ser usado de manera distribuida. Las juntas se pueden realizar con los participantes en el mismo lugar o diferentes lugares, al mismo tiempo o a distintos tiempos. Aunque no pretende reemplazar las juntas cara a cara, su uso permite reducir los costos de viaje, la rapidez de toma de decisiones lo que resulta en una mejor eficiencia y productividad de las juntas. El sistema funciona en terminales de trabajo que pueden estar o no en el mismo lugar, la interacción se realiza a través del teclado y el monitor de la computadora.

Otro sistema es el CRUISER cuyas siglas son para Computer Supported Spontaneous Interaction. La importancia de este sistema se basa en la interacción informal. CRUISER está diseñado alrededor del concepto de comunidad o grupo virtual que existe sólo en un mundo virtual, donde las distancias geográficas entre los participantes no son importantes. Por sus características este sistema provee acceso instantáneo a cualquier persona y cualquier lugar.

La importancia del sistema está basada en dos ideas. La primera, los usuarios pueden navegar a través del mundo virtual en búsqueda de encuentros sociales. La segunda, el mundo virtual es independiente del mundo físico y puede ser organizado de acuerdo a las necesidades del usuario. En la práctica el usuario recorre pasillos, oficinas y áreas comunes, todas ellas generadas por computadora. Los usuarios se comunican a través de audio y video. CRUISER ataca uno de los problemas de los trabajos en equipo, reconoce la importancia de la comunicación informal. Provee además características de la práctica de trabajo permitiéndole diferentes niveles de privacidad.

Sistema de ejecutivos:

ESS, executive support system, o sistemas de apoyo a ejecutivos.

Un ejemplo es el sistema comprado por Pratt & Whitney, una corporación que se dedica a la producción de motores de propulsión a chorro. Ellos compraron el sistema denominado Commander EIS que permite representaciones a todo color y un menú imaginativo que puede aprenderse intuitivamente, con variaciones y excepciones que son destacadas mediante colores. Los usuarios pueden acceder datos mediante una pantalla táctil, ratón o teclado y pueden agrandar las imágenes para mayores niveles de detalle, ya sea navegando por sí mismos o siguiendo caminos previamente definidos.

El Commander EIS permite a la organización hacer el seguimiento de los parámetros de la calidad y factibilidad de las medidas tomadas para cada motor a reacción por tipo de cliente. Los datos aparecen de los sistemas actuales de producción y proporcionan información sobre la confiabilidad, disponibilidad de motores y partes, y sobre las entregas.

Otro ejemplo es el sistema implantado por la New York State Office of General Services que es responsable de dar servicio a otras dependencias en Nueva York. El sistema permite que los ejecutivos verifiquen el estado por programa, comparando el presupuesto con el gasto real y mostrando el gasto estimado hasta el final del año fiscal. La administración puede bajar para ver los detalles específicos en cada categoría. El sistema sólo contiene datos crudos, permitiendo a los usuarios una gran flexibilidad para agregarlos y analizarlos para satisfacer sus necesidades. El sistema es operado por medio de un menú muy fácil de usar. Los nuevos usuarios son capacitados mediante una demostración que dura media hora, y la experiencia ha demostrado que es todo lo que necesitan. No se cuenta con un manual del usuario.

4.10. – Construcción de un sistema de información

En este proceso se genera el código de los componentes del Sistema de Información, se desarrollan todos los procedimientos de operación y seguridad y se elaboran todos los manuales de usuario final y de explotación con el objetivo de asegurar el correcto funcionamiento del Sistema para su posterior implantación.

Para conseguir dicho objetivo, en este proceso se realizan las pruebas unitarias, las pruebas de integración de los subsistemas y componentes y las pruebas del sistema, de acuerdo al plan de pruebas establecido. Asimismo, se define la formación de usuario final y, si procede, se construyen los procedimientos de migración y carga inicial de datos.

En la actividad Preparación del Entorno de Generación y Construcción (CSI 1), se asegura la disponibilidad de la infraestructura necesaria para la generación del código de los componentes y procedimientos del sistema de información. Una vez configurado el entorno de construcción, se realiza la codificación y las pruebas de los distintos componentes que conforman el sistema de información, en las actividades:

- Generación del Código de los Componentes y Procedimientos (CSI 2), que se hace según las especificaciones de construcción del sistema de información, y conforme al plan de integración del sistema de información
- Ejecución de las Pruebas Unitarias (CSI 3), dónde se llevan a cabo las verificaciones definidas en el plan de pruebas para cada uno de los componentes
- Ejecución de las Pruebas de Integración (CSI 4), que incluye la ejecución de las verificaciones asociadas a los subsistemas y componentes, a partir de los componentes verificados individualmente, y la evaluación de los resultados.

Una vez construido el sistema de información y realizadas las verificaciones correspondientes, se lleva a cabo la integración final del sistema de información en la actividad Ejecución de las Pruebas del Sistema (CSI 5), comprobando tanto las interfaces entre subsistemas y sistemas externos como los requisitos, de acuerdo a las verificaciones establecidas en el plan de pruebas para el nivel de pruebas del sistema.

En la actividad Elaboración de los Manuales de Usuario (CSI 6), se genera la documentación de usuario final o explotación, conforme a los requisitos definidos en el proceso Diseño del Sistema de Información.

La formación necesaria para que los usuarios finales sean capaces de utilizar el sistema de forma satisfactoria se especifica en la actividad Definición de la Formación de Usuarios Finales (CSI 7).

Si se ha establecido la necesidad de realizar una migración de datos, la construcción y pruebas de los componentes y procedimientos relativos a dicha migración y a la carga inicial de datos se realiza en la actividad Construcción de los Componentes y Procedimientos de Migración y Carga Inicial de Datos (CSI 8).

Tras la descripción de las actividades los siguientes esquemas muestran las actividades mencionadas, la conexión entre ellas, así como su ubicación temporal para realizar la construcción del sistema de información.

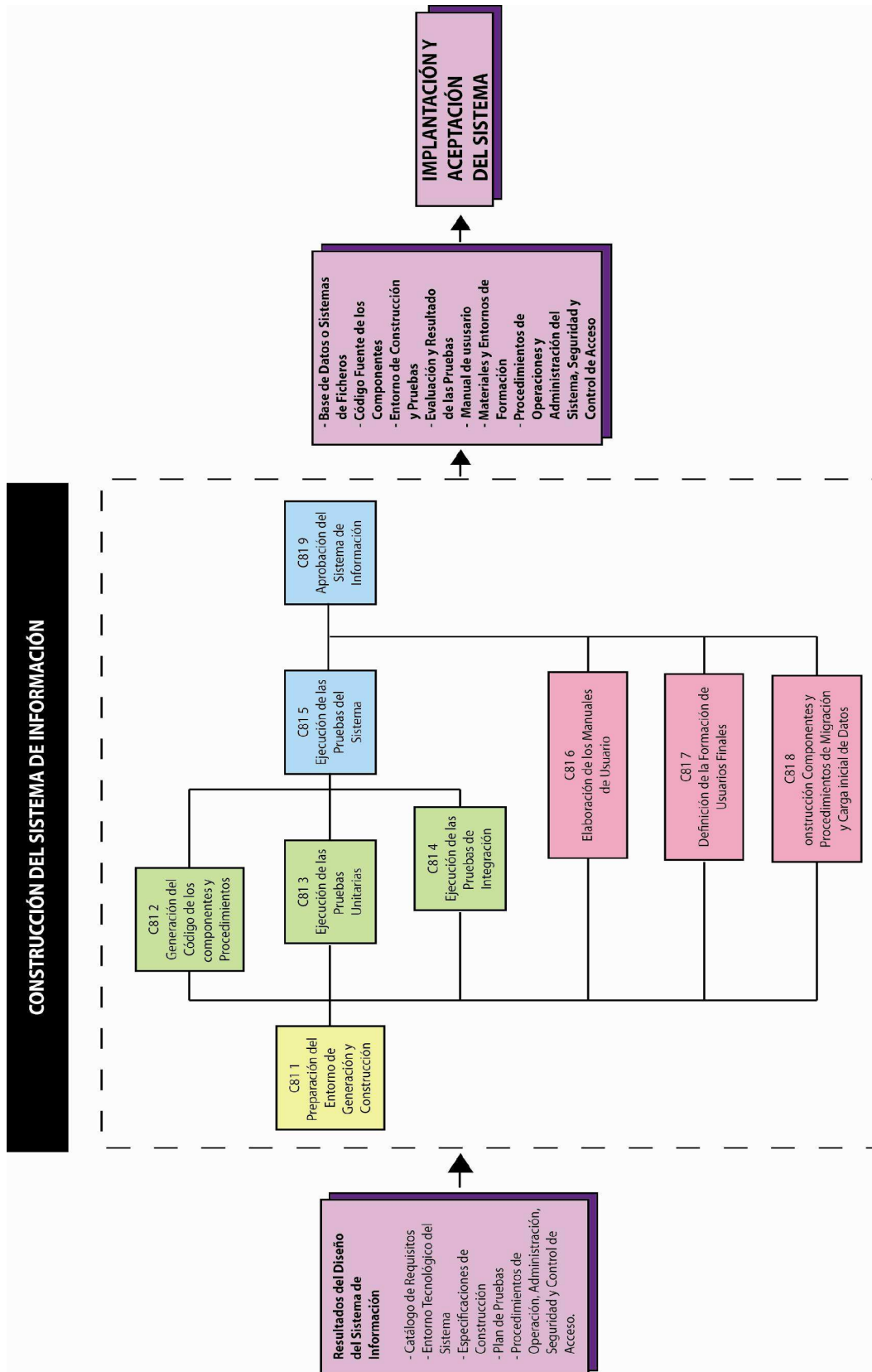


Figura 4.4.- Construcción de un sistema de información. Fuente: Ministerio de Administraciones Públicas

4.11.- Evolución de los sistemas de información

Del capítulo 4.7 se desprende la evolución que tienen los Sistemas de Información en las organizaciones. Con frecuencia se implantan en forma inicial los Sistemas Transaccionales y, posteriormente, se introducen los Sistemas de Apoyo a las Decisiones. Por último, se desarrollan los Sistemas Estratégicos que dan forma a la estructura competitiva de la empresa.

En la década de los setenta, Richard Nolan, un conocido autor y profesor de la Escuela de Negocios de Harvard, desarrolló una teoría que impactó el proceso de planeación de los recursos y las actividades de la informática.

Según Nolan, la función de la Informática en las organizaciones evoluciona a través de ciertas etapas de crecimiento, las cuales se explican a continuación:

- Comienza con la adquisición de la primera computadora y normalmente se justifica por el ahorro de mano de obra y el exceso de papeles.
- Las aplicaciones típicas que se implantan son los Sistemas Transaccionales tales como nóminas o contabilidad.
- El pequeño Departamento de Sistemas depende en la mayoría de los casos del área de contabilidad.
- El tipo de administración empleada es escaso y la función de los sistemas suele ser manejada por un administrador que no posee una preparación formal en el área de computación.
- El personal que labora en este pequeño departamento consta a lo sumo de un operador y/o un programador. Este último podrá estar bajo el régimen de honorarios, o bien, puede recibirse el soporte de algún fabricante local de programas de aplicación.
- En esta etapa es importante estar consciente de la resistencia al cambio del personal y usuario (ciberfobia) que están involucrados en los primeros sistemas que se desarrollan, ya que estos sistemas son importantes en el ahorro de mano de obra.
- Esta etapa termina con la implantación exitosa del primer Sistema de Información. Cabe recalcar que algunas organizaciones pueden vivir varias etapas de inicio en las que la resistencia al cambio por parte de los primeros usuarios involucrados aborta el intento de introducir el ordenador a la empresa.

Etapa de contagio o expansión. Los aspectos sobresalientes que permiten diagnosticar rápido que una empresa se encuentra en esta etapa son:

- Se inicia con la implantación exitosa del primer Sistema de Información en la organización. Como consecuencia de lo anterior, el primer ejecutivo usuario se transforma en el paradigma o persona que se habrá que imitar.
- Las aplicaciones que con frecuencia se implantan en esta etapa son el resto de los Sistemas Transaccionales no desarrollados en la etapa de inicio, tales como facturación, inventarios, control de pedidos de clientes y proveedores, cheques, etc.
- El pequeño departamento es promovido a una categoría superior, donde depende de la Gerencia Administrativa o Contraloría.
- El tipo de administración empleado está orientado hacia la venta de aplicaciones a todos los usuarios de la organización; en este punto suele contratarse a un especialista de la función con preparación académica en el área de sistemas.
- Se inicia la contratación de personal especializado y nacen puestos tales como analista de sistemas, analista-programador, programador de sistemas, jefe de desarrollo, jefe de soporte técnico, etc.
- Las aplicaciones desarrolladas carecen de interfases automáticas entre ellas, de tal forma que las salidas que produce un sistema se tienen que alimentar en forma manual a otro sistema, con la consecuente irritación de los usuarios.
- Los gastos por concepto de sistemas empiezan a crecer en forma importante, lo que marca la pauta para iniciar la racionalización en el uso de los recursos computacionales dentro de la empresa. Este problema y el inicio de su solución marcan el paso a la siguiente etapa.

Etapa de control o formalización. Para identificar a una empresa que transita por esta etapa es necesario considerar los siguientes elementos:

- Esta etapa de evolución de la Informática dentro de las empresas se inicia con la necesidad de controlar el uso de los recursos computacionales a través de las técnicas de presupuestación base cero (partiendo de que no se tienen nada) y la implantación de sistemas de cargos a usuarios (por el servicio que se presta).

- Las aplicaciones están orientadas a facilitar el control de las operaciones del negocio para hacerlas más eficaces, tales como sistemas para control de flujo de fondos, control de órdenes de compra a proveedores, control de inventarios, control y manejo de proyectos, etc.
- El departamento de sistemas de la empresa suele ubicarse en una posición gerencial, dependiendo del organigrama de la Dirección de Administración o Finanzas.
- El tipo de administración empleado dentro del área de Informática se orienta al control administrativo y a la justificación económica de las aplicaciones a desarrollar. Nace la necesidad de establecer criterios para las prioridades en el desarrollo de nuevas aplicaciones. La cartera de aplicaciones pendientes por desarrollar empieza a crecer.
- En esta etapa se inician el desarrollo y la implantación de estándares de trabajo dentro del departamento, tales como: estándares de documentación, control de proyectos, desarrollo y diseño de sistemas, auditoría de sistemas y programación.
- Se integra a la organización del departamento de sistemas, personal con habilidades administrativas y preparadas técnicamente.
- Se inicia el desarrollo de interfases automáticas entre los diferentes sistemas.

Etapa de integración. Las características de esta etapa son las siguientes:

- La integración de los datos y de los sistemas surge como un resultado directo de la centralización del departamento de sistemas bajo una sola estructura administrativa.
- Las nuevas tecnologías relacionadas con base de datos, sistemas administradores de bases de datos y lenguajes de cuarta generación, hicieron posible la integración.
- En esta etapa surge la primera hoja electrónica de cálculo comercial y los usuarios inician haciendo sus propias aplicaciones. Esta herramienta ayudó mucho a que los usuarios hicieran su propio trabajo y no tuvieran que esperar a que sus propuestas de sistemas fueran cumplidas.
- El costo del equipo y del software disminuyó por lo cual estuvo al alcance de más usuarios.
- En forma paralela a los cambios tecnológicos, cambió el rol del usuario y del departamento de Sistemas de Información. El departamento de sistemas evolucionó hacia una estructura descentralizada, permitiendo al usuario utilizar herramientas para el desarrollo de sistemas.

- Los usuarios y el departamento de sistema iniciaron el desarrollo de nuevos sistemas, reemplazando los sistemas antiguos, en beneficio de la organización.

Etapa de administración de datos. Entre las características que destacan en esta etapa están las siguientes:

- El departamento de Sistemas de Información reconoce que la información es un recurso muy valioso que debe estar accesible para todos los usuarios.
- Para poder cumplir con lo anterior resulta necesario administrar los datos en forma apropiada, es decir, almacenarlos y mantenerlos en forma adecuada para que los usuarios puedan utilizar y compartir este recurso.
- El usuario de la información adquiere la responsabilidad de la integridad de la misma y debe manejar niveles de acceso diferentes.

Etapa de madurez. Entre los aspectos sobresalientes que indican que una empresa se encuentra en esta etapa, se incluyen los siguientes:

- Al llegar a esta etapa, la Informática dentro de la organización se encuentra definida como una función básica y se ubica en los primeros niveles del organigrama (dirección).
- Los sistemas que se desarrollan son Sistemas de Manufactura Integrados por Computadora, Sistemas Basados en el Conocimiento y Sistemas Expertos, Sistemas de Soporte a las Decisiones, Sistemas Estratégicos y, en general, aplicaciones que proporcionan información para las decisiones de alta administración y aplicaciones de carácter estratégico.
- En esta etapa se tienen las aplicaciones desarrolladas en la tecnología de base de datos y se logra la integración de redes de comunicaciones con terminales en lugares remotos, a través del uso de recursos computacionales.

Enterprise Resources Planning

5.-Enterprise Resources Planning (ERP)

5.1.- Introducción

5.2.- Definición

5.3.- Estructura de un ERP

5.3.1.- El sistema básico de un ERP

5.3.2.- Módulos de gestión de compras

5.3.3.- Módulo de producción

5.3.4.- Módulos de ventas

5.3.5.- Módulo de finanzas

5.3.6.- Módulo de recursos humanos

5.3.7.- Módulo de gestión de medios técnicos y mantenimiento

5.4.- Características generales de un ERP

5.4.1.- Capacidad de personalización

5.4.2.- Adaptación a la estructura de la empresa

5.4.3.- Interfaz de usuario avanzada y flexible

5.4.4.- Integración con otras aplicaciones

5.4.5.- Capacidad de acceso a información

5.4.6.- Otras características

5.5.- El ERP como cadena de valor extendida

5.6.- Evolución histórica de los sistemas ERP: de la gestión de materiales a la empresa digital

5.6.1.- Antecedentes del software de gestión

5.6.2.- Primera etapa: la gestión informatizada de las listas de materiales (BOM)

5.6.3.- La gestión de necesidades de material: el MRP

5.6.4.- El MRP a ciclo cerrado: la gestión de cargas y capacidades

5.6.5.- El MRP II: la gestión de recursos de fabricación

5.6.6.- ERP: planificación de recursos de empresa

5.6.7.- SCM: la gestión de la cadena de suministros

5.6.8.- Los retos actuales: CRM Y PLM

5.7.- Situación de los ERPs en la empresa

5.8.- Proceso de implantación de un ERP

5.8.1.- Introducción

5.8.2.-Aspectos a considerar

5.8.2.1.- Expectativas generadas ante la implantación de un ERP

5.8.2.2.-Costes asociados a la implantación de un ERP

5.8.2.3.- Componentes de una implantación de un ERP

5.8.2.4.- Fases de una implantación.

5.8.2.5 Ventajas e Inconvenientes de su implantación

5.8.3.- Enfoque metodológico

5.8.3.1.- Análisis de la situación actual

- 5.8.3.2.- Análisis de Requisitos
- 5.8.3.3.- Identificación Alternativas
- 5.8.3.4.-Selección Alternativa
- 5.8.3.5.- Planificación Implantación
- 5.8.4.- Elección del ERP: Parámetros

5.1.- Introducción

Los llamados Enterprise Resources Planning (ERP) o traducido, sistemas de Planificación de Recursos Empresariales han sido sistemas claves para la mejora de la efectividad de las organizaciones. En el presente apartado trataremos de esclarecer algunos conceptos y exponer consideraciones generales a tener en cuenta al momento de seleccionar soluciones de este tipo.

Todas las definiciones de ERP concuerdan en que se trata de sistemas de información empresariales que integran y automatizan procesos de negocio, a lo largo de toda la cadena de suministros y en todas las funciones de la organización. Estos sistemas prometen una alta integración de la información clave en una sola base de datos, una sola aplicación y una interfaz unificada. En general, se habla de integración, de procesos de negocios, de la cadena de suministros, del “backbone” de información, etc.

Los retos de disponer de sistemas apropiados a la nueva economía afectan a todas las empresas, las cuales deben responder a las oportunidades que nacen en la nueva tecnología o bien desaparecerán.

Todas las empresas necesitarán la actualización de sus infraestructuras empresariales y cambiar el modo en el que trabajan para responder a las necesidades de los clientes. Las infraestructuras internas existentes de las empresas de hoy representan una gigantesca inversión en tecnología, en formación, en investigación en la ingeniería de los negocios que, en algunos casos, ha estado funcionando incluso cientos de años. En los últimos 15 años, esta inversión ha contribuido a unas mejoras de eficiencia que han sido las más importantes desde que ese inventó el ordenador hace 50 años.

Las empresas que triunfarán serán aquellas que se apoyen en esta inversión, con inversiones en sistemas ERP, en soluciones E-Business que funcionen de forma totalmente integrada.

5.2.- Definición

La Planificación de Recursos Empresarial (ERP), es mucho más que simplemente "Software de gestión", permite la integración y optimización de todos los procesos y recursos de la organización.

Los ERPs se inician en los años 70, cuando Gartner Group y AMR dan el primer paso para definir las bases de las ERP. En un principio las organizaciones estaban orientadas a las funciones y sólo se cubría las áreas de manufactura y finanzas. La información generada por las empresas era procesada por medio de Mainframe's.

Posteriormente, en los años 80, se cambio el paradigma de estar orientados a funciones y se orientó hacia los procesos, por medio del cual se identificaron los procesos críticos de las empresas y su integración entre sí. La información comenzó a utilizar el modelo Cliente - Servidor.

Una de las mayores fortalezas de las soluciones ERP es su poder integrador de diferentes procesos como son: Logística, Inventarios, Compras, Ventas, Distribución, Producción, Recursos Humanos, entre otros.

Asimismo, existen varios autores que explican que una ERP no es un proceso evolutivo de dicha metodología, tal es el caso de Don Ralston y David Turbide que explican respectivamente:

"El ERP no es un paso mayor, pero si una expansión del MRP II, con nuevas aplicaciones a través de la incorporación de tecnología moderna a la empresa, como el concepto Cliente-Servidor y EDI (Intercambio de Información Electrónica), que han provisto de mayor alcance al uso de la misma filosofía." (Ralston, citado por Camacho, Salvador. (1997). Planificación de Recursos Empresariales.)

"... es un intento por renombrar al MRP II sin ningún cambio real en su naturaleza, mas bien, es una extensión que incorpora tecnología moderna." (Turbide, citado por Camacho, Salvador. (1997). Planificación de Recursos Empresariales.)

"ERP representa un amplio espectro de funciones que intenta abarcar todas las entidades de una empresa. Requiere de la profundidad organizacional y funcional de una gran variedad de empresas de manera que se pueda examinar y modificar un concepto de empresa único" (Gartner Group. E.R.P. vendor Guide 1997).

"Solución de software que se enfoca a las necesidades de la empresa, tomando una visión de los procesos para cumplir todos los objetivos corporativos, buscando integrar todas las funciones de la empresa." (Hernández, José Antonio. (1999). SAP R/3. Ed. Mc GrawHill)

Cuando una empresa decide adoptar una ERP, toda su información queda integrada en el sistema, por medio de los módulos: Logística, distribución, inventarios, compras, ventas, recursos humanos, producción entre otros. (Revista Dinero, Octubre de 1999)

Los beneficios aportados por una solución ERP están basados en mayor productividad, información integrada y a tiempo para una mejor toma de decisiones. (PC MAGAZINE, Noviembre de 1999)

Por otra parte, Glovia International describe una ERP como:

"Un Sistema de planificación diseñado para reducir el tiempo de respuesta, ciclo de producción, optimizar calidad, mejorar el manejo de activos, reducir los costos, optimizando la comunicación" (Glovia, citado por Camacho, Salvador. (1997). Planificación de Recursos Empresariales.).

Por último, el entendimiento de lo que es ERP, desde la visión del grupo investigador de la Universidad Javeriana, lo define como:

"...Conjunto de Aplicaciones empresariales para compañías manufactureras, que permiten balancear funciones operacionales dispersas, como lo son finanzas, manufactura y producción... De esta manera, ERP es visto como un sistema para la planificación, control y operación total de una empresa..." (Torres, Diego; Nieto Raúl y Silva, Jorge. (2001). Aplicación de Sistemas Integrados ERP en el contexto Empresarial Colombiano)

Después de analizar cada una de las definiciones encontramos como elementos comunes: solución, necesidades de la empresa, visión y optimización de los procesos, integrar, planificar, reducir tiempo, toma de decisiones y control, que son rasgos preponderantes en este tipo de soluciones. Con lo que podremos definir un ERP anexando las palabras clave anteriormente citadas como una solución necesaria para la empresa, tomando una visión y optimización de los procesos, a través de su integración y planificación en el sistema podremos lograr una reducción de tiempos y ayudar a la toma de decisiones a través del control de los factores dependientes en una empresa.

Con lo planteado anteriormente, se aclara el entorno y se entiende con claridad las implicaciones de implantar en una organización un proyecto de Planificación de Recursos Empresariales (ERP), como también las empresas grandes que manipulan un gran volumen de datos, usuarios y complejidad de transacciones escogen este tipo de soluciones, porque establecen que el aumento en el rendimiento de las actividades corporativas y mejora en el servicio al cliente, calidad de la producción, provocan una mejora en la imagen corporativa, manejo de la integración de procesos y calidad de los mismos.

5.3.- Estructura de un ERP

5.3.1.- El sistema básico de un ERP

La mayoría de los ERPs adoptan una estructura modular que soporta los diferentes procesos de una empresa: el modulo de gestión financiera, el módulo de compras, el módulo de gestión de ventas, el módulo de recursos humanos, etc.

Todos estos módulos están interconectados y comparten una base de datos común, garantizando de este modo la coherencia e integración de los datos generados.

El hecho de que estos productos sean modulares posibilita la implantación del sistema por etapas, reduciendo el impacto global en la organización al facilitar la transición desde los sistemas anteriores. Normalmente, el primer módulo que se pone en marcha es el financiero y, posteriormente, se van integrando los restantes, dependiendo de las características particulares de cada empresa

El sistema básico del ERP está formado por las aplicaciones técnicas y la arquitectura necesaria para servir de plataforma al resto de los módulos. Proporciona herramientas de administración para controlar tanto el sistema en sí (rendimiento, comunicación con otras aplicaciones y otros sistemas, etc.), como la base de datos que constituye el núcleo del producto.

Las principales plataformas de servidores son Microsoft y UNIX, mientras que las bases de datos más utilizadas son Oracle, Microsoft SQL Server e IBM. Asimismo, la mayor parte de los sistemas ERP disponen de lenguajes de programación propietarios de cuarta generación, que facilitan el desarrollo y adaptación de aplicaciones a la medida de cada cliente.

Por otra parte, las últimas versiones de los ERPs incluyen el soporte a las tecnologías derivadas de Internet, como el estándar XML o el lenguaje de programación JAVA. Cada proveedor de ERP define la modularización de la solución atendiendo a razones comerciales o técnicas. En la tabla siguiente se muestran, a modo de ejemplo, la modularización de dos ERPs: uno de ellos es el paquete R3 perteneciente a la multinacional SAP y el otro es LIBRA, ERP desarrollado por la empresa española EDISA.

Procesos principales	Módulos de SAP R3	Módulos de LIBRA
Gestión Financiera y Control.	Finance. Treasure Management. Enterprise Controlling. Invest Management.	Libra Financiera.
Compras y Logística.	Material Management.	Libra compras. Libra Amacenes.
Ventas y Logística Externa.	Sales and Distribution.	Libra Producción.
Producción.	Production Planning.	Libra Producción.
Gestión de Medios Técnicos.	Plant Manitenance.	Libra Gestión de Medios Técnicos y Mantenimiento.

Gestión de Recursos Humanos.	Human Resources.	Libra Recursos Humanos.
------------------------------	------------------	-------------------------

Tabla 5.1.- Tabla comparativa de modularización del ERP SAP y el ERP LIBRA

Seguidamente, se muestran algunas de las funcionalidades incluidas en los principales módulos que constituyen un sistema ERP.

5.3.2.- Módulos de gestión de compras

El proceso de compras en una empresa comprende la gestión de materiales y la relación con los proveedores.

En el apartado de gestión de materiales el sistema debe dar soporte a la definición de los datos necesarios para el tratamiento de los materiales a lo largo de toda la cadena logística, así como las transacciones realizadas con ellos, facilitando el control de los stocks, la generación de nuevos pedidos, la valoración de inventarios de acuerdo con distintos criterios, etc.

En lo que se refiere al apoyo a la relación de la empresa con los proveedores, el sistema debe proporcionar toda la información sobre precios y condiciones de entrega, historial de compras, disponibilidad, etc., facilitando de este modo el proceso de toma de decisiones de compra.

Asimismo, mediante distintas opciones de análisis, el sistema puede realizar una valoración de los proveedores: cumplimiento de plazos de entrega, estado de los materiales, fiabilidad, etc.

Este módulo se apoya en dos bases de datos fundamentales:

- La base de datos de materiales, que permite registrar para cada referencia su código, descripción, peso, dimensiones, calidad, cantidad en stock, etc.
- La base de datos de proveedores, que almacena los datos sobre cada uno de los proveedores seleccionados: nombre, personas de contacto, dirección de pedido, datos fiscales para facturación, etc., así como precios y condiciones de entrega de los productos que ofrece.

El módulo de gestión de compras facilita la planificación de los pedidos a proveedores a partir de las necesidades de compra de la empresa, que pueden venir determinadas por la demanda de productos terminados o por el control de unos stocks mínimos de producción.

Además, este módulo puede ofrecer la posibilidad de consultar el historial de los proveedores y de los movimientos de materiales que se han realizado.

En definitiva, el módulo de gestión de compras deberá dar soporte a todos los procesos de compra, desde la gestión de proveedores y tarifas hasta el control de los procesos de pedidos, conciliación de facturas y otras fases implicadas en el gestión de compras, tanto de productos como de materias primas, bienes de inversión o servicios, así como la gestión de contratos de suministro.

5.3.3.- Módulo de producción

El módulo de producción se encarga de gestionar los materiales y servicios empleados en la cadena de producción de una empresa, así como los recursos (máquinas, utillaje, personal) utilizados en ésta.

Este módulo facilita la planificación de los materiales y de las capacidades de los recursos, lanzando los órdenes de montaje o de fabricación y adaptándose a las características específicas de los distintos sistemas de fabricación: fabricación contra stock, fabricación a medida contra pedido (build to order) o montaje (únicamente se realiza el ensamblaje final de las distintas piezas que componen el producto).

Para contribuir a una adecuada gestión de los stocks de materiales, este módulo debe estar totalmente integrado con el módulo de gestión de compras. Además, este módulo puede incorporar diferentes funcionalidades adicionales como la planificación a capacidad finita, la captura de datos en planta, la gestión de subcontrataciones, etc.

5.3.4.- Módulos de ventas

El módulo de ventas se ocupa de la relación de la empresa con los clientes, dando soporte a todas las actividades comerciales preventa y postventa.

Asimismo, facilita la gestión y configuración de los pedidos, la logística de distribución, la preparación de entregas, la expedición y el transporte.

Para un correcto funcionamiento, el módulo de ventas deberá estar integrado con los módulos de almacén, logística, módulo financiero, etc. Asimismo, cada vez se exige un mayor nivel de integración entre ventas y compras, reflejo de una progresiva orientación a una operativo “bajo pedido”.

5.3.5.- Módulo de finanzas

El módulo de finanzas se encarga de la contabilidad y de la gestión financiera de la empresa. Se trata de un módulo esencial dentro del sistema ERP, ya que va a estar totalmente integrado con los restantes módulos. Por este motivo, resulta fundamental para la correcta implantación del ERP.

Este módulo proporciona herramientas flexibles y aplicaciones orientadas tanto a la contabilidad financiera, como a la contabilidad analítica o de costes.

Entre sus múltiples funciones relacionadas con la operativa financiera y contable podemos destacar las siguientes:

- Contabilización de las operaciones de la empresa (generación de asientos contables).
- Elaboración de los balances y de la cuenta de resultados.
- Elaboración de presupuestos, generación de informes y análisis de desviaciones.
- Gestión de la tesorería (control de flujos de cobros y pagos, gestión de cuentas corrientes, etc.)
- Gestión de activos.

Asimismo, este módulo proporciona funciones específicas para el departamento de administración de una empresa:

- Facturación
- Liquidación de los impuestos
- Gestión de cobros y reclamación de impagados.

En general todos los sistemas ERP disponen de un gran número de informes financieros y contables estándar e incorporan herramientas de diseño a medida para facilitarles la generación de informes adaptados a las necesidades de cada cliente, como en el caso de la liquidación de impuestos de cada país.

5.3.6.- Módulo de recursos humanos

El módulo de recursos humanos de un ERP permite gestionar la información relacionada con los empleados de una organización (datos personales, formación recibida, experiencia, ocupación, etc.). Entre las múltiples funciones que facilita podemos destacar las siguientes:

- Definición de estructuras organizativas.
- Planificación de las necesidades de personal.

- Soporte al proceso de evaluación y selección de personal.
- Control de presencia, relacionado generalmente con el módulo de producción.
- Soporte a la contratación de personal.
- Gestión de las acciones formativas.
- Registro de gastos de representación y de dietas por desplazamientos.
- Soporte a la generación de nóminas.

5.3.7.- Módulo de gestión de medios técnicos y mantenimiento

Este módulo facilita el control de los recursos materiales y técnicos de la empresa, maquinaria, elementos de transporte y repuestos, integrando las funciones empresariales de compras y mantenimiento para asegurar la disponibilidad de estos recursos en las operaciones empresariales.

5.4.- Características generales de un ERP

A continuación se presentan de forma detallada algunas características comunes a los principales ERPs del mercado:

5.4.1.- Capacidad de personalización (customize)

Se trata de la característica diferencial de los ERPs frente a la mayor parte de las soluciones de gestión orientadas a pequeñas empresas.

La personalización (algunos textos la denominan parametrización, que es un termino no reconocido en la Real Academia de la Lengua española) de un ERP permite adaptar el funcionamiento del sistema a las necesidades concretas de cada empresa así como incorporar nuevas funciones o modos de funcionamiento a medida que la empresa en cuestión lo requiera.

La personalización del ERP exige un gran conocimiento tanto del producto como de las necesidades de la empresa y, por ello este trabajo requiere de un importante esfuerzo de consultoría, que supone un capítulo fundamental en un proyecto de implantación de un ERP.

5.4.2.- Adaptación a la estructura de la empresa

Otra de las características comunes de los ERPs es su capacidad para adaptarse a la estructura organizativa de la empresa, a las funciones asignadas a cada uno de los usuarios, las políticas de venta y de compra, los centros de fabricación, los centros de distribución, los almacenes, las zonas de carga, etc.

5.4.3.- Interfaz de usuario avanzada y flexible

Normalmente, los ERPs incorporan las últimas tecnologías y avances en la interfaz de usuario, con facilidades gráficas o la posibilidad de definir diversos dispositivos de acceso: ordenadores personales, terminales de radiofrecuencia, PDAs, etc.

5.4.4.- Integración con otras aplicaciones

Esta característica facilita la comunicación e intercambio de datos por medio de interfaces estandarizadas con paquetes de software EDI, herramientas de Internet, aplicaciones ofimáticas, soluciones de Business Intelligence, etc.

5.4.5.- Capacidad de acceso a información

Los ERPs cuentan con un conjunto de salidas e informes predefinidos y, además, posibilitan la interacción desde distintas herramientas de acceso a datos: OLAP, aplicaciones ofimáticas, paquetes software DSS o EIS, etc.

5.4.6.- Otras características

Entre estas otras características de los ERPs, podríamos citar la incorporación de herramientas de seguridad, ayuda online, etc.

5.5.- El ERP como cadena de valor extendida

La tecnología basada en la Red mueve la información a lo largo de la cadena de valor logística, uniendo grupos separados previamente, que pueden comunicarse más rápida y eficientemente por teléfono, fax o correo electrónico, que cara a cara. Esta tecnología tiene la capacidad de proporcionar información instantáneamente a un coste muy bajo.

El sistema ERP de cada compañía se conecta directamente a los sistemas ERP de proveedores y clientes. El EDI ofrece un modelo similar, aunque en el modelo EDI cada empresa debe crear protocolos únicos para cada proveedor o cliente. En un modelo basado en Internet con estándares

abiertos, una compañía puede conectar a través de la tecnología basada en la red con cada proveedor y cada cliente.

Debido a que la información es más fácilmente disponible usando la tecnología basada en Internet para conectar ambos, proveedores y clientes, la oportunidad existente para una empresa de crear nuevas estrategias de negocio basadas en transformar una cadena de valor en una red integrada de valor.

La razón para esto reside en los atributos únicos de información:

- La información puede ser consumida múltiples veces.
- La información puede ser condensada.
- El valor de la información cambia con el tiempo y el uso.
- La información abre las puertas.

Cuanto más empresas estén conectadas a la red integrada, mayor valor hay para cualquier otra empresa en conectarse a esta red. A la vez que los socios de la red integrada aprenden a trabajar juntos mejor, el beneficio del cliente se incrementa a la vez que los costes decrecen y los niveles de servicio mejoran.

Esta dinámica llega a ser un círculo vicioso, conduciendo a los miembros de la empresa extendida a mejorar constantemente sus propios procesos internos así como los procesos de la red extendida de empresas. Los miembros de la red integrada se esfuerzan por la eficacia y reducción de costes.

A la vez que las compañías adopten los estándares abiertos que incluyen el ERP dentro de las tecnologías Internet, sus arquitecturas de sistemas cambiarán espectacularmente. La siguiente figura (figura 5.1) muestra la arquitectura del sistema en su estado final para la compañía del siglo XXI.

Arquitectura de sistemas del siglo XXI



Figura 5.1.- Arquitectura de sistemas del siglo XXI

(Fuente: El futuro tecnológico de las Terminales Marítimas de Vehículos: La integración de sus sistemas de información, UPC – Departament de Ciència i Enginyeria Nàutiques Barcelona, 2004)

En el centro está el sistema ERP de la compañía, que es su motor de transacciones y generador de sus datos internos. Estos datos, son almacenados en un Almacén de datos (Data Warehouse), pueden ser cortados y analizados en cualquier número de formas por el software que soporta las decisiones de la compañía, y que esta utiliza para realizar sus análisis de negocio.

Con la tecnología basada en Internet, la compañía puede transferir información hacia y desde sus clientes, proveedores y socios del negocio. En suma, la tecnología permite a la empresa utilizar fuentes de investigación externa para sumar calidez y robustez a sus análisis de negocio. La tecnología para soportar las decisiones asiste a los gerentes a todos los niveles de la organización de manera que puedan tomar decisiones coherentes, dándoles una clara imagen de la información relevante desde ambos sitios, dentro y fuera de la compañía. En un sistema como tal, los datos de fuentes internas y externas pueden ser consolidados y comparados

con los objetivos de una compañía como parte del sistema de medida del rendimiento, convirtiendo los datos en información de gestión.

5.6.- Evolución histórica de los sistemas ERP: de la gestión de materiales a la empresa digital

5.6.1.- Antecedentes del software de gestión

Los primeros ordenadores fueron fruto de grandes proyectos de desarrollo tecnológico desarrollados durante la segunda guerra mundial para cubrir necesidades de cálculo militares (generación de tablas balísticas, investigación de los procesos de fisión nuclear, etc.). Estas primeras máquinas eran demasiado caras para ser utilizadas en la industria, pero generación tras generación de computadoras, la tecnología fue mejorando, aumentando la velocidad y capacidad de cálculo y disminuyendo los costes como en ningún otro sector industrial.

En la década de los 50 los ordenadores comienzan a expandirse por las universidades y ya en 1955 se crea la asociación SHARE (Society to Help Allieve Redundant Effort, primer grupo de usuarios de ordenadores) para compartir conocimientos y evitar en la medida de lo posible labores redundantes (History of Computing, Computer Society, 2003). A finales de esta década, los ordenadores para uso industrial comienzan utilizarse en el entorno empresarial (An Overview of the History of the Software Industry, Software History Center, 2003).

A comienzos de los 60 se fundan numerosas empresas dedicadas al desarrollo de software. En esta época, la práctica habitual es incluir el software básico gratis con la venta del hardware, teniendo que contratar desarrollos a medida para cubrir cualquier otra necesidad. De todas formas, se empiezan a crear las primeras librerías de utilidades, en las que se pueden conseguir ciertas aplicaciones gratuitamente.

En este caldo de cultivo, van surgiendo los primeros intentos de aplicar la tecnología a la problemática de gestión de materiales y en 1959 Bosch desarrolla una aplicación que puede considerarse la primera aproximación a lo que posteriormente se conoció como Material Requirement Planning (MRP) o Planificación de Necesidades de Materiales.

El concepto de software como producto comienza a considerarse viable comercialmente y en 1967, la compañía International Computer Programs, Inc. (ICP) crea el primer catálogo de software con 49 aplicaciones (Software History Center, 2003). Como fecha significativa, cabe citar que IBM anuncia que a partir del uno de enero de 1970 ciertos paquetes de software iban a comenzar a venderse por separado, dando por finalizada la era en la que el

software se consideraba un derecho ilimitado inherente a la compra del hardware.

5.6.2.- Primera etapa: la gestión informatizada de las listas de materiales (BOM)

Las prácticas de gestión utilizadas en los años 60, se basaban en los modelos tradicionales de punto de pedido y lote económico de compra. La disponibilidad comercial de computadoras propició el inicio de una nueva era del procesamiento de la información de negocios, con un impacto profundo de las nuevas tecnologías en la dirección de operaciones. Probablemente, en ningún área ha supuesto un impacto mayor (al menos potencialmente) que en el área de logística de fabricación, por ejemplo en la gestión de inventarios y en la planificación de la producción (Orlicky, Joseph (1975), MRP, *The New Way of Life in Production and Inventory Management*. McGraw-Hill Book Company). Hasta la llegada de la computadora, estas funciones constituían un problema crónico e intratable para todas aquellas empresas que se dedican a la fabricación de productos que requieren múltiples etapas en su proceso de transformación. Las soluciones conocidas y disponibles eran imperfectas, parciales y generalmente insatisfactorias desde el punto de vista de gestión.

Las primeras aplicaciones informáticas, hacia 1960, orientadas a la gestión de inventarios, representaron el comienzo de la ruptura con la tradición. La disponibilidad de computadoras, capaces de manejar un gran volumen de información a velocidades previamente inimaginables, supuso la eliminación de las fuertes restricciones relacionadas con el procesamiento de la información y la súbita obsolescencia de muchos métodos y técnicas desarrollados en base a estas restricciones. Los planteamientos tradicionales en los días previos a las computadoras, no podían ir más allá de los límites impuestos por las herramientas. Debido a esto, casi todas aquellas técnicas eran imperfectas.

Funcionaban a modo de muleta e incorporaban métodos aproximados, a menudo basados en asunciones poco realistas, otras veces forzando la aplicación de conceptos a la realidad para poder utilizar las técnicas.

El salto cualitativo, en esta área, radica en el simple hecho de que una vez que se dispone de un ordenador, el uso de dichos métodos y sistemas ya no es obligatorio. Es posible evitar, revisar o descartar las técnicas previas e instaurar nuevas que hasta el momento había sido imposible utilizar.

Analizando los casos de las compañías pioneras en la gestión computerizada de inventarios (años 60), puede verse que los mejores resultados no fueron obtenidos por aquellos que eligieron mejorar, refinar y acelerar las técnicas existentes, sino por aquellos que plantearon una

completa revisión de sus sistemas. En este contexto, surgen los primeros sistemas que tratan la gestión de demanda dependiente, es decir, la gestión de productos cuya descomposición implica que la cantidad demandada de un componente depende de las cantidades demandadas de todos los productos finales en los que toma parte. Estos primeros intentos, basados en iniciativas de empresas individuales y con las carencias propias de la falta de experiencia previa y por lo tanto la inexistencia de metodologías estandarizadas, son catalogadas hoy en día bajo la denominación de gestores de listas de materiales o gestores del BOM (Bill Of Materials).

En el área de gestión de inventario industrial, las innovaciones más exitosas están englobadas en lo que se ha dado a conocer como sistemas MRP (Material Requirements Planning o Planificación de Necesidades de Materiales).

5.6.3.- La gestión de necesidades de material: el MRP

Joseph A. Orlicky está considerado como el padre del MRP moderno. En la siguiente figura (figura 5.2) se muestra el diagrama de definición del sistema MRP de su obra “MRP, The New Way of Life in Production and Inventory Management “(1975).

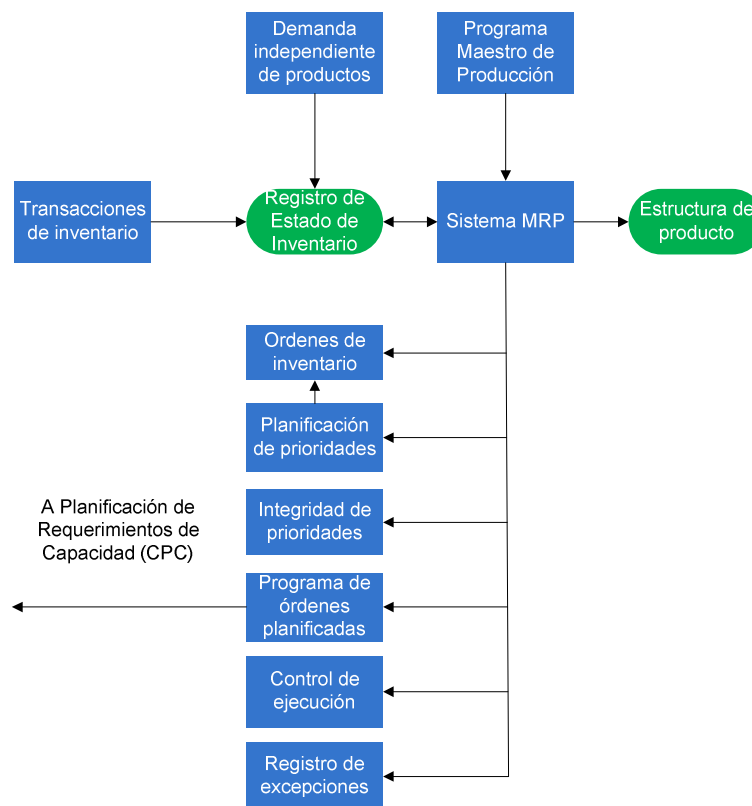


Figura 5.2.- Diagrama de definición del MRP

Fuente: MRP, The New Way of Life in Production and Inventory Management, Joseph A. Orlicky

Según la definición de Orlicky, el MRP consiste en una serie de procedimientos, reglas de decisión y registros diseñados para convertir el Programa Maestro de Producción en Necesidades Netas para cada Periodo de Planificación.

El objetivo con el que se desarrolló la metodología MRP, fue sustituir los sistemas de información tradicionales de planificación y control de la producción (Cooper, R.B. y Zmud, R.W. (1990), “Information technology implementation research: a technological diffusion approach”).

Las dos hipótesis de base de los sistemas MRP son las siguientes (Orlicky, Buffa, E.S. y Miller, J.G. (1979), “Production-Inventory Systems Planning and Control”, 3rd ed., Richard D. Irwin, Homewood, IL):

- La planificación y el control de la producción no dependen de los procesos.
- Los productos terminados son determinísticos.

Es decir, el sistema MRP está construido alrededor del BOM (lista de materiales) y su validez depende de la exactitud del mismo (Chung S.H. y Snyder C. A. (2000) “ERP adoption: a technological evolution approach”. International Journal of Agile Management Systems 2/1). Según George Plossl, uno de los padres del MRP, «el MRP calcula qué necesito, lo compara con lo que tengo y calcula qué voy a necesitar y cuándo». Este es el verdadero avance del MRP I: por primera vez la planificación de necesidades de materiales es capaz de dar respuesta al CUÁNDO (Ptak, C.A. y Schragenheim, E. (2000), “ERP: Tools, Techniques, and Applications for Integrating the Supply Chain”, CRC Press-St Lucie Press).

Debido a las limitaciones de capacidad de cálculo de los ordenadores de la época, la metodología MRP I asume ciertas simplificaciones. Para realizar estos cálculos, los órdenes se planifican sobre la última fecha posible para así minimizar el stock. Este método de programación hacia atrás provoca que al no disponer de tiempos de sobra, todas las actividades forman parte del camino crítico. Así pues, al no disponer de margen para recuperar el tiempo perdido, cualquier retraso o problema causa inevitablemente un retraso en la entrega al cliente. Esta limitación del sistema condujo a definir tiempos de entrega holgados para prevenir los efectos negativos de los pequeños problemas ocasionales.

5.6.4.- El MRP a ciclo cerrado: la gestión de cargas y capacidades

Una vez asumidos los conceptos propuestos por la metodología MRP I, resulta evidente que no es sólo necesario calcular los lanzamientos con una antelación más o menos holgada. También es necesario calcular si se dispone de suficiente capacidad para realizar la tarea planificada. La idea básica es cerrar el ciclo de planificación con una comparación entre la carga de trabajo propuesta para un periodo y la capacidad productiva de los recursos involucrados en los procesos, de modo que el nuevo sistema recibió el nombre de «MRP a ciclo cerrado». La siguiente figura (figura 5.3) muestra un esquema del concepto.

Gracias a la introducción de los cálculos de las cargas de trabajo por máquina o por centro de trabajo, fue posible prever con la suficiente antelación conflictos de exceso de trabajo, de modo que la planificación pasó a ser una labor proactiva, consistente en alisar los excesos de carga de trabajo, adelantando para ello la cantidad mínima de pedidos necesaria. El ciclo cerrado supuso un gran paso adelante en el proceso de planificación de necesidades de materiales y de recursos.

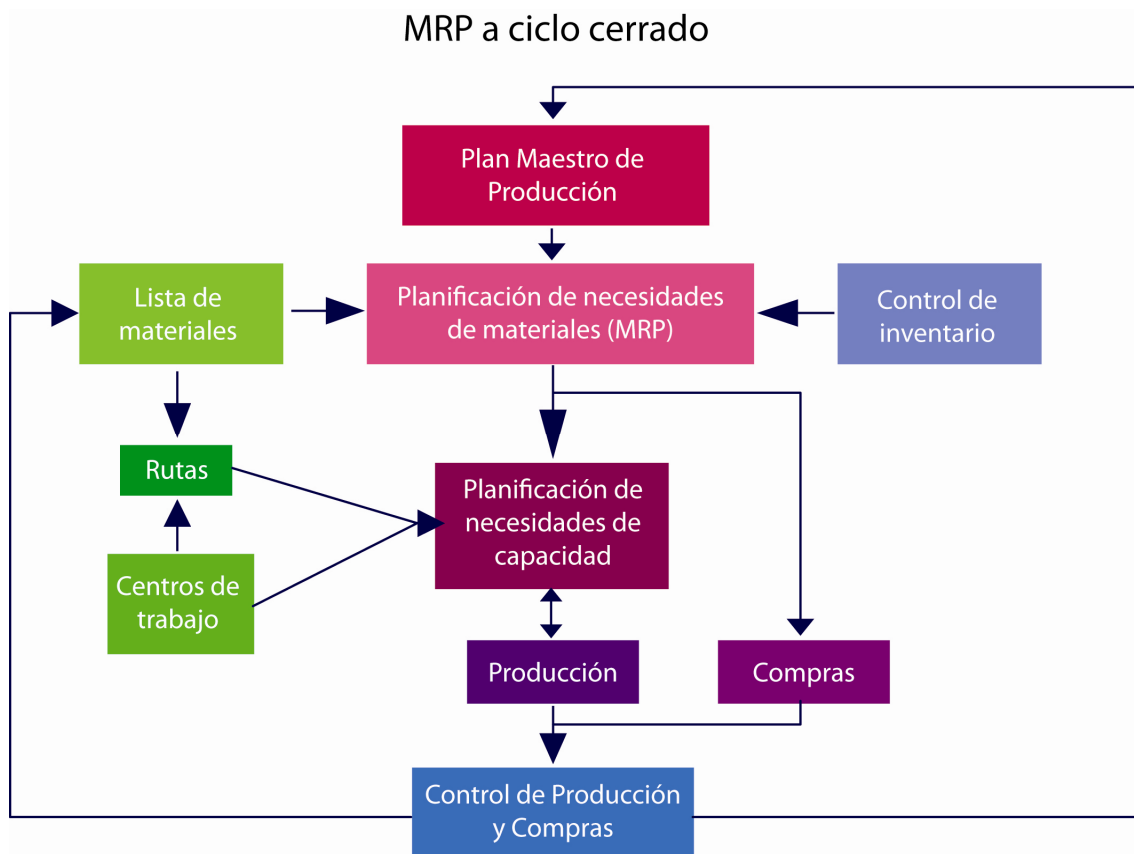


Figura 5.3.- MRP a ciclo cerrado.

Fuente: Delgado, J. y Marín, F.: “Evolución de los sistemas de gestión de materiales: del MRP al erp”, Economía industrial

5.6.5.- El MRP II: la gestión de recursos de fabricación

Tras integrar compras con fabricación, el siguiente paso fue integrar la información financiera. La gestión de materiales tiene una vertiente puramente logística, es decir, la mera necesidad de disponer del material suficiente en el momento apropiado para realizar una tarea. Este mismo material, sin embargo, supone un nuevo activo en el balance de la empresa y una deuda pendiente con el proveedor. Tirando del mismo hilo lógico de razonamiento, el resultado de la planificación del taller se convierte en el trabajo realizado por los operarios y los recursos productivos, por lo que las horas de trabajo empleadas en la transformación de las piezas suponen un coste que puede ser directamente imputado al material en curso. Estas mismas tareas implican la disminución de los stocks de materias primas y el aumento de productos terminados, por lo que el capítulo de existencias de contabilidad de la empresa debe variar a medida que se procesan los órdenes de trabajo.

Este concepto de sistema de información que integre producción inventario y finanzas, fue bautizado por Ollie Wight como MRP II, siendo las siglas las mismas que en el caso de su antecesor (el MRP I) pero cambiando las palabras Material Requirement Planning por Manufacturing Resource Planning (Ptak, C.A. y Schragenheim)

En esta familia de aplicaciones, se realizaron intentos de automatizar la toma de decisiones de modo que los conflictos carga-capacidad fueran resueltos por el ordenador en base a una serie de criterios pre-establecidos. Este tipo de enfoques, en los que se propugna la *toma automática de decisiones* por el sistema, ha provocado en ocasiones el rechazo a los sistemas MRP como consecuencia de lo que se conoce como *nerviosismo del MRP*: una excesiva sensibilidad en las acciones a emprender o modificar ante cualquier pequeño cambio en las condiciones de contorno (Delgado, J. y Marín, F.: “Evolución de los sistemas de gestión de materiales: del mrp al erp”, Economía industrial).

Por esta razón los sistemas MRP II han estado orientados principalmente a la identificación de los problemas de capacidad que presenta un plan de producción, fundamentalmente mediante la presentación gráfica de la disponibilidad de recursos y el consumo planificado, de forma que el planificador pueda llevar a cabo con facilidad las modificaciones oportunas. Para facilitar, no sólo la ejecución de medidas correctoras, sino la evaluación conjunta de diferentes acciones y su comparación con otras alternativas, los sistemas MRP II suelen ofrecer la posibilidad de analizar diferentes escenarios, respondiendo a preguntas del tipo «qué pasa si...». Posteriormente, puede hacerse efectivo el plan de producción que resulte más satisfactorio entre todos los planteados.

De todos modos, no existen grandes diferencias conceptuales entre el MRP II y el MRP a ciclo cerrado. Más que diferencias, puede decirse que se

trata de evoluciones y mejoras en aspectos como la información tratada, las herramientas informáticas disponibles y la mayor divulgación de las buenas prácticas empresariales. En este terreno debe mencionarse la labor de divulgación realizada por la APICS (American Production and Inventory Control Society). Durante los años 70 y 80, esta asociación llevó a cabo la denominada *Cruzada del MRP*, con el objetivo promover el cambio de los modelos de gestión de materiales en las empresas.

El diccionario de la APICS define el MRP II como *un método para la planificación efectiva de todos los recursos de una compañía de fabricación*.

La necesidad de este tipo de herramientas se vio reforzada por la evolución en las exigencias del mercado, debido a la creciente importancia del plazo de entrega y de la amplitud de gama como factores competitivos. En este escenario, las compañías se vieron obligadas a replantear sus sistemas productivos y a implantar modelos de fabricación *Just in Time*. Atrás quedaba el modelo de mejora tradicional basado en la automatización de procesos. En los años 40 y 50 entre un 40% y un 60% de los costes empresariales estaban relacionados con la mano de obra; a principios de los 90 muchas compañías se encontraron con una situación en la que los costes de materiales suponían entre un 60% y un 70% de sus costes, mientras que el coste de mano de obra bajaba a un 10 o un 20% (Ptak, C.A. y Schragenheim)

5.6.6.- ERP: planificación de recursos de empresa

La creciente importancia del plazo de entrega tuvo implicaciones más allá del departamento de producción. La “departamentalización” de las organizaciones supuso uno de los mayores obstáculos para lograr el servicio y los tiempos de respuesta reclamados por los clientes. Un sistema de información común a los diferentes departamentos de la empresa se convirtió en un requisito indispensable para dar respuestas coordinadas.

A diferencia de la evolución de conceptos tratada hasta el momento, el salto del concepto de MRP II al concepto de ERP no es una mera ampliación de las áreas departamentales cubiertas. Se trata de establecer un sistema de información que funcione como columna vertebral de las decisiones tomadas en la empresa. Según Delgado y Marín (2000), una de las principales claves para entender la expansión de los sistemas integrados es la difusión de la cultura RP (Resource Planning) en la empresa, es decir, la cultura de trabajo en base a una planificación de las necesidades de recursos previa y un control de la evolución del consumo de recursos.

Otro aspecto en el que inciden las aplicaciones ERP es la gestión por procesos.

En la medida que el sistema de información es la plataforma desde la que se gestiona el proceso, el sistema de información es también quien define cómo

debe ser dicho proceso (qué información debe introducirse, que personas deben ser informadas, qué orden lógico debe seguirse, etc.). En cierta medida, el sistema de información puede ser la mejor herramienta para modificar un proceso y para introducir mejoras en el mismo.

Así pues, la filosofía de base de los ERPs es la de ser el soporte de gestión de la empresa en su conjunto y no simplemente la extensión del modelo de gestión de la producción a otros departamentos. La mejor prueba de esto es que las aplicaciones ERP ya no sólo están destinadas a compañías en las que la fabricación es el punto fuerte, sino que han sido implantadas en todo tipo de empresas.

5.6.7.- SCM: la gestión de la cadena de suministros

Una característica destacable de la evolución empresarial en los años 90 ha sido la creciente importancia de la externalización de las operaciones en las que la empresa no esté especializada. La aplicación de esta filosofía a la producción ha supuesto que los proveedores hayan absorbido una parte importante de las operaciones productivas.

Por otro lado, factores ya mencionados como el acortamiento de los plazos de entrega y la necesidad de mantener una gama muy alta de producto (o incluso un producto individualizado para cada cliente) también impulsan la necesidad de una coordinación cada vez mayor con clientes y proveedores, provocando un cierto *desgaste* del término ERP. A modo de ejemplo, se puede mencionar que la consultora Gartner Group, mediante la publicación de un artículo con un título tan descriptivo como “ERP Is Dead - Long Live ERP II” (Bond B, Genoves Y., Miklovic D, Wood N., Zrimsek B y Rayner N, del mismo título anterior, 2000), remarcó la necesidad de adoptar sistemas de información capaces de cubrir las necesidades de la empresa extendida mediante la gestión de las cadenas de suministro o Supply Chain Management y por lo tanto superar el concepto que ella misma acuñó en los años 90. Gracias a las nuevas tecnologías de la comunicación y a estándares como EDI o XML, la información fluye entre los sistemas de información de las distintas empresas y es posible un funcionamiento coordinado y ágil.

A modo de resumen, la siguiente figura (figura 5.4) representa la evolución de los sistemas de gestión empresarial como un crecimiento concéntrico, en el que cada nuevo concepto engloba y extiende el anterior.

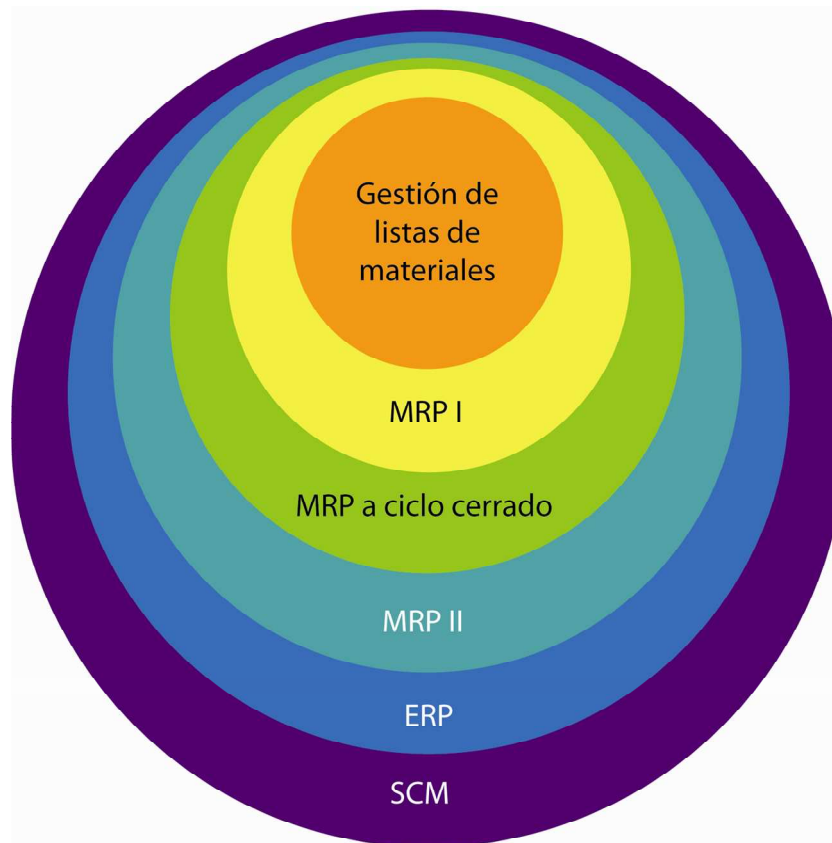


Figura 5.4.- Evolución de los sistemas de gestión empresarial
Fuente: Ptack y Schragenheim, 2000

5.6.8.- Los retos actuales: CRM Y PLM

En la actualidad, los sistemas de gestión empresarial descritos conviven y compiten con otros sistemas de información. De entre las diversas soluciones que ofrece el mercado, merece la pena destacar dos: el CRM y el PLM.

El CRM (Customer Relationship Management) es ante todo una estrategia y una modalidad operativa que tiene como objetivo mejorar y extender las relaciones con el cliente, generando nuevas oportunidades de negocio del que se hablará con mayor detalle en puntos posteriores. La implantación de un sistema CRM, afecta hoy día sobre todo a los puntos de contacto con el cliente dentro de la empresa en las áreas de ventas, marketing, servicios de atención al cliente y en un segundo plano a gestión de los pedidos, distribución y logística (Díaz de Basurto Uruga, Pablo. “BEDI Conocimiento y Satisfacción de los Clientes en la Empresa Digital Extendida y basada en el conocimiento”, 2004).

Es en estas últimas áreas donde surgen mayores solapamientos de funciones entre sistemas ERP y CRM, de modo que las empresas distribuidoras de uno y otro tipo de software defienden la idoneidad de su producto para gestionar las relaciones con los clientes. Mientras los

defensores de los ERPs destacan las ventajas de disponer de un sistema integrado, los defensores de los CRMs defienden la especialización de este tipo de aplicaciones como fuente de ventaja competitiva de la empresa.

Las aplicaciones utilizadas en los departamentos técnicos (CAD/CAM/CAE) han llevado un proceso paralelo de evolución. Para cubrir las crecientes necesidades de gestión de información técnica, han surgido las aplicaciones de tipo PDM (Product Data Management), orientadas principalmente a las necesidades de la Oficina Técnica (almacenamiento de ficheros, gestión de versiones, búsquedas, gestión de relaciones entre documentos de conjuntos, piezas y planos, control de acceso, etc.). Este concepto inicial ha derivado en un concepto más amplio que bajo las siglas PLM (Product Lifecycle Management) engloba una gestión completa de la información técnica a lo largo de todo el ciclo de vida de producto. La asociación CIMData define así el concepto: «Un planteamiento estratégico de negocio que aplica un conjunto robusto de soluciones de negocio colaborativas para soportar la creación, gestión, divulgación y uso de la información de producto a lo largo de la empresa extendida, desde el concepto hasta el fin de la vida del producto e integrando personas, procesos, sistemas de negocio e información (CIMData, “PLM to PDM: Empowering the Future of Business”, 2002).

Nuevamente nos encontramos con solapamientos de funciones, ya que tanto los sistemas PDM como los sistemas ERP trabajan con la estructura de datos de producto, unos desde un punto de diseño y los otros desde un punto de vista de fabricación (los sistemas PLM, por definición, abordan la perspectiva completa, por lo que contemplan ambos puntos de vista y sirven de puente entre ambos).

La integración adecuada de sistemas CRM basados en tecnología web, sistemas PLM colaborativos y sistemas SCM permiten una completa gestión informática del ciclo de diseño y el ciclo de pedido, cumpliendo así al cien por cien el ideal de Empresa Digital. En la medida que estas sofisticadas empresas aprovechen el uso de estas tecnologías para cumplir mejor las exigencias del mercado, será posible afirmar que la inversión y el esfuerzo realizados se han transformado en ventaja competitiva con respecto a los que no hayan querido o no hayan podido ir tan lejos. Si no, una vez más, se podrá afirmar que lo mejor es enemigo de lo bueno.

5.7.- Situación de los ERPs en la empresa

En la actualidad, las compañías buscan implementar sistemas para que manejen todas las áreas del negocio de tal forma que estén integrados.

Muchas han buscado nuevas herramientas tecnológicas para poder optimizar los procesos operativos internos para así ahorrar costes y ser más eficientes, lo que tiene como consecuencia un mejor posicionamiento y la

atracción o bien conservación de clientes. Los sistemas de ERP forman parte fundamental de las estrategias de las grandes empresas actuales.

En el estudio, “Soluciones ERP en la PYME española”, se desprenden las siguientes conclusiones:

- El 58,9% de las compañías tiene implantada una solución de gestión integrada, siendo mayor el porcentaje en aquellas que tienen decisión corporativa.
- Las principales razones que llevan a una empresa a plantearse la implantación de un ERP son la obsolescencia del sistema anterior (52%) y la ampliación del crecimiento de la empresa (49,6%).
- Una vez tomada la decisión de adquirir una solución ERP, el 45% consultan en primer lugar a su proveedor habitual, el 11,6% recurren a consultores y en un 11,5% de los casos recogen información por medios corporativos.
- La principal característica valorada, a la hora de elegir una solución ERP, fue las prestaciones de dicha solución, con un 72,2%, seguida de otros factores como fiabilidad, seguridad, precio y confianza en el integrador.

Si bien hasta finales de los años noventa la implantación de sistemas ERP se había llevado a cabo en su mayoría en empresas de gran tamaño, desde principios del nuevo milenio está extendiéndose cada vez más a empresas de tamaño mediano y pequeño, mediante el lanzamiento de sistemas más económicos y con tiempos de implantación más cortos.

El principal reto de los sistemas ERP sigue estando en su correcta implantación. No es meramente una cuestión de alta complejidad técnica, sino que suele conllevar un cambio de filosofía empresarial, por lo que muchas veces tiene que ser concebido dentro de un programa de gestión del cambio. De ahí que cada vez más, la implantación de un ERP deja de ser una cuestión de sistemas de información y se convierte en un aspecto de la estrategia de negocio.

Para tomar decisiones racionales acerca de cómo comprometer recursos para implantar un ERP, cualquier compañía necesita conocer su situación inicial así como su estado final deseado.

Teniendo en cuenta esto, se pueden definir cuatro posibles escenarios en los que se puede encontrar una empresa:

1. Carencia de sistemas o “desde cero”; no existen sistemas de información

2. Sistemas no integrados; existen gran número de sistemas de información no integrados, con varias plataformas hardware y sistemas operativos, numerosos programas de aplicación y lenguajes de programación. Se crea la necesidad de interfaces para paliar las limitaciones de acceso a datos y permitir que varios sistemas “hablen” entre sí.
3. ERP limitado a funciones individuales; existen sistemas ERP instalados para operar en un área funcional individual (finanzas, ventas o distribución) de una división o de toda la empresa.
4. ERP integrado para toda la empresa; existen procesos completos de uno a otro extremo y a lo largo de toda la compañía, elementos de datos comunes a través de las unidades de negocio y software de aplicación ERP estandarizado, con un único conjunto de aplicaciones aplicado a través de toda la compañía.

5.8.- Proceso de implantación de un ERP

5.8.1.- Introducción

La implementación de un sistema de ERP, por lo general, es larga y compleja, ya que implica rediseñar los esquemas de trabajo. Su implementación es de alto riesgo, ya que implica complejidad, tamaño, altos costos, un equipo considerable de desarrollo, además de inversión de tiempo. En la mayoría de las empresas, se requiere remplazar la infraestructura existente, lo que implica inversión de capital adicional, especialización y hasta la posibilidad de parar el negocio temporalmente para la implementación, por otra parte es importante señalar que el grado de experiencia de los proveedores es un factor importante para el buen funcionamiento del sistema.

Después de la implementación es importante centrarse en el aseguramiento de la calidad y en la mejora del desempeño, para que así el sistema funcione correctamente a largo plazo. También se debe analizar constantemente el retorno de inversión y aspectos clave como la optimización, la cual proporciona ideas que no fueron consideradas durante la implementación como por ejemplo la expansión del software implementado; es importante ver a la optimización como un proceso de mejora continua.

5.8.2.-Aspectos a considerar

5.8.2.1.- Expectativas generadas ante la implantación de un ERP

En la ponencia “Elección de ERP: Criterios y Costes de Implantación de un

ERP” expuesta en el salón Softgest 2003, se enumeran las expectativas que se generan ante la implantación de un ERP:

- Disponer de un sistema integral para todas las áreas de la empresa
- Disponer de información financiera y operativa on-line
- Definición y mejora de procesos y herramientas de control de gestión
- Reducir costes de operación (ROI)
- Incrementar ingresos operativos del negocio (ROI)
- Mejorar eficiencia operativa en todas las áreas (ROI)
- Mejorar la imagen de la empresa
- Mejora en procesos equivale a incremento de la competitividad
- Utilizar el ERP como base para proyectar nuevos negocios
- Implantar un ERP que disponga de herramientas E-Business

5.8.2.2.-Costes asociados a la implantación de un ERP

En la ponencia citada en el punto anterior, también se hace referencia a los costes asociados a la implantación de un ERP, tanto los externos como los internos:

Costes externos (Figura 5.5):

- Infraestructura técnica (hardware, red, comunicaciones).
- Software (licencias, módulos a implantar, actualizaciones).
- Servicios de consultoría, desarrollo, implantación y mantenimiento.

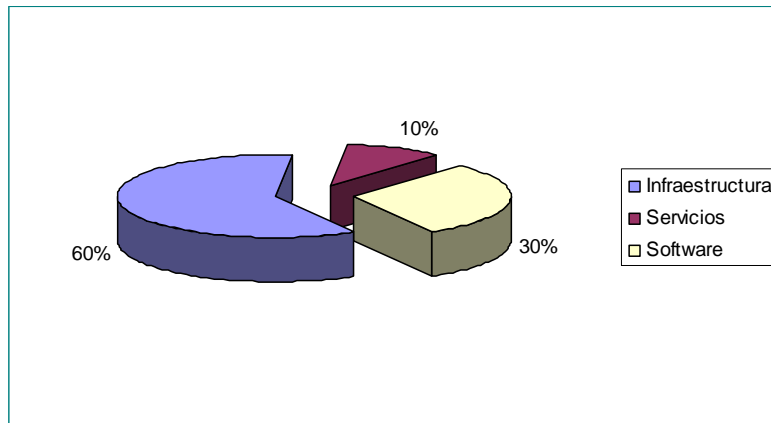


Figura 5.5.- Distribución de costes externos.

Fuente: El futuro tecnológico de las Terminales Marítimas de Vehículos: La integración de sus sistemas de información UPC – Departament de Ciència i Enginyeria Nàutiques Barcelona, 2004.

Costes internos (Figura 5.6):

- Dedicación necesaria por parte de los recursos de la compañía.
- Costes asociados a la aparición del ERP en la empresa.

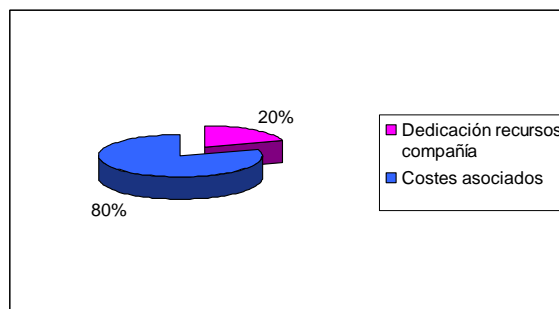


Figura 5.6.- Distribución costes internos

Fuente: El futuro tecnológico de las Terminales Marítimas de Vehículos: La integración de sus sistemas de información UPC – Departament de Ciència i Enginyeria Nàutiques Barcelona, 2004.

5.8.2.3.- Componentes de una implantación de un ERP

Debido a la complejidad asociada a la implantación de un ERP, es importante considerar todos aquellos aspectos que pueden afectar durante dicha implantación:

1. El ERP.

Existen multitud de ERPs, cada uno de ellos con unas características determinadas. Algunos ERPs sirven para cualquier tipo de organización y

para cualquier sector, es lo que se denomina, solución horizontal. También existen ERPs específicos para atender a las necesidades concretas de un sector, es lo que se denomina, solución vertical.

De hecho, hay ERPs que disponen de ambas soluciones: horizontal y vertical. Es importante tener en cuenta que cada organización tiene unas necesidades concretas y que el ERP y su personalización dependerán de estas necesidades, por ello, no existe una implementación “tipo”, lo que en una organización funciona puede no ser válido para otra organización.

2. Las personas y la gestión del cambio.

En función de cómo se enfoque, la gestión del cambio permitirá u obstaculizará el proceso de implantación del ERP.

Por ello, el correcto análisis de los requerimientos de los usuarios e integrarlos desde el primer momento de la implantación es clave para conseguir buenos resultados con el proyecto.

Además, se deben definir exactamente las mejoras que va a obtener cada una de las personas de la organización con la implantación y definir un plan de comunicación para “vender” el proyecto a todas las personas de la organización.

Además, es poco habitual que las organizaciones cuenten con personal con una visión tanto de negocio como de tecnología que consiga liderar el proyecto por lo que el trabajo de consultores externos, y en concreto del director de proyecto, es muy importante.

3. La estrategia.

El proceso “ideal” sería que el plan tecnológico, incluyendo el ERP y su hardware asociado, soporte la estrategia corporativa y no al contrario. Básicamente, la idea es que teniendo perfectamente definida la estrategia de la organización, se asocie a ella los recursos tecnológicos necesarios para que sea posible ejecutarla.

4. El hardware.

Aunque en principio el hardware no es la parte más compleja de la implantación, en algunos casos puede ocurrir que la mala elección del hardware o diseño del sistema haga disminuir el rendimiento global de la implantación.

En este sentido es básico definir exactamente los requerimientos del sistema y así diseñar la solución de manera que no se invierta ni más ni menos de lo necesario.

5. Los procesos.

Se ha de considerar que además de las personas, los procesos son los que definen la eficiencia y eficacia de la organización.

Por ello en el proyecto de implantación de ERP se deben redefinir los procesos para mejorar su eficiencia y eficacia.

El enfoque correcto es redefinir los procesos como un paso previo a la implantación y que los nuevos procesos sean soportados por el ERP.

Sin embargo, lo habitual es encontrar implantaciones de ERPs en los que, tras la implantación, se ejecutan los procesos exactamente igual que antes del ERP. Este es un gran problema ya que no se consigue ninguna mejora en los costes o tiempos de los procesos.

Aunque tengamos el mejor ERP del mundo, si los procesos no se remodelan, seguirán siendo igual de eficientes o ineficientes como lo eran hasta el momento de la implantación y entonces, la implantación del ERP tendrá bajo o nulo impacto en la eficacia y eficiencia.

6. El resto de aplicaciones de gestión existentes en la organización.

Cada vez es más usual que las organizaciones tengan distintas aplicaciones para la gestión. Entre las aplicaciones más habituales están las herramientas propias o sectoriales, las de Gestión de Relaciones con los clientes (CRM), Business Intelligence, Gestión de la cadena de suministro (SCM), etc.

En la mayoría de las ocasiones, todas las aplicaciones han de estar conectadas con el ERP para conseguir una gestión de la información eficiente. Por ello, la integración entre las distintas aplicaciones (EAI) es una tarea cada vez más compleja y que condiciona los resultados finales de la implantación.

5.8.2.4.- Fases de una implantación

En toda implantación de ERP hay dos fases totalmente distintas:

1. La "pre-implantación", es decir, el análisis previo para definir los objetivos del proyecto, alcance funcional, coste total, recursos necesarios,

necesidades concretas de la organización, calendarios, etc. para conseguir evaluar la rentabilidad que supondrá la implantación del ERP.

2. El proyecto propio de implantación incluyendo desarrollos, personalizaciones, formación, etc.

Habitualmente la fase de pre-implantación es infravalorada y en muchas ocasiones ni se realiza este análisis llevando a implantaciones con objetivos poco definidos y con multitud de problemas.

Es habitual encontrar organizaciones que no han desarrollado correctamente el análisis pre-implantación y por tanto no han elegido bien la solución. Por todo ello, el análisis previo debe contener al menos los siguientes apartados:

- a) Análisis inicial de la estrategia, tecnología, procesos, personas y organización. En esta fase, se debe realizar un profundo análisis de la estrategia, personas, procesos y tecnología para así plantear la mejor solución tanto desde el punto de vista tecnológico como de gestión del cambio asociado. En esta etapa se crearán equipos de trabajo para hacer este análisis y para el trabajo posterior.
- b) Definición de objetivos de la implantación del ERP. Claramente, habrán objetivos tangibles (reducción de costes, mejora de eficacia y eficiencia de procesos, reducción del plazo de entrega, reducción de los niveles de inventario, etc.) y otros intangibles como por ejemplo disponer de más cantidad de información y conocimiento para la toma de decisiones. Obviamente, todos estos objetivos deben estar integrados dentro de la estrategia de la organización.
- c) Definición de las mejoras en los procesos y organización que aportará la implantación del ERP. Esto no debe ser una declaración de intenciones sino que se deben haber modelado los procesos de la organización y reconocer el impacto sobre ellos de la implantación del ERP. En esta fase se deben definir objetivos cuantificados de mejora para cada uno de los procesos y deben estar integrados en el calendario del proyecto.
- d) Definición del plan de gestión del cambio para conseguir el cambio de manera no traumática. Dentro de este plan, la comunicación interna es muy importante para "vender" los beneficios del proyecto a los integrantes de la organización, y conseguir que todo el mundo perciba una mejora con el proyecto ERP.
- e) Elección de la solución tecnológica. Así como el implantador más adecuado en función del análisis realizado en la primera fase así como los módulos y personalizaciones necesarias.

- f) Definición de un calendario aproximado y presupuesto asociado. Obviamente esta fase estará directamente relacionada con la fase anterior ya que en función de la elección tecnológica y de los desarrollos anexos, el calendario y el presupuesto variarán.
- g) Definición del retorno de la inversión (ROI). Y los parámetros clave KPI, para definir el seguimiento de la implantación así como un análisis de sensibilidad ante la variación de determinados parámetros.
- h) Implantación del ERP. Seguimiento y control estricto de los objetivos previamente definidos así como de los elementos críticos para la rentabilidad del proyecto. Es muy importante que haya un estricto control del proyecto para que se cumplan los objetivos definidos en las primeras etapas.

5.8.2.5 Ventajas e Inconvenientes de su implantación

Ventajas

Los sistemas ERP, integran los procesos relevantes de una empresa. Las ventajas que ofrece la implementación de un sistema ERP son: control de la operación, eficiencia administrativa, productividad, servicio a clientes, ahorros en costos operativos, visibilidad de las operaciones, soporte a toma de decisiones, preparación para e-business, etc.

Los ERP otorgan a la empresa la posibilidad de reducir sus costos y de ser más competitivas además de tomar ventaja con respecto a su competencia si ésta no cuenta con un sistema como este.

Los sistemas ERP ofrecen un enorme potencial de ahorro tangible e intangible. Entre los primeros destaca la reducción de recursos humanos necesarios y de inventario.

Por otra parte, los ERP aportan un incremento la cantidad y la calidad de información de los consumidores, lo que representa un beneficio intangible muy valorado.

Estos sistemas permiten ver y gestionar la red extendida de la empresa, sus proveedores, alianzas, y clientes como un todo integral. Entre otros beneficios, esto repercute en una mejora de la cadena de procesos, una mayor estandarización y mayor eficacia en la respuesta a los clientes.

Inconvenientes

Por otra parte, la implantación de un ERP, conlleva una serie de inconvenientes que es muy importante valorar y tener en cuenta. Los más relevantes son:

- Hay que tener en cuenta que la implementación suele ser larga, cara y difícil.
- La implementación puede costar varias veces más que la licencia.
- La empresa tiene que adaptar sus procesos al sistema.
- Dependencia de un solo proveedor.
- La fijación de un estándar a veces lleva a adoptar el mínimo común denominador.
- Imponer un sistema ERP desde arriba puede ser un gran error.
- Empresas cambiantes y altamente descentralizadas no deben usar un ERP.
- Algunos proveedores se han especializado solo en ciertas industrias.

5.8.3.- Enfoque metodológico

Permitirá conocer los procesos de negocio de las compañías, identificar los criterios clave de selección de herramientas de gestión, analizar la oferta existente en el mercado, seleccionar aquella solución que mejor se adapta a las necesidades de la compañía (Figura 5.6).



Figura 5.6.- Enfoque metodológico

Fuente: El futuro tecnológico de las Terminales Marítimas de Vehículos: La integración de sus sistemas de información UPC – Departament de Ciència i Enginyeria Nàutiques Barcelona, 2004.

5.8.3.1.- Análisis de la situación actual

Objetivo: Conocer los procesos de negocio clave de las compañías, la interrelación entre ellos, y los Sistemas e Infraestructuras existentes.

Actividades:

- Identificación Procesos y Reglas de Negocio actuales.
- Identificación Arquitectura Tecnológica actual.

Resultados:

- Plan de Proyecto.
- Modelo de Procesos / Reglas de Negocio.
- Descripción de funciones.
- Esquema de Infraestructura Tecnológica.

5.8.3.2.- Análisis de Requisitos.

Objetivo: Conocer las necesidades de los usuarios con respecto al Modelo de Procesos de Negocio que tienen la compañía, y las expectativas respecto al nuevo sistema.

Actividades:

- Toma de requisitos.
- Selección de las “best practices” sectoriales.
- Identificación de los procesos necesarios para las “best practices”.

Resultados:

- Modelo de Procesos mejorado.
- Requisitos de usuario (de negocio, técnicos).

5.8.3.3.- Identificación Alternativas

Objetivo: Identificar los ratios que servirán para evaluar las diferentes alternativas y seleccionar aquellas que se ajustan a las necesidades establecidas.

Actividades:

- Definición de ratios de evaluación.

- Identificación de alternativas.
- Selección de alternativas posibles.

Resultados:

- Ratios de evaluación.
- Cuadro de alternativas posibles.

5.8.3.4.-Selección Alternativa.

Objetivo: Estudio de las alternativas identificadas a partir de los ratios de definidos y selección de aquella que mejor se adapte a las necesidades y requerimientos marcados.

Actividades:

- Análisis de Ratios y Alternativas.
- Análisis de Requerimientos y Alternativas.
- Análisis Coste / Beneficio.

Resultados:

- Análisis de alternativas.
- Valoración de alternativas.
- Selección de mejor alternativa.

5.8.3.5.- Planificación Implantación.

Objetivo: Elaboración detallada del Plan de Implantación de la alternativa seleccionad.

Actividades:

- Identificación de tareas y subtareas a desarrollar.
- Elaboración del calendario de implantación.
- Identificación de recursos necesarios.

Resultados:

- Plan de Proyecto de Implantación.

5.8.4.- Elección del ERP: Parámetros

Para la elección del ERP se emplearán al menos los siguientes parámetros:

- Capacidad de adaptación a las necesidades del cliente.
- Cantidad de módulos adaptables a las necesidades.
- Facilidad de uso.
- Coste de la solución.
- Experiencias y casos de éxito en el sector.
- Solidez financiera del vendedor.
- Tecnologías empleadas.
- Estabilidad en las tecnologías empleadas.
- Cantidad y perfil de clientes.
- Robustez tecnológica de la solución.
- Inversión en I+D.
- Metodología de implantación.
- Independencia de sistema operativo y de motor de base de datos.
- Usabilidad.
- Escalabilidad.
- Flexibilidad para la gestión de nuevas líneas de negocio.

Para la evaluación del implantador, se emplearán al menos los siguientes parámetros:

- Experiencia en el producto y en el sector de la compañía.
- Coste.
- Conocimientos y experiencia del personal, sobre todo del líder de proyecto en implantaciones del producto y en el sector de la compañía.
- Metodología de implantación.
- Metodología de formación.
- Proximidad geográfica.

- Presencia global.
- Compromiso en la implantación.
- Conocimientos y experiencia en integración de sistemas.
- Capacidad de disposición de personal.
- Estabilidad financiera del implantador.

Customer relationship Management

6.- CRM (Customer Relationship Management)

6.1.- Introducción

6.2.- Antecedentes de la Gestión de las Relaciones con el Cliente (CRM).

6.3.- Delimitación del concepto de CRM

6.4.- Principales factores para la implantación exitosa de CRM en las organizaciones

6.5.- Principales problemas y barreras para la implantación exitosa de CRM.

6.6.- Modelo de Implantación de CRM en la organización

6.7.- Conclusión

6.1.- Introducción

Debido a la competencia global entre las organizaciones, ha surgido la necesidad de establecer enfoques de ventas, para la atracción de clientes y retención de los ya existentes, cambiar el enfoque del negocio, en donde los clientes forman parte fundamental de la organización.

El concepto de *Customer Relationship Management (CRM)*, a los inicios de los 90's se enfoca en una mejora de un canal eficiente e integrado, el cual conduce a la satisfacción y retención del cliente, venta de productos e incremento de las ganancias.

El *CRM* está enfocado en predecir el comportamiento del cliente con respecto a la organización. Se puede definir de una manera clara y sencilla al *CRM* como la manera de identificar, adquirir y retener a los clientes.

La finalidad del *CRM*, es que las organizaciones tengan un trato personalizado con el mercado (con sus clientes), recolectando la mayor cantidad posible de información en relación a los clientes y a las necesidades de éstos, para anticiparse a sus deseos y así crear la lealtad de ellos hacia la organización.

El *CRM*, es una estrategia enfocada hacia el cliente, tratando de coordinar a las personas, a los procesos y a la tecnología. La relación con el cliente ha venido revolucionando, y gracias a los avances de las Tecnologías de Información estas estrategias pueden ser aplicadas en la organización. Es importante que la estrategia pueda implementarse de manera gradual, así se podrá incrementar el posible éxito en CRM.

6.2.- Antecedentes de la Gestión de las Relaciones con el Cliente (CRM).

La gestión de las relaciones con el cliente (CRM) es, actualmente, uno de los tópicos que mayor atención ha generado en los campos de la estrategia de negocios, tecnología de la información y en la gestión de marketing (Hall, O. (2001); *Mining the Store*; Journal of Business Strategy).

Una de las premisas que pueden justificar el surgimiento y actual auge de la Gestión de las Relaciones con los Clientes es una reflexión aportada por Kotler (Kotler, P. (2002); About CRM practices; Marketing Science Investigation Review), quien señala que numerosos mercados han llegado a su madurez y eso significa que no hay muchos clientes nuevos disponibles que puedan ingresar a la categoría del producto/servicio, aumentando los costes para atraerlos y la competencia. En esos mercados cuesta alrededor de cinco veces más atraer un nuevo cliente que conservar la buena disposición de los ya existentes.

De forma similar a lo anterior, Winer (Winer, R. (2001); *A Framework for Customer Relationship Management*) sostiene que la necesidad por comprender mejor el comportamiento del cliente y el interés de numerosos directivos por focalizarse hacia esos clientes, quienes pueden generar beneficios a largo plazo, han modificado la manera como los especialistas en marketing venían concibiendo el mundo.

Tradicionalmente, los especialistas estaban entrenados para conseguir clientes, tanto nuevos, es decir, quienes nunca habían adquirido el producto/servicio anteriormente, como aquellos que actualmente forman la cartera de clientes de la competencia. Por lo tanto, el énfasis de la conversión ha sido cambiado desde la denominada adquisición (captación) de clientes hacia la retención. Este paulatino cambio de, donde el enfoque transaccional que predominó en el marketing durante varias décadas parece moverse hacia un enfoque centrado en las relaciones.

Por otra parte, Dawson, Reichheld y Rigby (Dawson, C., Reichheld, F. y Rigby, D. (2003); *Winning Customer Loyalty Is The Key To a Winning CRM Strategy*) sostienen que con la aparición de Internet, es más difícil que nunca mantener a los clientes. Ellos disponen de amplias elecciones y con sólo marcar un par de teclas en el ordenador, les permite chequear la oferta de la competencia para formarse una visión más amplia. En efecto, una compañía con una impresionante tasa de retención cercana al 90% perderá más de la mitad de sus clientes en el lapso de cinco años. En esta misma línea, Albert, Goes y Gupta (Albert, T., Goes, P. y Gupta, A. (2004); *GIST: A model for design and management of content and interactivity of customer*) advierten que, con la amplia proliferación de las herramientas de la tecnología y la comunicación, los sistemas centrados en el cliente y basados en la Web, tales como sitios Web de comercio electrónico o sitios que apoyan las actividades de CRM son en sí mismos sistemas complejos, de múltiples componentes y de amplia variedad de información, pero su diseño y mantenimiento requieren seguir de manera amplia diferentes enfoques de los tradicionales sistemas del ciclo de vida. Mientras algunos usuarios finales pueden ser, ya sea clientes actuales o potenciales, muchos de ellos son visitantes no transaccionales, a menudo anónimos que son parte de la extensa población de usuarios de Internet. Esos usuarios pueden manifestar una amplia variedad de preferencias y motivos hacia el sitio, lo cual dificulta su captura.

Sin duda alguna, los conceptos de lealtad y retención, han sido los temas dominantes entre los especialistas interesados en la gestión de las relaciones con el cliente (CRM). Aunque los progresos se han centralizado, precisamente, en la gestión de dichas relaciones, todavía se observan altas tasas de deserción (Thomas, J., Blattberg, R. y Fox, E. (2004); *Recapturing Lost Customers*; *Journal of Marketing Research*). De hecho, los mismos autores sostienen que, aunque todos los aspectos de CRM necesitan ser asumidos y las estrategias y tácticas deben desarrollarse, un área que ha sido inmensamente ignorada en la literatura del marketing son las

estrategias de reconquista (recuperación) del cliente. La reconquista del cliente es el proceso de revitalización de las relaciones de la compañía con aquellos clientes que han desertado.

Por lo tanto, los antecedentes básicos que justifican el surgimiento de CRM deben buscarse en los estudios que apoyan la relevancia de fomentar la retención y lealtad de los clientes, así como también y cada vez con mayor intensidad, en la relevancia que implica la construcción y establecimiento de relaciones orientadas a recuperar a los clientes perdidos o que manifiestan comportamientos irregulares en la adquisición de bienes y servicios de una determinada compañía.

No obstante lo anterior, la principal complejidad que implica abordar el tema relacionado con CRM estriba en el hecho que su impulso y justificación responde a una serie de factores y variables que subyacen en su aplicación. Por una parte, se relaciona estrechamente con el concepto de *calidad de servicio*, pudiendo ser considerada como una extensión de las diversas aplicaciones orientadas a proporcionar un nivel de calidad óptimo en la provisión de determinado servicio. Dentro de la literatura en torno al concepto de calidad, se ha insistido reiteradamente en la necesidad de proveer productos y servicios que satisfagan e incluso superen las expectativas de los clientes, con los consecuentes beneficios que esta situación conlleva para las utilidades de la empresa.

En dicho sentido, Zeithaml (Zeithaml, V., Rust, R. y Lemon, K. (2001); *The Customer Pyramid: Creating and Serving Profitable Customers*) reconocen que hasta antes de los años noventa, la conexión general entre calidad de servicio y rentabilidad aún estaba siendo cuestionada, pero desde principios de los noventa esta relación ha sido persuasivamente asumida. De hecho, el estudio de Buzzell y Gale (Buzzell, R. y Gale, R. (1987); *The PIMS Principles*) puede ser considerado como un referente destacado que permitió demostrar la correlación entre calidad y beneficios tanto en compañías manufactureras como en empresas de servicios. Numerosos estudios han demostrado las estrechas relaciones entre calidad de servicio y rentabilidad. En esta línea, Bolton y Drew (Bolton, R. y Drew, J. (1991); *A Longitudinal Analysis of the Impact of Service Changes on Customer Attitudes*) demostraron que los esfuerzos de mejora en el servicio producían un incremento en los niveles de satisfacción del cliente, tanto a nivel de proceso como de atributo. Keiningham, Zahorik y Rust, (Keiningham, T., Zahorik, A. y Rust, R. (1994); *Getting Return on Quality*; *Journal of Retail Banking*) demostraron que el incremento de la satisfacción del cliente en el proceso o a nivel de atributo explicaban un incremento de la satisfacción global del cliente.

En línea con lo anterior, Zeithaml, Berry y Parasuraman (Zeithaml, V., Berry, L. y Parasuraman, A. (1996); *The Behavioral Consequences of Service Quality*) demostraron que una más alta calidad de servicio o satisfacción del cliente repercutían en un incremento en las intenciones de

conducta, especialmente en un aumento considerable de la intención de recompra. Bolton (Bolton, R. (1998); *A Dynamic Model of the Duration of the Customer's Relationship with a Continuous Service Provider: The Role of Satisfaction*) demostró que un incremento en la conducta de intencionalidad repercutía en el impacto conductual, incluyendo la recompra o la retención del cliente, una positiva disposición hacia la publicidad boca-oído (word-of-mouth) y un incremento en el nivel de uso. Además, Zahorik y Rust (Rust, R., Zahorik, A. y Keiningham, T. (1995); *Return on Quality (ROQ): Making Service Quality Financially Accountable*) demostraron que el impacto en la conducta genera un aumento en la rentabilidad y en otros resultados financieros.

Tal como señalan Zeithaml, para construir y mejorar sobre la segmentación tradicional, los negocios han estado intentando identificar segmentos – o, más apropiadamente, los niveles de rentabilidad de los clientes – que difieren en su nivel de rentabilidad actual y futura para la compañía. Incluso, tales autores llegan a identificar por medio de un ejemplo empírico, cuatro condiciones absolutamente necesarias para que los diversos niveles de clientes sean explotados por la compañía: en primer lugar, tales niveles tienen perfiles diferentes e identificables; en segundo lugar, los clientes en cada uno de los niveles aprecian la calidad de servicio de manera diferente; en tercer lugar, existen factores diferentes que influyen en la incidencia y volumen del nuevo negocio a través de tales niveles y, por último, la rentabilidad derivada del impacto de la mejora en la calidad de servicio varía ampliamente en los diferentes niveles de clientes.

Otro antecedente que se relaciona con el potencial de CRM como herramienta para fomentar las relaciones con los clientes y aumentar la rentabilidad de cada uno de los diversos grupos identificados, es lo que Zeithaml denomina la “alquimia del cliente”. Básicamente la definen como el arte de reconvertir a los clientes menos rentables en clientes más rentables. Esa reconversión tiene lugar en alguno de los niveles de la “Pirámide del Cliente”, advirtiendo, eso sí, que es más difícil concretarlo en unos niveles más que en otros. Por ejemplo, es muy difícil ascender a un cliente “plomo” hacia los niveles “oro” o “platino” y, frecuentemente, es necesario asumir el hecho de dejar a ese cliente en esa misma categoría más que intentar trasladarlo a las categorías superiores.

No es de extrañar, entonces, que una vez identificada y demostrada la conexión entre calidad de servicio y nivel de beneficios (rentabilidad), el desafío lógico a asumir consistía en comprender las necesidades de los clientes e identificar aquellos grupos de consumidores más rentables (Rigby, D.; Reichheld, F. y Berez, S. (2002b); *Custom Fit: Optimize*). Ello, indudablemente, conduce a centrar la atención en la construcción e intensificación de las relaciones con aquellos clientes o grupos de clientes que pueden resultar más rentables para la compañía, lo que puede considerarse el antecedente clave que lleva al surgimiento de CRM como

una herramienta de gestión que permita conseguir dicho objetivo de la manera más eficiente.

6.3.- Delimitación del concepto de CRM

Definitivamente, el ímpetu por el interés en el CRM se origina en una investigación de Reichheld, quien demostró un incremento dramático en los beneficios a partir de pequeños incrementos en la tasa de retención de clientes. Su estudio mostró que un incremento tan leve como un 5% en la retención de cliente generó impactos tan altos como el 95% sobre el valor actual neto generado por los clientes (Winer, R.).

De hecho, la importancia e impacto de la recuperación del cliente es un elemento clave en la estrategia de CRM de la compañía que no puede ser subestimado. Las investigaciones han mostrado que una compañía tiene entre 60% - 70% de posibilidad de repetir exitosamente la venta en el caso de un cliente activo; entre 20% - 40% de oportunidad de repetir la venta exitosamente en un cliente que se consideraba perdido y sólo entre 5%-20% de oportunidad de cerrar la venta exitosamente cuando se trata de un cliente nuevo (Thomas).

No obstante, aunque existe un interés considerable por las relaciones con el cliente, también existe un alto nivel de confusión e incertidumbre. Numerosas empresas están apoyando un importante número de iniciativas de CRM aún cuando la mayoría de ellas no sabe cuánto de esas iniciativas ayudará a incrementar o disminuir la rentabilidad de sus organizaciones (Battista, P. y Verhun, D. (2000); *Customer Relationship Management. The promise and the reality*; CMA Management).

A pesar del poderoso atractivo que generan las cifras anteriormente señaladas, Rigby, Reichheld y Schefter (Rigby, D.; Reichheld, F. y Schefter, P. (2002); *Avoid the Four Perils of CRM*; Harvard Business Review) señalan que la promesa de CRM es cautivante, pero en la práctica puede ser arriesgada. Cuando se trabaja en sintonía con CRM permite a las compañías recolectar información velozmente, identificar durante todo el tiempo a la mayoría de los clientes que reportan los más altos beneficios – valor – e incrementar la lealtad del cliente mediante la entrega de productos y servicios a la medida – customization-. Además, reduce los costes de servir a tales clientes y hace más fácil conseguir (captar) clientes similares conforme se avanza en el día a día. El riesgo, por lo tanto, es que CRM se convierta en una especie de “panacea” que dará solución, por la vía de una gestión eficaz apoyada en las múltiples opciones que despliegan las nuevas tecnologías de la información y la comunicación (TIC’s) al desafío de proporcionar productos/servicios a medida, aumentando con ello los niveles de satisfacción de los clientes, quienes retribuirían dicho esfuerzo mediante

una estrecha y permanente relación conducente a mejorar los niveles de fidelización y lealtad para con la empresa o compañía.

El primer modelo de CRM empezó a ser discutido hace menos de una década atrás. Desde entonces, trozos y piezas del modelo han sido desplegados en algunas organizaciones y se detecta un nivel considerable de experimentación, con resultados muy heterogéneos (Hansotia, Hansotia, B. (2002); *Gearing up for CRM: Antecedents to Successful Implementation*; Journal of Database Marketing). En este sentido, un punto de vista bastante crítico es el que proporciona Patron (Patron, M. (2002); *If Database Marketing was so good, why is CRM so bad?*; Journal of Database Marketing), quien señala que las cosas estaban mucho mejor antes de que se empezara a utilizar el término CMR. El Marketing de Base de Datos – Database Marketing- era consistente y tenía un auspicioso futuro. El panorama actual, sin embargo, demuestra que sobre el 80% de los proyectos de CRM fracasan en Europa.

Agrega que las empresas ya han sacado partido del marketing de base de datos y ahora ya conocen quienes son sus mejores clientes. Ahora las empresas están desarrollando capacidades de CRM para que puedan proporcionar la oferta correcta, a la persona correcta, en el momento adecuado y a través del canal más idóneo, situación que es mucho más compleja que el marketing de base de datos.

Nairn (Nairn, A. (2002); *CRM: Helpful or full of hype?*; Journal of Database Marketing) sostiene que gran parte del avance y proliferación de herramientas orientadas a maximizar los beneficios derivados de las relaciones entre proveedor y cliente, viene apoyado por el impresionante desarrollo tecnológico de los últimos años, proponiendo un esquema de evolución (Figura 6.1) en el cual es posible visualizar los avances previos que han posibilitado el surgimiento de la denominada gestión de las relaciones con el cliente:

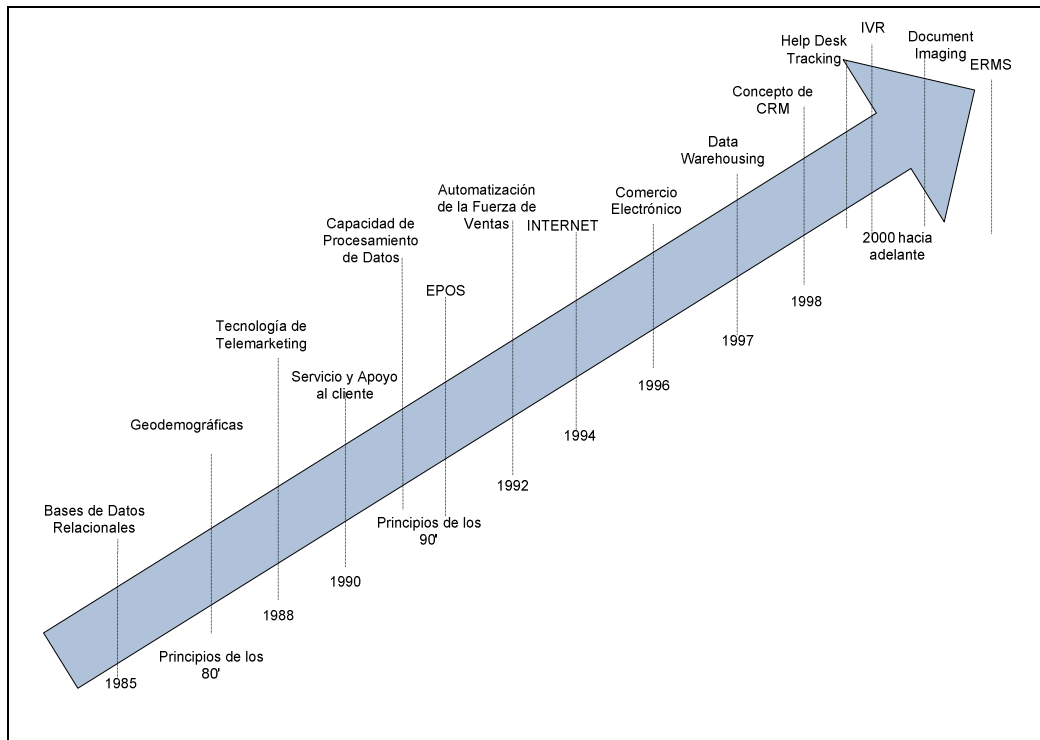


Figura 6.1.- Evolución de la tecnología. Fuente: Nairn, A. (2002); *CRM: Helpful or full of hype?* Journal of Database Marketing

Un aspecto que resulta notable de destacar es el vertiginoso crecimiento que ha experimentado la implantación de iniciativas de CRM en numerosas organizaciones en los últimos años. De hecho, los resultados de una investigación sobre herramientas globales de gestión desarrollado por la consultora Bain, aplicada a directivos experimentados en negocios con alto grado de tecnología, demostró que el uso de las diversas aplicaciones de CMR aumentó a casi el doble en el lapso de tan sólo una año – desde un 35% en 2001 a un 72% en 2002.

No obstante, el espectacular aumento de las aplicaciones de CRM en numerosas organizaciones parece ir acompañado de importantes tasas de fracaso, quizá como respuesta a una situación más bien guiada por la novedad que por una seria apuesta estratégica que permita alinear la organización con una nueva filosofía de gestión del negocio. Como señalan Gillies, Rigby y Reicheld (Gillies, C., Rigby, D. y Reichheld, F. (2002); *The Story Behind Successful Customer Relations Management*), son muy pocos los líderes que han sido capaces de resistirse a una tecnología que promete identificar rápidamente a sus clientes más rentables, seleccionándolos como objeto central de campañas orientadas a incrementar tanto sus compras y su lealtad, siempre al más bajo coste.

A modo de ejemplo, un estudio desarrollado el año 2001 por la empresa consultora Gartner sobre una muestra de grandes compañías, determinó que un 55% de las iniciativas de CRM fracasaron al analizar el retorno de la inversión que, habitualmente, oscila entre \$60 millones y \$130

millones de dólares. Además, el mismo año, una investigación desarrollada por Bain mostró que CRM ocupaba el lugar número veintiuno entre veinticinco herramientas evaluadas por altos directivos, identificando, además, que una quinta parte de los usuarios de CRM habían abandonado las herramientas en su conjunto.

Recurriendo a una revisión de la literatura disponible, son numerosos los trabajos que han intentado proporcionar una definición precisa acerca del concepto de CRM.

En una primera aproximación al concepto, Hansotia (Hansotia, B. (2002); *Gearing up for CRM: Antecedents to Successful Implementation*) señala que CRM es esencialmente un esfuerzo intensivo de información acerca del cliente. Además, puntualiza que en el núcleo central de CRM subyace la habilidad de la organización para tratar la información del cliente de modo creativo, eficaz y eficiente para diseñar e implementar estrategias focalizadas hacia el mismo.

Por su parte, Rigby (Rigby, D.; Reichheld, F. y Berez, S. (2002); *Custom Fit*) proponen que CRM consiste, simplemente, en comprender a los grupos de clientes y tratar a cada uno de una forma que se maximice su valor, recurriendo habitualmente al uso de softwares que apoyen y faciliten el proceso.

Para ellos, CRM es una herramienta más que forma parte de las numerosas aplicaciones enfocadas a fomentar la lealtad de los clientes. En este sentido, identifican que dentro de tales iniciativas existen aplicaciones más tradicionales como la segmentación de los clientes, el marketing one-to-one, la satisfacción de los clientes y la retención de los mismos. Y, al mismo tiempo, agregan que conforme a sus dilatadas observaciones, CRM ha sido la herramienta con la tasa más alta de abandono por parte de la mayoría de las compañías que han analizado (con una tasa cercana al 18%).

De manera similar, Gillies define CRM como una forma de enlazar las técnicas y elementos tratados y probados en la segmentación de clientes para determinar con cuáles clientes se quiere 'CR' (create relationships), es decir, establecer o crear relaciones y con cuáles se desea gestionar el coste ('CM', cost-manage).

Para Croteau y Li, CRM es un concepto que permite a una organización confeccionar productos y servicios específicos para cada cliente individual. Por su parte, Piccoli entienden CRM como una filosofía de gestión que permite a la empresa llegar a establecer relaciones más familiares con sus clientes. Agregan que la empresa que apoya iniciativas de CRM se esfuerza por proveer un servicio consistente y personal al cliente durante todo el tiempo y a través de múltiples puntos de contacto. En otras palabras, CRM es una filosofía de gestión que clama por la reconfiguración de las actividades de la compañía alrededor del cliente.

En un escenario más avanzado, CRM se puede entender como una herramienta que se utiliza para crear una experiencia personalizada, de uno en uno, que otorgará al cliente una sensación de estar bien tratado y cuidado, lo que permite generar nuevas oportunidades de marketing basadas en las preferencias e historial del cliente (Peppers Peppers, D., Rogers, M. y Dorf, B. (1999) ; *Is Your Company Ready for One-to-One Marketing*).

Morris Morris, T. (2000); *Custom Made. Adding Value to Your E-Business Strategy*) se refiere a CRM como un software, no como un proceso integrado de gestión, mientras que Nairn entiende CRM como una filosofía de negocio a largo plazo que se orienta, básicamente, a recolectar, comprender y utilizar de manera inteligente la información del cliente y tratar a los clientes diferentes de manera distinta, proporcionando un alto nivel de servicio para los mejores clientes; de la combinación simultánea de esas dos acciones se intenta alcanzar los objetivos de incrementar la lealtad del cliente y la rentabilidad.

Por último, para Piccoli CRM puede entenderse como un compromiso global de la empresa para identificar clientes individuales y crear relaciones entre la compañía y esos clientes durante todo el tiempo para que dicha relación sea mutuamente beneficiosa.

A partir de las definiciones anteriores, queda en evidencia que no existe un consenso general que permita delimitar exactamente el concepto de CRM y sus implicaciones para las organizaciones que la utilizan. En este sentido, quizá la dificultad de identificar una definición sobre qué es exactamente CRM se puede encontrar en las reflexiones planteadas por Reinartz (Reinartz, W. y Kumar, V. (2002); *The Mismanagement of Customer Loyalty*). Para ellos, no existe una clara evidencia en cuanto a las características de los enfoques exitosos de CRM ni de las razones por las que CRM puede potencialmente fracasar. Es más, la literatura académica existente y las aplicaciones prácticas de CRM no proveen una indicación clara sobre qué exactamente constituye la implantación de los procesos de CRM. Algunas compañías ven CRM principalmente como inversiones en tecnología y software, mientras otras tratan CRM más ampliamente y son agresivas en desarrollar estrechas y productivas relaciones con los clientes.

Mucho más crítica es la visión que aporta Nairn quien sugiere que CRM, en su conjunto, no es más que una extensión de la antigua y bien establecida idea de la segmentación, lo que lleva a plantearse la siguiente interrogante: ¿por qué repentinamente surge un marcado frenesí y por qué los analistas giran nerviosamente en torno al concepto con un afán excesivo de proliferación de acrónimos? Y agrega, ¿por qué se percibe un sentimiento de que CRM es un bombo publicitario?

Sin embargo, a pesar de lo anterior, se puede detectar que tras las numerosas definiciones acerca de CRM se puede extraer un objetivo común

que justifica su aplicación en la organización. Los objetivos clave que subyacen en las inversiones de CRM descansan en el intento de reforzar la afinidad de los clientes con marcas particulares o específicas, al mismo tiempo que se intenta disminuir el coste de dicha interacción.

Otra clara situación derivada del énfasis que se viene otorgando a las iniciativas de CRM en el ámbito del marketing de relaciones es el hecho que la notoria preocupación y atención sobre dicha herramienta ha provocado que numerosas empresas se enfrenten a una permanente lucha por la necesidad de desarrollar estrategias de relaciones y establecer procesos operacionales que sintonicen con la captación, servicio y retención de clientes (Battista y Verhun).

En consecuencia, parece que contrariamente a lo que sucede con numerosos conceptos tratados durante el desarrollo de la presente investigación, el término CRM, aunque universalmente utilizado, hasta la fecha carece de una definición formal. Así, un número importante de proveedores, aprovechando el movimiento del mercado en esa dirección empezaron a catalogar como CRM a las mismas aplicaciones que venían existiendo e implementándose desde hace varios años. Por este motivo, si se revisa la literatura disponible respecto al tema, la gran mayoría de las definiciones provienen de empresas consultoras y de asesoría en tecnologías y aplicaciones informáticas.

En dicho sentido, Winer, señala que el problema principal es que CRM significa diferentes cosas para diferentes personas. Para algunos, CRM significa correo electrónico directo. Para otros, implica customización masiva o el desarrollo de productos/servicios que se ajusten a las necesidades individuales de los clientes. Para los consultores de IT – tecnologías de la información-, CRM se traduce en una complicada jerga relacionada con términos tales como OLAP – on-line analytical processing – y CIC's - customer interactions centers -.

Considerando los antecedentes anteriormente señalados y para los efectos de la presente investigación, CRM puede entenderse como una estrategia de negocios destinada a focalizar los recursos de las empresas a partir del conocimiento real de todas las interacciones de la compañía con el cliente y de las respuestas que éste da a cada estímulo.

También se puede definir como un modelo de negocio que tiene por objetivo establecer relaciones con clientes de forma individual para luego utilizar las informaciones recogidas que permitan tratar clientes diferentes de manera diferente.

Dicha interpretación encuentra similitudes con los planteamientos del Gartner Inc.

(Gartner, Inc. (2004); Emerging Technologies) – empresa proveedora líder en investigación y análisis de la industria de la información y nuevas tecnologías – para quienes CRM es una estrategia de negocio volcada al

conocimiento anticipado de las necesidades de los clientes actuales y potenciales de una empresa. Desde el punto de vista tecnológico, CRM comprende la captura de datos del cliente a lo largo de toda la empresa, la consolidación de todos los datos, tanto interna como externamente, en un banco de datos central, el análisis de tales datos consolidados, la distribución de los resultados de dicho análisis a los diversos puntos de contacto con el cliente y, por último, utilizar esa información en beneficio de la interacción con el cliente a través de cualquier punto de contacto entre éste y la empresa.

Se refiere a aquellas aplicaciones que las empresas pueden utilizar para administrar todos los aspectos de sus encuentros con los clientes. Un sistema CRM puede incluir todo, desde tecnología para la recolección de datos en las llamadas telefónicas del área de ventas hasta sitios web de autoservicio donde los clientes pueden informarse y aprender acerca de los productos/servicios que la compañía les ofrece.

Desde otra óptica, CMR puede ayudar a lo que Zeithaml denominan “vínculos estructurales” con el cliente – learning relationships -. Estos vínculos estructurales se crean para proveer servicios al cliente que habitualmente están diseñados de manera correcta dentro del sistema de distribución del servicio para ese cliente. Por lo general, estos vínculos se crean para proveer servicios a medida de cada cliente y están basados en tecnología con la finalidad de hacer al cliente más productivo.

En dicho sentido, Day, Dean y Reynolds (Day, J., Dean, A. y Reynolds, P. (1998); *Relationship Marketing: Its Key Role in Entrepreneurship*) orientan sus observaciones hacia los beneficios que reporta el uso de CRM. En primer lugar, por medio del desarrollo de una relación más estrecha y cercana con el cliente, la empresa puede ganar una ventaja competitiva y, a través de un incremento de los costes de cambio, puede ser capaz de defenderla. De hecho, durante todo el tiempo, los clientes individuales proporcionan información a la compañía acerca de sus necesidades, deseos y preferencias individuales un proceso costoso que, por lo general, ellos son reacios a repetir con una empresa rival.

Por lo tanto, el hecho de conocer estrechamente a los clientes crea una barrera para la imitación de la estrategia del líder.

En segundo lugar, una iniciativa de CRM eficaz puede incrementar la satisfacción del cliente. Adecuadamente implementado, el diálogo cliente-compañía facilita la adaptación de los productos y servicios a las necesidades individuales y el desarrollo de nuevos productos y servicios satisfacen las necesidades cambiantes e incluso se anticipan a las necesidades futuras.

En tercer lugar, el uso de las técnicas de CRM contribuye a disminuir el gasto global de marketing. Se estima que adquirir nuevos clientes es más costoso que mantener a los ya existentes.

Por último, el desarrollo de relaciones más estrechas con los clientes permite incrementar la lealtad del cliente y los clientes leales permanecen con la compañía más tiempo y compran más a menudo (Dowling, G. (2002); *Customer Relationship Management: In B2C Markets, Often Less is More*).

Por lo tanto, extrayendo los aspectos principales que contienen cada una de las diversas definiciones y aportaciones analizadas, CRM es una herramienta tras la que subyacen una multiplicidad de objetivos y que, en su conjunto y desde un punto de vista estratégico, intentan alinearse con la finalidad de crear, fortalecer y proyectar las diversas relaciones que pueden existir entre la empresa y su cartera de clientes. A modo de síntesis se aporta la figura 6.2. en la que se intenta reflejar los objetivos claves que pretende conseguir una aplicación de CRM en la organización:

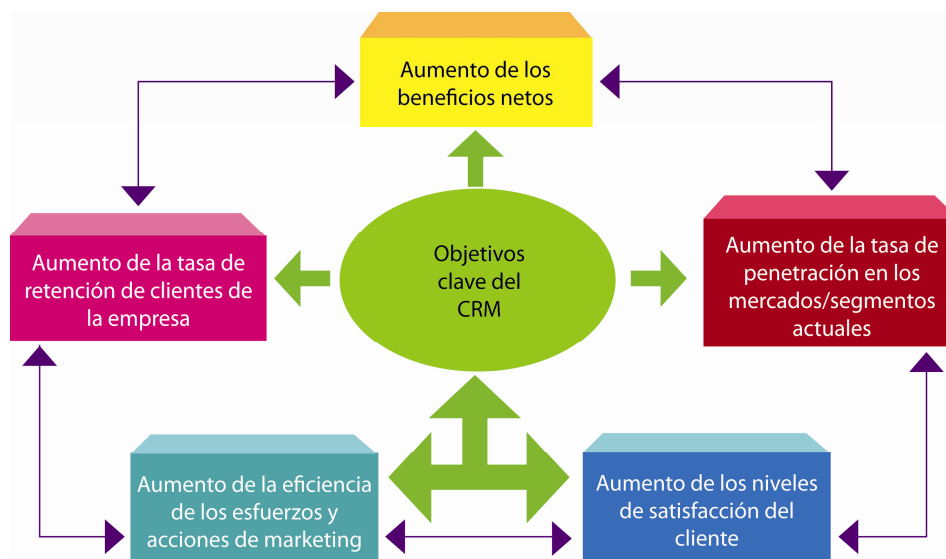


Figura 6.2.- Objetivos clave que subyacen en la aplicación de CRM. Fuente: La gestión de las relaciones con los clientes. Guillermo Nova Castillo

En síntesis, CRM se convierte en una herramienta de gestión que intenta proporcionar un conjunto de acciones y elementos concretos orientados a maximizar los beneficios derivados de la relación comprador-vendedor a partir del fomento de las tasas de lealtad y retención de los clientes ya existentes. Tal como señala Fernández López (Fernández López, J. (2000); *CRM o Cómo Aprovechar al Máximo los Datos del Cliente*), el activo más importante de cualquier organización que desee generar valor a largo plazo consiste en crear relaciones del tipo ‘gana-ganar’ (win-win) con cada uno de sus clientes. Sin embargo, la dificultad que entraña medirle hace que la mayoría de las empresas no recoja en sus estados financieros el valor de dichas relaciones. Por lo tanto, CRM se convierte en un conjunto de maneras eficaces de gestionar y medir cada relación con el cliente, cuyo máximo objetivo es desarrollar ideas para la mejora continua de dicha relación (Fernández López, J. (1999); *Antecedentes y Consideraciones de CRM en la Nueva Era*).

No se debe perder de perspectiva que el surgimiento del concepto de gestión de la relación con el cliente (CRM) es el resultado de una lenta evolución de la mentalidad de las empresas, de las reflexiones académicas y, sobretodo, de una transformación de los sistemas de marketing (Benavent, C. y Meyer-Waarden, L. (2004); *Programmes de Fidélisation: stratégies et pratiques*), por lo que la aparente confusión de términos, la inexistencia de un cuerpo conceptual extenso y consistente y la escasez de contraste empírico pueden ser entendidas como parte de un proceso que aún se encuentra en las etapas iniciales de desarrollo.

En sintonía con el párrafo anterior, Nairn sostiene que no es de extrañar que el nivel de satisfacción actual respecto a CRM no sea tan alto como debería serlo dada la oportunidad que representa. Una buena parte de esta insatisfacción puede ser atribuida a quienes forman parte de la industria de venta tras las tecnologías de CRM quienes están luchando por una porción de un mercado muy lucrativo, mientras que la otra, puede ser atribuida a la falta de comprensión dentro de las organizaciones que el hecho de combinar tecnología con una filosofía de gestión requiere un replanteamiento en las labores intra e inter-funcionales y una reevaluación de los sistemas de compensaciones.

6.4.- Principales factores para la implantación exitosa de CMR en las organizaciones.

Algunos autores han llevado a cabo un conjunto de investigaciones tendentes a identificar los factores que pueden contribuir al éxito de las iniciativas de CRM que emprenden determinadas organizaciones, el surgimiento el negocio electrónico, las dinámicas organizacionales y los cambios culturales están modificando dramáticamente las unidades funcionales de la organización hacia un enfoque centrado en el cliente. A partir de esta premisa, diseñan un modelo que ayuda a identificar los factores críticos de éxito de las iniciativas tecnológicas basadas en CRM, llegando a determinar que tales factores clave son el *desarrollo de las capacidades de la organización para gestionar el conocimiento* y el *apoyo de la alta dirección*, los que influyen significativamente en el impacto de CRM dentro de la empresa. Además identifican una relación positiva entre la disponibilidad tecnológica y las capacidades que dispone la empresa para gestionar el conocimiento.

Para Piccoli, el éxito de la implantación de CRM requiere, necesariamente, de una actitud diferente: la compañía debe fomentar una relación con los clientes a través de programas que abarcan actividades de marketing, operaciones, sistemas de información, contabilidad y otras funciones organizacionales. Agregan, además, que debido al incremento en la cantidad de información que debe ser gestionada, sumado al incremento de la escala y ámbito de acción de la compañía, una gestión de las relaciones

con el cliente exitosa requiere de inversiones significativas en *tecnología, rediseño de procesos y en recursos humanos.*

Desde 1990 hasta la fecha, el mundo ha experimentado considerables transformaciones en cuanto a las aplicaciones que las organizaciones pueden desplegar con el poder de los ordenadores. Las tecnologías de la información (TI) permiten que la información del cliente sea recolectada, consolidada, manipulada y analizada a una escala sin precedentes, lo que ha llevado a que numerosos autores identifiquen las TI como uno de los factores clave del éxito para la implantación de iniciativas de CRM.

El rol de las TI en los esfuerzos de CRM es tan importante que algunos autores (Copulshy, J. y Wolf, M. (1999); *Relationship Marketing: Positioning for the Future*) y numerosos ejecutivos y vendedores asocian CRM con la tecnología que se utiliza para apoyar el enfoque. Esto ha llevado a que muchos utilicen el término en el contexto específico del marketing de base de datos, donde un rango de antecedentes demográficos, estilos de vida y actividades de compra son registrados y monitoreados y, posteriormente, utilizados como base del diseño y oferta de productos/servicios altamente diferenciados para grupos de clientes seleccionados. No obstante, CRM es un concepto mucho más amplio que sólo tecnología.

En un artículo previo, Piccoli destacan que tras CRM subyacen una serie de conceptos clave. En primer lugar, la compañía buscará activamente al cliente adecuado (y, evidentemente, ello implica que la compañía no emprenderá acciones directas orientadas a quienes no representen una oportunidad de negocio atractiva para la misma); en segundo lugar, desarrollará relaciones mutuamente beneficiosas a largo plazo con cada cliente, construyendo y manteniendo un diálogo en dos direcciones; en tercer lugar, procurará satisfacer las necesidades de los clientes y resolver sus problemas y, por último, apoyará a los clientes a través del ciclo de vida de sus interacciones con la compañía.

Por lo tanto, dicho enfoque de CRM requiere mucho más que sistemas computacionales y tecnologías de la información. El cliente debe representar el punto focal de la organización. Todos los miembros de la misma deben comprender y apoyar la contribución de valores requeridos por CRM, su filosofía debe abarcar no sólo las acciones de marketing sino que también a la organización completa y debe ser utilizada para gestionar todos los aspectos de la relación con el cliente de una forma coordinada.

Desde otra perspectiva, Patron (Patron, M. (2002); *If Database Marketing was so good, why is CRM so bad?*, Journal of Database Marketing) insiste en la necesidad de que las compañías requieren una estrategia que ayude a toda a la organización a clasificar y comprender la amplia información disponible antes que este proceso se haga más engorroso con CRM. En otras palabras, no se puede gestionar aquello que no se puede

medir y muy pocas compañías miden el valor individual del ciclo de vida del cliente antes de empezar proyectos de que sintonizan con CRM.

También Rigby manifiestan su preocupación acerca de por qué las iniciativas de CRM frecuentemente fracasan. En relación a esta cuestión, señalan que han pasado los últimos diez años tratando de responder a dicha interrogante, analizando las iniciativas de fidelización/lealtad de clientes, ya sean exitosas o fracasadas, en más de doscientas compañías en un amplio rango de industrias. Su investigación sugiere que una de las principales razones por las que fracasa CRM descansa en el hecho que la mayor parte de los directivos simplemente no entienden lo que están implantando, dejando a la deriva los temas de costes y el tiempo que tomará el proceso.

Para aminorar el nivel de fracaso de las iniciativas de CRM implantadas en numerosas organizaciones empresariales, los autores proponen un conjunto de “imperativos” que debe procurarse cumplir de forma cabal (ver figura 6.3), traduciendo dichas prioridades en un conjunto de acciones que deberán ejecutarse para el éxito del proceso, destacando, además, de qué manera, concretamente, la tecnología puede ayudar a la consecución de dicho cometido.

Aspectos Imperativos del CRM	Conseguir al cliente correcto	Construir – dar forma- una correcta proposición de valor	Instituir los mejores procesos	Motivar a los empleados	Aprender de los clientes ya retenidos
<i>Se obtiene cuando...</i>	<ul style="list-style-type: none"> *Se han identificado los clientes con mayor valor *Se ha calculado la porción de su ingreso que se destina a los bienes y servicios de la compañía 	<ul style="list-style-type: none"> *Se ha estudiado que productos o servicios necesitan actualmente los clientes y cuales requerirán en el futuro *Se han analizado que productos o servicios de la competencia se están ofreciendo actualmente y serán ofrecidos en el futuro *Se ha reconocido que productos o servicios deberán ser ofrecidos 	<ul style="list-style-type: none"> *Se han estudiado las mejores formas de dirigir los productos y servicios a los clientes, incluyendo las alianzas que se necesitan para atacar en un momento dado, las tecnologías en las que se requiere invertir y las capacidades de servicio que se necesitan para desarrollar al actual portafolio de clientes y captar nuevos. 	<ul style="list-style-type: none"> *Se han determinado que herramientas se necesitan los empleados para fomentar las relaciones con los clientes *Se han identificado los sistemas de MR que requieren instituir para favorecer la lealtad de los empleados 	<ul style="list-style-type: none"> *Se ha detectado como los clientes desertan y como pueden ser nuevamente recuperados *Se ha analizado lo que los competidores están haciendo para ganar a los clientes de más alto valor
<i>Las tecnologías detrás de CRM pueden ayudar a....</i>	<ul style="list-style-type: none"> *Analizar el ingreso del cliente *Información de coste para identificar clientes actuales y futuros con un alto valor *Dirigir correctamente los esfuerzos de marketing directo 	<ul style="list-style-type: none"> *Capturar información relevante acerca del comportamiento del producto/servicio *Crear nuevos canales de distribución *Desarrollar nuevos modelos de fijación de precios *Construir comunidades 	<ul style="list-style-type: none"> *Procesar rápidamente las diversas transacciones *Gestionar con mayor eficiencia logísticas internas y la cadena de oferta *Catalizar el comercio colaborativo 	<ul style="list-style-type: none"> *Analizar incentivos y estándares de medida del desempleo *Desplegar sistemas de gestión del conocimiento 	<ul style="list-style-type: none"> *Recrear los defectos hacia los clientes y los niveles de retención *Rastrear los niveles de satisfacción del cliente respecto al servicio

Figura 6.3.- Imperativos empresariales para el éxito de las iniciativas de CRM. Fuente: *Avoid the Four Perils of CRM*; Harvard Business Review

Complementando los planteamientos anteriores, Rigby destacan que se requieren cuatro pasos básicos para implantar con éxito iniciativas de CRM: (i) iniciar el proceso con una *estrategia eficaz* orientada a los clientes que permita claramente identificar las oportunidades para crear valor en el negocio; luego (ii) *rediseñar el proceso clave de negocio* de manera que permita llevar a cabo la estrategia orientada al cliente; una vez realizados

estos pasos fundamentales, recién se procede con (iii) la *búsqueda de tecnología* que permita trazar la estrategia y proceso de negocio y, finalmente, (iv) utilizar dicha tecnología en donde se considere apropiado implementar el nuevo proceso de negocio, centrándose en las oportunidades que favorezcan la creación de valor para acabar con una medición de los resultados.

Un matiz relevante a dichas aportaciones es el que plantean Reinartz (Reinartz, W., Krafft, M. y Hoyer, W. (2004); *The Customer Relationship Management Process: Its Measurement and Impact on Performance*) et al. (2004), quienes distinguen tres niveles posibles en los que tiene lugar la aplicación de las prácticas de CRM:

1. Funcional
2. De cara al cliente
3. La compañía en su conjunto

Según el nivel que se trate, dependerá la forma en que se defina el concepto de CRM.

Luego de llevar a cabo una investigación empírica, dichos autores defienden la tesis de que la implantación de procesos de CRM está asociada con un mejor rendimiento de la compañía en las etapas de inicio y mantenimiento de las relaciones, siendo más notorio el efecto en el caso del mantenimiento de la relación.

Hansotia, defendiendo la convicción que una estrategia de CRM es más propicia para ciertos negocios y clientes, señala que el entorno ideal para una adecuada aplicación de CRM deberá tener:

- *frecuentes interacciones/compras del cliente*
- *alto nivel de experiencia necesaria para guiar la compra y la resolución de problemas*
- *multiplicidad de productos y servicios adquiridos por los clientes*
- *existencia de productos que proveen conveniencia, simplicidad o que reducen el riesgo*
- *productos que son intensivos en aprendizaje y requieren un sustancial servicio de apoyo a cliente después de la venta*
- *un departamento de marketing centralizado.*

Nairn también desarrolla un análisis enfocado a determinar cuales son los factores clave que deben existir previos a la implantación de iniciativas de CRM. Conforme a este autor, CRM comprende tres fases básicas:

1. *conocer y comprender el rango de necesidades y deseos de los clientes.*
2. *crear grupos homogéneos de clientes a partir de esta masa heterogénea.*
3. *concentrar el esfuerzo de marketing sobre el grupo que puede ser más rentable para la organización.*

6.5.- Principales problemas y barreras para la implantación exitosa de CRM.

En los apartados anteriores se insiste reiteradamente en las altas tasas de fracaso de las iniciativas de CRM en numerosos entornos en los que se ha apostado por su implantación. Por lo tanto, un aspecto relevante de analizar consiste en determinar cuales son los motivos o causas principales que llevan a esas elevadas tasas de fracaso.

En este sentido, Rigby llega a la conclusión que el fracaso de las iniciativas de CRM no depende de fallos o problemas con la tecnología en sí. Agregan que en una investigación llevada a cabo por CRM Forum, una asociación industrial independiente, cerca del 90% de los directivos de negocios tecnológicos atribuyó el fracaso de CRM a la *falta de liderazgo* y al *cambio de los mecanismos de gestión interna* dentro de sus compañías. Sólo un 4% mencionó los problemas de software como la causa principal del fracaso de las iniciativas enfocadas a los clientes.

Para Hansotia el éxito o fracaso de las iniciativas de CRM viene determinado por el hecho que su aplicación no es universal y tiende a ser más apropiada para determinados entornos y para empresas con características muy particulares. En este sentido, sostiene la tesis que CRM es una de las más poderosas estrategias de negocio y tiene el potencial para generar los mayores beneficios para cierto tipo de negocios más que para otros. Esencialmente, CRM permite gestionar las interacciones con el cliente y crear experiencias memorables que superen sus expectativas. Para que tales experiencias tengan un impacto a largo plazo, cada experiencia a favor del cliente debe traducirse en un retorno para la compañía.

Croteau y Li sostienen la idea que una estrategia de negocio de CRM es irreal sin un apropiado nivel de comprensión de los beneficios y oportunidades que proporciona la tecnología y viceversa. Asimismo, Battista y Verhun determinan cinco dimensiones básicas para implementar con éxito CRM: (1) estrategia de relación con el cliente, (2) gestión del acceso al

cliente, (3) gestión del proceso de cliente, (4) gestión de la experiencia del cliente y (5) gestión del conocimiento acerca del cliente.

Los mismos autores señalan, además, que la principal barrera para la implantación exitosa de las iniciativas de CRM no descansa en una cuestión relacionada con el enfoque estratégico, sino más bien, en una cuestión relacionada con el uso y aplicaciones de las nuevas tecnologías. De esta manera, las principales barreras relacionadas con la tecnología para la implantación de CRM son:

- *inconsistencia de la información y pérdidas de datos.*
- *la legalidad de los sistemas.*
- *la proliferación de canales de distribución y comunicación.*
- *el coste.*
- *la falta de habilidades empresariales.*
- *la gestión del cambio.*

También es importante destacar que algunos estudios se han centrado en determinar la aparente contradicción que se empieza a visualizar entre avance tecnológico y la necesidad por una atención más personalizada y cálida y las serias dificultades e incoherencias que suelen presentarse en la gestión de las bases de datos, de los call centres y de los sistemas de comunicación con el cliente, ya sea en los contactos cara a cara, por teléfono, por correo escrito tradicional o por correo electrónico, todos soportes ampliamente utilizados por las iniciativas de CRM (De Torcy, De Torcy, G. (2002); *A New Wave in Creating Customer Satisfaction*). Incluso, el mismo autor sostiene la necesidad por diferenciar claramente la aparente confusión que existe entre los conceptos de customización versus personalización.

Para McKim (McKim, B. (2002); *The Differences Between CRM and Database Marketing*) la principal dificultad de las iniciativas de CRM descansa en sus elevados costes requeridos para su adecuada implantación. Dicho autor apoya la utilización de los ampliamente difundidos principios y técnicas del marketing de base de datos en detrimento de CRM, justificando su visión en el hecho que ambas disciplinas tienen características y costes de implementación muy similares, permitiendo obtener una visión de 360º del cliente y manejar toda la información en un sistema común, pero cuyos resultados tienden a ser menos favorables en el caso de la herramienta más reciente.

Incluso, el factor tiempo se agrega como una variable en contra de las iniciativas de CRM.

Normalmente, los sistemas de CRM requieren una media de un año para su instalación y funcionamiento a pleno rendimiento, mientras que los sistemas de marketing de base de datos generarán información y resultados en un lapso comprendido entre cuatro a seis meses como promedio.

Siguiendo con las aportaciones de McKim, el autor clasifica los principales problemas con las iniciativas de CRM en tres áreas distintas conforme a las observaciones realizadas en los últimos años, los que se exhiben en la figura siguiente:

Una visión diferente es la que propone Hansotia, para quien el principal foco de atención respecto a CRM ha estado puesto en la plataforma tecnológica que apoya las interacciones con el cliente y la automatización de las ventas, sin embargo, son muy escasos y puntuales los estudios que se han centrado en analizar el éxito de la altas tasas de retorno sobre los esfuerzos e inversiones en infraestructura de marketing. El autor considera que la justificación clave que explica dicha situación descansa en una falta de inversión en los dos principales requisitos de CRM: *el diseño estratégico* y una adecuada *planificación y análisis*. Agrega que si una empresa está buscando desarrollar un proyecto de demostración de CRM, tal situación puede implicar que la empresa no desea realizar un esfuerzo firme por el diseño de su estrategia, lo cual significaría el mayor error si sólo se focaliza en la tecnología que permite la interacción con el cliente sin considerar modelos y análisis que ayuden a planificar y guiar la ejecución (Figura 6.4).



Figura 6.4.- Identificación de los Principales Problemas para la Implantación Exitosa de Iniciativas de CRM. Fuente: The Differences Between CRM and Database Marketing; Journal of Database Marketing

6.6.- Modelo de Implantación de CRM en la organización.

López Fernández (Fernández López, J. (2000); *CRM o Cómo Aprovechar al Máximo los Datos del Cliente*) propone un modelo de implantación de CRM que intenta favorecer su paulatina inserción dentro de la organización que pretende llevarlo a cabo, poniendo de relieve la necesidad de abordar aspectos y etapas clave si lo que se desea es alcanzar resultados beneficiosos para la empresa (Figura 6.5).

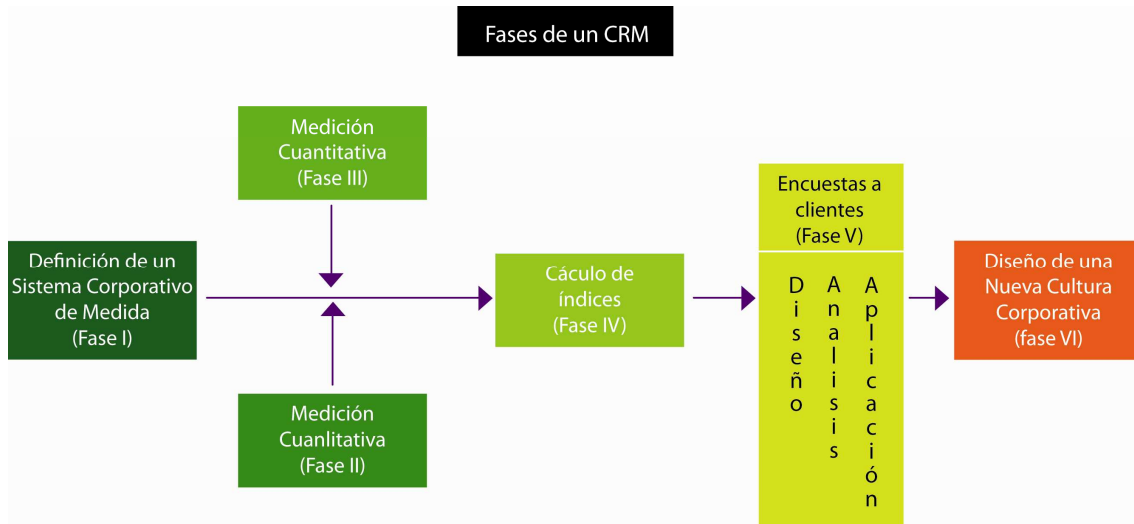


Figura 6.5.- Fases de un CRM. Fuente: CRM o Cómo Aprovechar al Máximo los Datos del Cliente; MK Marketing+Ventas

Dentro de cada una de estas fases y siguiendo se pueden detectar las siguientes tareas y funciones básicas a desarrollar:

Fase I: Diseño de un Sistema Corporativo de Medida.

La idea es que, para el cliente el valor se mide como el cociente entre la calidad percibida y el precio que ha debido invertir para su adquisición. El sistema corporativo de medida debe verificar lo que se hace en relación con la competencia del mercado y el beneficio que reporta al cliente.

Fase II: Medición Cualitativa.

Este tipo de medición revela cuál es la percepción del cliente respecto a los productos/servicios que le ofrece la empresa. Las herramientas más utilizadas en este proceso suelen ser tres: la realización de dinámicas de grupo, en donde se analiza lo que el cliente necesita; los paneles de consumo, donde se revisan los aspectos del producto/servicio que demandan y necesitan los consumidores y la técnica de incidentes críticos, que permite profundizar en la información generada a través de las técnicas anteriores.

Fase III: Medición Cuantitativa.

Consiste en dar inicio al proceso de medición por medio de un conjunto de unidades de métrica interna, respondiendo al principio de que no se puede mejorar externamente lo que no se mide internamente. Para la consecución de este objetivo, se debe intentar establecer una conexión entre las mediciones internas con las externas, siguiendo una secuencia similar a la que sigue:

- a) Determinar el valor percibido por el cliente.
- b) Descomponer el valor en factores de percepción, como el nivel de uso del producto/servicio a partir de experiencias tangibles, estrategias, entorno y prestación de servicio.
- c) Desarrollar los procesos: precios, créditos y garantías.
- d) Definir las expectativas de los clientes.
- e) Desarrollar métricas internas que permitan evaluar la conducta del personal.

Fase IV: Cálculo de Índices.

Básicamente se recurre al uso de dos tipos de índices: de satisfacción y de proceso.

Las mejores herramientas de medición son las que generan índices financieros de satisfacción al cliente y de porcentaje de cuota de mercado.

Fase V: Encuestas a clientes.

Su diseño y aplicación deberá responder al objetivo de permitir que sea el cliente quien defina el producto/servicio de cada uno de los procesos según sus necesidades.

Fase VI: Diseño de una Nueva Cultura Corporativa.

Por regla general, cuando por medio de un CRM se incentiva la participación del cliente para que valore y opine sobre los productos/servicios y de la estrategia global de la empresa, se suelen obtener buenas referencias y una respuesta de lealtad. Esta nueva cultura requiere, por lo tanto, que la empresa se dirija tanto a los clientes como a sus empleados, vale decir, la organización deberá establecer una operativa

interna y sistemas ágiles que faciliten y potencien la relación en ambas direcciones.

En una línea similar a los planteamientos anteriores, Winer sostiene que el paso inicial primordial para aplicar una solución CRM es la construcción de una potente y detallada base de datos del consumidor o un archivo de información. Esta tarea resulta relativamente simple en aquellos negocios que basan buena parte de sus ventas en el comercio electrónico, puesto que las transacciones del consumidor y la información de contacto están acumuladas como componente habitual de la interacción con los consumidores. No obstante, para las empresas que no han recolectado previamente información sobre sus consumidores/clientes, la tarea incluirá la búsqueda de información de contacto histórico del consumidor a partir de fuentes internas, tales como la contabilidad y el servicio al cliente.

No obstante lo anterior, algunos autores advierten que las inversiones sustanciales en CRM no son adecuadas para todos. En un pequeño negocio es relativamente fácil mantenerse ajustado a las preferencias de los clientes. De hecho, una gran aerolínea o una importante cadena hotelera internacional deben gestionar, sustancialmente, grandes cantidades de información, lo que un pequeño hostel o establecimiento de alojamiento puede alcanzar de modo similar en las relaciones con sus clientes.

Las compañías, tradicionalmente, han utilizado una amplia variedad de métodos para elaborar sus bases de datos. Las empresas manufactureras de bienes de larga duración utilizan información a partir de las tarjetas de garantía para configurar la información descriptiva básica. Desafortunadamente, las tasas de respuesta de tales herramientas están en un rango entre 20-30%, lo que deja enormes vacíos en las bases de datos. En contraposición, las empresas basadas en la explotación de servicios lo tienen mucho mejor, ya que la propia naturaleza de la actividad facilita la recolección de información. Por ejemplo, los bancos han estado a la vanguardia en la implantación y aplicaciones de herramientas CRM desde hace varios años. Las empresas de telecomunicaciones también disponen de una cantidad amplia de información acerca de sus clientes.

Por lo tanto, se trata de llegar a determinar un perfil exhaustivo y lo más cercano a la realidad de cada cliente. La información acerca del perfil del cliente puede incluir: cuantificación del valor mediante las ventas y margen bruto, historial de compra, ciclos de compra, preferencias de producto servicio, aplicaciones de producto, condiciones financieras, frecuencia y tipo de contacto, aspectos demográficos e intenciones futuras. También puede incluir registros individuales acerca de satisfacción del cliente, intención de recompra y recomendación e indicadores claves de lealtad que calibran la relación de la compañía con el cliente.

Para Piccoli CRM permite generar una más alta rentabilidad debido al incremento de las ventas, la disminución de los costes de adquisición de

clientes y al incremento de la rentabilidad de los clientes dispuestos a pagar un poco más a cambio de un mejor servicio. En este sentido, los autores proponen una representación gráfica respecto a cómo opera CRM:

Desde otra perspectiva, Hansotia considera tres componentes esenciales del CRM:

- *diseño estratégico y disponibilidad organizacional.*
- *planificación y análisis.*
- *ejecución de las interacciones con el cliente.*

Agrega, además, que cuando el cliente se convierte en el punto focal de la estrategia de la compañía, resulta inevitable que la implantación de CRM (Figura 6.6) se relacionará con casi todas las partes, áreas o unidades que integran una organización (finanzas, diseño y desarrollo de producto/servicio, servicio al cliente, comunicaciones, marketing, recursos humanos, entre otras). Por lo tanto, de manera permanente, todas las áreas o unidades de la organización deberán reinventarse a sí mismas – reingeniería – si la compañía en su globalidad pretende estar enfocada al cliente. En este sentido, los directivos necesitan crear una cultura cuyo enfoque esté centrado en el cliente y una organización basada en el aprendizaje constante (learning organization) que se adapte acertadamente a los nuevos procesos.

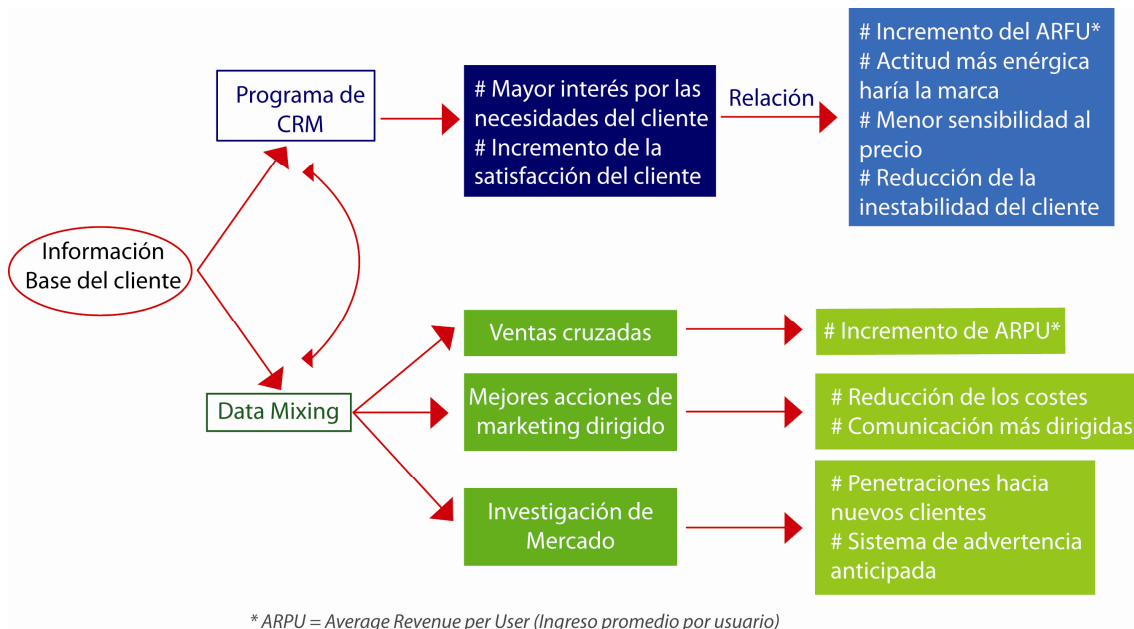


Figura 6.6.- Modelo de CRM. Fuente: Customer Relationship Management: A Driver for Change in the Structure of U.S. Lodging Industry

Por otra parte, para Croteau y Li tres funciones organizacionales principales están apoyadas en iniciativas tecnológicas dentro del escenario de CRM:

- apoyo y servicio al cliente.
- automatización de la fuerza de ventas.
- automatización del marketing de la empresa.

La función de apoyo y servicio al cliente (CSS, customer support and service) cubre el modo en el cual un producto es distribuido, empaquetado, descrito, pagado, instalado, reparado, renovado y rediseñado. Por su parte, la automatización de la fuerza de ventas (SFA, sales force automation) implica aplicar las mejores prácticas de venta en un paquete informático que puede ayudar a los equipos de ventas de la organización a atraer y retener los clientes rentables. Por último, la automatización del marketing de la empresa (EMA, enterprise marketing automation) busca el mismo impacto automático y poderoso que tiene la SFA sobre las ventas, pero orientado al marketing.

Considerando las características hasta ahora señaladas, (Hansotia Figura 6.7) propone el siguiente diseño estructural de alto nivel para apoyar la correcta ejecución de CRM:

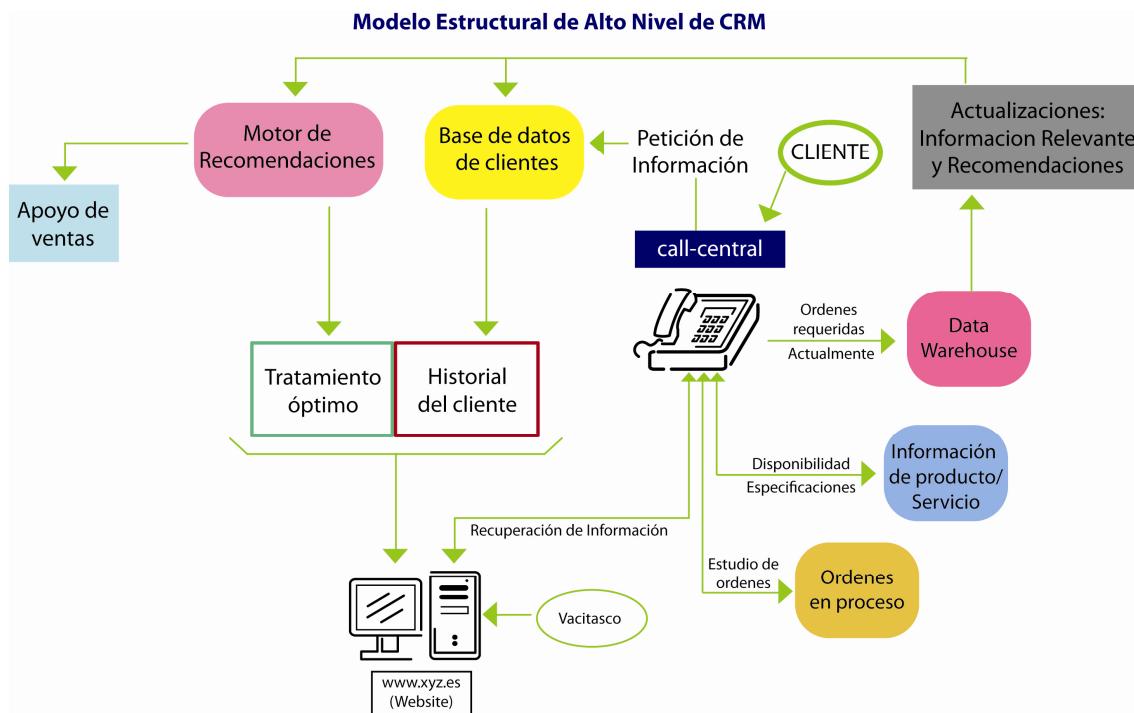


Figura 6.7.- Modelo estructural de alto nivel de CRM. Fuente: Hansotia, B. (2002). *Gearing up for CRM: Antecedents to Successful Implementation*; Journal of Database Marketing

En definitiva, CRM depende de una cuidadosa planificación y una favorable disposición organizacional. La tecnología que subyace en las aplicaciones de CRM es necesaria para gestionar con eficiencia y eficacia los contactos iniciados con los clientes, pero si las empresas no construyen actividades específicas que sirvan de apoyo al constante flujo de caja generado por los clientes, CRM puede convertirse en una acción financieramente inviable.

6.7.- Conclusión

Se puede entender entonces que para la implementación de un CRM es más que hacer una solicitud a un proveedor de software con las cotizaciones y una presentación de los beneficios que se obtendrán, hay que analizar si la organización está preparada y quiere un cambio de estrategia orientado hacia el cliente para poder así garantizar el éxito, o por lo menos minimizar el riesgo de fracaso de dicha implementación.

Con la implementación y el uso de CRM las organizaciones pueden conservar y conseguir más clientes, y de esa manera permanecer en el mercado competitivo que estamos viviendo.

Una implementación de CRM se hace y se planea de forma pausada, así tendremos la posibilidad de que los riesgos sean menores y evidenciaremos los resultados poco a poco; de esta forma se podrán incrementar los casos de éxito.

Debemos recordar que el CRM debemos verlo también, como una estrategia de negocio, es por eso que debemos aprender continuamente del comportamiento de nuestra herramienta, debemos observar los movimientos que la competencia está realizando, así como tener siempre presente que el cliente y su satisfacción son primero.

Criterios comparativos de paquetes ERP

7. - Criterios comparativos ERP

7.1.- Introducción

7.2.- Funcionalidad

7.3.- Flexibilidad

7.3.1.- Personalización

7.3.2.- Actualizaciones flexibles

7.3.3.- Internacionalización

7.3.4.- Facilidad de uso

7.3.5.- Arquitectura

7.3.6.- Escalabilidad

7.3.7.- Seguridad

7.3.8.- Interfaces

7.3.9.- Independencia del sistema operativo.

7.3.10.- Independencia del sistema de bases de datos

7.4.- Soporte

7.4.1.- Infraestructura de Soporte

7.4.2.- Formación

7.4.3.- Documentación

7.5.- Continuidad

7.5.1.- Estructura del proyecto

7.5.2.- Actividad de la comunidad

7.5.3.- Transparencia

7.5.4.- Frecuencia de las actualizaciones

7.5.5.- Otros efectos acordados

7.6.- Madurez

7.6.1.- Estado del desarrollo

7.6.2.- Lugar de referencia

7.1.- Introducción

En este apartado se presentan una serie de criterios que se deberían tener en cuenta a la hora de elegir el sistema ERP más adecuado para una organización.

Los criterios introducidos en este apartado están jerárquicamente estructurados y pueden ser usados como una base para una posterior adaptación personal. Muchos de los criterios no son medibles, con lo que se ha tratado de medir de la mejor manera posible para el lector. De manera que a partir de los mismos el lector sea capaz de entender la fortaleza, debilidad de cada uno, al igual que la diferencia de los sistemas de ERP de software libre presentados, todos los criterios se encuentran en el artículo publicado en *International Journal of Production Economics*: “An AHPbased approach to ERP system selection” (2005), nombrado en la bibliografía.

Los criterios son los siguientes:

- Tamaño
- Funcionalidad
- Flexibilidad
 - Personalización
 - Flexibilidad de las actualizaciones
 - Internacionalización
 - Facilidad de uso
 - Arquitectura
 - Escalabilidad
 - Seguridad
 - Interfaz
 - Independencia del S.O.
 - Lenguaje de programación
- Soporte
 - Soporte de la infraestructura
 - Formación
 - Documentación
- Continuidad
 - Estructura del proyecto
 - Actividad de la comunidad
 - Transparencia
 - Frecuencia de las actualizaciones
 - Otros efectos acordados
- Madurez
 - Estado del desarrollo
 - Lugares de referencia
- Otros
 - Licencia
 - Demos online
 - Alojado en sourceforge

- Acceso CVS
- Comprobación de la corrección de la descarga
- Comienzo del proyecto

Todos los criterios de evaluación están influenciados por el coste. La funcionalidad nos muestra los módulos que ofrece el sistema para dar soporte a las necesidades de las distintas áreas funcionales de la empresa. Si el ERP presenta carencias en alguna de estas áreas, es importante que pueda integrarse con otros productos que suplan dichas carencias, o bien, que facilite la realización de desarrollos a medida, en definitiva, el grado de personalización y de desarrollo adicional para una mejor adaptación del ERP a los procesos necesarios. La flexibilidad nos muestra las opciones de reducción de las limitaciones funcionales. Soporte nos indica los conocimientos para la solución de las incidencias. Continuidad trata sobre la sostenibilidad del proyecto así como la independencia del proyecto. Los puntos de madurez nos indica el riesgo de escoger un sistema no asentado o con la calidad suficiente.

La gran mayoría de ERPs de software libre están aplicados a PYMES, sobretodo en sus inicios, las aplicaciones ERP más consolidadas empiezan a dar servicio a grandes empresas, la dependencia en una comunidad activa y en Partners son fundamentales para su proliferación y expansión, aventurarse en la elección de una aplicación de software libre por el ahorro económico puede condenar al fracaso, las siguientes características genéricas aplicables a los ERPs, no servirán en absoluto si no existe un estudio previo de la situación del lugar en dónde se pondrá en funcionamiento la aplicación ERP. Una vez conocidos todos los factores, características y recursos empresariales propios, podremos realizar de manera inteligente la elección apropiada ayudándose del capítulo siguiente.

7.2.- Cobertura funcional

La cobertura funcional se utiliza para la perspectiva de la empresa. En gran parte de empresas prefieren el uso del término cobertura funcional que el de funcionalidad. El concepto de cobertura lleva implícito que la funcionalidad innecesaria, de alguna manera es inútil. Representa el grado en el que el sistema ERP elegido, en su versión estándar, encaja con los procesos de negocio. Cuanto más precisa sea la cobertura funcional, menores son los costes de implementación y personalización. La cobertura funcional tiene un alto impacto en el coste total, así como en el tiempo de implantación.

Como los requisitos funcionales cambian ampliamente dependiendo del área de negocio, no hay una manera general para medir la cobertura funcional. El número de tablas de la base de datos es, cuando está disponible, un indicador de la magnitud funcional de un sistema ERP, suponiendo que la estructura de la base de datos esté bien diseñada.

7.3.- Flexibilidad

Un sistema ERP es flexible de tal manera que responde a las constantes transformaciones de las empresas. La tecnología cliente/servidor permite al sistema ERP operar sobre diferentes bases de datos por las conexiones de bases de datos abiertas, pues es muy probable que el mismo producto migre de un área de producción para otra durante el ciclo total de producción.

La flexibilidad permite salvar la distancia existente entre la funcionalidad inicial y la precisa cobertura que proporciona un sistema personalizado. A parte de la oportunidad de adaptar los sistemas de forma óptima para los procesos empresariales, la flexibilidad también implica cuestiones de facilidad de uso y de la administración e independencia en la plataforma. La idea de flexibilidad se basa en conceptos técnicos y en el diseño software del sistema. La flexibilidad en sistema ERP se basa en los criterios nombrados a continuación.

7.3.1.- Personalización

Dependiendo del grado de personalización necesaria y de los niveles de competencia de los sistemas ERP, se deben proporcionar diferentes niveles de personalización. Por lo tanto el esfuerzo por personalizar el sistema puede ser distribuido entre un gran número de usuarios.

- **Alto nivel de personalización a través de la edición de metadatos.** En este contexto, significa que el sistema se puede personalizar a través de la edición de datos de fácil lectura, en vez de la realización de la codificación con un lenguaje de programación. Un experto en asunto de negocios debe ser capaz de personalizar el sistema sin tener un conocimiento detallado de programación. La finalidad es reducir la carga de aprendizaje proporcionando facilidades para una amplia gama de problemas. La posibilidad de un alto nivel de personalización constituye un factor de productividad importante para acortar la aplicación en tiempo y permitir la adaptación continua a los procesos.
- **La personalización a bajo nivel** (basado en la aplicación framework). Para los desarrolladores que quieran adentrarse más en los detalles y en la necesidad de mayor flexibilidad el sistema también debe poder ser utilizado como framework para el desarrollo de aplicaciones. Aquí el sistema ERP define la arquitectura de software y permite añadir operaciones a medida. Este código personalizado debe cumplir con el framework de la Application Programming Interface (API) o programación de la interfaz de la aplicación. La codificación esta asociado a un bajo nivel de personalización.

El siguiente nivel inferior de personalización/desarrollo sería la adaptación o ampliación del framework, por ejemplo, proporcionando un nivel más alto de instalaciones personalizadas.

7.3.2.- Actualizaciones flexibles

Como las personalizaciones las hemos definido a partir de metadatos y del código personalizado, los cuales deben cumplir con las especificaciones del framework, es posible proporcionar un procedimiento de actualizaciones sin afectar a las personalizaciones del ERP. Debido a esto, las actualizaciones del sistema central no deben provocar nuevas adaptaciones en el sistema anteriormente personalizado.

7.3.3.- Internacionalización.

Mide la capacidad de utilizar el mismo sistema, aunque con las necesarias adaptaciones, a otros países con diferentes idiomas, legislación y costumbres operativas. La forma más simple de internacionalización es proporcionar traducciones de la interfaz de usuario y de los sistemas de contabilidad de cada región.

El idioma se selecciona a nivel de usuario. Se puede distinguir entre la simple traducción de partes de la interfaz gráfica de usuario (GUI) a nivel estática (por ejemplo, menús, etiquetas de campo), la traducción de las partes del GUI a nivel dinámico (por ejemplo, flujo de trabajo, estados), y el contenido (por ejemplo, descripciones de los productos). Los requisitos legales a nivel nacional, especialmente en la contabilidad a menudo exigen un flujo de trabajo personalizado o adaptado a la lógica de negocios. Esto significa que poseer buenas opciones de personalización (criterios de flexibilidad 1 y 2) son una condición previa para la internacionalización.

Es muy importante para el código abierto de los sistemas ERP, aunque estos sean simples, destinados a servir únicamente a nivel local, proporcionar la flexibilidad suficiente para apoyar a muchas naciones con el fin de obtener una base más amplia de usuarios internacionales y reducir el riesgo del proyecto debido a las limitaciones de soporte a nivel internacional. La separación del proyecto significa dividir la base de código, lo que lleva a dos proyectos separados y por lo tanto la fragmentación de la comunidad que a su vez conlleva menor colaboración.

El soporte en múltiples lugares conlleva prestar servicios en varios lugares distribuidos internacionalmente, usando diferente contabilidad y esquemas de coste dentro de un sistema ERP.

Para administrar la funcionalidad específica de cada lugar y país, el sistema ERP debe permitir la generalización de la funcionalidad común. Los lugares (site) pueden ser servidos con un sistema central, teniendo una conectividad fiable a todos los lugares donde se realizan o con sistemas distribuidos y sincronizados.

7.3.4.- Facilidad de uso

El interfaz de usuario debe ser diseñado conforme a la información necesaria para una tarea. Una simple tarea no requerirá navegar a través de muchas pantallas. Es parte de la personalización, adaptar el sistema ERP a los procesos. Para los trabajos rutinarios se deberán realizar atajos de teclado. Algunos sistemas ERP soportan unos pocos elementos GUI (Interfaz gráfica de usuario). Facilidad de uso está relacionada con las posibilidades de personalización, la aceptación del usuario, los costes de formación y el coste de las operaciones.

7.3.5.- Arquitectura

La importancia de muchos factores de flexibilidad es la elección de arquitectura. Las soluciones de código abierto tienen 2 niveles o 3 de arquitectura. El nivel 2 o la arquitectura cliente-servidor consiste en que un cliente “grande” contiene el GUI, la lógica empresarial que se comunican directamente con la base de datos. En el caso de una arquitectura de nivel 3, el cliente es el responsable de la interfaz gráfica de usuario y de la validación simple de los datos.

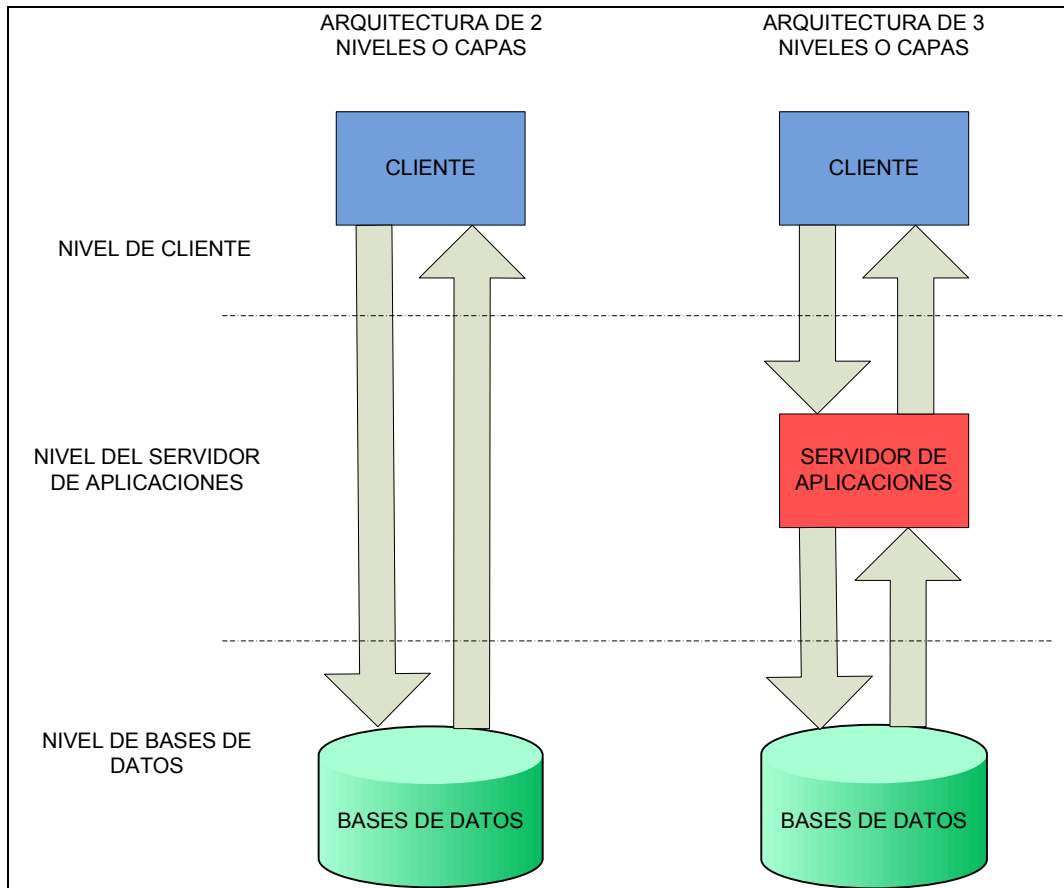


Figura 7.1 Arquitecturas por niveles o capas. Fuente: "An AHPbased approach to ERP system selection" (2005)

Toda la lógica se encapsula en la aplicación del servidor. La base de datos se encarga de almacenar los datos persistentes. La base de datos es la responsable del almacenamiento de los datos persistentes. Por lo general, en el caso de la arquitectura de 3 capas, el cliente es un navegador web y el servidor de aplicaciones es un servidor web de aplicaciones. Algunas arquitecturas aprovechan la funcionalidad de un "estándar" de uso general, como servidor de aplicaciones (servidor de aplicaciones J2EE para Java, Zope para Python), otros usan un servidor propietario o un servidor de web básico.

Teniendo arquitecturas avanzadas permiten ejecutar en el servidor de aplicaciones diferentes tipos de cliente o incluso cualquiera de ellos. Estos clientes pueden estar basados en web, en terminales, así como en una rica interfaz gráfica de usuario, ejecutándose en un dispositivo móvil o en un ordenador personal. Esto es posible debido a un diseño multi-capa. El nivel medio está más dividido de forma horizontal en los datos, la lógica de negocio y la capa de presentación. Así que sólo la capa de presentación necesitará cambios para dar soporte a varios tipos de clientes. Una mayor flexibilidad es posible por la separación vertical del sistema en los servicios que son conectados con flujos de trabajo flexibles. Para la integración con sistemas externos a estos servicios pueden ser publicados como servicios

web. Flujo de trabajo es la automatización de un proceso de negocio, durante el cual, la información se pasa a lo largo del sistema de acuerdo a un conjunto de reglas.

7.3.6.- Escalabilidad

El sistema debería aceptar grandes volúmenes de transacciones con tiempos de respuesta constantes. La escalabilidad es altamente dependiente de la arquitectura y por lo tanto del servidor de aplicaciones y de la tecnología de bases de datos. "Un sistema que no escala para apoyar a todos sus futuros usuarios es un desastre a punto de ocurrir".

7.3.7.- Seguridad

La seguridad se puede dar a nivel de usuario o de mejor manera a través de mecanismos de seguridad basados en roles que permiten la definición de los distintos niveles de derechos de acceso. Los usuarios están autorizados para ver y cambiar sólo los datos que necesitan para su trabajo. El nivel de detalle lo podemos definir a partir de la forma, materia y el nivel de fila. El nivel de seguridad por fila restringe el acceso a nivel de datos. Por ejemplo, un usuario solo puede ver las transacciones de las que él es responsable.

7.3.8.- Interfaces

El interfaz es el límite de comunicación de un sistema ERP. El interfaz de usuario se comento en el apartado de facilidad de uso mencionado anteriormente. Otro tipo de interfaces serán descritas a continuación. Se suele conectar el sistema ERP con otros sistemas o se usa para el intercambio de datos en general. El primero se conoce como Enterprise Application Integration (EAI) y usa interfaces de servidor estándar como CORBA (Common Object Request Broker Architecture), XML-RPC (XML-Remote Procedure Call) y SOAP (Standardized Object Access Protocol) para automatizar los procesos de negocio más allá de los límites del sistema. Pero también la integración a nivel de base de datos puede ser suficiente en especial para leer los datos únicos que no tiene se basa en la lógica de negocio. Como este tipo de integración es la única base de datos específica, no será evaluado aquí.

El envío y recepción de mensajes de correo electrónico y el manejo de archivos adjuntos de correo electrónico son importantes para las relaciones de comunicación de un CRM y las notificaciones de usuarios. Por parte del cliente, adjuntar archivos a los datos ERP, como documentos CAD, recepción de imágenes escaneadas o imágenes del producto deberán ser soportadas.

Otras interfaces a menudo utilizadas manualmente son la integración con office, la exportación e importación CSV e información general.

Las interfaces locales para las autoridades públicas y los bancos se abordarán cuando tomen parte del sistema. En general, la proporciona el soporte contratado.

7.3.9.- Independencia del sistema operativo.

La independencia del sistema operativo permita ejecutar el sistema ERP en varias plataformas. Es una característica necesaria para el cliente, si los usuarios tienen diferentes sistemas operativos.

7.3.10.- Independencia del sistema de bases de datos

Las bases de datos están altamente influenciadas por la escalabilidad del sistema, o dicho de otra manera por la habilidad para poder hacerse más grande sin perder calidad en sus servicios. Algunos prefieren las bases de datos de software libre para los sistemas ERP de software libre. Existen situaciones contrapuestas entre la independencia de las bases de datos y las características de las mismas. Una alta independencia de las bases de datos implica el uso mínimo de las características comunes proporcionadas por el soporte de bases de datos. Algunas características que se pierden por la independencia del sistema de bases de datos pueden ser proporcionadas a través de la aplicación o del servidor de aplicaciones.

7.3.11.- Lenguaje de programación

¿Qué es código abierto sin conocer la fuente del lenguaje? El lenguaje puede ser un criterio para aprovechar las posibilidades disponibles por el bajo nivel de personalización.

Los lenguajes de programación de los sistemas ERP seleccionados son lenguajes de scripting de código abierto (Python, Perl) y Java. Python es conocido por tener una sintaxis fácilmente legible y concisa así como sus capacidades integradas de refactorización (Refactoring es la reorganización del código fuente para mejorar la coherencia interna y la claridad). Perl es ampliamente utilizado, pero requiere más disciplina de desarrollo para obtener un código más profesional. Java cuenta con el apoyo fuerte de la industria y dispone de muchas herramientas de ingeniería de software.

El conteo de las líneas de código es un mal indicador de funcionalidad por las siguientes razones: los lenguajes de alto nivel de scripting necesitan menos líneas de código. La flexibilidad de los metadatos basados en el

diseño también necesita menos líneas de código a parte de que los metadatos pueden ser definidos en el código o externamente.

7.4.- Soporte

El soporte ayuda a acortar el tiempo de implementación debido a la transferencia de conocimientos de la empresa. Ayuda a desarrollar habilidades internas de la propia empresa o a través de consultores externos ayuda a implementar y mantener el sistema ERP open source.

7.4.1.- Infraestructura de Soporte

El soporte fiable y responsable es muy importante. Puede ser de manera local u on-line. La mayoría de proyectos de ERP de sw libre resuelven los problemas relativos a los diferentes requisitos a nivel nacional a través de redes asociadas al proyecto. Un socio (partner) local puede proporcionar servicios de consultoría, soporte, módulos complementarios y requisitos a nivel nacional tales como normas de contabilidad, interfaces con autoridades públicas y bancos. A parte de lo anteriormente citado también poseen conocimientos específicos de la industria. Soporte on-line en los foros públicos, sin censura y las listas de correo también son importante, porque ofrece a los usuarios y los desarrolladores oportunidad de leer y discutir temas.

7.4.2.- Formación

La calidad y la frecuencia de la formación técnica a nivel de usuario así como la organización de conferencias periódicas gozan de gran importancia.

7.4.3.- Documentación

Integridad, actualizaciones y documentación de desarrollo son completamente necesarias para el usuario. Muchos proyectos usan sistema de gestión de contenidos Wiki para la colaboración en la creación y mantenimiento de la documentación.

7.5.- Continuidad

La continuidad del proyecto asegura que los gastos del sistema ERP sean una inversión sostenida. Cuando una empresa se centra en un sistema, comentado en el libro (Advantages of using a flexible ERP Package), corre el riesgo de que éste nunca llegue a ponerse en práctica. Este problema lo

podemos denominar como Independencia de la estrategia de proveedor. La consolidación en el mercado ERP y los continuos cambios en la tecnología pueden forzar a los clientes a seguir la estrategia de producto del proveedor y por tanto el posible upselling (Estrategia de desarrollo de clientes que trata de maximizar la ganancia por venta y por cliente) o las costosas propuestas de migración de los vendedores de sistemas ERPs. Existe el riesgo de la suspensión del sistema a causa del proveedor, como por ejemplo una bancarrota del mismo o cambios en la tecnología. El software libre reduce el riesgo de la inversión ya que al ser código abierto el usuario es capaz de mantenerlo, aunque para obtener ventajas es importante el respaldo de empresas dedicadas a ese sistema en particular, así como una comunidad activa basada en el paquete ERP usado. Para actualizaciones peligrosas de las personalizaciones de los sistemas ERP se necesita un software de diseños flexible. Por otra parte cuando el proyecto es llevado por una única empresa, se corre el riesgo de que las nuevas versiones sean publicadas bajo licencias distintas. Incluso en este caso, las empresas open source tienen menos poder para llevar a cabo cambios de estrategia, porque existe el riesgo de que el proyecto se desvíe, cuando la estrategia cambia de rumbo, haciendo que los clientes quedan descontentos. Las compañías open source son muy dependientes de la comunidad de usuarios, ya que una pequeña parte de los usuarios están interesados en comprar servicios adicionales. Tanto las estrategias de venta como el blindaje de desarrolladores de la comunidad así como el frenado de las características de las versiones open source, junto con un fuerte enfoque en la venta de una versión comercial, puede perjudicar el crecimiento de la comunidad.

Una pequeña comunidad a su vez dificulta la venta de servicios tales como documentación adicional, formación, consultoría y certificaciones.

- La participación de la comunidad y el tamaño de la misma clasifican a los diferentes miembros de las comunidades on-line y los modelos de participación de la misma. Aplicado a los sistemas ERP open source existe cuatro categorías de socios o miembros pertenecientes a la comunidad que forman: los usuarios virtuales que son activos en foros, los testadores beta (beta testers) que proporcionan descripción de errores, los creadores de contenido que proporcionan documentación y especificaciones de requisitos, y los desarrolladores que proporciona mejoras del sistema. Cuanto más grande y activa es la comunidad de un proyecto ERP, menos es el riesgo de que el proyecto se abandone. No podemos tener en cuenta el número de clientes que utilizan sistemas de ERP de código abierto como indicador de continuidad ya que los clientes no necesitan dar cuentas a los creadores del proyecto. Si el proyecto se aloja en Sourceforge, una plataforma ampliamente utilizada para proyectos open source, entonces podemos usar las estadísticas proporcionadas como indicador. Para algunos resultados estadísticos sobre los proyectos ERP de código abierto alojados en Sourceforge, como el número de desarrolladores, vida útil, actividad CVS (sistema de control de

versiones), así como el número de descargas describen las medidas y las características del proyecto. La utilidad de las estadísticas de Sourceforge sin un análisis detallado del proyecto hace que sean cuestionables. Algunas razones como por ejemplo que hay proyectos o bien que no usan los servicios que les ofrecen o bien los usan sólo en parte, otra razón sería que no mantienen la web Sourceforge, esto lo podemos explicar con el siguiente ejemplo, los desarrolladores cambian las versiones del sistema de CVS que ofrece Sourceforge a una nueva versión sin borrar los datos antiguos del CVS.

- El número de mensajes en las lista de correo o en los foros es un indicador medible. Es más importante para la estimación de la continuidad, la satisfacción del usuario así como el desarrollo de la comunicación. Obteniendo también una pista de la madurez del proyecto. Un sistema que sea bueno y que tenga uso es complicado que desaparezca en la comunidad.

7.5.1.- Estructura del proyecto

La evaluación de los proyectos son impulsados por empresas o comunidades. Que una empresa los impulse quiere decir que una empresa es la responsable del desarrollo, proporciona servicios y asegura a los socios soporte a nivel local. Una típica empresa impulsadora tiene los siguientes participantes: empresa con proyecto open source, empresas asociadas, clientes con contratos de soporte, clientes sin contrato de soporte y usuarios que trabajan con el sistema.

Una comunidad “impulsadora” (community driven) significa que el desarrollo es cooperativo y no existe una única empresa responsable.

7.5.2.- Actividad de la comunidad

El tamaño de una comunidad no es medible, pero sí su actividad comunicativa en ciertos canales de comunicación. Aquí se utilizan el número de mensajes en foros y las listas de correo. A parte de la cantidad, las respuestas competentes y los tiempos de respuestas de las mismas son importantes. La creación de sitios web, así como las entradas en la Wikipedia son una parte del soporte y la documentación.

7.5.3.- Transparencia

Trata sobre las barreras que pueden tener los desarrolladores y las posibilidades para la comunidad de contribuir e influenciar los procesos, la calidad de la gestión del proyecto, así como la documentación del proceso de desarrollo. Un fiable plan de trabajo documentado ayuda a estimar el foco

actual y la orientación futura del proyecto. Una de las razones del porqué algunos proyectos no tienen un plan de trabajo detallado con horarios de trabajo es que la nueva funcionalidad necesita ser patrocinada. Como los desarrolladores son profesionales, un cliente necesita contratarlos para implementar ciertas funcionalidades, excepto si es visto como esencial para el proyecto inicial. Un sistema público de seguimiento informa sobre detalles de errores y el tiempo que se tardará en solucionarlos, las características previstas y sus prioridades sobre estas últimas.

Sobre todo cuando el proyecto es impulsado por una empresa, es interesante saber y de que manera se entiende la participación activa del desarrollo por la empresa. El grado de participación de la comunidad en el proceso de desarrollo constituye otro factor de independencia del fabricante, en este caso la independencia de la compañía open source o los líderes del proyecto. Con el acceso al código de versiones del sistema junto con la documentación técnica podemos estimar las posibilidades de participación de forma activa en el proceso de desarrollo. El código fuente tiene que ser legible y documentado.

Las versiones del código fuente del sistema necesitan un registro de los cambios de forma detallada y comprensible, para poder ser utilizadas. La documentación de herramientas de desarrollo y construir procedimientos de ayuda a los nuevos desarrolladores de manera que se involucren en el proyecto con mayor facilidad. La gestión del código aportado es especialmente importante para tener bien acoplados los sistemas ERP destinados a cubrir una amplia variedad de requisitos. Aparte del control de calidad del código y su encaje en la arquitectura del software que asegura que la nueva funcionalidad es para el uso general de la comunidad y no enfocada a un cliente en particular.

7.5.4.- Frecuencia de las actualizaciones

La introducción continua de nuevas funcionalidades y el arreglo de los errores son una prueba de la continuidad del desarrollo. Un documento de información sobre modificaciones informa acerca de las características de las nuevas versiones mostrando la actividad de actualización pasada. Considerando que la actividad de la comunidad se basa en la comunicación, las actualizaciones periódicas muestran la actividad del desarrollo.

7.5.5.- Otros efectos acordados

Además de los acuerdos en el mismo proyecto, los posibles efectos secundarios pueden derivar del uso (por ejemplo comerciales) de componentes, tecnologías o dependencias en otros proyectos open source. La independencia del sistema operativo, de las bases de datos y del lenguaje

programación, los cuales se comentaron en el apartado de flexibilidad citado anteriormente, son también criterios acordados.

7.6.- Madurez

Introduce el Modelo de madurez de open source (Open Source Maturity Model) como un proceso general para la selección, evaluación e implementación de productos open source. Aquí madurez se utiliza en un contexto más estrecho y su significado se basa en la calidad del software. Partiendo de que la flexibilidad se basa en conceptos técnicos y en el diseño del software, la madurez nos dice lo bien y libre de errores que ha sido implementado y probado.

7.6.1.- Estado del desarrollo

Algunos paquetes ERP de código abierto no están listos para la producción todavía. El concepto del estado de desarrollo de Sourceforge también es aplicado a los proyectos open source no alojados en Sourceforge. Pueden estar en el estado de planificación, alfa, beta o estables. Planificación implica que las especificaciones del software se han definido pero no está disponible el programa ejecutable. La primera versión de un programa de ordenador se llama versión alfa o alfa release. Es probable que sea inestable e incompleta, pero útil para fines de demostración y como prototipo que se seguirá desarrollando. La versión beta o beta release es una versión de una aplicación que está todavía en desarrollo, pero publicada para fines de comprobación o testeo. La funcionalidad no ha sido plenamente probada y pueden aparecer errores. Después de que una versión beta ha sido ampliamente probada y los errores importantes solucionados, el programa se convierte en una versión estable. Sólo entonces se admiten errores que no afecten a la funcionalidad.

7.6.2.- Lugar de referencia

La calidad de una versión estable puede ser probada por la implementación y el amplio testeo del software. Existe el riesgo que el sistema resulte ser inadecuado. Por lo tanto es mejor ver el sistema ERP en práctica y discutir las cuestiones de implementación y operaciones con el cliente que ya conoce el sistema. Los criterios relevantes son los lugares de referencia que aparecen en la página web del proyecto y la disponibilidad de casos de negocio documentados.

Características ERPs analizados en profundidad

8.- Características ERPs analizados en profundidad

- 8.1. – Introducción
- 8.2. – Tablas comparativas
- 8.3. - Características ERPs analizados
 - 8.3.1.- Abanq
 - 8.3.1.1.- Flexibilidad
 - 8.3.1.2.- Soporte
 - 8.3.1.3.- Continuidad
 - 8.3.1.4.- Madurez
 - 8.3.2.- Adempiere
 - 8.3.2.1.- Flexibilidad
 - 8.3.2.2.- Soporte
 - 8.3.2.3.- Continuidad
 - 8.3.2.4.- Madurez
 - 8.3.3.- Compiere
 - 8.3.3.1.- Flexibilidad
 - 8.3.3.2.- Soporte
 - 8.3.3.3.- Continuidad
 - 8.3.3.4.- Madurez
 - 8.3.4.- Oasis erp
 - 8.3.4.1.- Flexibilidad
 - 8.3.4.2.- Soporte
 - 8.3.4.3.- Continuidad
 - 8.3.4.4.- Madurez
 - 8.3.5.- Openbravo
 - 8.3.5.1.- Flexibilidad
 - 8.3.5.2.- Soporte
 - 8.3.5.3.- Continuidad
 - 8.3.5.4.- Madurez
 - 8.3.6.- Openerp
 - 8.3.6.1.- Flexibilidad
 - 8.3.6.2.- Soporte
 - 8.3.6.3.- Continuidad
 - 8.3.6.4.- Madurez
 - 8.3.7.- OpenXpertya
 - 8.3.7.1.- Flexibilidad
 - 8.3.7.2.- Soporte
 - 8.3.7.3.- Continuidad
 - 8.3.7.4.- Madurez
- 8.4. – Conclusiones a partir de los puntos anteriores
 - 8.4.1. – Flexibilidad
 - 8.4.2. – Soporte
 - 8.4.3. – Continuidad

- 8.4.4. – Madurez
- 8.4.5. - Conclusión
- 8.5. – Instalación Openbravo
 - 8.5.1. – Requisitos de intalación
 - 8.5.2. – Entorno
 - 8.5.2.1. - Componentes
 - 8.5.2.2. – Módulos
- 8.6. - Instalación Abanq
 - 8.6.1. – Requisitos de intalación
 - 8.6.2. – Entorno
 - 8.6.2.2. – Módulos
 - 8.6.2.2. – Área de facturación
 - 8.6.2.2. – Área financiera
- 8.7. – Conclusiones instalación

8.1. – Introducción

La información que se proporciona está basada en los recursos online proporcionados por cada sistema ERP. En primer lugar se realiza una comparación global de todos los sistemas a analizar, posteriormente se examinarán de forma más detallada. Los criterios comparativos están basados en el contexto de las definiciones del apartado 7, criterios comparativos ERPs. Los términos relativos se basan en las diferencias entre los sistemas analizados entre ellos, todas las afirmaciones de este capítulo son aportaciones o conclusiones propias, originadas o bien por la documentación hallada o bien por las experiencias observadas.

Con respecto a la elección de los paquete ERPs, está basado en una búsqueda bajo criterios de importancia a nivel mundial, nacional y regional, escogiendo las aplicaciones más interesantes a analizar, obviamente la existencia de más aplicaciones de las que aquí tratamos merecen un estudio, pero el muestreo intenta ser lo suficientemente significativo para lograr una idea global en la elección del mismo.

8.2. – Tablas comparativas

Leyendas:

Ok	→	sí
X	→	no
n/a	→	no está disponible
¿?	→	Desconocido
Medio	→	Situado entre sí y no

Criterio de evaluación		ERP DE SOFTWARE LIBRE						
	Sub-Criterio	Abanq	Adempiere	Compiere	Oasis ERP	Openbravo	Openerp	Openxpertia
TAMAÑO								
1	Micro	ok	ok	ok	ok	ok	ok	x
2	Pequeña	ok	ok	ok	ok	ok	ok	x
3	Media	x	ok	ok	ok	ok	ok	ok
4	Grande	X	x	ok	x	x	x	ok
FUNCIONALIDAD								
1	Número de Tablas	n/a	2631 DB calls detected	2056 DB calls detected	n/a	n/a	n/a	+700 tablas
2	e-Comercio	ok	ok	ok	ok	ok	x	ok
3	Contabilidad	ok	ok	ok	ok	ok	ok	ok
4	Mrp	ok	ok	ok	ok	ok	ok	ok
5	Pos (tpv o terminal de punto de venta)	ok	ok	ok	ok	Versión pos independiente	Ok (multi)	ok
6	Inventario y Almacén	ok	ok	ok	ok	ok	ok	ok



Criterio de evaluación		ERP DE SOFTWARE LIBRE						
	Sub-Criterio	Abanq	Adempiere	Compiere	Oasis ERP	Openbravo	Openerp	Openxpertia
FLEXIBILIDAD								
1	Personalización	Alto nivel (metadatos)	Alto nivel (metadatos)	Alto nivel (metadatos)	Bajo nivel	Alto nivel (metadatos)	Bajo nivel	Bajo nivel
2	Flexibilidad de las Actualizaciones	Medio	ok	ok	Medio	Ok	Medio	Medio
3	Internacionalización	Español, inglés	multiidoma	multiidoma	multiidoma	Multiidioma	multiidioma	Multilidioma a nivel español.
4	Facilidad de uso	ok	ok	ok	no	Ok	ok	ok
5	Arquitectura	A3d	2 capas cliente/servidor previsión 3 capas	2 capas cliente/servidor previsión 3 capas	3 capas	MVC (3 capas)	2 capas cliente/servidor	3 capas
6	Escalabilidad	Buena, limitada por PostgreSQL	ok	ok	ok	Altamente escalable	ok	ok
7	Seguridad	ok	ok	ok	ok	ok	ok	ok
8	Interfaz	csv	CSV, OAGSIS, OFX	csv	csv	csv	csv	n/a
9	Independencia del S.O	Linux, Mac OS X y Windows	Unix, Windows, Linux y Mac OS X	Unix, Windows, Linux y Mac OS X	Linux, Windows	GNU/Linux y windows	Linux/windows	Windows, Solaris, FreeBSD, Linux, UNIX, AIX, MacOS
10	Independencia de la Base de datos	PostgreSQL, MySQL	SQLJ	ORACLE	PostgreSQL	PostgreSQL y Oracle	PostgreSQL	Oracle, actualmente, PostgreSQL
11	Lenguaje de Programación	C++, Javascript	Java	Java	PHP-GTK2	JDK	Python, PyGTK	J2EE

Criterio de evaluación		ERP DE SOFTWARE LIBRE						
	Sub-Criterio	Abanq	Adempiere	Compiere	Oasis ERP	Openbravo	Openerp	Openxpertia
SOPORTE								
1	Infraestructura de soporte	ok	ok	ok	ok	ok	ok	ok
2	Formación	ok	ok	ok	ok	ok	ok	ok
3	Documentación	ok	ok	ok	ok	ok	ok	ok
CONTINUIDAD								
1	Estructura del proyecto	Compañía+socios	Compañía + Comunidad	Compañía + socios	Desarrollado por la comunidad de castilla la mancha	Compañía + socios + comunidad	Compañía + socios + comunidad	Compañía + socios (COMUNIDAD)
2	Actividad de la Comunidad	ok	ok	ok	n/a	ok	ok	ok
3	Transparencia	ok	ok	ok	n/a	ok	ok	ok
4	Frecuencia de Actualizaciones	X	ok	ok	n/a	ok	ok	ok
5	Otras efectos acordados	n/a	n/a	Herramientas de migración	n/a	n/a	n/a	n/a
MADUREZ								
1	Estado del desarrollo	Estable	ESTABLE	estable	Estable	Estable	Estable	Estable
2	Lugares de referencia	ok	ok	ok	ok	ok	ok	ok

Criterio de evaluación		ERP DE SOFTWARE LIBRE						
	Sub-Criterio	Abanq	Adempiere	Compiere	Oasis ERP	Openbravo	Openerp	Openxpertya
OTROS								
1	Licencia	GPL, GNU General Public License Version 2.	GPL	MPL,CPL	GNU/GPL	OBPL	GPL	lpo
2	Demos online	ok	ok	ok	ok	ok	ok	no
3	Alojado en sourceforge	ok	ok	ok	ok	ok	ok	ok
4	CVS (sistema de control de versiones) acceso	ok	ok	ok	Nx	ok	x	x
5	Descargas del checksum	-	ok	ok	x	ok	x	x
6	Comienzo del proyecto	2007 (a partir de facturalux)	2006 (separación Compiere)	1999	2009	2001 (2006 liberaliza el código)	2008	2001, 2005 asentado.

8.3. - Características ERPs analizados

8.3.1.-Abanq

Esta sección está basada en las características del sistema ERP de software libre Abanq. AbanQ es un ERP de código libre orientado a la administración, gestión comercial, finanzas y en general a cualquier tipo de aplicación donde se manejen grandes bases de datos y procesos administrativos. Su aplicación abarca desde la gestión financiera y comercial en empresas hasta la adaptación a procesos complejos de producción.

Abanq		
Sub-criterio		Descripción
FLEXIBILIDAD		
1	Personalización	Alto nivel, con la edición de metadatos.
2	Flexibilidad de las Actualizaciones	Dependiendo de las personalizaciones.
3	Internacionalización	Bajo nivel.
4	Facilidad de uso	Alta facilidad de uso a todos los niveles
5	Arquitectura	A3D
6	Escalabilidad	Buena, limitada por PostgreSQL
7	Seguridad	Buen nivel de acceso (usuarios y grupos)
8	Interfaz	XML, CSV
9	Independencia del S.O	Linux, Mac OS X y Windows
10	Independencia de la Base de datos	PostgreSQL, MySQL
11	Lenguaje de Programación	C++, JavaScript, QSA
SOPORTE		
1	Soporte de la infraestructura	Foros, web, contratos de soporte.
2	Formación	A nivel de usuario y de programador
3	Documentación	A nivel de usuario, programador e instalación.
CONTINUIDAD		
1	Estructura del proyecto	ok
2	Actividad de la Comunidad	ok
3	Transparencia	X
4	Frecuencia de Actualizaciones	X
5	Otras efectos acordados	-
MADUREZ		
1	Estado de desarrollo	Estable
2	Lugares de referencia	ok

8.3.1.1.- Flexibilidad

Personalización

El motor de Abanq lee e interpreta los metadatos (información sobre tablas, formularios, scripts, etc.) que el Sistema Gestor de Base de Datos (SGDB) le proporciona. Estos metadatos se organizan en lo que llamamos módulos. Cada módulo agrupa un conjunto de metadatos que implementan una funcionalidad concreta (facturación, almacén, etc.).

Flexibilidad de las actualizaciones

La flexibilidad de las actualizaciones en Abanq depende del grado de personalización propio que la aplicación tenga, es decir que a más personalización más complicada será la actualización a nuevas versiones.

Mientras que un usuario sólo haga uso de los módulos de la rama oficial sin ninguna modificación, el problema de actualizar a nuevas versiones oficiales es mínimo ya que sólo deberá cargar estas versiones de los módulos conforme sean liberadas. Es decir, estos nuevos módulos están preparados para funcionar correctamente siempre y cuando sólo se usen versiones oficiales sin ninguna modificación y de las que se sabe en todo momento su comportamiento.

La situación es algo más compleja si un usuario tiene versiones de módulos modificadas respecto a la rama oficial. Las nuevas versiones de la rama oficial sólo se instalan con total garantía cuando todas las versiones anteriores sean también oficiales. En el momento que tengamos alguna modificación estamos abriendo un nuevo camino, una nueva rama personalizada.

En una rama personalizada las actualizaciones tienen que tener en cuenta las diferencias con la rama oficial, y se deberá crear una versión que respete esas diferencias que cubren ciertas necesidades específicas y que a la vez aproveche las mejoras de la rama oficial. A las diferencias a nivel de código entre la rama oficial y una rama personalizada se denominan parche. Esta simple expresión aritmética nos ofrece una idea representativa de los que es un parche: rama oficial + parche = rama personalizada. En la práctica un parche es un conjunto de textos con código de distintos lenguajes utilizados por AbanQ y que contiene la funcionalidad o funcionalidades específicas.

La solución que adoptan, consiste en la detección de esos parches y tratar de resolver los conflictos que surgen con la nueva versión.

Internacionalización

El grado de internacionalización mostrado por la aplicación Abanq es a nivel de idioma, reflejando su página web, una traducción completa de los

módulos en alemán, castellano y catalán, al 90% al inglés y en proceso a idiomas como el euskera y el portugués. Los requisitos legales a nivel nacional, especialmente en la contabilidad no han sido especificados. Los criterios anteriormente citados ayudarán en un futuro a la internacionalización a un alto nivel, así como a su expansión.

Facilidad de uso

En este apartado hay mucho que comentar ya que la aplicación, a pesar de su completitud y relativa complejidad es muy fácil de usar incluso para usuarios noveles o con pocos conocimientos de contabilidad.

En cuanto a diseño la aplicación sigue un mismo patrón de diseño para todo, pero esto influye directamente en la funcionalidad, ya que los procesos usando el programa son siempre iguales, facilitando enormemente la tarea al usuario. La manera de moverse por los menús, de acceder a las opciones, de crear un dato (ya sea un cliente o un artículo) o de rellenar los campos son siempre iguales. Se trata de un sistema homogéneo. Además hay muchos detalles positivos que se pasan a comentar.

Los iconos para realizar las acciones no solo son representativos, si no que a veces se agrupan por la misma tonalidad de color, así, se sabe que las acciones en “verde” son las relativas a clientes y las “moradas” a proveedores, por ejemplo. Rellenar formularios puede ser uno de los quebraderos de cabeza de los empleados si tienen que crear decenas o cientos de entradas. Para ello, como se ha explicado, existe una conexión entre módulos que facilita el rellenado de los campos, mostrando listados de opciones ya creadas anteriormente para el campo que se quiere rellenar. Pero además si el usuario quiere más rapidez a la hora de rellenar estos campos, y hacer dos o tres clicks le resulta incómodo, puede introducir directamente el número de registro, que seguramente ya conozca si un campo se repite mucho o trabaja mucho con él. Para más facilidad, si al desplegar la lista no hay ninguna entrada creada, está la opción de crear en ese instante ese nuevo dato mediante un nuevo formulario, por lo que todo está indexado. Además AbanQ en sus formularios incluye siempre la opción de almacenar el registro que se acaba de rellenar y pasar a crear otro nuevo del mismo tipo. Así, se abre un nuevo formulario directamente para empezar con la creación de una nueva entrada, lo que facilita la vida si se tienen que crear, por ejemplo, 100 clientes. Otras virtudes de usabilidad en los listados donde aparecen todas las entradas creadas (por ejemplo una lista de artículos), son la inclusión de un campo para realizar búsquedas, o mostrar datos utilizando un filtro.

Otra opción útil que se incluye en la ventana de todos los módulos es la de saltar directamente a otro módulo instalado, lo que facilita la navegación por el programa.

Además para ciertos casos donde se necesitan datos creados para realizar acciones, la aplicación da la opción de trabajar con datos de prueba, lo que facilita la práctica con la aplicación sobre todo si la usamos por primera vez. Por ejemplo, se puede trabajar con una empresa de prueba.

Arquitectura

En la figura se observa el esquema general de la arquitectura Abanq. Vemos como todo se almacena en la base de datos y sólo el servidor puede acceder directamente a ella, sirviendo a los clientes los datos y los módulos de aplicación, y gestionando el control de acceso a los usuarios.

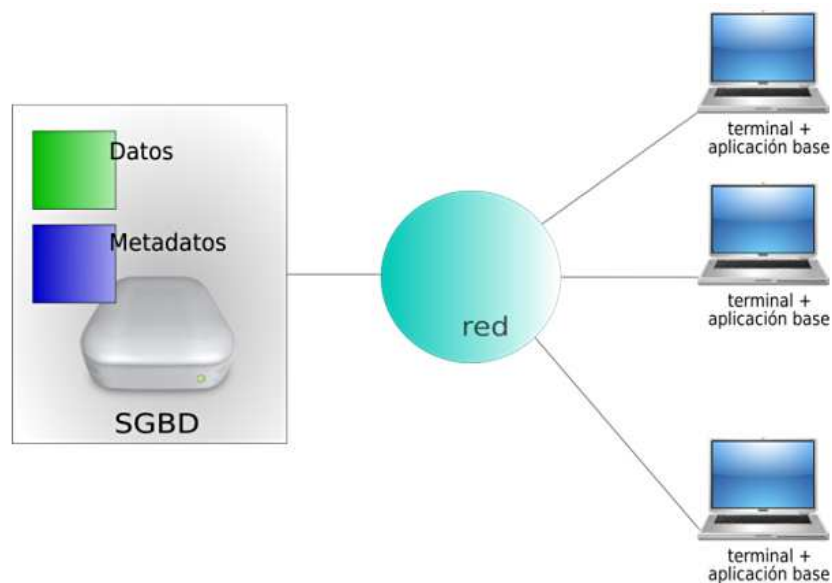


Figura 8.1.- Arquitectura Abanq

Estructura A3D

Cientes: Los clientes son las máquinas que se encuentran conectadas directamente al SGBD (sistema gestor de base de datos) pudiendo acceder a la base de datos. En cada terminal o máquina cliente se ejecuta el software que denominamos aplicación base.

SGBD: El gestor de base de datos se encarga de almacenar y mantener dos tipos de información (y aquí está la clave):

Datos: En esta zona de la base de datos se almacenan los datos concretos que la aplicación maneja y que tienen sentido para el usuario (datos de clientes, facturas, etc.). Son los datos "tradicionales".

Módulos de metadatos: Los módulos contienen la información necesaria para implementar las aplicaciones de usuario: formularios, definiciones de tablas y campos, código de los scripts que realizan los procesos, formato y definición de los informes.

Los metadatos residen en la base de datos, pero previamente deben ser cargados desde el directorio en disco en el que han sido alojados tras su descarga.

La estructura de directorios en los módulos presenta cuatro niveles:

- Nivel 1. Directorio raíz (ejemplo: directorio modulos)
- Nivel 2. Área (ejemplo: directorio facturacion)
- Nivel 3. Módulo (ejemplo: directorio almacen)
- Nivel 4. Metadatos (ejemplo: directorio tables)

En el nivel 4 existen varios directorios, uno por cada tipo de metadatos:

- tables. Definiciones de las tablas. Cada tabla se define en un archivo de extensión mtd
- forms. Definiciones de los formularios. Cada formulario se define en un archivo de extensión ui
- scripts. Definiciones de los scripts. Cada script se define en un archivo de extensión qs
- queries. Definiciones de las consultas. Cada consulta se define en un archivo de extensión qry
- reports. Definiciones de los informes. Cada informe se define en un archivo de extensión kut
- translations. Listados de traducciones. Cada listado de traducciones para un determinado idioma se define en un archivo de extensión ts

Seguridad

El módulo de control de acceso proporcionado para AbanQ permite establecer permisos sobre distintas partes de la aplicación dependiendo del usuario que accede. Este módulo sólo funciona a partir de la versión 2.0 de la aplicación base.

El módulo permite definir usuarios y grupos de usuarios y establecer para ellos permisos de lectura, escritura o lectura-escritura sobre las tablas de datos y formularios del sistema hasta el nivel de campo, permitiendo controlar el acceso de cualquier usuario a cualquier parte de la aplicación.

8.3.1.2.- Soporte

Soporte de la infraestructura

Hay un foro (<http://abanq.org/laboratorio/minibb/index.php>) para transmitir todo tipo de problemas, dudas o sugerencias, que además se divide en una zona para usuarios y otra para desarrolladores. También se

puede contactar vía formulario (<http://abanq.org/infosial/contactar.php9>) o con el webmaster <http://abanq.org/infosial/webmaster.php>.

A través de su página cualquier usuario registrado puede realizar preguntas, resolver dudas de la aplicación, en el caso de que esa información sea más especializada, la opción es el contrato de paquetes de horas de soporte, proporcionando mayor calidad en las actividades de soporte, pero con el coste económico que conlleva.

Formación

Al igual que el soporte anteriormente mencionado, en su página Web existen accesos a formación a distinto nivel, debiéndose poner en contacto, de manera que se acuerden las condiciones, ya sea de desplazamiento de uno o de otros. Otra opción es la de cursillos online, ambas opciones conllevan coste económico.

Documentación

Existe documentación a nivel de usuario y de programador, en mi opinión no es demasiado amplia y bastante costosa de entender, la cual cosa conlleva de una formación extra casi obligada.

8.3.1.3.- Continuidad

Estructura del proyecto

Bajo la marca Abanq G2 se ha constituido el primer Grupo Empresarial para el Desarrollo del Open Source, S.A., que ha sido fundado por empresas españolas de software para la investigación, desarrollo y colaboración en proyectos informáticos relevantes para las Empresas y la Administración y que hasta ahora sólo estaban al alcance de grandes corporaciones.

Inicialmente son cinco las empresas que forman Abanq G2: Calidae, Gestweb, InfoSiAL, Isolix y KLO, aunque la estructura es flexible y está orientada a facilitar nuevas incorporaciones, siempre que se demuestren capacidades e implicación decidida en el Proyecto.

Actividad de la comunidad

A través del foro alojado en su Web se deduce un flujo constante de información entre usuarios pertenecientes a la misma.

Frecuencia de las Actualizaciones

No existe suficiente información para comentar este punto ya que tanto en su página principal, así como en sourceforge, no queda claro este punto, la última versión data de 2008. Pero en su página ya existe la versión 2.3 para descargar, que data del mismo año. Tras la formación de Abanq G2 se espera una nueva versión.

8.3.1.4.- Madurez

Lugares de referencia

Basicamente son estos:

www.abanqg2.com
<http://www.abanq.org/>

8.3.2.- ADempiere

Esta sección está basada en las características del sistema ERP de software libre ADempiere.

Adempiere		
Sub-criterio		Descripción
FLEXIBILIDAD		
1	Personalización	ok
2	Flexibilidad de las Actualizaciones	ok
3	Internacionalización	Multiidoma
4	Facilidad de uso	ok
5	Arquitectura	2 capas cliente/servidor previsión 3 capas
6	Escalabilidad	ok
7	Seguridad	ok
8	Interfaz	CSV, OAGSIS, OFX
9	Independencia del S.O	Unix, Windows, Linux y Mac OS X
10	Independencia de la Base de datos	SQLJ
11	Lenguaje de Programación	Java
SOPORTE		
1	Soporte de la infraestructura	ok
2	Formación	ok
3	Documentación	ok
CONTINUIDAD		
1	Estructura del proyecto	Compañía + Comunidad
2	Actividad de la Comunidad	ok
3	Transparencia	ok
4	Frecuencia de Actualizaciones	ok
5	Otras efectos acordados	-
MADUREZ		
1	Estado del desarrollo	ESTABLE
2	Lugares de referencia	ok

8.3.2.1.- Flexibilidad

Personalización

Además de la posibilidad de personalizar las Interfaces de Usuario, Reportes y Extensiones, ADempiere proporciona capacidades de personalización adicionales:

- Preferencias Default o elecciones preseleccionadas
 - Preferencias de Login: Organización, Lenguaje, Fecha de Transacciones e Impresora.
 - Preferencias definidas por el Usuario, tales como tipos de transacciones específicas.
- Personalización de la Barra de Menú, permitiendo guardar cualquier entrada en la barra (Ventanas, Procesos, Reportes) como un acceso rápido.
- La Terminología puede ser cambiada. Por ejemplo si los usuarios en lugar de Productos utilizan Ítems
 - Artículos, o a la Organización la denominan Sucursal, etc.
- Los Textos de Ayuda pueden ser modificados y extendidos por el usuario para proporcionar sugerencias y ayudas específicas.

Las personalizaciones son definibles a diferentes niveles:

- Sistema o implementación
- Ventana, si es apropiado (por Ej. para preferencias)
- Cliente
- Organización
- Usuario Específico

Los niveles más específicos tienen preponderancia sobre los más bajos. Los cambios efectuados a nivel del sistema pueden ser guardados y definirse como personalizaciones si se requieren reaplicar luego de la instalación.

Las extensiones funcionales son implementadas utilizando la tecnología de "callout". Los clientes pueden proporcionar funcionalidad adicional en Java o inclusive funcionalidad nativa en C, por ejemplo para validaciones adicionales o alimentación de datos. Los callouts pueden ser invocados antes o después del ingreso de datos en cualquier campo. ADempiere asegura que los callouts no permitan la caída o corrupción del sistema.

Flexibilidad de las actualizaciones

Además del diccionario interno de la aplicación, ADempiere tiene también la posibilidad de extender la aplicación utilizando Java Business API's. A diferencia de otras aplicaciones, las extensiones de clientes son posibles en ambientes con datos centralizados y son preservadas durante las actualizaciones del producto a nuevas versiones.

Internacionalización

Adempiere está traducido a los siguientes idiomas: Árabe, Bosnio, Brasileño, Portugués, Catalán, Chino (Simplificado), Inglés, Alemán, Indonesio, Italiano, Japonés, Rumano, Ruso, Español, Tailandés.

A nivel de módulos de adecuación a la economía de cada país, están en proyecto o finalizados, a nivel español, existen reseñas del proyecto pero nada concluyente, el intento de contactar con las empresas españolas que en teoría llevan el proyecto, no ha dado fruto alguno.

Facilidad de uso

A causa del alto grado de personalización del ERP Adempiere, el interfaz de usuario puede ser diseñado con la información necesaria para cada tarea, es lo que llaman interfaz de usuario inteligente. Como resultado se obtiene una interfaz de usuario consistente, que permite navegar rápidamente en áreas de la aplicación que no son familiares. Este método también permite que el layout (esquema de distribución, lógico y ordenado de un sistema) de las pantallas pueda ser modificado o extendido y que se puedan generar nuevas pantallas, creadas por el administrador del sistema, sin necesidad de modificar el código; los usuario automáticamente ven las nuevas pantallas la próxima vez que ingresen a la aplicación.

ADempiere proporciona un sistema de ayuda multinivel integrado y personalizable. Cada tarea, reporte y ventana, tiene información general, cada campo tiene un “tool-tip” (ventana de ayuda contextual) hint y un texto de ayuda.

Si la ayuda no es suficiente, el usuario puede hacer un “zoom” desde la ventana de ayuda hacia el sistema de ayuda online de ADempiere, para obtener información actualizada, consejos y secciones de Preguntas Frecuentes (FAQ).

Arquitectura

ADempiere es una aplicación cliente-servidor escrita enteramente en Java, que soporta el procesamiento de grandes volúmenes de información y una interfaz gráfica de usuario con un alto grado de diseño.

Está prevista en un futuro la migración hacia una arquitectura de 3 capas.

El servidor de aplicaciones está implementado en Java, con la tecnología J2EE, utilizando la infraestructura del servidor de aplicaciones JBoss. Este servidor puede estar corriendo de manera stand alone o en el mismo equipo que el servidor de la base de datos. Para la administración del servidor se utiliza JMX (Java Management Extensions). El acceso a la base de datos se realiza mediante el protocolo JDBC (Java Database Connectivity). Está planificado que futuros releases de ADempiere soporten otros servidores de aplicaciones que cumplan con las especificaciones de J2EE (por Ej. IBM Websphere, Oracle Application Server, etc.). Además del estándar HTTP, se utiliza el protocolo SSL para la implementación de la funcionalidad Web Store.

Los componentes de la aplicación Cliente están escritos enteramente en Java, diseñados para utilizar las capacidades que brindan los PC's actualmente. La aplicación Java o cliente Java Applet es la elección ideal para altos volúmenes de datos y proporciona una interfaz gráfica de usuario alta en diseño. Se comunica vía thin JDBC (Java Database Connectivity) con la base de datos y mediante RMI (Remote Method Invocation) con el servidor de aplicaciones. El cliente puede acceder a los servidores a través de Internet o de una Intranet.

En aquellos casos donde la instalación o descarga de la aplicación no sea posible, es posible utilizar un cliente HTML. Este está implementado mediante Java Servlets y Java Server Pages almacenadas en Servidores de Servlet. Si bien este cliente proporciona mucha funcionalidad, es menor a la soportada por el cliente Java Applet.

Seguridad

ADempiere proporciona una infraestructura de seguridad completa y flexible para cumplir con las necesidades del usuario, y soporta función, seguridad de datos, como así también auditoria. La función de seguridad está basada en Roles de Usuario, la cual controla el acceso a Ventanas, Reportes y Procesos.

Por otro lado, la seguridad de los datos está basada en Cliente y Organización, y es mantenida a nivel del contexto de seguridad de la base de datos. Este es un nivel adicional de seguridad posterior al login normal de usuario de la base de datos. Antes de acceder a cualquier dato, el usuario debe identificarse mediante un store procedure con un nombre de usuario, contraseña, rol y opcionalmente la preferencia de lenguaje. Todas las contraseñas se almacenan de manera encriptada.

La funcionalidad de auditoria incluye registro de accesos (que funciones/datos fueron utilizados), registro de cambios (que datos fueron cambiados, que valor tenían y cuales se establecieron; incluyendo datos borrados), como así también archivos (documentos y reportes generados).

8.3.2.2.- Soporte

Soporte de la infraestructura

Una solución de software libre, muchas veces es asociada con un menor costo a lo largo de todo su ciclo de vida, pero también es percibida con un menor nivel de soporte y un alto riesgo, comparada con un sistema propietario. Este no es el caso justamente.

El nivel de soporte proporcionado por organizaciones de software libre, puede ser considerablemente superior que el proporcionado por un revendedor que distribuye aplicaciones de software propietarias. El primero motivo es que el código fuente está disponible, y por lo tanto puede ser modificado para resolver el problema localmente, a diferencia de los productos propietarios donde el código fuente normalmente no está al alcance de la organización que brinda el soporte; éstos dependen de su desarrollador para proporcionar una corrección. Y esto generalmente se hace en una nueva versión, unos seis a doce meses más tarde.

Adicionalmente, es posible obtener soporte entre la comunidad de desarrolladores, partners y usuarios del software, los cuales responden a las consultas realizadas en los foros, muchas veces en cuestión de horas e inclusive de minutos de realizado el requerimiento.

Además del soporte, la mayoría de las organizaciones buscan obtener “garantías” de que el software adquirido está libre de defectos, o en caso de existir alguno, el mismo se solucionará rápidamente. La historia reciente y la experiencia indican que comprar un software a un proveedor no es garantía de libertad de errores.

La realidad es que en soluciones de software libre, la lista de errores es conocida y el código fuente está disponible para la organización de soporte, lo cual le permite corregir cualquier error que surja. Este no el caso del software propietario, donde generalmente los errores no se publican y el código fuente no está disponible para las organizaciones que lo distribuyen y dan soporte.

Adempiere brinda soporte de dos maneras:

- 1) Soporte comercial brindado por muchas organizaciones.
- 2) Soporte gratuito, disponible en:

- a) Comunidad de Desarrolladores
- b) Listas de correo
- c) Archivos
- d) Base de datos de soporte

Formación

Al igual que el soporte anteriormente mencionado, en su página Web existen accesos a formación a distinto nivel, debiéndose poner en contacto, de manera que se acuerden las condiciones, ya sea de desplazamiento de uno o de otros. Otra opción es la de cursillos online, ambas opciones conllevan coste económico.

Documentación

En este caso nos ofrecen tanto tutoriales para la instalación como para el posterior manejo y una documentación bien estructurada clara y que hará más fácil la implantación de este tipo de solución. Tanto en su página web, como en todas las empresas de consultoría que trabajan con ADempiere en cada país existe una amplia gama de documentación, tanto a nivel usuario como a nivel de programación.

8.3.2.3.- Continuidad

Estructura del proyecto

Al ser un proyecto basado en la comunidad todos tienen el derecho de emitir su opinión y de hecho es algo que se promueve activamente. Sin embargo, al ser una comunidad con miembros en más de 17 países y a través de 5 continentes, para fines prácticos el proyecto está guiado por un Consejo de Contribuidores. Este Consejo es liderado por una persona que actúa como Director general del proyecto. El rol del consejo es:

- respaldar las decisiones del líder
- aceptar aportes
- definir el plan a seguir
- revisar y aprobar especificaciones
- votar por nuevas funcionalidades
- aprobar cambios en el núcleo

Actividad de la comunidad

Adempiere posee una comunidad muy activa, no sólo en su página oficial, sino que en todas las empresas consultoras que trabajan con este ERP, a través del foro o bien a través de correo electrónico.

Frecuencia de las Actualizaciones

Adempiere a través del repositorio sourceforge, muestra un movimiento constante de actualizaciones y parcheados, dando a entender un flujo de trabajo constante en mejorar el producto, también es posible la descarga de nuevas versiones a través de su Web oficial o sus representantes a nivel nacional.

8.3.2.4.- Madurez

Lugares de referencia

Entre otros son estos:

- www.adempiere.org
- www.adempiere.com
- http://sourceforge.net/project/showfiles.php?group_id=176962&package_id=215056
- www.openbiz.com.ar (Argentina)
- <http://demo.globalqss.com/webui/> (Colombia)
- www.egs.pe (Perú)

8.3.3.-Compiere

Esta sección está basada en las características del sistema ERP de software libre Compiere. Compiere ERP & CRM es una sofisticada solución de negocios de software libre que se ha posicionado como una fuerte alternativa a los productos propietarios, es decir a aquellas aplicaciones que son desarrolladas y comercializadas por una organización comercial. El verdadero poder de Compiere queda demostrado cuando, siendo funcionalmente rico, tiene la posibilidad de incorporar los cambios específicos de cada negocio y preservarlos en la liberación de nuevas versiones.

Compiere		
Sub-criterio	Descripción	
FLEXIBILIDAD		
1	Personalización	Alto nivel con la edición de metadatos
2	Flexibilidad de las Actualizaciones	ok
3	Internacionalización	Multisite
4	Facilidad de uso	ok
5	Arquitectura	2 capas cliente/servidor previsión 3 capas
6	Escalabilidad	ok
7	Seguridad	ok
8	Interfaz	Csv
9	Independencia del S.O	Unix, Windows,
10	Independencia de la Base de datos	Linux y Mac OS X
11	Lenguaje de Programación	ORACLE
SOPORTE		
1	Soporte de la infraestructura	Foros, web, contratos de soporte.
2	Formación	A nivel de usuario y de programador
3	Documentación	A nivel de usuario, programador e instalación.
CONTINUIDAD		
1	Estructura del proyecto	ok
2	Actividad de la Comunidad	ok
3	Transparencia	X
4	Frecuencia de Actualizaciones	X
5	Otras efectos acordados	-
MADUREZ		
1	Estado de desarrollo	Estable
2	Lugares de referencia	ok

8.3.3.1.- Flexibilidad

Personalización

Compiere posee un alto grado de personalización a través de metadatos. Los metadatos se almacenan en la base de datos den 114 tablas en lo que se denomina diccionario de la aplicación. Puede ser modificada con el cliente Compiere con tan sólo tener derechos de administrador. Para crear una nueva ventana en Compiere, debe ser definida con anterioridad la estructura de la base de datos. Una GUI (Interfaz gráfica de usuario) Compiere consta de una ventana con varias pestañas y campos, y los correspondientes elementos de la base de datos tal como se muestra en la figura.

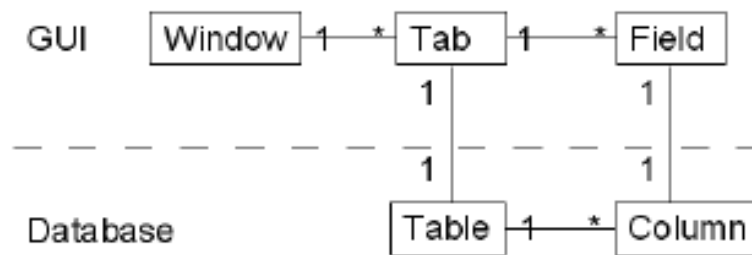


Figura 8.2 Personalización Compiere

El Diccionario de Datos de Compiere, alojado en la capa de metadatos, sabe como acceder a los datos y como se relaciona la información. Contiene definiciones de entidades de datos (tipos, validaciones, etc.), como se muestran (títulos sobre pantallas y reportes, ayudas, posición relativa con respecto a otros datos, etc.) y las reglas para mostrarlos. Permite a usuarios autorizados, agregar nuevas tablas, nuevas pantallas y datos adicionales sobre pantallas ya existentes en la aplicación.

Para la personalización a niveles inferiores se puede ampliar el modelo con callouts, proceso de documentos workflow y con procedimientos SQL.

A partir de los callouts los desarrolladores son capaces de realizar personalizaciones mayores siguiendo los patrones usados por la aplicación.

Flexibilidad de las actualizaciones

Como consecuencia de la arquitectura única de Compiere, las personalizaciones específicas no necesitaron una cara "re-instalación" y migración del software para cada una de las sucesivas actualizaciones del software.

Por ejemplo, dotBase desarrolló un proceso personalizado para la facturación por publicidad, permitiendo a Médecine & Hygiène justificar diferentes situaciones de facturación basadas en cada cliente. Las

funcionalidades y reglas de negocio personalizadas fueron desarrolladas y almacenadas en el diccionario de datos de Compiere, eliminando los tradicionales problemas asociados a la actualización de las personalizaciones como parte de cada futura actualización.

En definitiva Compiere tiene también la posibilidad de extender la aplicación. Estas extensiones son preservadas durante las actualizaciones del producto a nuevas versiones.

Internacionalización

Tanto la interfaz de usuario como los reportes están traducidos al inglés, chino, alemán, castellano, francés e italiano. El estado de traducción a otros idiomas se puede acceder a través de su página oficial. Los esquemas de cuentas son accesibles para Estados Unidos, España, Italia y Alemania.

Facilidad de uso

La interfaz de usuario posee pestañas de diseño y admite un subconjunto de opciones para adecuar su interfaz gráfica, de manera que el mismo usuario es capaz de realizar los ajustes adecuados para su propósito.

Arquitectura

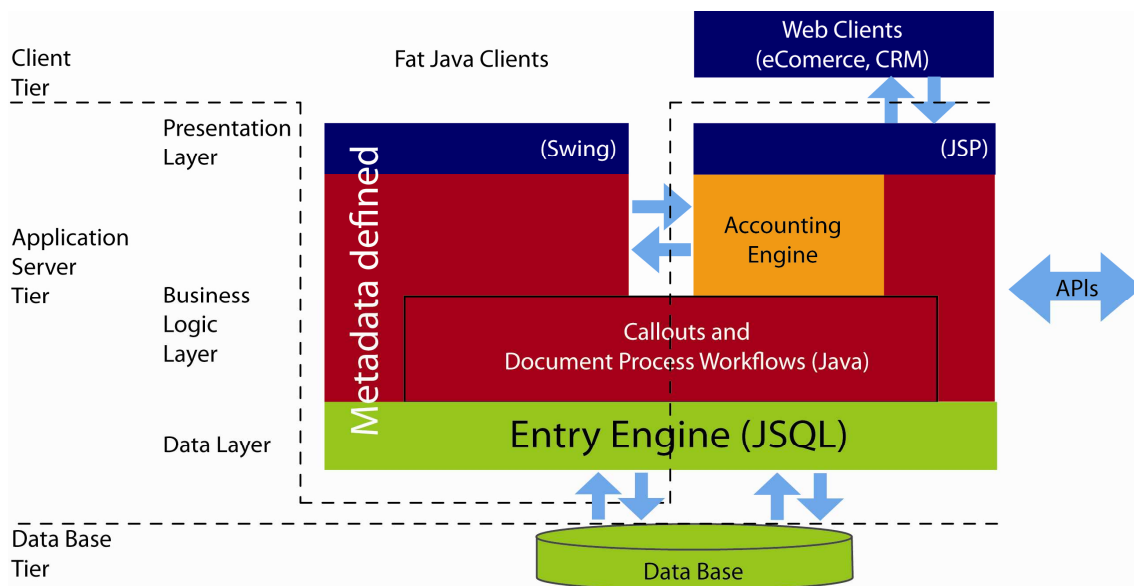


Figura 8.3.- Arquitectura Compiere

La arquitectura de Compiere, es una mezcla de arquitectura de 2 capas y 3 capas. Usa un cliente Java pesado (Java Webstart). El motor

contable, se encuentra en el servidor de aplicación (JBoss J2EE container). Los clientes menos pesados para la web store y para alguna funcionalidad del CRM. Existe una versión beta con clientes menos pesados, pero necesita un mayor desarrollo para equipararla a clientes más pesados.

Seguridad

La seguridad está basada en Roles de Usuario, la cual controla el acceso a Pantallas, Reportes y Procesos. La seguridad de los Datos para los Clientes y Organizaciones es mantenida a nivel del contexto de seguridad de la base de datos. Esto es un nivel adicional de seguridad posterior al login normal de la base de datos. Antes de acceder a cualquier dato, el usuario debe identificarse con un nombre de usuario, contraseña, rol de usuario y opcionalmente su preferencia de lenguaje. Todas las contraseñas son almacenadas en forma encriptada.

Roles: definen el primer nivel de seguridad en Compiere. El usuario se identifica en Compiere con un Rol específico y ve las organizaciones, pantallas, procesos, formularios, workflows y tareas a las que el Usuario puede acceder.

El usuario no ve los ítems de menú a los que no puede acceder y el Rol controla una serie de acciones que son habilitadas o deshabilitadas para cada Rol en particular (Por ejemplo ver información contable, poder exportar, poder ejecutar Reportes, poder actualizar información, acceso a organizaciones, etc.).

Roles para Acceso a Datos: es el segundo nivel de seguridad que dispone Compiere. La seguridad de acceso para un determinado Rol, puede ser refinada adicionalmente definiendo los accesos a tablas, columnas o registros específicos. Por ejemplo, un usuario específico que solo pueda crear Ordenes de Venta con la condición de pago inmediata; para este caso no dispondrá de la posibilidad de seleccionar cuenta corriente. Deshabilitar el acceso a un usuario a determinadas cuentas contables, en cuyo caso ellas no podrán ser utilizadas por ese usuario, ni podrá ver los balances para esas cuentas.

8.3.3.2.- Soporte

Soporte de la infraestructura

1. Soporte comercial brindado por las empresas asociadas al proyecto.
2. Soporte gratuito, disponible en:

- Comunidad de Desarrolladores
- Listas de correo
- Archivos
- Base de datos de soporte

Las experiencias de soporte son generalmente más responsables que con las aplicaciones propietarias.

Formación

Al igual que el soporte anteriormente mencionado, en su página Web existen accesos a formación a distinto nivel, debiéndose poner en contacto, de manera que se acuerden las condiciones, ya sea de desplazamiento de uno o de otros. Otra opción es la de cursillos online, ambas opciones conllevan coste económico.

Documentación

La documentación tanto a nivel de usuario como a nivel de desarrollador es amplia, una de las razones es la antigüedad del proyecto, base de otros proyectos erp (adempiere, entre otros), existen manuales de funcionamiento e instalación, así como de desarrollo de nuevas funcionalidades de la aplicación.

8.3.3.3.- Continuidad

Estructura del proyecto

Compiere es un proyecto dirigido a empresas. Compiere Inc., dirigida por Jorg Janke, principal desarrollador, tiene su sede en Portland (EE.UU). Sen concentra en el desarrollo y en el soporte y formación a segundo nivel (Partners). Compiere no proporciona servicios de implementación y por lo tanto no compite con sus Partners. Más de 70 Partners certificadas venden la implementación, así como servicios de consultoría en por lo menos 25 países. La mayoría del desarrollo esta realizado por Jorg Janke y otros empleados de Compiere. Algunas Partners y usuarios ayudan al proyecto en la recopilación de requisitos, control de calidad, testeo y parchado. Las Partners determinan las prioridades de desarrollo.

Actividad de la comunidad

Compiere tiene una comunidad de usuarios muy activa. Los foros también son muy activos. Se utilizan principalmente foros Sourceforge y servicios de seguimiento para la comunicación con alrededor de 500

mensajes de los foros al mes. Las peticiones de soporte y nuevas funcionalidades tienen un seguimiento separado, y tienen aproximadamente unos 150 mensajes al mes.

Frecuencia de las Actualizaciones

La última versión data de junio de 2010, alojada en sourceforge, muestra una amplia actividad este último año. En su página oficial existe la posibilidad de descarga de las últimas versiones, aunque la única gratuita es la versión con funcionalidad básica (3.2), versiones con nuevas funcionalidades, sólo son descargables previo pago o con caducidad limitada.

8.3.3.4.- Madurez

Lugares de referencia

La aplicación inicial fue construida para una empresa de distribución de neumáticos. Compiere asegura que existen por lo menos 100 compañías que usan su software.

Algunas Partners tienen en la página principal comentarios a cerca de sus clientes y de forma más detallada en la página del propio partner.

El uso de Compiere en dos empresas alemanas fue presentado en una conferencia de software libre:

Destacar que la primera empresa utilizó Compiere para el cálculo de precios de fabricación sobre pedidos en tiempo real de placas frontales para dispositivos electrónicos. Un cliente puede pedir una placa frontal, proporcionando su diseño con un software libre. El proceso completo de pedidos, cálculo de precios con la ayuda del archivo de diseño, control de la viabilidad técnica, generación de programas para el control de la máquina, producción de la placa frontal, control de calidad, embalaje y envío lo cubre la aplicación Compiere. Un programador interno perteneciente al Partner de Compiere apoyó a la personalización de la aplicación. El código fuente principal fue personalizado de manera que lograron un mantenimiento fácil de las actualizaciones.

Por otra parte Compiere se utilizó para la gestión de productos, proporcionando una funcionalidad virtual del almacén, y de los procesos de compras y ventas. El proceso completo que recorre desde el contacto inicial con el cliente hasta la contabilidad, ofreciendo un sistema muy flexible para la generación de presupuestos y seguimiento de los clientes. Necesitando interfaces con la contabilidad externa así como gestión de proyectos. Ambos sistemas fueron muy personalizados, quedando muy satisfechas con el sistema y afirmando que realizarían la misma elección.

El coste, la flexibilidad, la seguridad de la inversión y el conocimiento de la empresa, fueron las principales razones para la selección de Compiere.

Los lugares de referencia destacables son básicamente estos:

- www.compiere.com (página oficial)
- A nivel europeo:
- www.audaxis.com (Bélgica y Francia)
 - www.viventu.de (Alemania)
 - www.audaxis.com (Suiza y Luxemburgo)
 - ...

A nivel nacional:

<http://www.directivesoft.es/asociados.jsp>

8.3.4.-Oasis erp

Oasis nació como respuesta a la necesidad de Software Libre para la gestión empresarial en el entorno de las PYMES de Castilla la Mancha (España). En su desarrollo han participado varias empresas agrupadas bajo 'DESERTIC' Desarrollos y Servicios en Tecnología de la Información y las Comunicaciones. Tras 3 años de desarrollo, el producto resultante fue un nuevo ERP.

Actualmente Oasis ERP consta de versiones para Windows y Linux; Incluso dispone de una versión portable que permiten su ejecución desde unidades USB. La última versión disponible es la 3.0.7 de Enero 2010. Oasis ERP dispone del sello de certificación OPSOA (Open Source Assessment), concedido por el CESLCAM (Centro de Excelencia del Software Libre de Castilla la Mancha). Se trata de un sello de confianza que permite acreditar la calidad de las aplicaciones de Software Libre, basada en los parámetros de calidad definidos por ISO 91261, especialmente respecto a funcionalidad, usabilidad, mantenimiento, portabilidad y seguridad.

Oasis erp		
Sub-criterio	Descripción	
FLEXIBILIDAD		
1	Personalización	ok
2	Flexibilidad de las Actualizaciones	Depende del grado de personalización
3	Internacionalización	multiidoma
4	Facilidad de uso	no
5	Arquitectura	3 capas
6	Escalabilidad	ok
7	Seguridad	ok
8	Interfaz	csv
9	Independencia del S.O	Linux,Windows
10	Independencia de la Base de datos	PostgreSQL
11	Lenguaje de Programación	PHP-GTK2
SOPORTE		
1	Soporte de la infraestructura	ok
2	Formación	ok
3	Documentación	ok
CONTINUIDAD		
1	Estructura del proyecto	Desarrollado por la comunidad de castilla la mancha
2	Actividad de la Comunidad	ok
3	Transparencia	ok
4	Frecuencia de Actualizaciones	ok
5	Otras efectos acordados	-
MADUREZ		
1	Estado de desarrollo	Estable
2	Lugares de referencia	ok

8.3.4.1.- Flexibilidad

Personalización

La personalización de la aplicación OasisERP se puede realizar a nivel de usuario configurando las características del entorno para cada usuario en particular, también se ofrece la activación o desactivación de módulos por grupo de usuario, pudiendo realizar la misma opción con los menús.

Los menús siguen una estructura jerárquica en árbol con diferentes niveles de profundidad. Cada rama del menú puede contener a su vez otras ramas de menú que contengan opciones u otras ramas a su vez (estructura padre-hijo).

Mediante el programa `AdmMenusGruposUsuarios.php` accesible desde el propio menú de Oasis se pueden incorporar las nuevas opciones siguiendo la jerarquía comentada. En el nivel inicial aparece siempre el módulo hasta llegar al máximo desarrollo de opciones. Del mismo modo con el programa `AdmModulos.php` se mantiene la tabla 'Módulos' para la definición de nuevos módulos.

La modificación de los módulos se realizará a través de su lenguaje de programación PHP, por lo que podremos que tiene un grado de personalización de bajo nivel.

Flexibilidad de las actualizaciones

El grado de flexibilidad en las actualizaciones es proporcional al grado de personalización realizado con anterioridad, antes de realizar una actualización a una nueva versión habrá que realizar un estudio previo de los módulos modificados anteriormente para no perder la funcionalidad adquirida con las modificaciones previas.

Internacionalización

El grado de internacionalización mostrado por la aplicación OasisERP está adaptado en la actualidad a su origen español, así como a la normativa y esquemas contables españoles, su expansión en la actualidad se basa en el mercado iberoamericano.

Facilidad de uso

La modificación de los menús por usuario y grupos, así como la personalización del interfaz son las armas más potentes de la aplicación. OasisERP también ofrece el uso de teclas rápidas para el acceso a los diferentes menús y acceso al manual de usuario a través de la opción de ayuda.

Arquitectura

Arquitectura modular de la aplicación cliente-servidor (2 Capas)

Seguridad

Accesos

Los accesos en Oasis están perfectamente controlados y referenciados en la tabla 'logmenus', aquí se llevan los siguientes conceptos:

Campo	Descripción
Id.Usuario	Código de Identificación del Usuario.
Id.Menu	Código de Identificación del Menú
Id_Username	Código de Identificación de la máquina
Fecha	Fecha de acceso
Título	Nombre de opción
Accion	Nombre del programa.

De este modo es posible saber qué, quién y cuando se han realizado los diferentes procesos. El programa que permite la consulta del log se llama 'AdmLogMenus.php' y esta se realiza por usuario, rango de fechas y terminal.

Permisos

Los permisos para los módulos y opciones se definen por grupo y usuario, y como en anteriores puntos es el programa AdmMenusGruposUsuarios.php el encargado de realizar el proceso.

En la ventana se presentan dos listas de menús. El Menú General (en azul) con todas las opciones disponibles en la aplicación y el Menú del Usuario (en verde) con las opciones a las que tiene permiso el usuario que se define.

Las passwords de paso a las diferentes opciones están definidas por usuario con su correspondiente fecha de caducidad.

8.3.4.2.- Soporte

Soporte de la infraestructura

Oasis nació a partir de tres empresas HispaFuentes, Infodasa y LogicComputers, son las encargadas de ofrecer el soporte a nivel nacional y las encargadas de las modificaciones y lanzamientos de nuevas versiones. Albergado en su página oficial contiene un foro de soporte de escaso movimiento, así como enlaces vía mail.

Formación

- Los servicios que ofrece para los usuarios con contrato de Soporte:
 - Asesoramiento de negocio e integración en las TIC.
 - Asesoramiento en la implantación el ERP.
 - Mantenimiento y Soporte de la Aplicación via Internet (on-line) y telefónico.

- Los servicios que ofrece para los partners:
 - Formación Técnica y de Usuario
 - Soporte Técnico
 - Desarrollo a medida.
 - Consultoría especializada.

Documentación

Existe documentación a nivel de usuario y de la aplicación, la documentación para el desarrollo de la misma no es lo bastante específica.

8.3.4.3.- Continuidad

Estructura del proyecto

OASIS nace como consecuencia de que tres empresas de Castilla La Mancha, expertas en desarrollo de aplicaciones informáticas de gestión y en sistemas operativos y lenguajes de programación, después de analizar el estado actual del mercado de “Programas de gestión para empresas”, componen el grupo DESERTIC.

Actividad de la comunidad

A través del foro alojado en su Web podemos observar un flujo escaso de mensajes entre sus componentes.

Frecuencia de las Actualizaciones

La última versión de OasisERP es la versión 1.0 que es similar Oasis 3.0.7 que ha obtenido la Certificación OPSOA. Por este motivo se ha decidido inicializar el número de versión del producto. Ahora Oasis 3.0.7 es Oasis 1.0.

Versiones previas datan del 2000 en el repositorio sourceforge hasta 2007 versión 2.3, a partir de aquí no existe movimiento alguno en el repositorio anteriormente mencionado.

8.3.4.4.- Madurez

Lugares de referencia

Basicamente su enlace oficial:

www.oasis-clm.es

8.3.5.-Openbravo

Openbravo es una aplicación de software libre de gestión empresarial tipo ERP enfocado a PYMES, Su origen es español y actualmente está llevando un proceso de expansión a nivel mundial. El software es una aplicación completamente basada en Web, lo que facilita su administración e interacción con los usuarios al encontrarse toda la información, incluido la aplicación en un solo lugar. Sumado a esto la facilidad de que el equipo cliente solo necesite un navegador Web para interactuar con el aplicativo es mucho mas funcional.

El software posee diferentes módulos integrados con el fin de globalizar la información dentro de la empresa y al mismo tiempo controlar su disponibilidad e integridad. La empresa desarrolladora brinda soporte a los usuarios a través de consultorías estratégicas, de implantación y mantenimiento presencial.

Openbravo		
Sub-criterio		Descripción
FLEXIBILIDAD		
1	Personalización	Alto (metadatos)
2	Flexibilidad de las Actualizaciones	ok
3	Internacionalización	Multiidioma
4	Facilidad de uso	ok
5	Arquitectura	MVC (3 capas)
6	Escalabilidad	Altamente escalable
7	Seguridad	ok
8	Interfaz	csv
9	Independencia del S.O	GNU/Linux y windows
10	Independencia de la Base de datos	PostgreSQL y Oracle
11	Lenguaje de Programación	JDK
SOPORTE		
1	Soporte de la infraestructura	ok
2	Formación	ok
3	Documentación	ok
CONTINUIDAD		
1	Estructura del proyecto	Compañía + socios + comunidad
2	Actividad de la Comunidad	ok
3	Transparencia	ok
4	Frecuencia de Actualizaciones	ok
5	Otras efectos acordados	
MADUREZ		
1	Estado de desarrollo	Estable
2	Lugares de referencia	ok

8.3.5.1.- Flexibilidad

Personalización

En el diccionario de la aplicación se definen los metadatos. En este módulo se dan de alta los informes, procesos y formularios que se usen en la aplicación, para poder acceder a ellos mediante el menú o botones en ventanas. También se definen todas las ventanas que se construyen automáticamente al compilar la aplicación, para ello se registran las tablas y columnas correspondientes de la base de datos y se define el diseño de la ventana. Supone un modelo de diseño de software que depende de metadatos almacenados en un diccionario para modelar el comportamiento de la aplicación. Esto conlleva una reducción drástica en cuanto a codificación manual y número de errores se refiere, permitiendo que expertos de negocio con poca experiencia a nivel de codificación puedan configurar la aplicación para satisfacer las necesidades de cada empresa.

Flexibilidad de las actualizaciones

Las interfaces públicas estables y la capa de modularidad de Openbravo proporcionan el mejor método para conservar las personalizaciones durante las actualizaciones del ERP. Pudiendo obtener la funcionalidad específica del negocio necesaria sin quedarse atrapado en una versión de ERP obsoleta por el alto coste de actualizar las personalizaciones.

Internacionalización

- Soporte para múltiples monedas.
- Soporte para múltiples esquemas contables, lo cual permite que la misma transacción sea contabilizada según reglas distintas, esquemas contables varios, distintas monedas o incluso diferentes calendarios.
- Soporte para números de cuentas bancarias internacionales.
- Soporte para múltiples idiomas árabe, portugués de Brasil, holandés, inglés, francés, gallego, alemán, italiano, polaco, ruso, español y tailandés., definidos a nivel de usuario.

Facilidad de uso

- Menú principal configurable por rol de usuario.
- Idioma de trabajo configurable a nivel de usuario.
- Alarmas programables por rol de usuario o usuario concreto.
- Navegación a través de teclas rápidas para una operativa más rápida.
- Interfaz de usuario modificable a través de skins o temas.
- Ayuda contextual (actualmente disponible en español e inglés).

- Posibilidad de anexas documentos, imágenes u otro tipo de ficheros a cualquier entidad de la aplicación.
- Información navegable (historial, documentos relacionados, etc.).
- Generación de informes en múltiples formatos: excel, pdf y html.
- Filtros configurables y búsquedas flexibles.
- Selectores incrustados en los formularios para las entidades más usadas (productos, terceros, cuentas, pedidos, facturas ...).
- Procesos en lote configurables para tareas que deban ser procesadas a intervalos periódicos.

Arquitectura

En la figura se observa el esquema general de la arquitectura del ERP Openbravo.

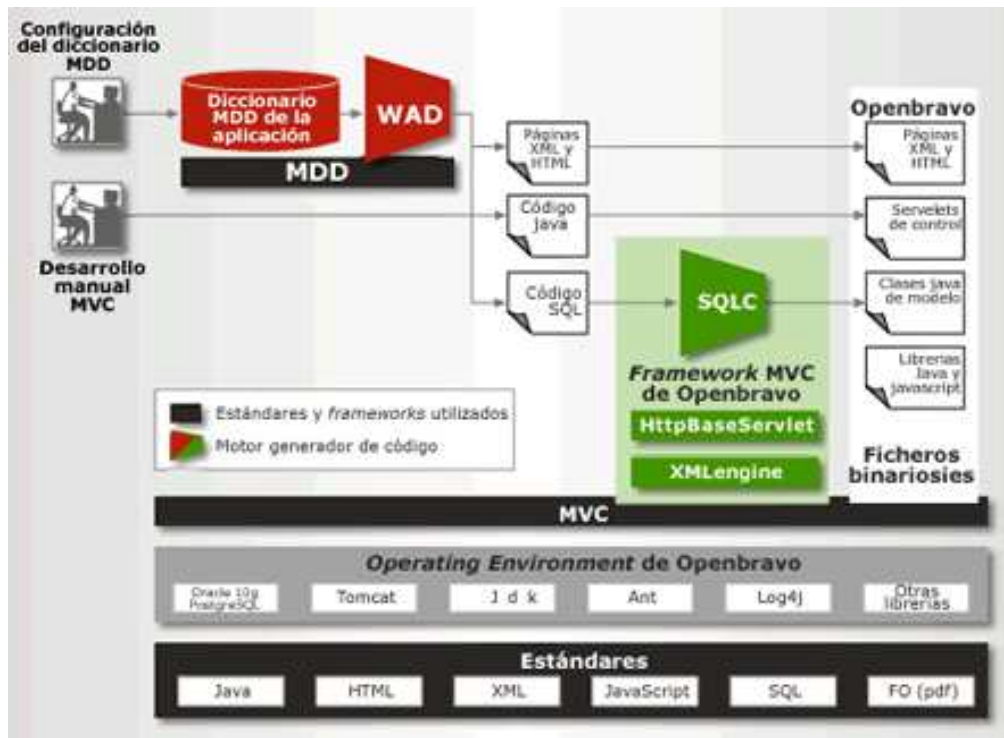


Figura 8.5.- Arquitectura Openbravo.

La mayor parte del código se genera automáticamente por el motor que denominado WAD (Wizard for Application Development), basándose en la información contenida en el Diccionario del modelo de datos (Data Model Dictionary). Esta característica proporciona una mejor calidad del código al reducir drásticamente la codificación manual, al tiempo que mejora la productividad y eficiencia del desarrollo. El motor ejecuta y recompila la aplicación cada vez que el administrador modifica la configuración para adaptarla a un nuevo requerimiento.

Openbravo se ha diseñado sobre la base de una arquitectura revolucionaria que resulta en una manera más eficiente de desarrollar aplicaciones. Por una parte, el modelo MVC (Model, View, Control) facilita el desacoplamiento de las áreas de desarrollo, permitiendo el crecimiento sostenible de la aplicación y una mayor facilidad en el mantenimiento del código. Por otra parte, MDD (Model Driven Development) proporciona una mejor calidad del código al reducir drásticamente la codificación manual, al tiempo que mejora la productividad y eficiencia del desarrollo. (Business Intelligence).

Toda la aplicación ha sido construida siguiendo estándares abiertos: J2EE, SQL, JDBC, HTML, CSS, MDD, XML Engine y SQLC para el desarrollo y XML, FOP, PDF, RTF para el intercambio y presentación de datos. El lenguaje de desarrollo es Java y la base de datos tanto Oracle como PostgreSQL. Los métodos de desarrollo se pueden encontrar en publicaciones de la IEEE, en algunas de las cuales participa Openbravo. El uso de estándares y tecnologías abiertas garantiza un ciclo de vida de la solución más largo que el de otras soluciones basadas en tecnologías propietarias.

En la siguiente figura, de forma más esquemática, se observa la interconexión entre los diferentes módulos.

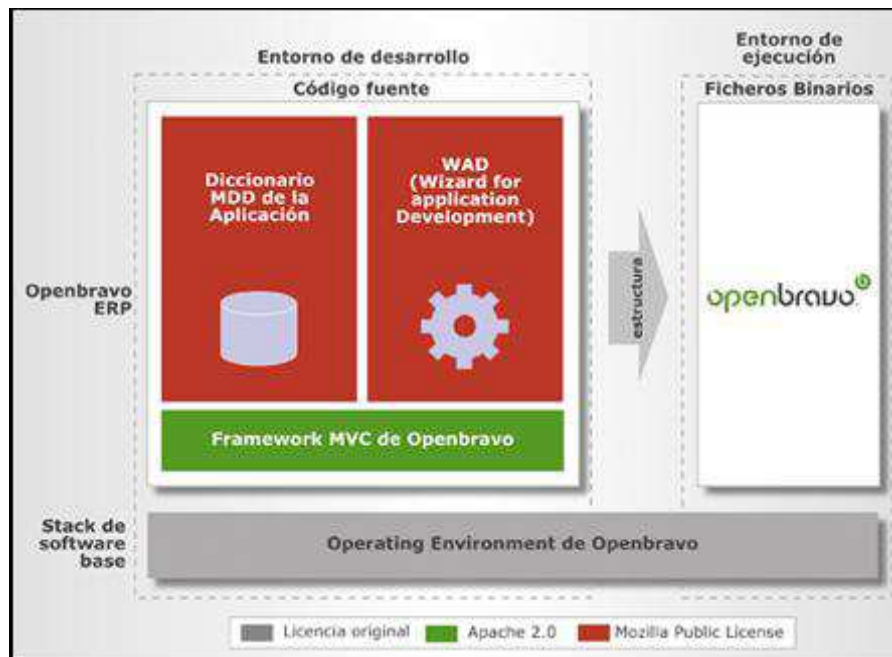


Figura 8.6 Esquema de Modularización de OpenBravo

El motor WAD es el motor, desarrollado por Openbravo, como se ha comentado con anterioridad genera automáticamente el código binario de la aplicación a partir del diccionario MDD. Los ficheros generados por el WAD se generan conforme al estándar MVC.

Posee un diccionario MDD que almacena los metadatos que describen cada elemento de la aplicación incluyendo el comportamiento del mismo.

El MVC Foundation Framework es el conjunto de utilidades de programación robustas seleccionadas entre los mejores candidatos en software libre disponibles o desarrollados por Openbravo en el caso que no exista candidato alguno en ese momento. Estas herramientas facilitan el desarrollo web de la aplicación según el esquema MVC.

El entorno operativo utiliza aplicación de terceros bien conocidas como lo son Apache http Server y Tomcat, y una base de datos PostgreSQL u Oracle, que pueden ser instalados en multitud de sistemas operativos, incluyendo GNU/Linux o Microsoft Windows.

Para el cliente final también se ofrece la consultoría para implementación y es recomendable que se trabaje con los “business partners certificados”, según la complejidad de la implementación.

Seguridad

- Niveles de acceso por usuario definidos según roles.
- Auditoría de cada transacción.
- Soporte para conexión segura a través de https.

Los usuarios de diversos perfiles pueden acceder a Openbravo ERP mediante roles diseñados a medida de sus hábitos de trabajo y que garantizan la seguridad de la información que pueden consultar y modificar. Los roles permiten controlar qué pantallas son accesibles desde el menú y son visibles para los usuarios de una determinada organización y accesibles en modo de edición o bien de sólo lectura.

También es posible configurar para cada usuario el idioma y otros valores predeterminados.

8.3.5.2.- Soporte

Soporte de la infraestructura

El soporte ofrecido por Openbravo, lo diferenciamos a dos niveles, el primero a nivel de usuario, basando sus soluciones a través de la multitud de partners que existen en cada región, el segundo nivel es el ofrecido a los Partners asociados a la aplicación.

Los servicios que ofrece para los usuarios son:

- Consultoría estratégica.
- Consultoría de implantación.
- Mantenimiento presencial (a través de los partners).

Los servicios que ofrece para los partners:

- Pack de evaluación.
- Formación.
- Soporte de 2º nivel (más especializado).
- Desarrollo a medida.
- Consultoría especializada.

Albergada en su página web existen foros de soporte respondido por los técnicos de openbravo donde responden a las dudas de los usuarios, la brevedad en tiempo de respuesta, así como de calidad dejan bastante que desear, remitiendo al soporte de los partners previo pago.

Formación

La formación de openbravo a nivel de usuario es ofrecida a través de las Partners y dependerá del nivel de desarrollo del ERP. La formación dirigida a Partners se puede dar de tres maneras, a nivel de clases presenciales (desplazamiento al lugar del curso), online, o bien a medida, ésta última será la de mayor coste económico.

Documentación

Existe documentación a todos los niveles y de gran amplitud en su página Wiki oficial, desde manuales de iniciación en la aplicación hasta manuales de desarrollo.

8.3.5.3.- Continuidad

Estructura del proyecto

Bajo una gran base de usuarios, diferentes casos de éxito en múltiples ámbitos y socios en diferentes partes del mundo, Openbravo se consolida como una solución con continuidad a largo plazo.

Actividad de la comunidad

Actividad constante a través de los foros por parte de la comunidad, a través de enlace http://forge.openbravo.com/plugins/espforum/?group_id=100, desde el cual existe la opción de elegir las siguientes subcategorías que se adaptan a nuestras dudas, desde funcionalidad hasta ayuda a traducciones, todo esto de manera participativa.

Frecuencia de las Actualizaciones

La última actualización es la 2.5, y cuenta con gran variedad de actualizaciones para los módulos, la actividad es constante y sobre todo este último año con la nueva versión.

8.3.5.4.- Madurez

Lugares de referencia

Básicamente son estos:

- www.openbravo.com
- <http://openbravo.grupoconasa.com>
- <http://www.everis.es>
- <http://www.tictech.es>

La primera es la web oficial y posee traducciones a multitud de idiomas, originalmente en inglés, los siguientes enlaces hacen referencia a Partners españoles con la certificación gold, que teóricamente ofrecen un servicio de la máxima calidad, la siguiente certificación es “certified” y la última “registered”.

Destacar el enlace <http://wiki.openbravo.com>, donde encontrar la documentación y toda la información necesaria, foros y demás acerca de la aplicación. Así como en el repositorio de sourceforge, están disponibles las distintas versiones y actualizaciones.

8.3.6.-Openerp

Es un ERP, basado íntegramente en la licencia pública GPL y libremente descargable. Aunque desarrollado inicialmente en Bélgica, existe traducción al español. OpenERP está orientado al uso en las PYME, aunque dispone de módulos como gestión de proyectos o estadísticas, más habituales de empresas de mayor tamaño. OpenERP se encuentra en estado funcional sobre Linux y Windows, con más de 350 módulos en desarrollo. OpenERP internamente usa un modelo de flujos de trabajo (*workflow*), con arquitectura en tres capas. Está desarrollado en Python, PyGTK y sobre PostgreSQL, y también tiene clientes en librería qt y un *frontend web* basado en TurboGears. GestiCAM es una versión localizada en español de este ERP, promocionada bajo Molinux, una distribución regional de Linux española.

Openerp		
Sub-criterio	Descripción	
FLEXIBILIDAD		
1	Personalización	Bajo nivel
2	Flexibilidad de las Actualizaciones	Medio
3	Internacionalización	multiidioma
4	Facilidad de uso	ok
5	Arquitectura	2 capas cliente/servidor
6	Escalabilidad	ok
7	Seguridad	ok
8	Interfaz	csv
9	Independencia del S.O	Linux/windows
10	Independencia de la Base de datos	PostgreSQL
11	Lenguaje de Programación	Python, PyGTK
SOPORTE		
1	Soporte de la infraestructura	ok
2	Formación	ok
3	Documentación	ok
CONTINUIDAD		
1	Estructura del proyecto	Compañía + socios + comunidad
2	Actividad de la Comunidad	ok
3	Transparencia	ok
4	Frecuencia de Actualizaciones	ok
5	Otras efectos acordados	-
MADUREZ		
1	Estado de desarrollo	Estable
2	Lugares de referencia	ok

8.3.6.1.- Flexibilidad

Personalización

La aplicación OpenERP posee un grado de personalización de los módulos a distintos niveles. Existiendo la opción de programación de sus módulos en lenguaje Python, siempre partiendo de un módulo ya existente, el desarrollador tiene la opción a mediante el lenguaje de programación anteriormente nombrado, modificar la funcionalidad del mismo, implicando un alto grado de especialización del desarrollador, el desarrollo de modificaciones se puede realizar tanto con Unix como con Windows, pese a que se recomienda el uso del primero. Otra opción es, a través del interfaz el administrador es capaz de:

- Modificar menús (por roles o usuario).
- Página de inicio.
- Asignar valores por defectos a campos de la base de datos.
- Cambiar terminologías para adaptarla a la empresa.
- ...

Estos son unas pocas características que entran más en el capítulo comentado posteriormente de facilidad de uso.

Flexibilidad de las actualizaciones

Existen cuatro opciones a analizar en la actualización de la aplicación OpenERP:

- Parches suministrados por Tiny (empresa desarrollador OpenERP, anteriormente se denominaba a la aplicación TinyERP) para la corrección de fallos, tras la validación de estos parches no debería causar ningún efecto secundario.
- Las actualizaciones de menor importancia, que agrupan la corrección de errores en un solo paquete, y son generalmente nombradas como modificación del número de versión (por ejemplo 5.0.0 a 5.0.1).
- Las actualizaciones que incluyen corrección de errores y mejora de funcionalidad (por ejemplo 5.0.0 a 5.1.0).
- Nuevas funciones publicadas con nuevos módulos.

Es recomendable el establecimiento de procedimientos con el proveedor para definir la manera de responder a los cambios en el código de la aplicación OpenERP.

Para las actualizaciones sencillas el equipo de mantenimiento propio evaluará los parches para determinar si son beneficiosos para el uso dentro de nuestra empresa, recomendando una prueba en una versión de pruebas antes de su instalación en la versión final.

El equipo de mantenimiento también debe hacerse cargo de las actualizaciones regulares del software. Los parches y las actualizaciones

sólo podrán ser instalados en el caso que tenga los derechos necesarios sobre la aplicación.

En el caso del lanzamiento de una versión mejorada recomiendan cautela. La mayoría de las actualizaciones requieren la migración de datos, a causa de que las bases de datos antes y después de la actualización pueden ser un poco diferentes.

OpenERP tiene un sistema para gestionar las migraciones de forma semiautomática, ofreciendo documentación así como los scripts necesarios para la realización de forma óptima de la migración.

OpenERP recomienda las actualizaciones siempre y cuando sean importantes las nuevas funcionalidades y no garantiza la funcionalidad de los módulos modificados no oficiales, es decir que no garantiza las modificaciones que se hayan realizado expresamente para nuestra aplicación en particular, en las nuevas versiones.

Internacionalización

OpenERP soporta múltiples monedas y contabilidades, ha sido traducido a 50 lenguas distintas, entre ellas el castellano. El esquema contable español está muy avanzado y adaptado a cada cambio existente, a través de módulos descargables.

Facilidad de uso

OpenERP permite las siguientes modificaciones para conseguir un entorno adecuado para el usuario final, nombraremos algunas:

- Modificar menús.
- Personalizar la página de inicio para cada usuario.
- Asignar valores por defectos a campos de la base de datos.
- Cambiar terminologías para adaptarla a la empresa.
- Puede utilizar la funcionalidad de traducción de idiomas de OpenERP para sustituir la terminología estándar con la terminología que se adapte a su mejor compañía.
- Traducción a través de un archivo CSV.
- Configuración de informes.

Arquitectura

Para acceder a OpenERP lo haremos de dos maneras:

- Usando un navegador web apuntando al cliente al servidor cliente-web de OpenERP
- Usando una aplicación cliente (cliente GTK) instalada en cada equipo.

Ambos métodos de acceso ofrecen unos servicios muy similares, y pueden ser usados en el mismo servidor al mismo tiempo. Es preferible utilizar el navegador web si el servidor de OpenERP está a cierta distancia (por ejemplo, en otro continente) porque es más tolerante a los retrasos de tiempo entre ambos que el cliente GTK. El cliente web también es más fácil de mantener, porque por lo general debe instalarse en los equipos de los usuarios. Por el contrario sería mejor el uso de la aplicación cliente (llamado GTK cliente porque la tecnología está basada en ello) si se usa un servidor local (por ejemplo, en el mismo edificio). En este caso el cliente GTK será más productivo, y por lo tanto tendrá un uso más aprovechable.

En la figura se observa el esquema general de la arquitectura de OpenERP.

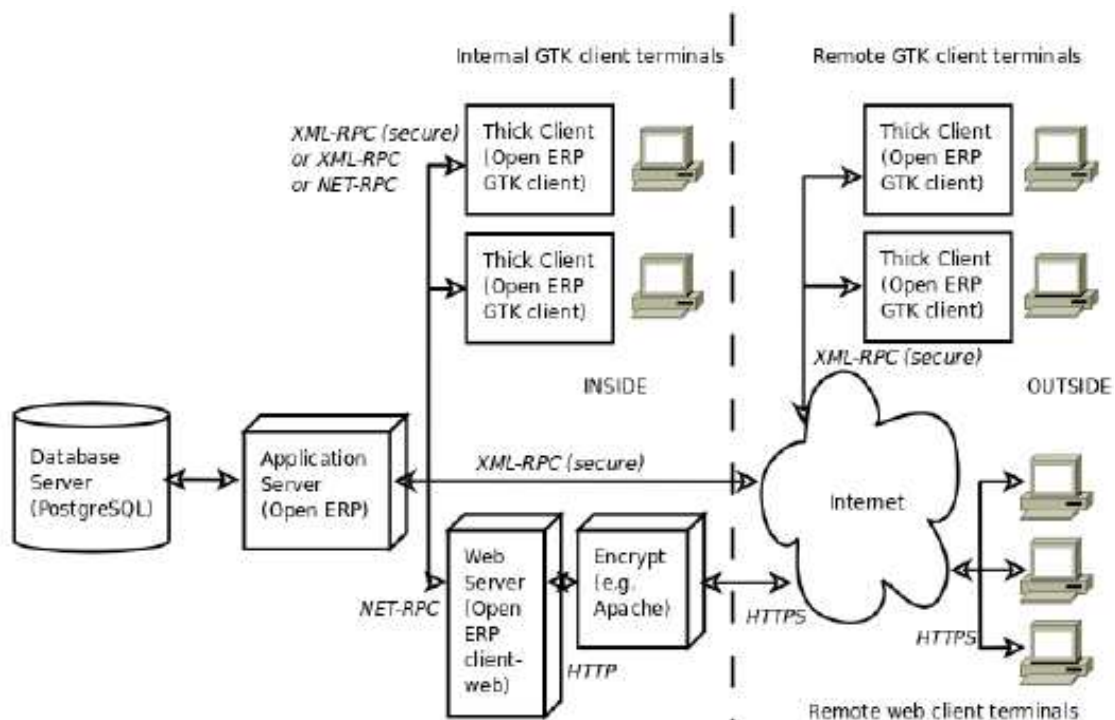


Figura 8.7.- Esquema general de la Arquitectura de OpenERP

El sistema openerp está formado por tres componentes principales:

- El servidor de Base de datos PostgreSQL, que contiene todas las bases de datos, conteniendo a su vez cada una de ellas todos los datos y la mayoría de elementos de la configuración del sistema ERP.
- El servidor de aplicaciones de OpenERP, contiene toda la lógica de la empresa y asegura que el OpenERP funcione de manera óptima.

- El servidor web, una aplicación independiente denominada Open Object client-web, que le permite conectarse a OpenERP desde un navegador web estándar y no es necesario cuando se conecta con un cliente GTK.

Estos tres componentes pueden ser instalados en el mismo servidor o se pueden distribuir en servidores separados, si existen consideraciones de rendimiento que así lo requieran.

Si opta por el uso de clientes GTK no será necesario el tercer componente (cliente-servidor web). En este caso cliente GTK de OpenERP debe estar instalado en la estación de trabajo de cada usuario de OpenERP de la empresa.

Seguridad

Una de las áreas más importantes en la configuración de Open ERP es cómo gestionar los derechos de acceso a la información que contiene.

La aplicación debe contener toda la información significativa del negocio en el sistema, pero la mayoría de usuarios deberán tener sólo acceso a lo necesario para desempeñar su trabajo. La gestión de derechos en el sistema OpenERP se realiza a través de la definición de usuarios que pertenecen a uno o más grupos determinando:

- La visibilidad de los elementos del menú.
- La accesibilidad de cada tabla en la base de datos

Los usuarios de OpenERP también pueden pertenecer a diferentes roles. Al igual que el grupo da derechos de acceso a usuario, cada rol determina los deberes de cada usuario. Esto se logra a través de flujos de trabajo (workflow), que forman los procesos de negocio de la empresa.

8.3.6.2.- Soporte

Soporte de la infraestructura

El soporte suministrado por las empresas especializadas tiene el objetivo de garantizar que los usuarios finales obtengan la máxima productividad de su uso de la aplicación OpenERP, respondiendo a las dudas sobre su uso. El soporte puede ser de carácter técnico o funcional

El mantenimiento tiene por objeto garantizar que el sistema en si, sigue funcionando según las necesidades. Incluyendo las actualizaciones del sistema, que dan acceso a las últimas funcionalidades disponibles. Algunas Partners ofrecen un mantenimiento preventivo. Este mantenimiento

asegura que todos los desarrollos específicos para el sistema son revisados y probados para cada nueva versión, de manera que sean compatibles con la base de OpenERP. Los desarrolladores oficiales recomiendan contratos de soporte para garantizar los costes surgidos tras una migración de una versión muy antigua a una nueva.

Formación

La formación de OpenERP a todos los niveles posee un alto grado de calidad, desde cursos a nivel de usuario, hasta cursos más completos de instalación, actualización, mantenimiento y programación de módulos, son llevados por las empresas asociadas a nivel nacional. En España webs como www.openersite.com o www.openerpweb.es entre otras se dedica a cursos y ponencias sobre las novedades de la aplicación.

Documentación

OpenERP a través de su página web oficial y de las webs de las Partners oficiales, ofrece una gran cantidad de información de instalación y configuración de la aplicación, así como manuales de usuario final, pero se echa en falta manuales de desarrollo de los módulos, ofreciendo las empresas asociadas cursos de formación.

8.3.6.3.- Continuidad

Estructura del proyecto

Una comunidad muy activa, así como un equipo de profesionales y de empresas asociadas forman un proyecto que goza de buena salud y que día a día trata de mejorar y expandirse.

Actividad de la comunidad

OpenERP goza de una comunidad altamente activa como se ha nombrado anteriormente, que no sólo se encarga a través de sus foros oficiales solucionar y tratar sobre temas del nombrado erp, también tratan de mejorar la versión que existe en la actualidad, compartiendo modificaciones, así como traducciones de manuales a los lenguajes de cada región.

OpenERP ofrece lo que llaman Launchpad, que es un sitio Web donde los colaboradores de OpenERP de todo el mundo, comparten y publican sus proyectos. Cualquier módulo en desarrollo está en el launchpad, las

traducciones, los errores, las sugerencias de mejora o de desarrollo de nuevos módulos están allí.

Para que esto esté un poco organizado, requiere cierto control y cierta estructura, por lo que el proyecto OpenERP en launchpad usa un léxico con palabras similares a las siguientes:

- **STABLE:** Se etiqueta así todo lo que haga referencia a la versión estable “oficial” publicada.
- **TRUNK:** Se etiqueta de esta forma, todo lo que vendrá en la próxima versión.
- **QUALITY:** Es el equipo de OpenERP SA y algunos partners oficiales que “tocan” el núcleo de la aplicación.
- **COMMITERS:** Son ramas que están siendo desarrolladas tanto por OpenERP SA como por partners oficiales, o miembros de la comunidad que tienen permisos para trabajar en estas ramas.
- **COMUNITY:** Son los módulos desarrollados por cualquier persona o empresa que quiera publicarlo.
- **ADDONS:** Módulos oficiales de una versión
- **EXTRA-ADDONS:** Módulos extra o adicionales no incluidos en la versión “oficial” estable.

Cualquier módulo publicado en ramas committers o community es susceptible de ser incorporado en nuevas versiones “estables” posteriores, siempre y cuando sean certificados y aprobados por el equipo quality de OpenERP.

Frecuencia de las Actualizaciones

OpenERP tiene alojada la versión completa 5.0.10, no existe una fecha de cuando se finalizó esa versión, pero en la actualidad existe una versión 6.0 Alpha disponible a partir del 6 de Marzo de este año. Esto muestra una actividad constante en las actualizaciones.

8.3.6.4.- Madurez

Lugares de referencia

Básicamente son estos:

- www.openerp.com
- <http://openobject.com/>
- <http://thymbra.com>
- <http://www.kemet-ingenieria.com>
- <http://www.nan-tic.com/>

Las dos primeras referencias son paginas web oficiales, donde tendremos acceso a las versiones, tanto online, como la versión estandar, las siguientes tres referencias, son los Partners oficiales en España, para dar soporte y formación de forma más cercana geográficamente.

8.3.7.-OpenXpertya

openXpertya es una solución de gestión integral para la empresa en español de código abierto que engloba ERP y CRM, con integración de servicios en línea de B2B o B2C (en función del tipo de cliente final) e incluso B2E (servicios internos) y con soporte de exportación de datos (enlaces) al estándar EDI (intercambio electrónico de información entre empresa: facturas, albaranes, pedidos: EDIFACT, estándar mundial de la ONU) y con posibilidad de trabajar con cubos multidimensionales OLAP (análisis exhaustivo de resultados). Todo ello adaptado muy de cerca a la legislación española e hispanoamericana, tanto fiscal, como mercantil, civil, contable, etc.

OpenXpertya		
Sub-criterio		Descripción
FLEXIBILIDAD		
1	Personalización	ok
2	Flexibilidad de las Actualizaciones	Medio
3	Internacionalización	Multilidioma a nivel nacional¿?
4	Facilidad de uso	ok
5	Arquitectura	3 capas
6	Escalabilidad	OK
7	Seguridad	ok
8	Interfaz	n/a
9	Independencia del S.O	Windows, Solaris, FreeBSD, Linux,UNIX, AIX, MacOs
10	Independencia de la Base de datos	Oracle, actualmente, PostgreSQL
11	Lenguaje de Programación	J2EE
SOPORTE		
1	Soporte de la infraestructura	ok
2	Formación	ok
3	Documentación	ok
CONTINUIDAD		
1	Estructura del proyecto	Compañía + socios (COMUNIDAD)
2	Actividad de la Comunidad	ok
3	Transparencia	ok
4	Frecuencia de Actualizaciones	ok
5	Otras efectos acordados	
MADUREZ		
1	Estado de desarrollo	Estable
2	Lugares de referencia	ok

8.3.7.1.- Flexibilidad

Personalización

Usa **diccionario de datos** propio, lo que permite una estructura de base de datos altamente dinámica. Así, el implementador o incluso **el usuario pueden agregar campos** nuevos a las tablas y **nuevas tablas** a la base de datos siendo interpretados y usados por la aplicación desde el primer momento.

Todos las interfaces de la aplicación son configurables. Incluso el usuario final puede decidir que partes de la aplicación se ven o no, cuales son los valores predefinidos para ciertos campos.

La visualización de los elementos se organiza por área, de tal manera que se puede ocultar un área rápidamente a cierto tipo de usuario, ya por cuestiones de seguridad o por cuestiones ergonómicas.

La utilidad del zoom hace que se pueda acceder rápidamente a todos aquellos elementos relacionados con una determinada vista.

Flexibilidad de las actualizaciones

El proyecto openXpertya a basado gran parte de sus esfuerzos en el lanzamiento de la versión 3 de su paquete ERP, más completa y con un entorno visual más cuidado y agradable para el usuario, ya que las versiones anteriores eran pobres en su entorno.

A nivel de flexibilidad sugiere una independencia absoluta con versiones anteriores, a pesar de los intentos de contactar con los responsables de la aplicación (vía mail), en ningún momento nombra la nueva versión como actualización de las anteriores, sino como una versión nueva e independiente, por lo que el análisis de la flexibilidad de las actualizaciones para openXpertya se hace complicado. Si a esto unimos la imposibilidad de la descarga de versiones anteriores (link roto) en su página web, así como en la inexistencia de las mismas en el repositorio sourceforge, llega a ser más que complicado el análisis de este punto.

Internacionalización

A pesar de la divulgación en su página web oficial de varios idiomas, en la actualidad la nueva versión solo cuenta con licencia en castellano y existen interfaces (por orden de desarrollo) en Español de España, Español de México, Español de Argentina, Mallorquí, Catalán y Gallego, por lo que el multiidioma deja bastante que desear, pese a esto, en la actualidad se ha orientado el proyecto a España así como Hispanoamérica, la falta de información provoca la incerteza en amplitud de mercado, sobretodo al anglosajón.

Facilidad de uso

La opción de configuración del interfaz de la aplicación, tanto por el administrador como por el usuario final, junto con la configuración de los elementos, así como el uso del zoom (buscador), dotan a la aplicación de un grado aceptable en lo que facilidad de uso se refiere.

Arquitectura

La metodología utilizada actualmente en el desarrollo de soluciones de software empresarial depende en una gran medida de las características del proyecto, de los integrantes del equipo de desarrollo y de las tecnologías a utilizar. En realidad, no existe una metodología concreta que nos permita un desarrollo eficaz. El continuo proceso de cambio en el proyecto, los integrantes del equipo y la renovación y mejora de las tecnologías implica un continuo replanteamiento en los pasos a seguir. Esta falta de continuidad en la metodología conlleva graves inconvenientes para garantizar la obtención de una solución fiable, fácilmente mantenible y sobretodo una solución que pueda evolucionar en el tiempo.

- El modelado UML intenta paliar este problema aportando una notación para la **definición de las necesidades** a cubrir, pero únicamente es una **notación**, nada más. Realmente, no es el establecimiento de una metodología clara de desarrollo, sino una herramienta que nos posibilita para el modelado previo a la metodología
- Aunque no es un estándar reconocido, está apoyado en gran medida por la OMG (Object Management Group). Este **grupo** definió en su día el estándar CORBA para el desarrollo sobre sistemas distribuidos, **necesario para definir las comunicaciones entre aplicaciones** (por ejemplo entre cliente y servidor).

OpenXpertya, a través del entorno de diseño en tres capas (3LD) aporta una metodología de declaración de los conceptos de negocio, definición de la interacción con el sistema, procesos a realizar sobre los conceptos, y finalmente nos permite establecer restricciones a este modelo y validaciones.



Figura 8.8.- Arquitectura OpenXpertia

Una vez realizado los pasos anteriores, disponemos directamente de la solución e implícitamente de toda la información para generar automáticamente una documentación en detalle de qué hace el sistema en cada momento y cómo lo hace.

Así las tres capas quedan definidas en nuestro caso de la siguiente manera:

1. En la **capa de datos** tenemos el motor de base de datos relacional, independiente de la aplicación y escalable en función de las necesidades de la empresa final. Inicialmente trabajamos sobre Oracle, por su potencia y por ser un estándar del mercado, pero adicionalmente hay disponibilidad para la utilización de otros motores de base de datos, que pese a no disponer de tanta fiabilidad, tienen como baza a su favor la disponibilidad absoluta del software libre (Daffodil One\$DB, PostgreSQL, MaxDB, Firebird y Sybase ASE Express Edition sobre Linux).
2. En la **capa del Servidor de Aplicaciones o de Negocio**, tenemos el servidor de aplicaciones JBOSS y las clases que interactúan directamente con la base de datos (vía JDBC).
3. En la **capa de Presentación** disponemos de varios clientes posibles. El principal y sus variantes de empaquetado (distribución directa, vía Java Web Start o applet Java), realizado directamente en Java; pero adicionalmente también disponemos de cliente ligero sobre navegador web (contra las páginas JSP servidas desde el servidor Apache Tomcat integrado en JBOSS) con diversas configuraciones posibles basadas en las necesidades de los procesos de negocio

de la empresa usuaria y en función del tipo de rol del usuario que abre sesión en cada momento concreto.

Seguridad

El software usa un sistema multiperfil de acceso. El administrador define las vistas del usuario y aquellos elementos que puede modificar o visualizar. Los usuarios se agrupan en perfiles para facilitar su gestión. El administrador define los accesos según el perfil y luego adjudica perfiles; así controla hasta los accesos a la BBDD.

Por otra parte, la implantación conlleva un plan avanzado de copias de seguridad. El software además permite su congelación. Se puede sacar un *snapshot* (para copias de seguridad) del sistema en cualquier momento y volver a él si es necesario. Todo el sistema está centralizado en el servidor, a pesar que la carga de proceso está distribuida. Por ello, es la única parte que necesita ser salvaguardada.

8.3.7.2.- Soporte

Soporte de la infraestructura

El sitio web **openXpertya.org** ofrece a sus usuarios una serie de foros y espacios de comunicación con el personal del proyecto y las empresas que forman parte de él y prestan servicios relacionados con la herramienta. En estos foros se publica información gratuita y se resuelven dudas y cuestiones de forma también gratuita. Pero la respuesta a estas comunicaciones depende del conocimiento, y la voluntad de los usuarios y moderadores de responder a estas cuestiones, y en ningún caso se pueden entender como una relación de tipo comercial.

En la actualidad el foro no está disponible, partiendo de la temporalidad de la situación y añadiendo que la opción de ponerse en contacto con el personal del proyecto vía mail, como se ha comentado con anterioridad, carece de respuesta alguna.

Actualmente el proyecto openXpertya a través de sus Partners ofrecen servicios de consultoría, desarrollo e implementación de nuevas funcionalidades, así como la puesta a punto del sistema en la empresa.

Formación

En cualquier proyecto de implantación de software, esta es la última fase, pero no por ello en absoluto la menos importante conocer a fondo una herramienta como openXpertya, tanto a nivel funcional como técnico requiere un tiempo y una curva de aprendizaje que puede resultar muy

ardua si se realiza en solitario. Por ello, los partners openXpertya ofrecen formación a tres niveles:

- formación a colaboradores: ofrece seminarios para el conocimiento de la aplicación, estado y desarrollo, enfocado a la formación de partners específicas en el proyecto OpenXpertya.
- formación técnica: ofrece a nivel de partner o a nivel de cliente la posibilidad de conocer de manos del equipo de desarrollo, la arquitectura, las políticas de desarrollo y todas las herramientas de desarrollo usadas en el proyecto.
- formación a usuarios finales: Dentro de las implantaciones a clientes finales, las empresas partners openXpertya ofrecen formación específica a los clientes finales. Esta formación se realiza de manera totalmente personalizada para la empresa.

Documentación

No existe una amplia documentación a la que atenerse, ni a nivel de desarrollo ni de usuario, en su nueva versión aparece documentación para la correcta instalación, pero no existen ni guías de usuario ni de desarrollo, se echa de menos accesos a zona Wiki, que en la actualidad está inactiva, así como al foro desactivado en la actualidad.

8.3.7.3.- Continuidad

Estructura del proyecto

OpenXpertya es un proyecto de I+D+I seleccionado por la Fundación FICYT en el marco del Plan de Investigación, Desarrollo Tecnológico e Innovación I+D+I del Principado de Asturias.

OpenXpertya no es un proyecto dirigido desde el Gobierno o entidades estatales, sin embargo OpenXpertya ha sido subvencionado por el Gobierno del Principado de Asturias.

Actividad de la comunidad

Como hemos comentado con anterioridad la actividad de la comunidad es escasa, no existe ningún canal de comunicación que de pie a un reconocimiento de la actividad.

Frecuencia de las Actualizaciones

La última actualización es la versión 3.0, las versiones anteriores no están disponibles en el repositorio sourceforge pero sí en su página web oficial. De la versión anterior el flujo de actualizaciones era constante, aproximadamente era posible descargar cada dos meses una nueva actualización, de esta nueva versión no existen actualizaciones nuevas.

8.3.7.4.- Madurez

Lugares de referencia

Tan solo encontramos su página web oficial:

- www.openxpertya.com

En la actualidad no existe referencia de Partners a nivel hispanoamericano pero a nivel español no he encontrado ninguna referencia.

Destacar el enlace <http://wiki.openxpertya.com>, que en un principio debería existir pero que en la actualidad se encuentra inactivo.

8.4. – Conclusiones a partir de los puntos anteriores

Cada ERP analizado contiene características y atributos que lo hacen distinto a los demás, sea su arquitectura, su facilidad de uso o el nivel de soporte que posee, puede decantar la balanza hacia un lado u otro en la elección del software.

El posterior análisis basado en la experiencia que atesoro tanto a nivel laboral, como tras la realización del proyecto, tratando de mostrar de la manera más objetiva posible y en base a los factores a continuación nombrados la mejor solución en cada caso

La gran mayoría de ERPs analizados están diseñados para pequeñas y medianas empresas, esto no quiere decir que el uso de los mismos en empresas de mayor tamaño no alcance el éxito, pues a pesar de su diseño previo, muchos de ellos lo han alcanzado en grandes empresas

8.4.1. – Flexibilidad

Sobre el criterio de flexibilidad, el primer punto a diferenciar y a tener en cuenta es el uso de metadatos, a excepción de los ERPs Oasis ERP, OpenERP y OpenXpertia, las demás aplicaciones ERPs se apoyan en ellos, de esta manera ayudan al desarrollador teniendo un modelo de programación más sencillo, la carencia de los mismos obviamente dificulta en mayor o menor grado la flexibilidad de los ERPs

La flexibilidad en las actualizaciones es desde mi punto de vista un factor muy importante, depender del grado de personalización que se haya alcanzado en la aplicación para poder actualizar sin riesgos es un paso atrás al escoger un ERP. En esta característica Adempiere, Compiere y Openbravo permiten una actualización exenta de riesgos en lo que se refiere a modificaciones de la aplicación

La internalización es una característica que a priori no parece tan relevante como pudieran ser las dos anteriores, pero si partimos de la base de que un software libre depende en un alto grado de su comunidad y sobre todo de su uso, es entonces cuando dotamos de importancia a esta característica, pues una mayor internalización basada no sólo de traducciones del interfaz, sino que también de las características contables de cada región genera un mayor número de usuarios, así como una comunidad más amplia, logrando un crecimiento al más alto nivel.

En lo que se refiere a la arquitectura del ERP, la mayor parte de las aplicaciones analizadas, contienen una arquitectura en 3 capas o bien está en camino, la clásica cliente/servidor, deja paso a la arquitectura de 3 capas

que aporta robustez a la aplicación. Adempiere y Compiere están en camino de adoptar su aplicación, OpenERP continua trabajando con una arquitectura de cliente/servidor.

La facilidad de uso depende también del alto grado de personalización que posea la aplicación, una aplicación que sea complicada o “peligrosa” de personalizar, obliga al usuario final a acoplarse a la aplicación y no al contrario, independientemente de lo anterior, muchas aplicaciones poseen ventanas de ayuda e incluso personalización a nivel de usuario, de manera que cada usuario tiene una interfaz más acoplada a sus necesidades funcionales. En este punto la aplicación Oasis ERP es la que menos da la talla con respecto a sus competidores analizados.

Tanto en escalabilidad, como en seguridad todas las aplicaciones son altamente escalables y admiten seguridad a nivel de usuario y grupo, pero con detalles que las diferencian. AbanQ permite establecer permisos sobre distintas partes de la aplicación dependiendo del usuario que accede. Adempiere y Compiere ofrece un nivel adicional en el que la seguridad de los datos está basada en Cliente y Organización. Oasis ERP genera logs de acceso por usuario, con toda la información de las acciones realizadas. OpenBravo contiene auditoría por transacción (similar a la de Oasis ERP), así como soporte para la conexión https. Por su parte OpenERP se basa en seguridad por roles. OpenXpertia usa seguridad multiperfil muy similar a la seguridad por roles.

En lo referente a la independencia del sistema operativo todos los sistemas analizados permiten el uso con las dos grandes plataformas (Linux y Windows), aunque unos pocos dan la opción para el sistema operativo de Apple, quizás mi desconocimiento o quizás la realidad, pero no recuerdo ninguna empresa que use Mac a nivel de cliente servidor, por lo que siempre desde mi punto de vista, que no contenga la opción no le doy una gran importancia. OasisERP, Openbravo y Openerp en la actualidad no dan la opción de instalación con Mac. Cada aplicación usa un sistema de gestión de bases de datos gratuito que más se acopla a sus necesidades.

8.4.2. – Soporte

En lo que a soporte nos referimos distinguimos tres conceptos, el soporte de la infraestructura, la formación y la documentación.

En general el soporte de la infraestructura, se basa en los foros, la página web oficial de la aplicación y la de los Partners, en mayor o menor medida todos ofrecen un servicio aceptable, pero con matices. Oasis y OpenXpertia tienen el foro con poco uso o deshabilitado temporalmente, desde mi propia experiencia, las dudas que he tenido analizando su aplicación no han sido respondidas, por lo que el soporte a este nivel no llega

al mínimo requerido. Pese a todo, todas las aplicaciones ofrecen soporte más especializado a nivel de Partners, previo pago.

En lo que respecta a la formación, todas las aplicaciones estudiadas ofrecen formación a todos los niveles, tanto de desarrollo como a de usuario. La formación conlleva siempre en mayor o menor medida un coste.

Al hablar de documentación específica, no nos referimos a cómo es la aplicación, o qué funciones tiene, sino a la documentación necesaria para poder implantar la aplicación a todos los niveles, y es aquí cuando las aplicaciones más potentes marcan diferencia. Openbravo, Adempiere y Compiere tiene una documentación extensa a todos los niveles y bien estructurada. En lo que respecta a las demás aplicaciones, se echan en falta documentación a nivel de desarrollo sobretodo y en menor medida a nivel de instalación.

8.4.3. – Continuidad

Cuando hablamos de continuidad tenemos en cuenta tres factores, la estructura del proyecto, la actividad de la comunidad y frecuencia de las actualizaciones. Un proyecto con una estructura débil ya sea de software libre como propietario está destinado al fracaso. En el ámbito del software libre la actividad de la comunidad es fundamental, es tan importante que muchos autores la utilizan como factor de medida del uso de aplicaciones de software libre. En lo que respecta a las actualizaciones, y en lo que nos atañe que son aplicaciones ERP, las actualizaciones son muy importantes, planes contables, nuevas funcionalidades, son entre otros factores que obligan a actualizar a versiones mejoradas que contengan los requisitos necesarios para que la aplicación no se quede obsoleta.

Todas las aplicaciones ERP analizadas tienen una estructura fuerte y robusta, aunque Adempiere, Compiere y Openbravo, al estar más asentados son probablemente los más fuertes cuando nos referimos a la estructura del proyecto.

Al hablar de actividad en la comunidad, la aplicación ERP de Openbravo es con diferencia la más activa, Adempiere y Compiere, estarían probablemente un escalón por debajo, AbanQ y OpenERP irían a continuación y OpenXpertya y OasisERP irían a la cola en este sentido.

Si nos referimos a la frecuencia de actualizaciones, en Abanq existe un estancamiento considerable desde hace dos años, pese a esto tras la formación de Abanq G2, se esperan nuevas versiones. Tanto Adempiere como Openbravo, alojados en sourceforge y en sus respectivas webs oficiales muestran un flujo constante de actualizaciones, Compiere se retrasa en este aspecto ya que existe un flujo constante de versiones de nueva funcionalidad, pero con un coste económico. Cuando hablamos de Oasis ERP

es más complicado, ya que en el repositorio antes mencionado no se datan de actualizaciones reciente en los últimos años, pese a esto este mismo año está a la disposición de los usuarios una nueva versión estable. Open ERP al igual que el ERP anterior, carecía de un gran movimiento en lo que a actualizaciones se refiere, pese a esto, desde marzo de este año, existe una importante actualización disponible. OpenXpertya ofrecía actualizaciones regulares, y a partir de este año existe una nueva versión.

8.4.4. – Madurez

Todos los proyectos analizados se mantienen en un estado de desarrollo que podemos definir como estable (quizá Abanq, esté en un grado menos). Adempiere como disgregación de Compiere, Compiere y Openbravo, son los proyectos que más años llevan asentados en el paradigma de las aplicaciones ERPs, con lo que su estado es más estable que los demás, un factor importante a tener en cuenta.

Los lugares de regencia se echan en falta en los ERPs AbanQ, OasisERP y OpenXpertya, con tan solo enlaces a sus páginas oficiales. Los restantes ERPs poseen una gran variedad de lugares de referencia, no sólo a lo que respecta a las páginas oficiales de calidad, sino a un gran número de Partners.

8.4.5. - Conclusión

La tarea de elección de un ERP, como ya se ha comentado con anterioridad, es una tarea ardua y compleja, el éxito o fracaso del proyecto consiste en una elección adecuada, partiendo de un estudio previo del lugar en donde se implantará el ERP, para seleccionar la aplicación que más apropiada para el entorno, un error muy general es “matar moscas a cañonazos”, no las aplicaciones más caras (en nuestro caso podríamos decir las mas completas) son las mejores.

Desde mi punto de vista las aplicaciones más maduras y más extendidas estudiadas son obviamente Openbravo, Adempiere y Compiere. Pese a esto Compiere a adoptado una filosofía empresarial respetable, pero que salta las normas en algunos casos de las bases del software libre. La elección entre Openbravo o Adempiere debería ser a priori exitosa, ya que poseen las características necesarias para llegar a una implantación exitosa. Pese a ello Openbravo con respecto a las características mencionadas y analizadas supera en algunos aspectos a Adempire. Y es probablemente el ERP con mejor “salud” de los que hemos mencionado, siempre en el ámbito del software libre, compararlo con las grandes empresas dedicadas a la gestión empresarial, sea SAP o Microsoft, en la actualidad es complicada. Pese a esto el crecimiento de las aplicaciones de software libre, augura un futuro prometedor.

8.5. – Instalación openbravo

8.5.1. – Requisitos de intalación

Openbravo esta basado en un entrono web, existe una versión simple y otra ampliada que incluye CRM y BI, además de funciones estándar de compras, almacenamiento, proyecto, fabricación, dirección de ventas y gestión financiera entre otros.

Para ejecutar el software, la aplicación debe estar instalada en un servidor con MVC-FF (MVC Foundation Framework), para proporcionar soporte a la arquitectura MVC, la cual facilita el desacoplamiento de las áreas de desarrollo, permitiendo el crecimiento sostenible de la aplicación y una mayor facilidad en el mantenimiento del código.

Una vez instalado el MVC, se realizará la instalación de la plataforma java, apache, el sistema de gestión de bases de datos y la actualización de los navegadores web pertinentes.

Puede trabajar bajo los siguientes sistemas operativos:

- Microsoft Windows 7, Vista, XP, 2000 o 2003 server.
- Linux: Red Hat, CentOS, OpenSuse, Debian, Ubuntu, Fedora

Software requerido:

- Plataforma Java 2 edición estándar 5.0 o superior.
- Apache-Tomcat versión 5.5 o Superior
- Apache-ant 1.6 o Superior.

Bases de datos (una de ellas):

- Oracle 10g release 2 (Express, Standard and Enterprise editions).
- PostgreSQL Database Server 8.1.4 o superior.

Navegadores web:

- Firefox 2.0 o Superior
- Internet Explorer 7.0 o Superior

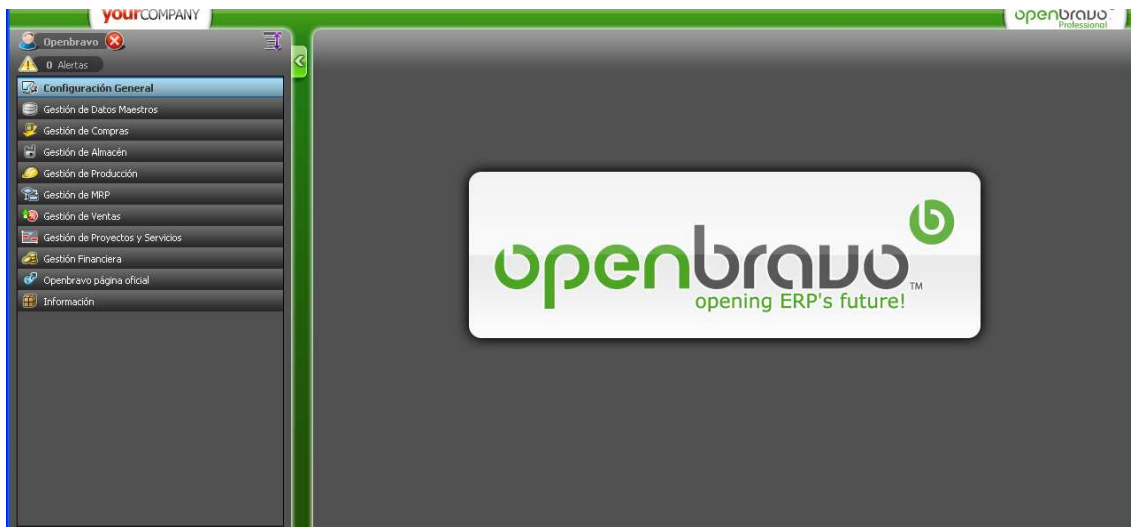
Estos requisitos son los necesarios para la instalación del servidor, para los equipos cliente basta con un navegador compatible.

Tras la instalación de los requisitos anteriormente citados, ya es posible la instalación del software ERP openbravo, bien desde la página oficial o bien desde el host sourceforge.net, desde la página oficial pide una serie de información personal, así como el mail, de esta manera existe la

posibilidad de estar informado de todas las novedades que surgen en este entorno.

La instalación del mismo no conlleva dificultad, tras la inserción de la clave de administrador, es cuando el sistema puede arrancar.

A través del navegador web escribimos la dirección `http://localhost:8080/openbravo/` y tras insertar la contraseña (por defecto usuario Openbravo, password: openbravo) aparecerá una pantalla similar a esta:



Interfaz de Openbravo

En la captura de pantalla anterior podemos ver el interfaz inicial con el acceso a los módulos instalados

8.5.2. – Entorno

8.5.2.1. - Componentes

El manual de usuario de OpenBravo contiene toda la información sobre:

1. Tipos de componentes
2. Selectores
3. Barra de herramientas
4. Barra de pestañas
5. Ventana de identificación
6. Menús
7. Ventanas
8. Diccionario de la aplicación

8.5.2.2. – Módulos

a) Gestión de datos maestros:

Al escoger la opción de gestión de datos maestros dentro de los datos maestros de un negocio se tratan los valores de terceros como pueden ser clientes, proveedores, productos, tarifas, así como toda la información referente a cada entidad. La creación, modificación y edición de cada entidad, se puede hacer de forma sencilla, a través de la barra de herramientas situada en la parte superior del interfaz de la aplicación.

b) Gestión de compras:

A través de la navegación del menú de gestión de compras es posible controlar la administración de la gestión de compras, desde la necesidad de material, incluyendo los pedidos, albaranes y facturas. Incluye una importante herramienta de análisis, por medio de filtros hasta encontrar la información deseable ofreciendo la posibilidad de la extracción en PDF o en HTML.

c) Gestión de almacenes:

El módulo de gestión de almacenes se subdivide en transacciones y consulta de informes. En la parte de transacciones nos da la opción del mantenimiento del almacén, inventario, movimientos, manteniendo el control de las existencias. Brinda la posibilidad de definir de forma personalizada la estructura del almacén. En la opción de herramientas de análisis podemos visualizar informes de todo tipo referente a la gestión del almacén, desde caducidades, producción, stock, etc.

d) Gestión de producción:

Toda la gestión de la producción se puede acondicionar a la estructura empresarial adecuada, permitiendo la configuración de área, puestos de trabajo, procesos, utillajes, etc. Además permite controles de producción, partes de trabajo,.. Y como cada módulo contiene herramientas de análisis a través de informes.

e) Gestión de proyectos y servicios:

Toda la gestión de proyectos se puede configurar a través de éste módulo, sobretodo si la empresa se dedica a proyectos y servicios.

f) Gestión de ventas:

El diseño de Openbravo se realizó para lograr la máxima agilidad y flexibilidad en los procesos comerciales. Pedidos, facturas, albaranes y todas

las herramientas de análisis, así como la configuración de ventas, comisión y marketing, incluyendo punto de venta externo.

g) Gestión financiera:

Toda la gestión financiera fue diseñada para evitar tareas pesadas repetitivas y pesadas en la introducción de datos. Pudiendo gestionar todas las tareas propias del departamento financiero, desde contabilidad, finanzas, hasta gestión de cobros y pagos, así como la gestión de los activos

h) Gestión MRP:

A través de este modulo Openbravo brinda la posibilidad de la planificación en la producción, compras y ventas

i) Business Intelligence:

El componente BI de Openbravo le ayuda a realizar un continuo seguimiento del negocio proporcionando información de ayuda para la toma de decisiones, los cuadros de mando predefinidos permite monitorear diferentes indicadores clave del desarrollo empresarial.

8.6. – Instalación abanq

8.6.1. – Requisitos de intalación

Abanq esta basado en un entorno cliente/servidor, es necesaria la instalación del gestor de bases de datos:

- PostgreSQL 8.2 o superior
- MySQL 4.1 o superior

Puede trabajar bajo los siguientes sistemas operativos:

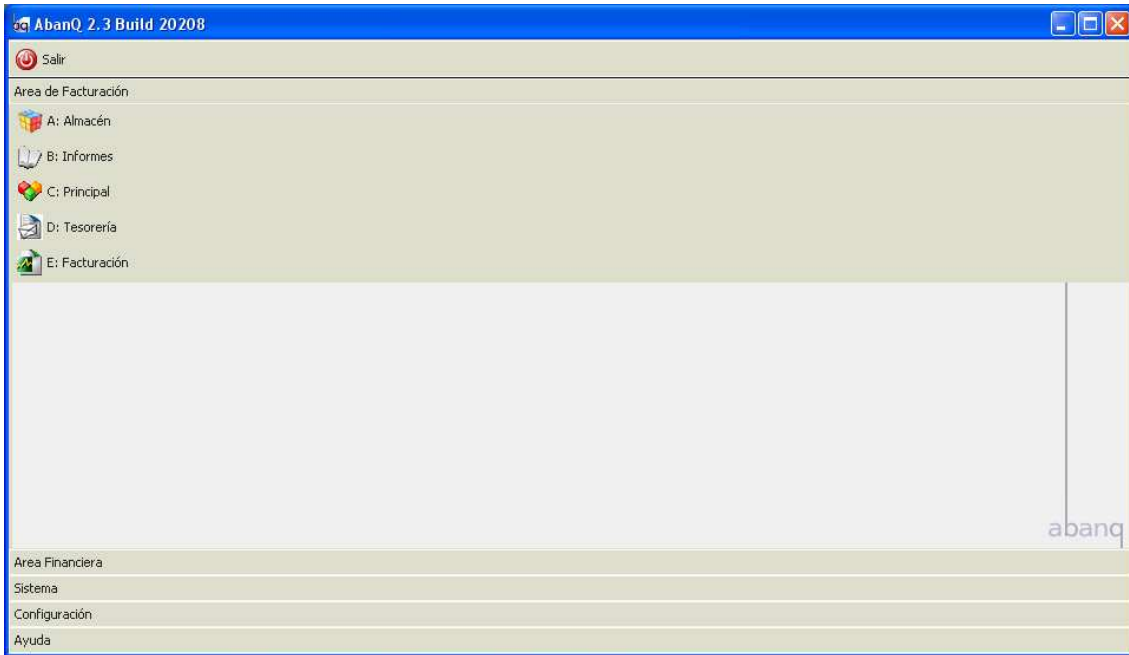
- Microsoft Vista, XP, 2000 o 2003 server.
- Linux
- Mac os

Estos requisitos son los necesarios para la instalación del servidor, existe más información de la instalación básica en el siguiente enlace: http://www.aloj.us.es/tmra1/material_adicional/abanq/InstalarClienteAbanQ.pdf

Tras cumplir los requisitos anteriormente citados, ya es posible la instalación del software ERP openbravo, bien desde la página oficial o bien desde el host sourceforge.net.

La única dificultad existente está en la conexión a la base de datos postgree, una vez instalada, la instalación de la aplicación carece de dificultad.

Tras iniciar la aplicación en el equipo aparecerá un interfaz similar a esta:



Interfaz de Abanq

En la captura de pantalla anterior podemos ver el interfaz inicial con el acceso a los módulos instalados

8.6.2. – Entorno

8.6.2.1. – Módulos

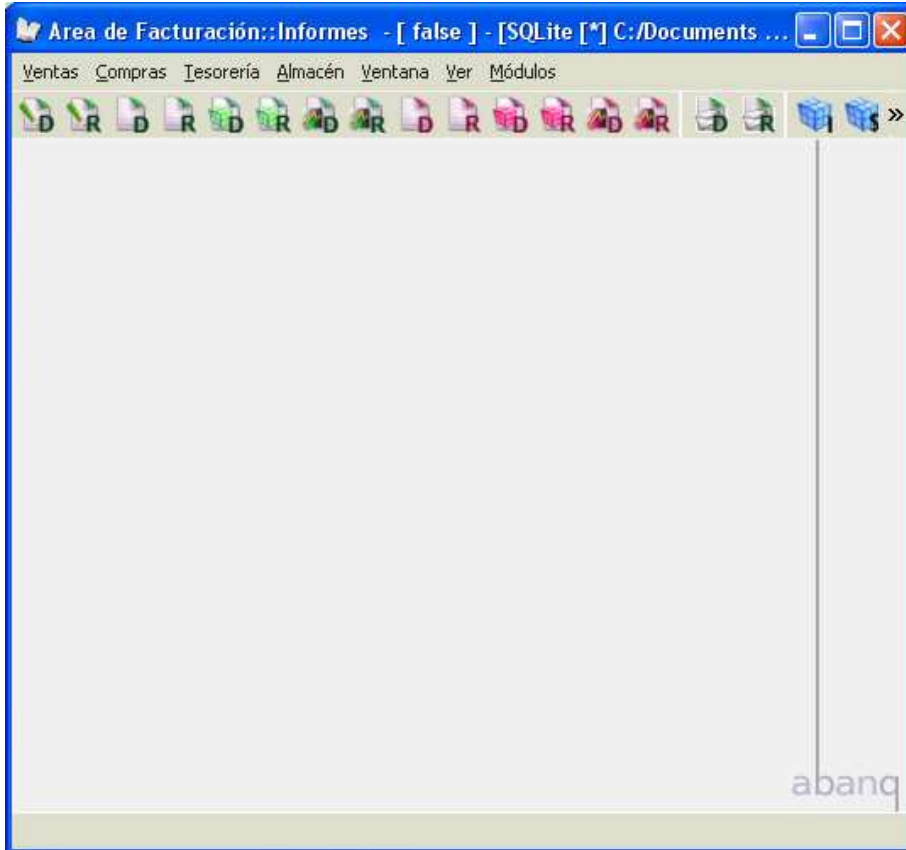
Básicamente la aplicación se subdivide en:

1. Área de facturación
2. Área financiera

8.6.2.2. – Área de facturación

En la que es posible la inserción inicial de la empresa en la que la aplicación funcionará, así como los datos de clientes y proveedores.

En este punto se incluye la gestión de almacén, así como toda el área de tesorería y de facturación. La gestión de almacén permite crear productos y familias de ellos, en cada caso el valor de venta o de compra, con sus correspondientes descuentos. También contiene una alta gama de informes como se ve a continuación.

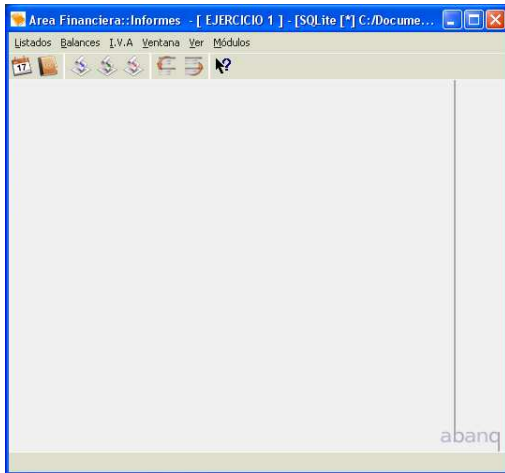


Continuando con el área de facturación vemos la gestión de tesorería y la gestión de la facturación, que contiene la generación de pedidos, la gestión de facturas y albaranes entre otras.

8.6.2.3. – Área financiera

Esta área aporta toda la información respecto al área del departamento financiero, cuentas, diario de iva, generación y edición de asientos.

El otro punto perteneciente es la generación de informes financieros, a través de listados, balances, IVA.



8.7. – Conclusiones instalación

Hoy en día la dificultad en la instalación de aplicaciones se ha reducido enormemente, con la existencia de manuales de usuarios, ayuda en foros y setup más sencillos, la ardua labor de los técnico informáticos resulta más amena, pese a esto la dificultad estriba en la correcta configuración de los mismos, en nuestro caso en particular, no ha existido una gran complicación, ya que la configuración ha sido básica, sobre una base de datos con apenas datos de prueba.

La elección de Openbravo y Abanq para su correspondiente instalación y análisis ha sido básicamente debida a que Openbravo ha sido el ERP más laureado en lo que se refiere al análisis previo y la comparativa con un ERP de menor calidad, intenta de esta manera mostrar las diferencias entre paquetes ERP de distinto nivel.

Tanto a nivel de interfaz, como de modularización, de seguridad y de facilidad de uso, Openbravo ha demostrado ser una aplicación muy superior a Abanq, Abanq tiene un interfaz menos intuitivo y menos agradable a la vista.

Las opciones que muestra Openbravo como por ejemplo el módulo de Business Intelligence es otro factor que mejora a su competidor.

Pese a esto existen usuarios que anteponen la sencillez de Abanq por delante de la complejidad y completitud. Abanq es una evolución del paquete de gestión Facturalux, una aplicación que sin llegar a ser un ERP, estuvo y está muy distribuido a nivel nacional, es más, todavía muchas PYMEs, siguen gestionando la empresa con Facturalux y hojas de cálculo. Por lo que la gran parte de usuarios que usan o usaron ese tipo de aplicación, les será más sencilla adaptarse al paquete ERP Abanq.

Finalmente, el paquete ERP Openbravo es una aplicación de software libre, capaz de soportar todas las necesidades de una forma limpia y

ordenada. Con todo esto y tras el análisis de todos los datos el ERP Openbravo es una gran elección a nivel empresarial.

Business Process Management (BPM)

9. - Business Process Management (BPM)

- 9.1. - Introducción
- 9.2. - Definición
- 9.3. - El BPM clave en las organizaciones
- 9.4. - Alcance del BPM
- 9.5. - Arquitectura Empresarial – Modelos de Negocio
- 9.6. - Automatización y orquestación de procesos, organización y sistemas
- 9.7. - Beneficios
- 9.8. - Monitorización de procesos y recursos empresariales
- 9.9. - Diez prácticas recomendadas de BPM
- 9.10. - Los diez escollos a evitar en BPM
- 9.11. - Conclusiones



9.1.- Introducción

Hace unos cuantos años nadie había oído hablar de Business Process Management (BPM), pero ha irrumpido en la escena global hasta convertirse en la tendencia de gestión empresarial y tecnológica más popular de la década. En la actualidad en cualquier empresa o sector industrial, ya sea público o privado, se empieza a estudiar el movimiento hacia el proceso, o la gestión de procesos e incluso la mejora de los procesos. Existen métodos conocidos de mejora de los procesos como Six Sigma de nuevas tecnologías como Business Activity Monitoring (BAM), supervisión de la actividad de negocio, o Service-Oriented Architecture (SOA), la arquitectura orientada a servicios.

BPM representa la culminación de la experiencia, pensamiento y desarrollo profesional de todo un colectivo en la gestión empresarial durante las pasadas décadas. Coloca al cliente en primer lugar. Se centra en el negocio. Faculta a los individuos de cualquier rincón de una empresa para alcanzar un mayor éxito.

Reúne a personas y sistemas. BPM es donde se condensan todas las elevadas ambiciones y mejores estrategias. Al unir todo esto se obtiene una mezcla que puede parecer bastante confusa. Pero en realidad, BPM es un concepto muy sencillo. Es un conjunto de métodos, herramientas y tecnologías utilizados para diseñar, representar, analizar y controlar procesos de negocio operacionales; un enfoque centrado en los procesos para mejorar el rendimiento que combina las tecnologías de la información con metodologías de proceso y gobierno.

9.2 - Definición

Las empresas necesitan constantemente adaptar y mejorar sus procesos, pero frecuentemente están frenadas por aplicaciones y sistemas que no están preparados para explotar nuevas oportunidades y adaptarse a los cambios de forma ágil. El BPM, con sus enfoques evolucionados y sus tecnologías punta, ha emergido como el elemento clave para proveer a las organizaciones de la “Agilidad” y “Flexibilidad” necesaria para responder de forma rápida a los nuevos cambios y oportunidades de mercado.

Popularmente se llama Gestión de Procesos de Negocio (BPM – Business Process Management) a “la metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. Como su nombre sugiere, BPM se enfoca en la administración de los procesos del negocio”.

Matizando la definición anterior de BPM como “Un conjunto de herramientas, tecnologías, técnicas, métodos y disciplinas de gestión para la identificación, modelización, análisis, ejecución, control y mejora de los

procesos de negocio. Las mejoras incluyen tanto cambios de mejora continua como cambios radicales.

Resaltar que no consiste en una solución tecnológica. Es mucho más, es un conjunto de herramientas, tecnologías, técnicas, métodos y disciplinas de gestión. Esto permite identificar procesos, modelizar, analizar el comportamiento, ejecutar los procesos (automatización), control la ejecución de los procesos y optimizar los procesos para la mejora continua.

En un mundo donde las tres “C”, Comunicación, Colaboración y Coordinación ya es la normalidad, se requieren de tecnologías que orquesten los procesos, la organización, los sistemas, y los clientes, colaboradores y otros entes externos.

Pero a su vez, las empresas exigen un alto ROI (Retorno de la Inversión), y ya muchas de ellas han comprobado que este tipo de tecnologías y enfoques lo aporta, consiguiendo espectaculares mejoras y beneficios.

Al hablar de BPM 360° se hace referencia a cubrir la mejora continua de los procesos de una empresa (Ver Figura 1). Normalmente se parte de un análisis de la situación actual de los procesos empresariales (Monitorización de los Procesos Actuales, recogiendo algunos indicadores de referencia) que indica qué se desea mejorar para conseguir unos resultados empresariales. Tras conocer que tenemos que desarrollar un proyecto BPM, se comienza a Modelizar y Diseñar Procesos de Negocio, creando lo que se denomina como Arquitectura Empresarial (se detecta el mapa de procesos de la empresa y se modelizan los procesos para su automatización, así como se definen los nuevos indicadores a controlar para orientarnos hacia los objetivos de negocio). En la Automatización e Integración, se ejecutan los procesos de negocio utilizando motores de Workflow y soluciones de integración de aplicaciones (para conectarnos con los aplicativos ya existentes) y de datos. Según se van ejecutando los procesos de negocio, se irá controlando el comportamiento mediante la monitorización (detectando cargas de trabajo, cuellos de botella, ineficiencias, buenos resultados, puntos de mejora...). En la monitorización se detectan mejoras a realizar, por lo que se empieza de nuevo el ciclo revisando la modelización y haciendo los ajustes necesarios de diseño. Siendo con todo esto un proceso de mejora continua.

Con el término BPM360° (Figura 9.1), significa que en BPM tenemos diferentes fases:

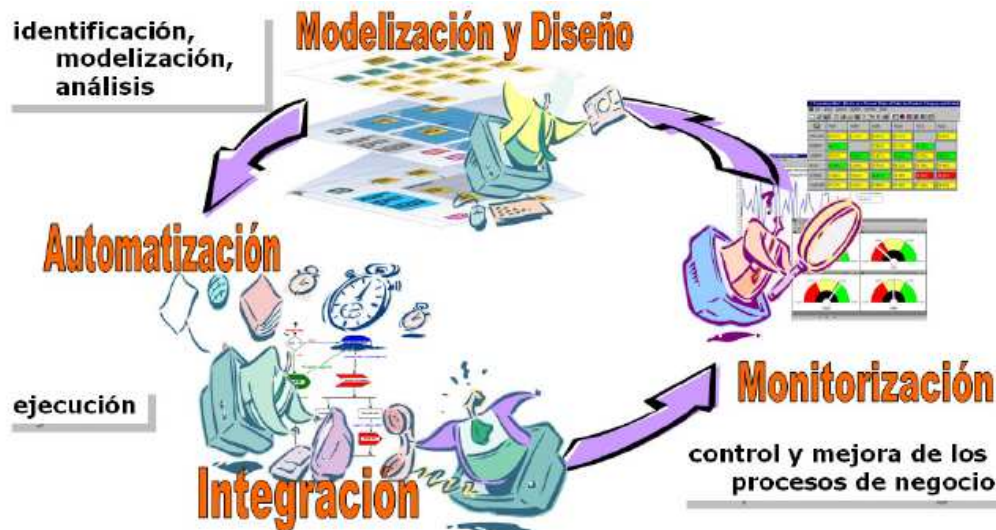


Figura 9.1.-BPM 360°. Fuente: BPM Business Process Management – Gestión de Procesos de Negocio

- 1- Análisis de Procesos: Analizar los procesos actuales o nuevos para conocer cómo definirlos (definición de tareas, cómo ejecutar dichas tareas, quién realiza las tareas, dónde se realizan, qué datos utiliza, qué reglas de negocio deben cumplirse...)
- 2.- Diseño de Procesos: Diseñar los procesos de negocio siguiendo una notación BPM
- 3.- Ejecución de los procesos de negocio: automatizar los procesos con un motor de workflow (flujo de trabajo) e integrar las aplicaciones y datos para que exista una orquestación adecuada.
- 4.- Monitorización y Análisis: Monitorizar las actividades de negocio y relacionar la información de los procesos con la estrategia empresarial para conocer la dirección a los objetivos o no, y así tomar decisiones reactivas.

9.3. – El BPM clave en las organizaciones

Uno de los principales retos de las organizaciones es conseguir la flexibilidad y agilidad necesarias para adaptarse a los rápidos y continuos movimientos del mercado, gestionando los riesgos operacionales y financieros, incrementando a su vez la rentabilidad empresarial y la satisfacción de sus clientes. Para ello, hoy en día, las experiencias de muchas organizaciones que han implantado Business Process Management (BPM) reportan grandes beneficios, con altísimos ahorros en costes y reducciones importantes en tiempos de servicios a sus clientes, dándose cuenta que BPM junto con sus tecnologías se hacen imprescindibles para convertir los retos en una realidad.

Los procesos y recursos empresariales deben dirigirse hacia la meta estratégica de la empresa, pero debemos ser capaces de conocer qué está impidiendo el no llegar a los objetivos marcados, qué cuellos de botella están

ocurriendo, cómo solventar las excepciones y cómo orquestar los procesos y recursos para conseguir el reto buscado. Para lograr tener un conocimiento y control absoluto de los procesos y recursos empresariales, se requieren de tecnologías que orquesten los procesos, la organización y los sistemas con los clientes, colaboradores y otros entes externos que garanticen el buen funcionamiento de la empresa hacia los objetivos empresariales. La solución hay que buscarla en BPM y sus tecnologías SOA (Services-Oriented Architecture), BPA (Business process automation), BRMS (Business Rule Management System), BAM (Business Activity Monitoring), y BI (Business Intelligence, comentado en capítulos anteriores).

Para tener éxito en la implantación del BPM, las organizaciones no deben de cometer el gran error de centrarse solo en las tecnologías, sino en el conocimiento, dominio y mejora continua de sus procesos, datos, y recursos empresariales. Se sugiere detectar una necesidad de mejora en la empresa para la primera experiencia en BPM, de forma que se haga un análisis del proceso actual, se optimice, y se fijen los indicadores clave que muestren los hitos conseguidos. La monitorización del proceso lleva a una mejora continua.

La gestión de procesos es cada vez una prioridad en el 65% de las empresas. Las organizaciones buscan una agilidad empresarial, que optimice los procesos de negocio, que controle los riesgos operativos, que gestione los recursos y se encamine hacia el cumplimiento de objetivos empresariales.

9.4. - Alcance del BPM

El alcance del BPM está conformado por un conjunto de soluciones de software especializado que logra automatizar, a día de hoy y de una manera eficiente, todo el ciclo de vida de los procesos, reglas y servicios de negocio, desde la identificación y modelización, hasta la monitorización, permitiendo así un entorno de Mejora Continua totalmente automatizado.

La siguiente figura (figura9.2) muestra las distintas tecnologías del BPM por cada una de las etapas del ciclo de vida de la gestión de los procesos del negocio, definiendo así el alcance del mismo.

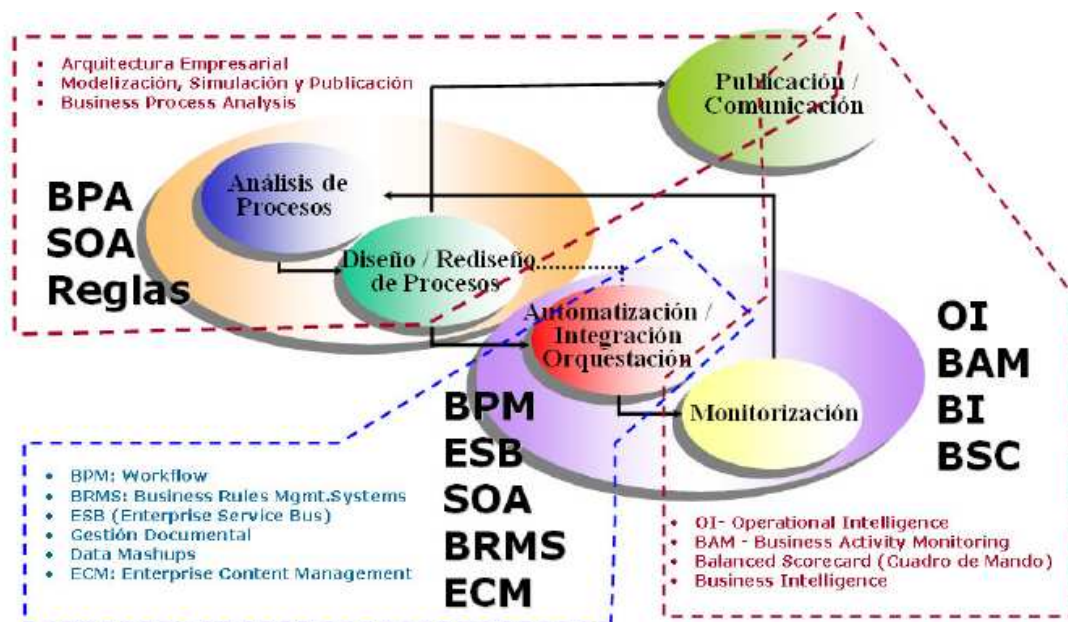


Figura 9.2.- Tecnología del BPM. Fuente: BPM Business Process Management – Gestión de Procesos de Negocio

9.5. - Arquitectura Empresarial – Modelos de Negocio

Es determinante que para poder gestionar cualquier elemento empresarial, hay que:

- Tenerlo adecuadamente identificado y definido.
- Asignarle objetivos y metas.
- Disponer de medidas para valorar su actuación.

El Proceso es ese elemento empresarial fundamental e intangible que está presente en toda la organización, pero que aún muchas empresas no lo están gestionando. Por esta y muchas otras razones tales como competitividad, nuevos canales, compras y fusiones, y nuevas tecnologías y soluciones, cada vez hay más empresas que implementan la Gestión de Procesos en sus organizaciones. Para lograr implementar esta gestión, se requiere de un elemento fundamental que se denomina “Modelos de Negocio”. Dichos modelos son un conjunto de técnicas y representaciones gráficas plasmadas sobre una base de datos orientada a objetos, y basados en estándares, que permiten representar y entender cuáles son:

- Los puntos de encuentro con los clientes
- Los puntos de encuentro con proveedores, colaboradores y otros entes externos

- Los problemas y oportunidades de mejora
- Los procesos, datos y flujos de información
- La organización
- Los sistemas informáticos
- Los indicadores de gestión y calidad

Y como gestionar y optimizar éstos de forma que asegure el más alto grado de satisfacción al cliente, manteniendo un balance entre nivel de calidad y costes. La utilidad que se les da a los Modelos de Negocio varía de empresa a empresa según sus necesidades, objetivos y prioridades.

No obstante, desarrollándolos con los enfoques y técnicas adecuadas, tienen muchas utilidades las cuales se enumeran las más relevantes a continuación:

- Hacer Análisis de Impacto Funcionales, Organizativos y de Sistemas.
- Desarrollar y Evolucionar Sistemas más Integrados, más de Negocio.
- Disponer de una base más sólida al Plan de Sistemas y Tecnología.
- Implantar tecnología BPM / WORKFLOW.
- Mejora continua de Procesos de Negocio (Reingeniería - Rediseño).
- Apoyar a los procesos de “Benchmarking”.
- Diseño y Reestructuración Organizativa.
- Formar y Guiar al personal de la Organización.
- Calidad Total - ISO 9000.
- Diseño y Lanzamiento de Nuevos Productos y Servicios.
- ABM / ABC (Activity Based Management / AB Costing).
- Gestión de Competencias.
- Control Interno.
- Implantar ITIL.

9.6. - Automatización y orquestación de procesos, organización y sistemas

Muchas organizaciones se han dado cuenta de que aunque han hecho grandes inversiones en tecnologías, sistemas y aplicaciones, aún no han alcanzado el control total de cada proceso, de principio a fin, además de la flexibilidad y agilidad necesaria.

Parte de estas tecnologías, conocida tradicionalmente como WorkFlow (flujo de trabajo), ha evolucionado desde la simple automatización del enrutamiento de documentos y actividades entre personas, a la coordinación y orquestación de los procesos de negocio utilizando todos los recursos (trabajadores, proveedores, organizaciones, aplicaciones, documentos, imágenes, datos, comunicaciones y otros). Además, las tecnologías para la integración de aplicaciones, motores de reglas de negocio, servicios web, ESB (enterprise service bus o en castellano bus de servicios de empresa (BSE) consiste en un combinado de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos), SOA (Services-Oriented Architecture, en castellano arquitectura orientada a servicios, es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio) y otras tecnologías complementarias, están permitiendo implementar soluciones cada vez más eficientes y más ágiles.

Para entenderlo mejor, a través la figura 9.3 podemos ver que existen diferentes capas en la arquitectura empresarial: Bases de datos, Sistemas y Aplicaciones, Procesos de Negocio y Roles (Clientes, personal, proveedores, partners, etc.). El objetivo de un sistema de flujo de trabajo (workflow) es, a través de un motor gestionar de forma automatizada los procesos y flujo de actividades, documentos, imágenes y datos, orquestando e integrando los Recursos Informáticos y los Roles:

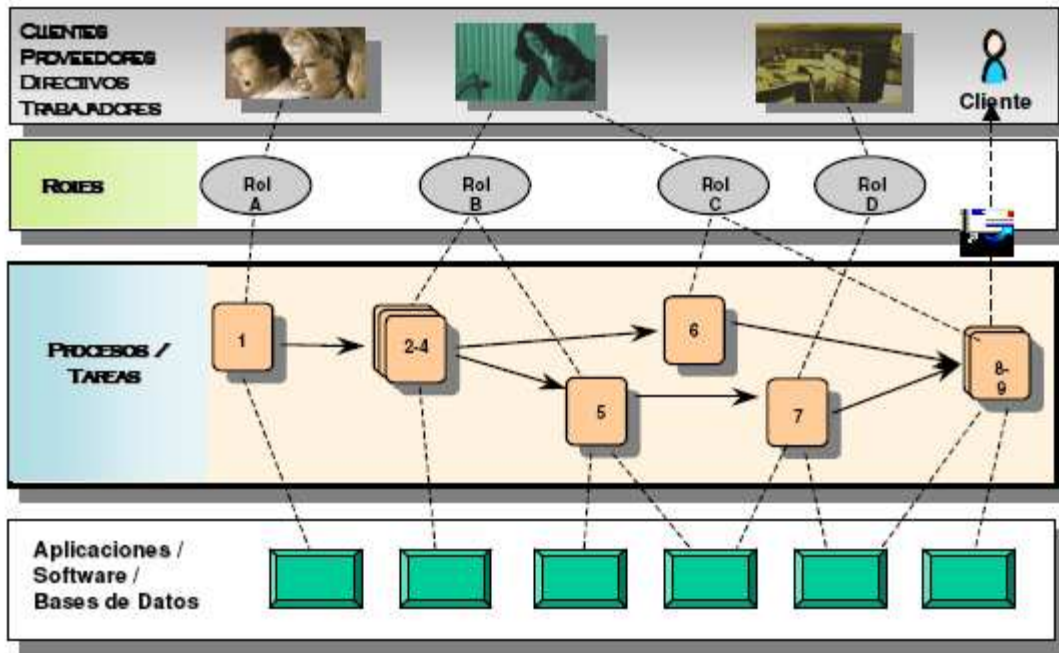


Figura 9.3.- Capas en la arquitectura empresarial. Fuente: BPM Business Process Management – Gestión de Procesos de Negocio

Con BPM:

- El trabajo no queda atascado o extraviado.
- Los jefes pueden enfocarse más en los problemas del negocio y del personal, tal como el rendimiento y capacitación individual, mejoras de procedimientos, y casos especiales, más que en la rutina de asignación de tareas.
- Los procedimientos son formalmente documentados y seguidos de forma exacta y estándar, asegurando que el trabajo es llevado a cabo en la forma planificada, cumpliendo a su vez todos los requerimientos y normas del negocio y externos.
- La persona adecuada, dispositivo o sistema es asignado a cada caso, y los casos más importantes o críticos en el tiempo, son asignados primero. Los usuarios no gastan tiempo escogiendo sobre cual caso trabajar, aplazando quizás aquellos casos más importantes pero de mayor dificultad.
- Se logra el procesamiento paralelo, donde dos o más actividades no dependientes pueden ser realizadas concurrentemente, generando así beneficios en cuanto a reducción de tiempo de los procesos, mejor servicio al cliente y reducción de costes.
- Se convierte el entorno de trabajo de “Reactivo” a un entorno “ProActivo”, con todas las ventajas y beneficios que esto conlleva.

- Asignación de actividades a las personas de forma automática y según cualquier criterio, o según cargas de trabajo.
- Recordar a las personas sus actividades, las cuales son parte de una cola de WorkFlow.
- Optimizar la colaboración entre personas que comparten actividades.
- Automatizar y controlar el flujo de documentos, datos e imágenes.
- Asignarle proactivamente a las personas que deben ejecutar las actividades, todos los recursos necesarios (Documentos, información, Aplicaciones, etc.) en cada una de ellas.
- Definir y controlar “alertas” según criterios de tiempo, de evento o de condición, provocando así algún mensaje a un supervisor, un “escalado” de actividades a otras personas para que las resuelvan, y/o una resignación automática.
- Modificar los procesos y gestionar excepciones “en vivo”, o “al vuelo”, y desde cualquier lugar, es decir, permitir modificar cualquier instancia de proceso ya iniciada, sin necesidad de volver a iniciarla..
- Proveer una vista “on-line” para supervisores del estado e histórico de cada instancia de proceso, de cada actividad, y del desempeño de las personas.
- Hacerles llegar a cada persona sus actividades y alertas, independientemente de su ubicación geográfica, a través de la WEB, Email, SMS, o cualquier otro dispositivo móvil.
- Proveer métricas para responsables de áreas, organizadores, gestores de procesos y calidad, tanto para efectos de mejora continua como de indicadores de calidad y de gestión.
- Integrarse fácilmente con otros sistemas, aplicaciones y ERPs.
- Proveer un alto nivel de soporte para la interacción humana.

9.7. - Beneficios

Los beneficios, tanto tangibles como intangibles, son numerosos. A continuación se describen los más importantes:

- Mejora la atención y servicio al cliente.

- Incrementa el número de actividades ejecutadas en paralelo.
- Minimiza el tiempo requerido por los participantes para acceder a la documentación, aplicaciones y bases de datos.
- Disminuye “drásticamente” el tiempo de transferencia de trabajo, información y documentos entre actividades.
- Asegura la continua participación y colaboración de todo el personal en el proceso.
- Disminuye “drásticamente” el tiempo que los participantes, supervisores y administradores necesitan para conocer la situación de un ítem de trabajo
- Simplificación de salidas automáticas. Documentos Word, Faxes, e-mails, mensajes cortos a móviles, etc.
- Disponibilidad de mecanismos para una mejor gestión y optimización de procesos.

9.8. - Monitorización de procesos y recursos empresariales

La gestión del rendimiento empresarial es el proceso de medir y analizar indicadores claves con el fin de ser más eficientes y lograr objetivos tácticos y estratégicos, bajo cuatro perspectivas: el aprendizaje y el crecimiento, los procesos de negocio, el cliente, y las finanzas. Los procesos y recursos empresariales deben ser monitorizados con el fin de saber cuáles son sus resultados, su rendimiento, y su comportamiento. Sólo de esta manera se puede implementar en la empresa u organización un ciclo permanente de Mejora Continua, y tener realmente dominados nuestros procesos de negocio.

Las diferentes tecnologías que se están implantando en esta etapa del ciclo, son el BAM (Business Activity Monitoring), el Business Intelligence y el Cuadro de Mando Integral o Balanced Scorecard, y el CPM (Corporate Performance Management). La monitorización puede ser activa o en “tiempo real”, o más bien pasiva o de análisis posterior. Sea una u otra, ésta permite a las empresas reaccionar a tiempo, cambiar procesos y recursos existentes, e incluso, cambiar sobre la marcha la terminación de aquellos procesos que están fallando.

9.9. – Diez prácticas recomendadas de BPM

Estas son diez recomendaciones a tener en cuenta para estar en el buen camino del éxito de BPM:

- Establecer una organización interdisciplinaria que impulse y respalde la orientación de los procesos. Los equipos de procesos y los propietarios de los procesos deben elaborar planes juntos, reunirse periódicamente y trabajar en colaboración, así como asegurarse de que el principal stakeholder (parte interesada) participe directamente.
- La dificultad de un BPM estriba en que los conocimientos pueden ser difíciles de adquirir, pese a todo, la mayor parte de la información está en el entorno, la formación tanto de los empleados como del encargado del estudio de los procesos es fundamental para el éxito del mismo.
- La búsqueda y compromiso de los ejecutivos con mayor experiencia es fundamental para el éxito de la implantación de un BPM.
- Los programas BPM involucran a muchas partes interesadas, cada una con su propia perspectiva y definición del éxito. Ser claro en lo referente a las propuestas de valor de sus programas y proyectos, es fundamental para evitar que aumenten las expectativas.
- Seleccionar la metodología de mejora y gestión de los procesos que sea más adecuada, y convertirla en la piedra angular de su arquitectura de procesos, debe ser la base efectiva para el BPM.
- La tecnología adecuada. Es importante elegir la tecnología que mejor satisfaga las necesidades y requisitos. La tecnología varía ampliamente de un proveedor a otro.
- La finalidad de BPM es crear valor para el cliente. Utilizar el BPM para ver al negocio como lo ven los clientes. A los clientes no les importa cómo se hacen las cosas, lo único que les preocupa es experimentar un servicio excepcional y recibir lo que han pedido, al mejor precio y en el momento en que lo desean.
- La importancia de seleccionar un proyecto que proporcione la mayor rentabilidad para el negocio y que se pueda completar en tres meses o menos, frente a abarcar mas procesos.
- Es fundamental primero diagnosticar el estado actual de los procesos antes de implementar cambios en los mismos, sólo entonces se actúa desde la posición adecuada del conocimiento.

- BPM es un sistema para el cambio. Está concebido para ayudar a identificar dónde se necesita el cambio y para realizar cambios rápidamente y pasar al siguiente nivel en cuanto a rendimiento operacional se refiere. Con un conjunto de herramientas tan potente, es necesario que el BPM se rodee de elementos de apoyo para el cambio. Gestionar el cambio, implementar políticas para realizar cambios, articular aprobaciones de cambio, reconocer eventos de cambio, medir el cambio y finalmente recompensar el cambio.

9.10. - Los diez escollos a evitar en BPM

De esta manera es posible evitar los diez errores que hagan fracasar un BPM

- El entusiasmo precoz de la tecnología BPM puede provocar el avance sin metodología, sin arquitectura y sin procesos, esto puede provocar el fracaso, es obligatorio el tomar el tiempo adecuado para asentar las dimensiones de negocio, de procesos y de gestión de BPM antes de atacar la tecnología.
- Otro escollo a evitar es el pensamiento funcional, es obligatorio un pensamiento integral, acerca de cómo se junta una cadena de valor y cómo el papel que desempeña, el rendimiento y la productividad crean valor dentro del proceso final.
- Hacer del BPM una iniciativa velada de reducción del personal, lleva al fracaso. Las personas son las que hacen que los procesos funcionen. La finalidad de BPM es ayudar a la gente a trabajar de manera más efectiva y a generar más valor.
- Resolver los problemas de manera puntual es otro escollo para el fracaso. La búsqueda de formación y desarrollo profesional de todo el personal desde los altos ejecutivos hasta los usuarios ayudará al cambio global de los procesos en beneficio de la empresa.
- Otro escollo es dejar a los usuarios sin apoyo. Un BPM faculta a los propietarios de los procesos y a los que participan en estos a implementar el cambio, compartiendo responsabilidades.
- Ignorar a los usuarios finales puede llevar al fracaso de un BPM, en cambio, tratándolos como si fueran clientes y haciendo que sean más productivos de forma que sus tareas adquieran más valor.
- No valorar lo realizado. Al Instalar algo en tres meses cuando suele tardar dos años es motivo de celebración. Conseguir un elevado objetivo empresarial de productividad o de satisfacción del cliente es

motivo de celebración. Los proyectos BPM pueden ser más cortos y las mejoras más graduales, pero debe medir y celebrar el éxito.

- Definir el marco de forma inamovible. BPM fue creado para ayudar en la creación de procesos de adaptación. Pero tiene que estar diseñado para la flexibilidad. No limitarse únicamente a definir de un modo inamovible la respuesta de hoy a costa de crear la flexibilidad que garantice que la respuesta puede cambiar y ser efectiva en el mundo del mañana.
- Basarse en el instinto y no en la visibilidad proporcionada por el BPM para la toma de decisiones basada en hechos. Si no se permite que los datos controlen las decisiones, y si se retrocede y se deja que la intuición y la tradición sean las que las controlen, se está desperdiciando la inversión y las oportunidades.
- La automatización de los errores. Si un proceso se estropea, solo generará automáticamente errores con más rapidez. Que BPM permita un nivel de automatización sin precedentes en lo referente a acciones, actividades y decisiones, no significa que se tenga que comenzar a automatizar las cosas.

9.11.-Conclusiones

La implantación del BPM y los BPMS, está emergiendo como un factor clave y estratégico, el cual las organizaciones están adoptando con más frecuencia para mejorar sus procesos y recursos empresariales. Constituirán uno de los principales ejes de inversión de TI en las Empresas y Administración Pública en los próximos años.

Claramente la tecnología BPM, combinada con una adecuada Gestión de Procesos, debe tener características específicas para ofrecer flexibilidad y agilidad en la evolución y dinamismo de los procesos de negocio y sistemas informáticos asociados.

El primer requisito es que el proceso automatizado debe ser fácil de modificar sin ayuda de un programador, de forma que la barrera del cambio disminuya. La tecnología BPM ha evolucionado en esta dirección con la introducción de descripciones gráficas de los procesos, motores de reglas de negocio, y otros mecanismos, y la posibilidad de modificar el proceso de forma inmediata, sobre la marcha y sin interrupciones.

Business Intelligence

10.-Business Intelligence

- 10.1. - Introducción
- 10.2. - Definición
- 10.3. - Componentes de Business Intelligence.
 - 10.3.1. - Fuentes de información
 - 10.3.2. - Calidad de los datos
 - 10.3.3. - Proceso de extracción, transformación y carga (ETL).
 - 10.3.4. - Herramientas ETL
 - 10.3.5. - *Datawarehouse* o almacén de datos
 - 10.3.6. - Gestión del *datawarehouse*
 - 10.3.7. - Herramientas de *Business Intelligence*
 - 10.3.8. - Visualización
 - 10.3.9. - ¿Quiénes son los usuarios de las herramientas de *Business Intelligence*?
- 10.4. - Necesidad y fases de planificación de un proyecto BI
- 10.5. - Elementos clave para el éxito o fracaso de un proyecto BI
- 10.6. - Usos de Business Intelligence y nuevas tendencias.
- 10.7. - Conclusiones

10.1. – Introducción

Cada vez es más importante saber qué está pasando en nuestro mercado y en nuestras propias organizaciones. El tiempo de que disponemos para acceder a esa información es cada vez menor; consecuentemente, necesitamos obtener la información más rápidamente para analizarla y tomar decisiones a partir de ella.

En muchos casos no hemos prestado suficiente importancia a la información para la toma de decisiones, nos hemos centrado en aquellos sistemas transaccionales que deben soportar el día a día de nuestras organizaciones. La implementación de sistemas transaccionales no es sencilla por la complejidad y la casuística de los mismos. El verdadero valor de la información se revela cuando a partir de ella somos capaces de descubrir conocimiento.

Este es el verdadero objetivo de la *Business Intelligence*.

10.2. – Definición

La traducción más habitual es la de “Inteligencia de Negocio”. El objetivo básico de *Business Intelligence* es apoyar de forma sostenible y continuada a las organizaciones para mejorar su competitividad, facilitando la información necesaria para la toma de decisiones.

El primero que acuñó el término fue Howard Dresner (“A Brief History of Decision Support Systems, version 4.0”) que, cuando era consultor de Gartner, popularizó *Business Intelligence* o BI como un término paraguas para describir un conjunto de conceptos y métodos que mejoraran la toma de decisiones, utilizando información sobre que había sucedido (hechos).

Mediante el uso de tecnologías y las metodologías de *Business Intelligence* pretendemos convertir datos en información y a partir de la información ser capaces de descubrir conocimiento.

Para definir BI partiremos de la definición del glosario de términos de Gartner (www.gartner.com):

“BI es un proceso interactivo para explorar y analizar información estructurada sobre un área (normalmente almacenada en un datawarehouse), para descubrir tendencias o patrones, a partir de los cuales derivar ideas y extraer conclusiones. El proceso de Business Intelligence incluye la comunicación de los descubrimientos y efectuar los cambios. Las áreas incluyen clientes, proveedores, productos, servicios y competidores.”

En esta definición podemos descomponerla de forma detallada:

- **Proceso interactivo:** al hablar de BI estamos suponiendo que se trata de un análisis de información continuado en el tiempo, no sólo en un momento puntual. Aunque evidentemente este último tipo de análisis nos puede aportar valor, es incomparable con lo que nos puede aportar un proceso continuado de análisis de información, en el que por ejemplo podemos ver tendencias, cambios, variabilidades, etc.
- **Explorar:** En todo proyecto de BI hay un momento inicial en el que por primera vez accedemos a información que nos facilita su interpretación. En esta primera fase, lo que hacemos es “explorar” para comprender qué sucede en nuestro negocio; es posible incluso que descubramos nuevas relaciones que hasta el momento desconocíamos.
- **Analizar:** Pretendemos descubrir relaciones entre variables y tendencias, es decir, cuál puede ser la evolución de la variable, o patrones. Si un cliente tiene una serie de características, cuál es la probabilidad que otro con similares características actúe igual que el anterior.
- **Información estructurada y *datawarehouse*:** La información que utilizamos en BI está almacenada en tablas relacionadas entre ellas. Las tablas tienen registros y cada uno de los registros tiene distintos valores para cada uno de los atributos. Estas tablas están almacenadas en lo que conocemos como *datawarehouse* o almacén de datos.
- **Área de análisis:** Todo proyecto de BI debe tener un objeto de análisis concreto. Nos podemos centrar en los clientes, los productos, los resultados de una localización, etc., qué pretendemos analizar con detalle y con un objetivo concreto: por ejemplo, la reducción de costes, el incremento de ventas, el aumento de la participación de mercado, el ajuste de previsiones de venta, el cumplimiento los objetivos de venta presupuestados, etc.
- **Comunicar los resultados y efectuar los cambios:** Un objetivo fundamental del BI es que, una vez descubierto algo, sea comunicado a aquellas personas que tengan que realizar los cambios pertinentes en la organización para mejorar nuestra competitividad.

El origen de *Business Intelligence* va ligado a proveer acceso directo a la información a los usuarios de negocio para ayudarles en la toma de decisiones, sin intervención de los departamentos de Sistemas de Información

10.3. – Componentes de Business Intelligence.

En la figura 10.1 vemos los distintos componentes de Business Intelligence.

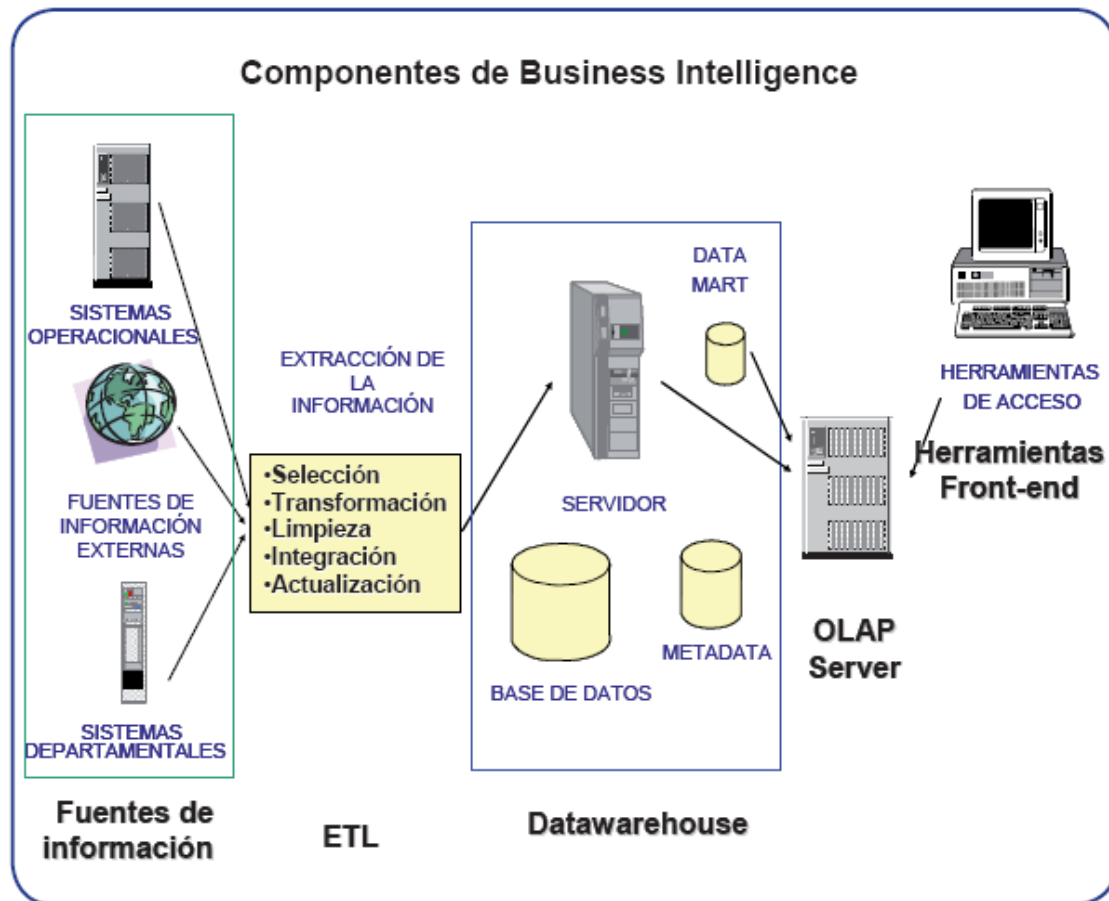


Figura 10.1.- Componentes de Business Intelligence. Fuente: Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

Los componentes son:

- Fuentes de información, de las cuales partiremos para alimentar de información el *datawarehouse*.
- Proceso ETL de extracción, transformación y carga de los datos en el *datawarehouse*. Antes de almacenar los datos en un *datawarehouse*, éstos deben ser transformados, limpiados, filtrados y redefinidos. Normalmente, la información que tenemos en los sistemas transaccionales no está preparada para la toma de decisiones.
- El propio *datawarehouse* o almacén de datos, con el Metadata o Diccionario de datos. Se busca almacenar los datos de una forma que maximice su flexibilidad, facilidad de acceso y administración.

- El motor OLAP, que nos debe proveer capacidad de cálculo, consultas, funciones de planeamiento, pronóstico y análisis de escenarios en grandes volúmenes de datos. En la actualidad existen otras alternativas tecnológicas al OLAP.
- Las herramientas de visualización, que nos permitirán el análisis y la navegación a través de los mismos.

Para describir los distintos componentes vamos a comenzar primero por las fuentes de información, seguiremos con el resto de componentes y finalizaremos por las herramientas de visualización. Seguiremos este orden a fin de conocer los distintos componentes que forman la solución, aunque éste no será el orden que seguiremos en un proyecto real.

En un proyecto real debemos definir primero cuáles son los objetivos y el alcance de la solución, qué modelos de negocio queremos analizar. Con esta información es mucho más fácil tomar las decisiones necesarias en cada uno de los componentes.

10.3.1. – Fuentes de información

Las fuentes de información a las que podemos acceder son:

- Básicamente, de los sistemas operacionales o transaccionales, que incluyen aplicaciones desarrolladas a medida, ERP, CRM, SCM, etc.
- Sistemas de información departamentales: previsiones, presupuestos, hojas de cálculo, etcétera.
- Fuentes de información externa, en algunos casos comprada a terceros, como por ejemplo estudios de mercado (Nielsen en distribución de gran consumo, IMS de la industria farmacéutica). Las fuentes de información externas son fundamentales para enriquecer la información que tenemos de nuestros clientes. En algunos casos es interesante incorporar información referente, por ejemplo, a población, número de habitantes, etc. Podemos acceder a información de este tipo en la web del Instituto Nacional de Estadística (www.ine.es).

Existen muchos factores que contribuyen a la complejidad de cargar la información en un *datawarehouse*. Uno de los principales es el número de fuentes de información distintas de las que cargamos la información. Además, el número de fuentes de información varía de una organización a otra: en grandes corporaciones se habla de una media de 8 bases de datos, y en algunos casos puede llegar a 50.

Acceder a distintas bases de datos requiere distintas habilidades y el conocimiento de distintas sintaxis de SQL. Si el número de bases de datos a las que debemos acceder es elevado, puede provocar que tanto las definiciones como las codificaciones en los distintos entornos sean diferentes, lo que añadirá dificultad a nuestro proyecto; por ello, un aspecto clave será conocer el modelo de información transaccional y el significado de cada uno de sus elementos.

La definición de los distintos componentes de nuestro sistema de información no siempre es consistente a través de distintas aplicaciones, que no están integradas. Si las aplicaciones han sido desarrolladas normalmente, no están suficientemente documentadas para ser interpretadas correctamente. En la mayoría de los casos son aplicaciones que han sido modificadas a lo largo del tiempo por distintos programadores, y normalmente no se han actualizado.

La información que cargamos en un *datawarehouse* normalmente es estructurada, es decir, aquella que se puede almacenar en tablas: en la mayoría de los casos es información numérica. Cada vez más, la tecnología nos permite trabajar con información no estructurada, y se espera que este tipo de información sea cada vez más importante. Dentro de la información no estructurada tenemos: correos electrónicos, cartas, informes, videos, etc.

Una encuesta ha indicado que el 60% de los directores de Sistemas de Información y de los de Tecnología considera que la información semi-estructurada es crítica para mejorar las operaciones y para la creación de nuevas oportunidades de negocio.

En esta fase, el punto clave es identificar las fuentes más apropiadas de las cuales recuperaremos la información, deberemos analizar los formatos, la disponibilidad y la calidad de la información.

Tendremos que analizar si la información de la que disponemos es la que necesitamos para alimentar los modelos de negocio que hemos definido anteriormente. En este punto, muchas veces descubrimos que no disponemos de la información necesaria para completar el modelo de negocio que habíamos planteado, circunstancia que nos puede llevar a modificar nuestras aplicaciones transaccionales para conseguirla.

Un ejemplo podría ser el siguiente: en un concesionario de coches es muy importante saber el número de visitas que hemos recibido en la exposición y su relación con las ventas. Si hemos diseñado un modelo de negocio que nos relacione el número de visitas, el número de ofertas y el número de ventas, y no disponemos del número de visitas, tendremos que modificar los procesos y las aplicaciones para recoger esta información. Si las ventas disminuyen, se debe atribuir a que ha disminuido el número de clientes interesados en nuestros vehículos o bien a que nuestra efectividad ha disminuido.

Una vez decididas las fuentes de información debemos verificar la calidad de los datos.

10.3.2. – Calidad de los datos

La calidad de los datos en un *datawarehouse* es fundamental, como afirma Bill Inmon en su artículo (<http://www.b-eye-network.com/>), aparecido en “*Business Intelligence Network*” sobre Calidad de Datos:

“Las organizaciones actúan bajo la suposición de que la información de la que disponen es precisa y válida. Si la información no es válida, entonces no pueden responder de las decisiones basadas en ella.”

Consecuentemente, es necesario asegurar que la calidad de los datos es máxima. Si en el *datawarehouse* hay errores, éstos se propagarán a lo largo de toda la organización y son muy difíciles de localizar. Además, pueden ocasionar que se tomen decisiones erróneas que afecten a los resultados de la organización. Los costes derivados de que la calidad de los datos no sea la correcta pueden llegar a ser muy elevados.

Si los usuarios perciben que no tenemos suficiente calidad de datos rápidamente desprestigiarán el proyecto de *Business Intelligence*. En una de las previsiones de Gartner se afirma que:

“A lo largo de 2007, más del 50% de los proyectos de datawarehouse experimentarán una aprobación limitada, si no un pleno fracaso, ya que no habrán actuado proactivamente sobre la calidad de los datos.” (Con una probabilidad del 80%)

Los errores en los datos pueden provenir de los sistemas transaccionales de los que recuperamos los datos, del proceso ETL, o del propio *datawarehouse*.

Asumir que la calidad de los datos es buena puede ser un error fatal en los proyectos de *Business Intelligence*. Normalmente, cuando se construye un *datawarehouse* la mayoría de las organizaciones se focalizan en identificar los datos que necesitan analizar, los extraen y los cargan en el *datawarehouse*. Generalmente no se piensa en la calidad de los datos, permitiendo que los errores sean cargados al *datawarehouse*. Debería por tanto establecerse un control o conjunto de controles en el proyecto que localizara los errores en los datos y no permitiera la carga de los mismos. Las comprobaciones se deberán llevar a cabo, de forma manual o automatizada, teniendo en cuenta distintos niveles de detalle y variando los periodos de tiempo, comprobando que los datos cargados coinciden con los de las fuentes de datos origen; por ejemplo, comprobando que las ventas totales o el número de pedidos coinciden diariamente con la información cargada en el *datawarehouse*. De forma menos frecuente, comprobar que los agregados

de un periodo coinciden entre los dos entornos, el del *datawarehouse* y el de las fuentes originales de datos.

En algunos casos se detectan errores que se originan por fallos en los sistemas transaccionales, lo que debería provocar proyectos de mejora en los mismos. Muchos de estos casos se deben a que los usuarios pueden introducir datos sin ningún tipo de control. Siempre que se pueda, es recomendable que los usuarios elijan entre distintos valores, en lugar de introducirlos libremente ellos. No es una buena opción corregirlos en el proceso ETL y no modificar las aplicaciones origen. Esta alternativa es mucho más rápida inicialmente, pero mucho más costosa a largo plazo.

Los errores también se pueden producir, por ejemplo, en el proceso de ETL o al integrarlos en el *datawarehouse*.

En el gráfico mostrado (figura 10.2) a continuación se presenta el proceso en el que se señalan los puntos de control: en la carga, la auditoría y reconciliación, y por los usuarios de *Business Intelligence*. El proceso debe ser continuo para conseguir la mejora en la calidad de los datos. Este proceso nos puede ayudar a mejorar nuestros sistemas transaccionales, corregir errores en el *datawarehouse*, mejorar el proceso ETL o incluso mejorar los modelos de negocio por parte de los usuarios de Business Intelligence.

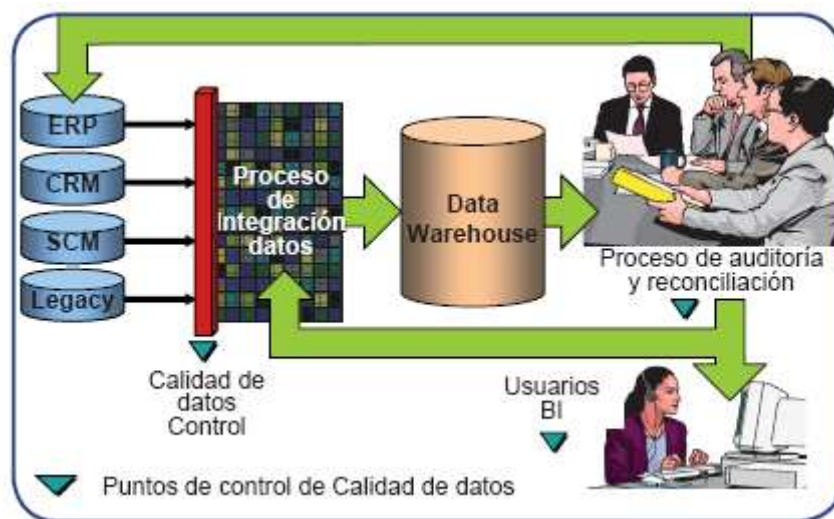


Figura 10.2.- Proceso de puntos de control. Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

La responsabilidad de la calidad de los datos no pertenece sólo a los departamentos de tecnología: Debe asumirse la parte correspondiente en cada uno de los propietarios de los procesos y de las aplicaciones que los soportan. Desde el proyecto debemos velar por la calidad de los datos, puesto que si la calidad no es la adecuada nunca podremos obtener los beneficios esperados del proyecto. Debemos entender que la problemática de la calidad de datos no es un problema de los departamentos de tecnología, sino un

problema estratégico al que debemos asignar objetivos, recursos y planificación.

No hay demasiadas organizaciones que tengan un plan de calidad de datos; en una encuesta de The datawarehouse Institute realizada en el año 2001, los resultados obtenidos fueron contundentes: El 48% de las organizaciones encuestadas no tenían un plan para gestionar o mejorar la calidad de los datos.

Los problemas de calidad de datos son un problema de negocio, no de los departamentos de tecnología; en base a ello, las recomendaciones que deberíamos seguir para mejorar la calidad de los datos son:

- Conocer los datos es la clave para el éxito en muchos negocios e iniciativas de tecnología:
 - Realizar una auditoria incluyendo una evaluación de la calidad.
 - Conocer dónde están los datos y su nivel de calidad.
 - Incluir la calidad de los datos en la estrategia de metadata.
- Establecer un programa formal de calidad de datos:
 - Construir el acuerdo para aplicarla en toda la gestión de las fuentes de datos.
 - Establecer acciones de calidad de datos en la gestión de la información de la organización.
- Desarrollar las habilidades necesarias y organizar un equipo, tanto a nivel de los usuarios de negocio como de los de tecnología.
- Definir las políticas y las métricas de la calidad de datos:
 - Definir los estándares de direcciones, como calcular el beneficio, los ingresos, etc.
 - Establecer y usar métricas para alcanzar la calidad de los datos.
- Implementar tecnologías de calidad de datos, reconociendo que tan sólo son una parte de la solución.

10.3.3. - Proceso de extracción, transformación y carga (ETL).

Siguiendo el modelo que hemos propuesto al inicio del presente capítulo vamos analizar el proceso de extracción, transformación y carga y las herramientas que nos facilitan este proceso y que nos permitirán alimentar un *datawarehouse*. El proceso trata de recuperar los datos de las fuentes de información y alimentar el *datawarehouse*.

El proceso de ETL consume entre el 60% y el 80% del tiempo de un proyecto de Business Intelligence, por lo que es un proceso clave en la vida de todo proyecto.

Esta parte del proceso de construcción del *datawarehouse* es costosa y consume una parte significativa de todo el proceso, por ello requiere recursos, estrategia, habilidades especializadas y tecnologías.

La extracción, transformación y carga (el proceso ETL) es necesario para acceder a los datos de las fuentes de información al *datawarehouse*. El proceso ETL se divide en 5 subprocesos:

1. *Extracción*: Este proceso recupera los datos físicamente de las distintas fuentes de información. En este momento disponemos de los datos en bruto.
2. *Limpieza*: Este proceso recupera los datos en bruto y comprueba su calidad, elimina los duplicados y, cuando es posible, corrige los valores erróneos y completa los valores vacíos, es decir se transforman los datos -siempre que sea posible- para reducir los errores de carga. En este momento disponemos de datos limpios y de alta calidad.
3. *Transformación*: Este proceso recupera los datos limpios y de alta calidad y los estructura y resume en los distintos modelos de análisis. El resultado de este proceso es la obtención de datos limpios, consistentes, resumidos y útiles.
4. *Integración*: Este proceso valida que los datos que cargamos en el *datawarehouse* son consistentes con las definiciones y formatos del *datawarehouse*; los integra en los distintos modelos de las distintas áreas de negocio que hemos definido en el *mismo*. Estos procesos pueden ser complejos.
5. *Actualización*: Este proceso es el que nos permite añadir los nuevos datos al *datawarehouse*.

Vamos ahora a describir con más detalle cada una de los distintos subprocesos.

Extracción

La extracción de los datos se puede realizar bien de forma manual o bien utilizando herramientas de ETL. De forma manual significa programar rutinas utilizando lenguajes de programación (por ejemplo: COBOL) que extraigan los datos de las fuentes de datos origen, aunque en otros casos se opta por las utilidades de replicar la base de datos que tienen los motores de

bases de datos. La alternativa más rentable es la que provee las herramientas especializadas de ETL, ya que han sido diseñadas para llevar a cabo esta función y nos permiten visualizar el proceso y detectar los errores durante el proceso o durante la carga. Cada vez más los motores de bases de datos tienen mejores funcionalidades de ETL.

En el cuadro siguiente (Figura 10.3) mostramos los principales problemas que nos podemos encontrar al acceder a los datos para extraerlos: básicamente se refieren a que provienen de distintas fuentes, BBDD, plataformas tecnológicas, protocolos de comunicaciones, juegos de caracteres, y tipos de datos.



Figura 10.3.- Problemática del acceso a datos. Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

El principal objetivo de la extracción es extraer tan sólo aquellos datos de los sistemas transaccionales que son necesarios y prepararlos para el resto de los subprocesos de ETL.

Para ello se deben determinar las mejores fuentes de información, las de mejor calidad. Con tal finalidad, deberemos analizar las fuentes disponibles y escoger aquellas que sean mejores.

Normalmente hablamos de almacenes de datos intermedios (*Data staging*) mientras que estamos en el proceso de limpieza de los datos. Se trata de un paso intermedio entre la extracción y las etapas posteriores: Acumulamos datos de distintas fuentes, en un momento determinado todos estos datos se cargarán en el *datawarehouse*. Los usuarios finales nunca acceden a este entorno.

Limpieza

Los sistemas transaccionales contienen datos que no han sido depurados y que deben ser limpiados.

Las herramientas ETL tienen funcionalidades de limpieza de datos, aunque existen herramientas especializadas para ello. En proyectos de CRM, la limpieza de los datos es clave: los nombres y las direcciones de los clientes siempre necesitan ser limpiados, eliminar duplicados, etc. Si no llevamos a cabo este subproceso de forma exquisita, crearemos escépticos al mostrar los resultados si, por ejemplo, mostramos los mejores clientes de nuestra organización y aparecen duplicados; en tal caso, lo más habitual es que se cuestione la validez del modelo.

Pero ¿cuáles son las causas que provocan que los datos estén “sucios”?, veamos algunos ejemplos:

- Valores por defecto: En la caja no saben la referencia de un producto e introducen el código 999 y el precio a mano.
- Ausencia de valor.
- Campos que tienen distintas utilidades: Para algunos clientes ponemos una información y para otros, otra distinta.
- Valores crípticos.
- Valores contradictorios.
- Uso inapropiado de los campos, por ejemplo en las direcciones de los clientes.
- Vulneración de las reglas de negocio.
- Reutilización de claves primarias con valores que se habían utilizado en el pasado.
- Identificadores que no son únicos.
- Problemas de carga de antiguos sistemas o de integración entre sistemas.
- Selección del primer valor de una lista por defecto.

La limpieza de datos se divide en distintas etapas, que vamos a describir a continuación:

- Depurar los valores (*Parsing*): Este proceso localiza e identifica los elementos individuales de información en las fuentes de datos y los aísla en los ficheros destino. Por ejemplo: separar el nombre completo en nombre, primer apellido, segundo apellido, o la dirección en: calle, numero, piso, etcétera.

- Corregir (*Correcting*): Este proceso corrige los valores individuales de los atributos usando algoritmos de corrección y fuentes de datos externas. Por ejemplo: comprueba una dirección y el código postal correspondiente.
- Estandarizar (*Standardizing*): Este proceso aplica rutinas de conversión para transformar valores en formatos definidos (y consistentes) aplicando procedimientos de estandarización y definidos por las reglas del negocio. Por ejemplo: trato de Sr., Sra., etc. o sustituyendo los diminutivos de nombres por los nombres correspondientes.
- Relacionar (*Matching*): Este proceso busca y relaciona los valores de los registros, corrigiéndolos y estandarizándolos, basándose en reglas de negocio para eliminar duplicados. Por ejemplo: identificando nombres y direcciones similares.
- Consolidar (*Consolidating*): Este proceso analiza e identifica relaciones entre registros relacionados y los junta en una sola representación.

Transformación

La transformación de los datos se hace partiendo de los datos una vez “limpios”. Transformamos los datos de acuerdo con las reglas de negocio y los estándares que han sido establecidos. La transformación incluye: cambios de formato, sustitución de códigos, valores derivados y agregados.

Los agregados, como por ejemplo la suma de las ventas, normalmente se precálculan y se almacenan para conseguir mayores rendimientos cuando lanzamos las consultas que requieren el cálculo de totales al *datawarehouse*.

En este proceso también ajustamos el nivel de granularidad o detalle. Podemos tener detalle a nivel de líneas de factura en los datos extraídos, pero en el *datawarehouse* lo que almacenamos son las ventas semanales o mensuales.

Integración

La última etapa es la de integración en el *datawarehouse*: es el momento en el que cargamos los datos y debemos comprobar si, por ejemplo, los totales de ventas que hemos cargado coinciden con la información que residía en nuestro sistema transaccional, así como si los valores que tienen los registros cargados corresponden a los definidos en el *datawarehouse*. Es

fundamental comprobar que se ha desarrollado correctamente, ya que en caso contrario pueden llevar a decisiones erróneas a los usuarios.

Actualización

Este proceso determina la periodicidad con el que haremos nuevas cargas de datos al *datawarehouse*.

10.3.4. - Herramientas ETL

Debido a la criticidad del proceso de extracción, transformación y carga, las herramientas ETL son claves en los proyectos de Business Intelligence.

El mercado demanda herramientas ETL más completas y con más funcionalidades, que aceleren la extracción y carga de datos, que puedan acceder a diversos formatos y fuentes de datos, que soporten mayor complejidad y que se acerquen a cargas en tiempo real.

Las herramientas ETL deberían contar con:

- **Diseño gráfico:** Entorno que permite a los desarrolladores establecer la relación entre las fuentes de datos, las transformaciones, los procesos y las tareas para desarrollar la carga. Los diseños se deben almacenar en un repositorio Metadata.
- **Gestión del Metadata:** Proveer un repositorio donde definir, documentar y gestionar la información del proceso ETL y su ejecución. El Metadata debería ser accesible también desde otras aplicaciones.
- **Extracción:** Extracción de la información mediante conectores, como ODBC, SQL nativos de los distintos motores de bases de datos o ficheros planos. Los conectores deberían acceder al Metadata para determinar qué información extraer y cómo.
- **Transformación:** Deberían proveer de librerías de transformación que permitan a los desarrolladores transformar los datos origen en los destino con las nuevas estructuras y crear las tablas de agregación para mejorar el rendimiento.
- **Carga:** Utilizar adaptadores para poder insertar o modificar los datos en el *datawarehouse*.
- **Servicios de transporte:** Las herramientas ETL utilizan las redes y sus protocolos (por ejemplo: FTP, File Transport Protocol) para mover los datos entre las distintas fuentes y los sistemas destino.

- Administración y operación: Las herramientas ETL deben permitir a los administradores programar, ejecutar y monitorizar los trabajos de ETL, los resultados, gestionar los errores, recuperar los fallos y reconciliar los resultados con los sistemas originales.

10.3.5. - *Datawarehouse* o almacén de datos

Cuando queremos analizar un problema empresarial, normalmente la información que necesitamos proviene de distintos sistemas, pero nosotros la requerimos en un mismo entorno para facilitar su análisis. Normalmente, en los sistemas transaccionales no tenemos preparada para ser analizada: sólo la tenemos la información de las transacciones actuales, pero no la de los periodos anteriores o la de las previsiones. Si queremos estudiar la evolución de las ventas necesitamos saber:

- Ventas actuales.
- Ventas del/os periodo/s anterior/es.
- Presupuesto de ventas del ejercicio.

Si seguimos con la tendencia actual, incluso quizá sea preciso las ventas estimadas hasta el final del período.

Inicialmente, podemos sentir la tentación de recuperar esa información, introducirla en una hoja de cálculo y a partir de ahí comenzar el análisis. ¿Es este el camino correcto? Obviamente no, pero vamos a justificar las razones por las que no es la mejor opción:

- Al introducir la información proveniente de distintos sistemas podemos cometer errores.
- Debemos invertir una cantidad de tiempo considerable en la introducción de la información.
- Cada vez que queramos hacer el análisis de ventas deberemos repetir el proceso.
- Si alguien necesita más detalle de las ventas, probablemente no dispondremos del mismo y quizás respondamos que nos llevaría mucho tiempo introducir toda la información a nivel de detalle.
- Cuando hagamos las consultas para extraer la información del entorno transaccional, penalizaremos el rendimiento de las aplicaciones.

- Cuando se produzcan modificaciones en el sistema transaccional, deberemos actualizar nuestra hoja de cálculo.
- Cada uno de los usuarios de información querrá los informes con un diseño determinado: Pueden aparecer “entornos paralelos” de información.
- Si no hay un acuerdo común, es posible que tengamos distintas versiones de una misma realidad: La cifra de ventas de uno de los informes igual no coincide con la de otro.
- Los usuarios finales no siempre saben dónde reside la información que necesitan.

Podríamos aducir muchas más razones, pero será mejor que analicemos las causas que las provocan y veamos cómo solucionarlas.

La aparición de los *datawarehouse* o Almacenes de datos son la respuesta a las necesidades de los usuarios que necesitan información consistente, integrada, histórica y preparada para ser analizada para poder tomar decisiones.

Al recuperar la información de los distintos sistemas, tanto transaccionales como departamentales o externos, y almacenándolos en un entorno integrado de información diseñado por los usuarios, el *datawarehouse* nos permitirá analizar la información contextualmente y relacionada dentro de la organización.

Hay muchas definiciones de *datawarehouse*; una primera aproximación es la del Profesor Hugh J. Watson, que lo define en su esencia como:

“Un datawarehouse es una colección de información creada para soportar las aplicaciones de toma de decisiones”

Bill Inmon fue el que definió las características que debe cumplir un *datawarehouse*, que son: Orientado sobre un área, integrado, indexado al tiempo, es un conjunto no volátil de información que soporta la toma de decisiones.

Analicemos cada una de estas características detalladamente:

- “Orientado a un área” significa que cada parte del *datawarehouse* esta construida para resolver un problema de negocio, que ha sido definido por los tomadores de decisiones. Por ejemplo: Entender los hábitos de compra de nuestros clientes, analizar la calidad de nuestros productos, analizar la productividad de una línea de fabricación, etc. Para poder analizar un problema de negocio necesitamos información que proviene de distintos sistemas y la

organizamos entorno a áreas: ventas, clientes, elementos de transporte, etc. Provee a los tomadores de decisiones de una visión completa y concisa sobre una problemática de negocio, obviando toda aquella información que no necesitan para la toma de decisiones.

- “Integrado”: La información debe ser transformada en medidas comunes, códigos comunes y formatos comunes para que pueda ser útil. La integración permite a las organizaciones implementar la estandarización de sus definiciones, por ejemplo: La moneda en la que están expresados los importes es común.
- “Indexado en el tiempo” significa que se mantiene la información histórica y se almacena referida a determinadas unidades de tiempo, tales como horas, días, semanas, meses, trimestres o años. Ello nos permitirá analizar, por ejemplo, la evolución de las ventas en los periodos que queramos.
- “No volátil” significa que los usuarios no la mantienen, como lo harían en los entornos transaccionales. La información se almacena para la toma de decisiones. No se va actualizando continuamente, sino periódicamente, de forma preestablecida.

Ralph Kimbal define los objetivos que debería cumplir un *datawarehouse*:

- El *datawarehouse* da acceso a la información de la corporación o del área funcional. El alcance del *datawarehouse* puede ser bien un departamento o bien corporativo.
- La información del *datawarehouse* es consistente.
- La información en el *datawarehouse* puede ser separada y combinada para analizar cada una de las posibles medidas del negocio.
- El *datawarehouse* no es sólo información sino también las herramientas de consulta, análisis y presentación de la información.
- Es el lugar donde publicamos la información.
- La calidad de la información en el *datawarehouse* es el motor del *business reengineering*.

El Profesor Hugh J. Watson amplía la definición anterior y describe el concepto de *datawarehousing*, es decir, la acción de construir *datawarehouses* y utilizar su información:

“Datawarehousing es el proceso completo de extraer información, transformarla y cargarla en un datawarehouse y el acceso a esta información por los usuarios finales y las aplicaciones”

Este entorno de *datawarehousing* nos debería permitir acceder a información que ha sido estructurada para hacer consultas, y estas consultas deberían permitir a los usuarios percibir el valor de esa información.

Muchas veces el valor aparece en un proceso secuencial de consultas, análisis y más consultas. Normalmente, las primeras consultas devuelven una gran cantidad de información que será refinada posteriormente con nuevas consultas. No todos los caminos de análisis escogidos serán los adecuados, y algunos los abandonaremos volviendo a modificar la consulta inicial. Al analizar formularemos hipótesis, que serán confirmadas o desmentidas por la información residente en el *datawarehouse*.

Si el *datawarehouse* está construido adecuadamente proporciona un entorno de información que nos permitirá encontrar nuevo conocimiento y generar valor.

Como afirma Sharon Sibigtroth, Vice President of Equitable Assurance, en unas declaraciones aparecidas de un estudio de IDC:

“Descubres el valor real de un datawarehouse cuando alguien puede encontrar los detalles importantes en la información, y te dice algo que puede generar la diferencia.”

Los *datawarehouses* se representan habitualmente como una gran base de datos, pero pueden estar distribuidos en distintas bases de datos.

El trabajo de construir un *datawarehouse* corporativo puede generar inflexibilidades, o ser costoso y requerir plazos de tiempo que las organizaciones no están dispuestas a aceptar. En parte, estas razones originaron la aparición de los Data Mart. Los Data Mart están dirigidos a una comunidad de usuarios dentro de la organización, que puede estar formada por los miembros de un departamento, o por los usuarios de un determinado nivel organizativo, o por un grupo de trabajo multidisciplinar con objetivos comunes.

Los Data Mart almacenan información de un número limitado de áreas; por ejemplo, pueden ser de marketing y ventas o de producción. Normalmente se definen para responder a usos muy concretos.

Normalmente, los Data Mart son más pequeños que los *datawarehouses*.

Tienen menos cantidad de información, menos modelos de negocio y son utilizados por un número inferior de usuarios.

Los Data Mart pueden ser independientes o dependientes (Figura 10.4). Los primeros son alimentados directamente de los orígenes de información, mientras que los segundos se alimentan desde el *datawarehouse* corporativo. Los Data Mart independientes pueden

perpetuar el problema de los “silos de información” y en su evolución pueden llegar a generar inconsistencias con otros Data Mart.

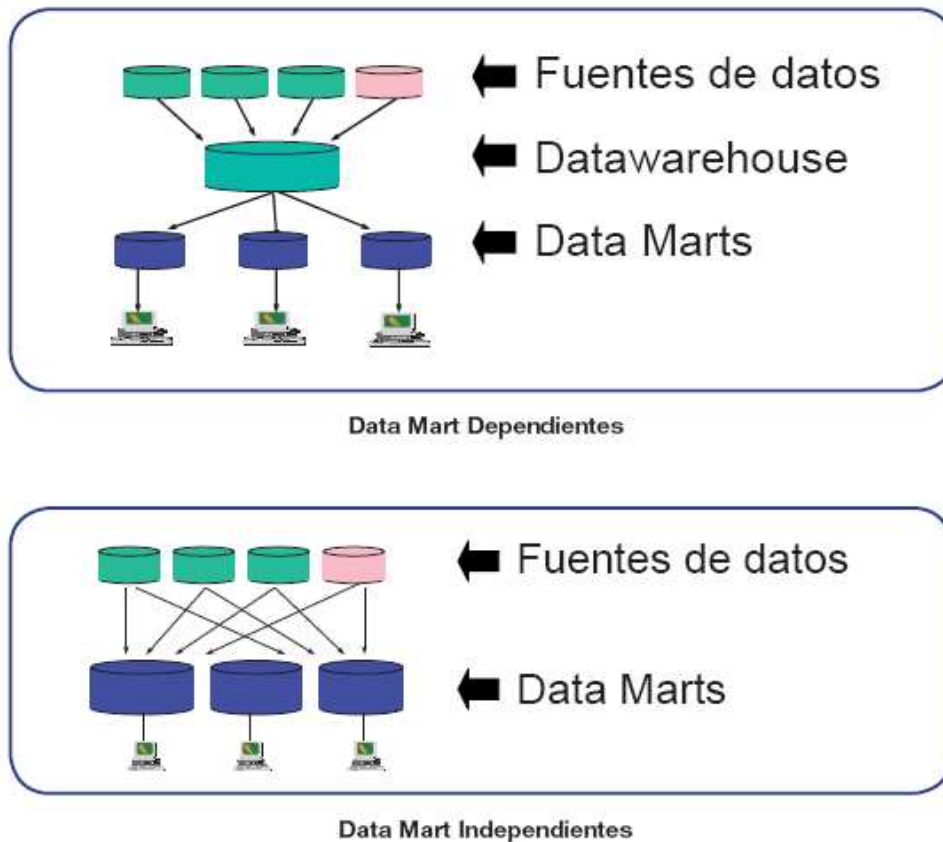


Figura 10.4.- Data Mart. Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

Para la construcción de un *datawarehouse* se han definido dos estrategias básicas:

- La defendida por W.H. Inmon, que propone definir un *datawarehouse* corporativo y a partir de él ir construyendo los modelos de análisis para los distintos niveles y departamentos de la organización; es decir, una estrategia de arriba abajo, desde la estrategia a lo más operativo.
- La defendida por R. Kimball es la de construir distintos Data Marts que cubran las distintas necesidades de la organización, sin la necesidad de construir un *datawarehouse*.

Como afirma el Profesor Hugh J. Watson, cuando se desarrollan correctamente las dos estrategias son válidas.

Con la estrategia de definir un *datawarehouse* corporativo, el *datawarehouse* es desarrollado en fases y cada una de las mismas debe ser diseñada para generar valor para el negocio. Se construye un *datawarehouse* corporativo, del que se cuelga un Data Mart dependiente con

una parte de la información del *datawarehouse*. En fases posteriores se van desarrollando Data Marts usando subconjuntos del *datawarehouse*. Igual que los proyectos complejos, es caro, necesita mucho tiempo y es propenso al fracaso. Cuando tenemos éxito conseguimos un *datawarehouse* integrado y escalable.

Si optamos por la estrategia más común, la de construir distintos Data Marts, el proyecto comienza con un Data Mart único al que posteriormente se irán añadiendo otros Data Marts que cubrirán otras áreas de negocio. Normalmente no requiere de grandes inversiones y es fácil de implementar, aunque conlleva algunos riesgos; de entre ellos, cabe destacar fundamentalmente dos: puede perpetuar la existencia del problema de “silos de información” y posponer la toma de decisiones que conciernen a la definición de criterios y modelos de negocio. Si seguimos esta estrategia debemos tener claro el plan de acción, es decir, qué áreas cubriremos y la integración de los distintos modelos. Esta estrategia se utiliza a veces como un paso previo al desarrollo de un *datawarehouse* corporativo.

Las dos aproximaciones abogan por construir una arquitectura robusta que se adapte fácilmente a los cambios de las necesidades de negocio y que nos proporcione una sola versión de la verdad.

En esta línea, en el artículo referenciado en el párrafo anterior se presentan dos alternativas más: la híbrida, que toma lo mejor de las dos aproximaciones, y la federada.

Un componente crítico de un *datawarehouse* es el Metadata. El Metadata es el repositorio central de información de la información. Nos da el significado de cada uno de los componentes y sus atributos que residen en el *datawarehouse* (o Data Mart). La información que contiene el Metadata es útil para los departamentos de tecnología y los propios usuarios. Puede incluir definiciones de negocio, descripciones detalladas de los tipos de datos, formatos y otras características.

El personal de los departamentos de Tecnología necesita saber los orígenes de la información: bases de datos de las que obtenemos los datos, qué transformaciones realizamos, criterios de filtros de información, nombre de las columnas y de las tablas, plazos de carga, utilización, etcétera.

Los usuarios necesitan saber las entidades y sus atributos, cómo han sido calculados, quiénes son los responsables de los datos, los informes disponibles, los flujos de distribución de la información, etcétera.

La construcción del Metadata supone que se defina el significado de cada una de las tablas y cada uno de los atributos que se cargan en el *datawarehouse*. Este es un punto complejo de todo proyecto, ya que obliga a que se definan los conceptos de negocio y se homogeneicen entre los distintos departamentos, filiales, etc. Obliga a que todos los componentes de la

organización hablen utilizando la misma terminología y con el mismo significado, lo cual no siempre es sencillo. Cuando alguien hable de “margen bruto” o “margen de contribución” deberá estar absolutamente definido para la organización. Evidentemente, organizaciones distintas tendrán normalmente definiciones distintas.

Existe un componente tecnológico, los *Operational Data Store* (ODS) que a veces se confunden con los *datawarehouses*. Los ODS son una extensión de la tecnología de los *datawarehouses*. Los ODS consolidan datos de múltiples fuentes provenientes de distintos sistemas de información no integrados y facilitan un acceso *online* integrado sobre esa información. Su objetivo es proporcionar información integrada, con el fin de facilitar la toma de decisiones en entornos operacionales. Algunas veces se utilizan para evitar integraciones o implementaciones de soluciones ERP. La información que reside en los ODS es volátil y normalmente tiene, como máximo, una antigüedad de dos o tres meses. La principal diferencia con los *datawarehouses* es que los datos de los ODS son volátiles y se actualizan en tiempo real. Los ODS habitualmente se convierten en una fuente de datos para el *datawarehouse*.

Las razones de negocio para construir un ODS son:

- Proveer informes integrados a través de distintos procesos, múltiples aplicaciones o módulos ERP.
- Agrupar los datos de dimensiones de negocio: clientes, productos, etc.
- Facilitar la integración de datos (*Data hub*)
- Conseguir una mayor actualización de la información que en el *datawarehouse*.

Preparar los datos para cargar un *datawarehouse* es complejo. Como ya hemos comentado anteriormente, requiere recursos, una estrategia, conocimientos específicos y el uso de tecnologías para llevarlo a cabo. Datos provenientes de distintas fuentes deben ser integrados en un modelo de negocio. Los sistemas transaccionales se actualizan normalmente en tiempo real (en algunos casos todavía pueden actualizarse en procesos *batch*, por ejemplo por la noche, pero esta alternativa cada vez es menos habitual). Para la actualización de los *datawarehouses* los plazos suelen ser mensuales, semanales o diarios.

Dado que los son diseñados usualmente para soportar la toma de decisiones estratégicas, los datos deben ser consistentes y no cambiar continuamente. El análisis estratégico normalmente necesita datos históricos para poder identificar tendencias y patrones. Si los datos cambian continuamente, es muy difícil detectar tendencias y patrones.

No es muy habitual que los datos se tengan que cargar en el *datawarehouse* continuamente, lo que se define en la literatura como “Real Time”. Debemos decidir los plazos en función de las necesidades de los usuarios de negocio, teniendo en cuenta los comentarios anteriores. Colin White en su artículo define el concepto de “Right Time” como aquel que implica que diferentes situaciones de negocio y eventos requieren diferentes respuestas o tiempos para actuar. En algunos casos necesitaremos respuestas en tiempo real, mientras que en otros un retraso de minutos u horas es aceptable.

Los factores que deberíamos tener en cuenta cuando estamos evaluando una alternativa tecnológica para la construcción de un *datawarehouse* son:

- **Tamaño del *datawarehouse*:** Es el volumen de datos que contiene el *datawarehouse*.
- **Complejidad de los esquemas de datos:** Si el modelo de datos es complejo, puede dificultar la optimización y el rendimiento de las consultas.
- **Número de usuarios concurrente:** Éste es un factor determinante. Si distintos usuarios pueden lanzar consultas concurrentes (a la vez), el *datawarehouse* debe gestionar sus recursos para poder dar respuesta a las distintas consultas.
- **Complejidad de las consultas:** Si las consultas necesitan acceder a un número elevado de tablas y los cálculos a realizar son complejos, podemos poner en dificultades al motor de la base de datos del *datawarehouse*.

En el presente apartado hemos introducido distintos componentes tecnológicos de las soluciones de *Business Intelligence*. En el siguiente esquema (Figura 10.5) los presentamos conjuntamente para facilitar su comprensión:

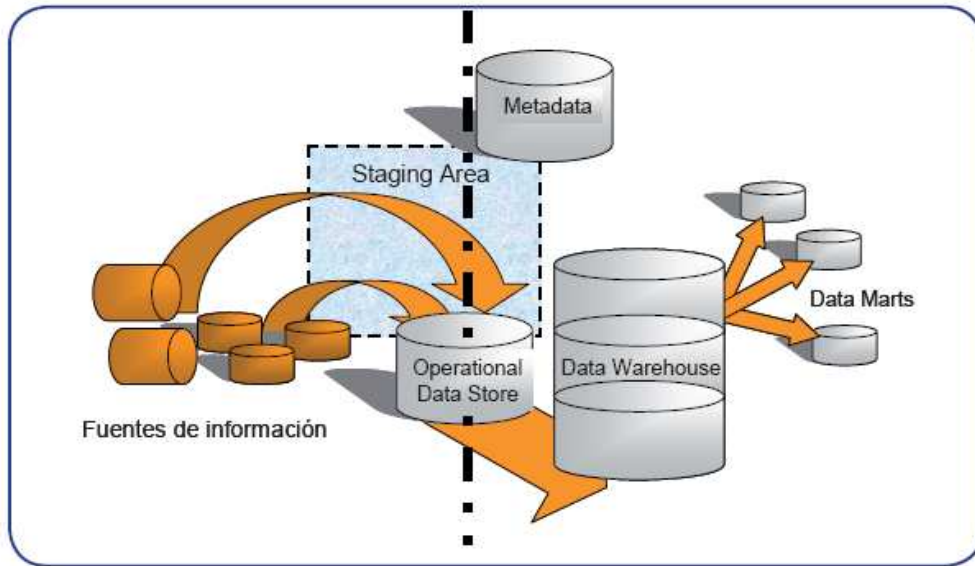


Figura 10.5.- Soluciones de Business Intelligence. Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

10.3.6. - Gestión del *datawarehouse*

Los usuarios de negocio necesitan tomar decisiones basadas en la información de los *datawarehouses*, por lo que debemos asegurar:

- Alta disponibilidad.
- Rendimiento.
- Copias de seguridad y recuperación.
- Recuperación física en caliente.

10.3.7. - Herramientas de *Business Intelligence*

Siguiendo el modelo que hemos propuesto al inicio, vamos a analizar las tecnologías que nos permitirán tratar y visualizar la información que reside en un *datawarehouse*. Tratamos conjuntamente estos dos componentes, ya que se da así en la mayoría de productos comerciales.

Existen distintas tecnologías que nos permiten analizar la información que reside en un *datawarehouse*, pero la más extendida es el OLAP.

Los usuarios necesitan analizar información a distintos niveles de agregación y sobre múltiples dimensiones: Por ejemplo, ventas de productos por zona de ventas, por tiempo, por clientes o tipo de cliente y por región

geográfica. Los usuarios pueden hacer este análisis al máximo nivel de agregación o al máximo nivel de detalle. OLAP provee de estas funcionalidades y algunas más, con la flexibilidad necesaria para descubrir las relaciones y las tendencias que otras herramientas menos flexibles no pueden aportar.

A estos tipos de análisis les llamamos multidimensionales, porque nos facilitan el análisis de un hecho desde distintas perspectivas o dimensiones. Esta es la forma natural que se aplica para analizar la información por parte de los tomadores de decisiones, ya que los modelos de negocio normalmente son multidimensionales.

La visualización de la información es independiente respecto de cómo se haya almacenado.

El OLAP Council formuló las 12 reglas de Codd en lo que ellos llamaban el concepto FASMI que los productos OLAP deben cumplir. El concepto FASMI proviene de las siglas de las iniciales en inglés:

- FAST (Rápido): Debe ser rápido, necesitamos lanzar consultas y ver los resultados inmediatamente.
- ANALYSIS (Análisis): Debe soportar la lógica de negocio y análisis estadísticos que sean necesarios para los usuarios.
- SHARED (Compartido): Tiene que manejar múltiples actualizaciones de forma segura y rápida.
- MULTIDIMENSIONAL (Multidimensional): Tiene que proveer de una visión conceptual de la información a través de distintas dimensiones.
- INFORMATION (Información): Debe poder manejar toda la información relevante y la información derivada.

La representación gráfica del OLAP son los cubos.

Existen distintos tipos de herramientas OLAP. La diferencia entre ellas, básicamente, depende de cómo acceden a los datos:

- ROLAP: *Relational* OLAP
 - Las capacidades OLAP acceden directamente a la base de datos relacional. Se accede por tanto a una base de datos relacional (RDBMS). Accede habitualmente sobre un modelo “estrella”. La principal ventaja es que no tiene limitaciones en cuanto al tamaño, pero es más lento que el MOLAP, aunque algunos productos comerciales nos permiten cargar cubos virtuales para acelerar los tiempos de acceso.

- MOLAP: *Multidimensional OLAP*
 - La implementación OLAP accede directamente sobre una base de datos multidimensional (MDDDB Multi Dimensional Data Base). La ventaja principal de esta alternativa es que es muy rápida en los tiempos de respuesta y la principal desventaja es que, si queremos cambiar las dimensiones, debemos cargar de nuevo el cubo.
- HOLAP: *Hybrid OLAP*
 - Accede a los datos de alto nivel en una base de datos multidimensional y a los atómicos directamente sobre la base de datos relacional. En esencia utiliza las ventajas del ROLAP y del MOLAP.

Las formas de acceso de las herramientas OLAP pueden ser:

- Cliente/Servidor, lo que significa tener las instalaciones locales en los ordenadores de los usuarios.
- Acceso web: cliente, cliente ligero, o sólo con el navegador. En este tipo de acceso el navegador comunica con un servidor web, el cual habla con la aplicación del servidor, que es la que conecta con el *datawarehouse*. En el caso de acceder con el navegador sin ningún tipo de cliente o con cliente ligero (por ejemplo JAVA), normalmente se descargan pequeñas aplicaciones para aumentar la funcionalidad.

Algunos autores también hablan del Virtual o Desktop OLAP (DOLAP). En este caso creamos un cubo con las dimensiones que le interesan al usuario, lo cargamos en memoria en su ordenador, trabaja y, cuando acaba, lo eliminamos de la memoria. La ventaja es que el usuario sólo recibe los hechos y las dimensiones en los que está interesado y los analiza en forma local.

Algunas de las herramientas del mercado permiten programar actividades, como por ejemplo ejecutar consultas, publicar en web, lanzar alertas a través de la red, mediante correo electrónico o sobre agendas personales (PDA8).

Una alternativa al OLAP son las herramientas que utilizan consultas de lógica asociativa: “Cuando se carga la información, se comprime y se normaliza al máximo para que no haya información redundante. Cada valor único para todos los datos se almacena una sola vez y se referencia a través de punteros. Por ejemplo, si el primer registro de una fuente de datos incluye el campo “coche rojo” y la segunda incluye el valor “coche negro” sólo se almacena “coche” una sola vez. En lugar de almacenar dos veces “coche”, un contador asociado a un puntero referencia el incremento de ese valor.”

Cuando realizamos una consulta, se accede directamente de los datos al visor, se accede al máximo detalle de la información sin dimensiones y jerarquías predefinidas y sin restricciones en cuanto al volumen de información.

El modelo de almacenamiento interno proporciona una visión “vertical” (basada en columnas) de los datos así como una visión “horizontal ampliada” (basado en filas) que va más allá de la tecnología de bases de datos relacionales. Cada columna almacena cada valor diferente de forma separada con la frecuencia y usos de cada valor. Las consultas son altamente eficientes por el nuevo modelo de almacenamiento y el conjunto de operaciones utilizados para resolver las consultas.

Se indexan automáticamente el 100% de los datos y se eliminan automáticamente los datos redundantes y los valores nulos, lo que significa un menor uso de espacio de disco y menores tiempos de escritura y lectura. Las principales herramientas de *Business Intelligence* son:

- Generadores de informes: Utilizadas por desarrolladores profesionales para crear informes estándar para grupos, departamentos o la organización.
- Herramientas de usuario final de consultas e informes: Empleadas por usuarios finales para crear informes para ellos mismos o para otros; no requieren programación.
- Herramientas OLAP: Permiten a los usuarios finales tratar la información de forma multidimensional para explorarla desde distintas perspectivas y periodos de tiempo.
- Herramientas de *Dashboard* y *Scorecard* (*Cuadros de mando*): Permiten a los usuarios finales ver información crítica para el rendimiento con un simple vistazo utilizando iconos gráficos y con la posibilidad de ver más detalle para analizar información detallada e informes, si lo desean.
- Herramientas de planificación, modelización y consolidación: Permite a los analistas y a los usuarios finales crear planes de negocio y simulaciones con la información de *Business Intelligence*. Pueden ser para elaborar la planificación, los presupuestos, las previsiones. Estas herramientas proveen a los *dashboards* y los *scorecards* con los objetivos y los umbrales de las métricas.
- Herramientas *datamining*: Permiten a estadísticos o analistas de negocio crear modelos estadísticos de las actividades de los negocios. *Datamining* es el proceso para descubrir e interpretar patrones desconocidos en la información mediante los cuales resolver problemas de negocio. Los usos más habituales del *datamining* son:

segmentación, venta cruzada, sendas de consumo, clasificación, previsiones, optimizaciones, etc.

10.3.8. – Visualización

La visualización de la información del *datawarehouse* se puede hacer utilizando hojas de cálculo, herramientas específicas o desde un simple navegador. Depende en cada caso de las características del producto seleccionado. Por ejemplo:

- Analysis Services de Microsoft SQL Server 2000:
- La herramienta de *Business Intelligence* de QlikView
- Microstrategy

Existen además las herramientas denominadas Group Decision Support Systems, que están pensadas para aquellos casos en las que se trata de un grupo de usuarios el que debe acceder a la información y tomar decisiones conjuntas.

10.3.9. – ¿Quiénes son los usuarios de las herramientas de *Business Intelligence*?

Se pueden dividir los usuarios de Business Intelligence en dos grandes grupos (Figura 10.6), para ello vamos a utilizar la clasificación y las definiciones que proponen W.W. Eckerson y C. Howson:

- Los productores de información: Normalmente se trata del 20% de los usuarios y utilizan herramientas *desktop* para crear informes o modelos. Normalmente se trata de estadísticos que utilizan herramientas *datamining* o autores de informes que utilizan herramientas de diseño o de programación para crear informes específicos. Habitualmente los autores de informes son: técnicos de sistemas de información o usuarios de negocio avanzados que son capaces de entender la información y la informática. Los usuarios avanzados pueden crear o utilizar informes, por lo que en el gráfico están a medio camino entre los productores y los consumidores de información. Usualmente utilizan hojas de cálculo, herramientas de consulta y de informes para acceder y analizar la información.
- Los consumidores de información: La mayoría de los consumidores de información son usuarios no habituales que regularmente consultan informes para la toma de decisiones, pero no acceden a los números o hacen análisis detallados diariamente. Los usuarios no habituales son directivos, gestores, responsables, colaboradores y usuarios externos.

Este numeroso grupo está bien servido con cuadros de mando con análisis guiados, informes interactivos (por ejemplo: OLAP, informes parametrizados, vinculados,...) e informes de gestión estandarizados. La mayoría de estas herramientas proveen ahora acceso vía web para promover el acceso desde cualquier lugar y facilitar el uso y minimizar los costes de administración y mantenimiento.

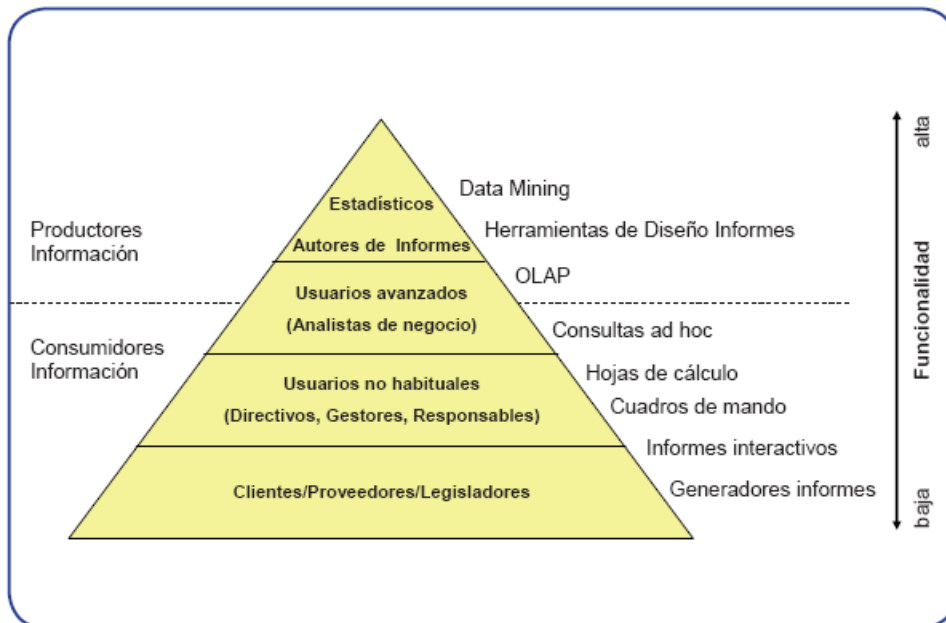


Figura 10.6.- Usuarios de Business Intelligence. Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

Es importante señalar que se diferencian los roles, no los individuos. La mayoría de los usuarios no se adecuan absolutamente a una de las dos categorías. Un ejecutivo podría querer ver información financiera utilizando un informe estático publicado semanalmente, pero ver información operacional en tiempo real utilizando un cuadro de mando. De esta manera, es crítico conocer la información a la que acceden los usuarios y las funcionalidades que necesitan en función de los roles.

Consecuentemente, deberemos tener en cuenta los distintos tipos de usuarios que tenemos en nuestra organización al elegir las herramientas de *Business Intelligence*.

10.4. – Necesidad y fases de planificación de un proyecto BI

La principal causa de fracaso en los proyectos de Sistemas de Información es la falta del uso de una metodología en su desarrollo. El número de tareas a realizar es muy elevado, y consecuentemente es imposible gestionarlas sin disponer de una metodología.

Las actividades de la planificación de un proyecto comprenden distintas etapas: Inicio, Planificación, Ejecución y Finalización.

El **inicio** del proyecto es el origen del proyecto y su razón de ser. En esta primera etapa deberemos decidir si seguimos adelante con el mismo o no.

La **planificación** del proyecto comprende: La organización, la dirección y el control de unos recursos de una empresa/departamento/unidad, para alcanzar un objetivo en un plazo, coste y calidad preestablecidos. No debemos olvidar que el proyecto se desarrolla para algunos usuarios de nuestra organización, a los que además se les pide que colaboren con él. Debemos conseguir que las relaciones con estos usuarios sean excelentes, ya que ello nos asegurará su participación y mejorará su evaluación del proyecto, por lo que será necesario comunicarles tanto el inicio del proyecto como su posterior evolución.

Objetivos del proyecto

El primer paso es definir los objetivos del proyecto, los cuales deberían ser cuantificados y estar alineados con la estrategia de la organización. El hecho que los cuantifiquemos nos permitirá confirmar el éxito o el fracaso del mismo. Deberíamos ser capaces de mostrar cómo apoyamos la estrategia de la organización con el proyecto. En ocasiones, en los proyectos de sistemas de información se confunde su éxito o su fracaso con el cumplimiento del plazo y de los recursos del proyecto, olvidando que la razón de ser de todo proyecto es aportar valor a la organización.

Si un proyecto no está alineado con los objetivos de la organización es muy difícil que se apruebe, y además se complica enormemente el poderlo llevar a cabo: Cuando aparezcan los primeros contratiempos, ¿quién lo defenderá? En tal caso, deberíamos preguntarnos si tiene sentido.

Cuando en lugar de tener un único proyecto tenemos varios hablamos de un portafolio de proyectos. Debemos gestionar el portafolio de proyectos de manera que éstos maximicen su aportación a la consecución de los objetivos de la organización. Una buena práctica para formalizar la gestión del portafolio de proyectos es establecer un procedimiento para: Identificar los nuevos, seleccionarlos, priorizarlos y asignar los recursos para llevarlos a cabo.

Los criterios de aprobación de los proyectos normalmente se basan en su alineamiento estratégico, los beneficios que aportan, el nivel de riesgo y los plazos.

Alcance del proyecto.

Debemos definir el alcance del proyecto. En los proyectos generales de sistemas lo definimos por las áreas de la organización que se ven involucradas, los procesos que soportarán el nuevo sistema y las prestaciones. En el caso de los proyectos de *Business Intelligence*, el alcance viene determinado por los modelos de negocio que queramos soportar y por los datos necesarios para soportarlos. También deberemos definir en este punto las funcionalidades que incorporará el sistema.

Los factores críticos de éxito para definir el alcance son: Definir correctamente los requerimientos e identificar qué está dentro y qué está fuera del proyecto. Estos dos componentes son fundamentales para poder estimar correctamente los plazos y los recursos que necesitaremos.

En caso de que se produzcan cambios de requerimientos que afecten al alcance deberemos gestionar los cambios, lo que significa: Identificarlos, analizarlos, valorarlos, tomar la decisión y comunicarla.

Riesgos

Al evaluar la posibilidad de llevar a cabo un proyecto debemos analizar cuidadosamente los riesgos asociados al proyecto, las probabilidades que se de, y cuáles son las señales que nos permitirán detectar los riesgos. En caso de que las probabilidades sean elevadas, deberemos crear un plan de contingencia, que incluirá las acciones concretas a realizar en caso de que se produzca el riesgo.

Normalmente los riesgos de los proyectos están relacionados con tres áreas:

1. Tamaño del proyecto: Si el proyecto es muy grande, el riesgo aumenta. Para reducir el tamaño deberemos fraccionar el proyecto en varios más pequeños.
2. Grado de estructuración. Se refiere a si sabemos exactamente lo que queremos del proyecto: Si no está claramente definido, es muy difícil que sea apoyado desde la Dirección o por los propios usuarios. La mejor alternativa para reducir este riesgo es construir el Business Case (caso de negocio del proyecto).
3. Conocimiento de la tecnología: Si no tenemos conocimiento de la tecnología que vamos a utilizar el riesgo es elevado. Para mitigarlo, deberemos formar a nuestro equipo o subcontratar parte de él para conseguir los conocimientos necesarios. En caso de que lo subcontratemos, deberíamos ser capaces de que se produzca la

transferencia de conocimientos tecnológicos entre la empresa subcontratada y nuestro equipo.

Deberemos llevar un seguimiento detallado de los riesgos, que se deberán identificar y analizar, lo cual implica revisarlos periódicamente. Normalmente el seguimiento de estos riesgos es semanal.

Limitaciones

En todos los proyectos tenemos limitaciones; la fundamental es el nivel de calidad que le pedimos al proyecto, que obviamente depende de: el alcance, el tiempo, los recursos y el presupuesto que le asignemos. Normalmente niveles muy elevados de calidad exigen el uso de más tiempo, más recursos y más presupuesto. Evidentemente no es posible disponer de recursos ilimitados, por lo que deberemos definir previamente cuáles son los estándares de calidad para el proyecto, fijándolos por anticipado, y llevar a cabo revisiones durante el proyecto y, si fuera necesario, auditorías.

Supuestos

Los proyectos no están aislados de la idiosincrasia de la organización: necesitamos la colaboración de personas de la organización, que tienen que facilitar o bien información o desarrollar tareas para el proyecto. Su colaboración es crucial y, debe producirse en un momento concreto del tiempo para que el proyecto no se retrase. Los retrasos también se pueden originar, por ejemplo, porque estamos pendientes de la entrega del *hardware* por parte de los proveedores: Si está previsto para una fecha y esta fecha no se cumple, evidentemente nos afectará a la duración total del proyecto.

Planificación de las actividades del proyecto.

En este momento el gestor del proyecto ya puede estimar una planificación detallada de los recursos (equipo), plazo y coste del proyecto. También deberá definir cuáles serán los procedimientos de seguimiento.

Podríamos resumir la planificación utilizando el siguiente esquema (Figura 10.7):

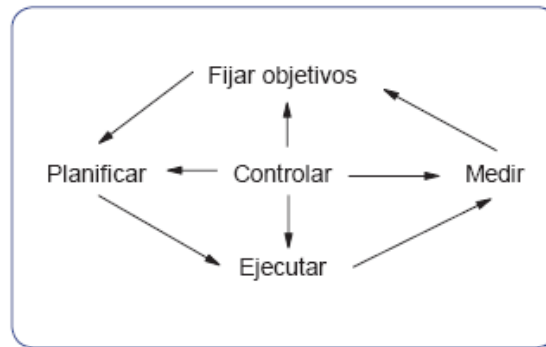


Figura 10.7.- Esquema de planificación de las actividades. Fuente: Fuente: Libro de Business Intelligence: Competir por la información. Josep Lluís Cano.

Debemos fijar los objetivos, planificar, medir, controlar y ejecutar las acciones correctivas que sean necesarias.

La planificación de un proyecto es una estimación que deberá ser revisada y controlada continuamente, y ajustada en los casos en que sea necesario.

Deberemos definir:

- Qué hay que hacer, cuáles son las actividades. Las actividades se pueden dividir en tareas.
- Cómo debe hacerse, es decir, definir la secuencia en la que deben realizarse las actividades. Normalmente, en los proyectos hay algunas actividades que deben preceder a otras, o algunas que no pueden iniciarse sin la finalización de las anteriores.
- Cuándo y por quién debe hacerse, es decir, definir una planificación detallada de la asignación de tareas para guiar al equipo del proyecto, lo que además nos dará un plazo total del proyecto. Para estimarlo nos podemos basar en experiencias pasadas o en la intuición. En la secuencia de las actividades deberemos analizar cuáles de ellas forman el camino crítico, es decir, aquel en que, si se retrasa una de las actividades, se alarga la duración del proyecto, y las actividades que disponen de holgura y que, aunque se retrasen dentro de unos límites, no alargan la duración total del proyecto.
- Cómo controlar la evolución del proyecto: Se definen hitos durante el proyecto que nos permiten evaluar los posibles retrasos en el mismo.

Con toda la información sobre las actividades, tareas, precedencias, asignación de actividades y tareas, calendario, etc. debemos construir un diagrama de Gantt.

La selección del equipo del proyecto es un factor crítico de éxito en la consecución del mismo. Si es posible, el equipo debe ser experimentado y estar equilibrado en relación al tiempo. Normalmente, si aumentamos el número de personas, disminuye el tiempo, pero no lo podemos llevar al extremo (muchas personas en muy poco tiempo), ya que el proyecto se hace inmanejable.

Durante la fase de ejecución para poder realizar el seguimiento del proyecto deberemos anotar las dedicaciones de las personas al mismo, hacer un seguimiento semanal, identificar las desviaciones y poner especial atención a los hitos del proyecto. El seguimiento incluye las desviaciones en plazo y costes.

En la última etapa del proyecto, la **finalización**, deberemos evaluar si hemos cumplido los objetivos dentro del plazo estimado y, utilizando los recursos humanos y los costes esperados, analizando cuáles han sido las desviaciones y las razones que las han originado, aprender para próximos proyectos. Esta es la finalización formal del proyecto, pero los sistemas siempre están vivos: Se pasa, por tanto, a la fase de mantenimiento. Las principales causas del fracaso en los proyectos pueden ser originadas por:

- Por incumplimiento de los resultados: Generará insatisfacción de los usuarios-clientes, elevados costes de mantenimiento, mala imagen de los participantes en el proyecto, pérdida de competitividad, etc.
- Por una deficiente gestión del proyecto: Por incumplimiento del plazo, por incumplimiento del presupuesto.

En todo proyecto los elementos clave de gestión son:

- Los objetivos del proyecto.
- El Jefe del proyecto.
- El equipo de trabajo.
- El soporte estratégico de la Dirección al proyecto.
- La asignación de recursos.
- Los canales de comunicación.
- Los mecanismos de control.

Hasta este punto los hemos abordado todos detalladamente, excepto el Jefe o el Responsable del proyecto. La dirección del proyecto debe recaer sobre una sola persona con autoridad real sobre el mismo, para hacerlo avanzar decididamente y orientado al objetivo final. El Jefe del proyecto debe ser un líder.

Las funciones que debe llevar a cabo el jefe del proyecto son:

- Concretar OBJETIVOS.

- Organizar y Planificar el PROYECTO.
- Controlar RESULTADOS
- Cumplir PLAZOS y PRESUPUESTO.
- Resolver INCIDENCIAS.
- Coordinar RELACIONES.
- Dirigir el EQUIPO de trabajo.

Las actividades y tareas que debemos plantearnos en todo proyecto de *Business Intelligence* son:

1. Planificación del proyecto:

- 1.1. Definir el proyecto.
- 1.2. Definir la planificación y la gestión del proyecto.
- 1.3. Establecer la finalización del proyecto.

2. Arquitectura tecnológica:

- 2.1. Revisar los requerimientos de negocio (usuarios, tiempos).
- 2.2. Definir la arquitectura tecnológica (*hardware*).
- 2.3. Definir las recomendaciones de configuración.
- 2.4. Estimar requerimientos de escalabilidad.
- 2.5. Implementar el *hardware* y el *software*.

3. Diseño:

- 3.1. Desarrollar los modelos de datos.
- 3.2. Analizar las fuentes de datos.
- 3.3. Diseñar la base de datos.
- 3.4. Diseñar el análisis de los usuarios finales.

4. Construcción:

- 4.1. Revisar el alcance y la planificación.
- 4.2. Implementar la base de datos.
- 4.3. Diseñar y desarrollar la integración de datos.
- 4.4. Cargar y validar la base de datos.
- 4.5. Construir el análisis de los usuarios finales.
- 4.6. Probar el sistema.
- 4.7. Ajustar el rendimiento.

5. Despliegue:

- 5.1. Entregar la documentación del proyecto.
- 5.2. Formar a los usuarios.
- 5.3. Entregar la aplicación.
- 5.4. Mantener el *datawarehouse*.

6. Operación:

- 6.1. Definir los procedimientos de soporte.
- 6.2. Monitorizar el rendimiento.
- 6.3. Mantener y mejorar la aplicación.

Deberemos, en cada caso, adaptar el listado anterior en función del tamaño, disponibilidad de recursos y dificultad del proyecto.

10.5. – Elementos clave para el éxito o fracaso de un proyecto BI

Una última recomendación para que los proyectos tengan éxito son los 10 errores¹⁰⁴ que deben evitar los *Project Managers* de los proyectos de *datawarehouse*:

1. Fallar en el uso de una metodología.
2. Definir una estructura organizativa del equipo inefectiva.
3. Fallar en lo involucrados que estén los usuarios de negocio.
4. No entregar evoluciones de la solución a los usuarios de negocio.
5. No tener una buena definición del proyecto.
6. Falta de una correcta estimación de las necesidades del proyecto.
7. Realizar pruebas inadecuadas.
8. Subestimar la limpieza de los datos.
9. Ignorar los Metadatos.
10. Ser un esclavo de las herramientas de gestión de proyectos.

10.6. – Usos de Business Intelligence y nuevas tendencias

Para analizar los actuales usos de *Business Intelligence* tenemos que fijarnos tanto en cuál ha sido la evolución de las herramientas como en cuál ha sido la evolución y las tendencias en el uso por parte de las organizaciones, así como los movimientos que se han producido en el mercado.

Evolución de las herramientas

Históricamente, los distintos usos que soportan las herramientas han sido:

- *Reporting*: Elaboración de informes.
- *Análisis*: Herramientas de consultas *ad hoc* y OLAP.
- *Planificación y modelización*.
- *Monitoring*: *Dashboards* y *Scorecards*.

- Análisis avanzado: *Datamining*, *Text Mining* y Visualización avanzada.

Cada vez más, las distintas herramientas soportan un mayor número de funcionalidades, son más maduras, escalables y más fáciles de usar, desde la carga de datos (procesos ETL) hasta el análisis. También se han producido cambios tecnológicos significativos referentes a acceso vía *web*, tan sólo con un navegador o con un cliente ligero o corriendo sobre procesadores de 32 bits a 64 bits, herramientas de *Text Mining*, localización, alertas, compatibilidad XML, etc.

Un factor determinante en la evolución de las herramientas de *Business Intelligence* será la adopción de arquitecturas SOA (*Service Oriented Architecture*), las mejoras en la visualización de la información y sin duda el aumento del uso de las herramientas de *Open Source*¹²³ o de código abierto.

Evolución y nuevas tendencias del uso empresarial

Cuando hablamos de cualquier Tecnología de la Información de la Comunicación, los más escépticos siempre muestran la duda de si a partir de ella podemos obtener los rendimientos esperados para las organizaciones que las utilizan. El precursor de esta línea de pensamiento fue Robert Solow, Premio Nobel de Economía de 1987, que expresó su opinión en el *New York Times* con la siguiente afirmación: “Veo ordenadores en todas partes, excepto en las estadísticas de productividad”, cuestionando los aumentos de productividad que aportaban los ordenadores. Esta idea fue la que distintos autores posteriormente denominaron la “paradoja de la productividad”, y ha sido explicada ampliamente en los trabajos llevados a cabo por el profesor Eric Brynjolfsson del MIT Sloan School of Management. El profesor Brynjolfsson afirma que en los estudios anteriores no se detectaban los incrementos de la productividad, ya que se producían en áreas que no se tenían en cuenta en los análisis convencionales de productividad, como, por ejemplo, en el incremento de la calidad de los productos y en mejoras en la relación con clientes, etc. Además, no todas las organizaciones aumentan su productividad, sino que en algunas empeora (con lo que los valores se pueden llegar a compensar); además, el factor tiempo es fundamental (los aumentos no se producen de forma inmediata, sino a medio y largo plazo), lo que explica, según el autor, que la productividad de las TIC no se refleje en el ámbito agregado y en un plazo de tiempo determinado.

Las organizaciones que tienen más experiencia en el uso de las herramientas de *Business Intelligence* están planteando los proyectos a nivel corporativo en lugar de hacerlo a nivel departamental. A partir de la experiencia de los proyectos a nivel departamental, se generan nuevos a nivel corporativo y se comienza a hablar de *Enterprise Business*

Intelligence, es decir, proyectos a nivel corporativo o la aparición de Centros de Competencia de *Business Intelligence* a nivel corporativo.

Una de las consecuencias del aumento del mercado y del número de licencias, junto con las nuevas tendencias de gestión en las que se da cada vez más importancia a que los distintos empleados de la organización accedan a aquella información que es clave para el desempeño de sus tareas, están provocando una mayor democratización en el uso de la información en las organizaciones. El objetivo es dar acceso a los usuarios a la información relevante que les permita tomar mejores decisiones, y por lo tanto aportar el máximo valor a la organización.

El uso de las herramientas de *Business Intelligence* está provocando la mitigación de los “silos de información”. Tradicionalmente hablamos de “silos de información” cuando dentro de las organizaciones no se comparte toda la información necesaria entre los distintos departamentos o centros, lo que evidentemente a la larga genera conflictos entre los mismos.

Como hemos comentado anteriormente, el uso de la información proveniente de las herramientas de *Business Intelligence* está invirtiendo la tendencia de enviar la información (*Push*) a los destinatarios; este sistema era el más común ya que suponía un menor coste de licencias, pero debido a la aparición de las herramientas que soportan las tecnologías *web* se está produciendo un aumento de los usuarios que utilizan estas tecnologías para acceder (*Pull*) a las herramientas de *Business Intelligence*.

Uno de los factores que ha provocado un incremento del uso de herramientas de *Business Intelligence* es la aparición de las normativas internacionales de presentación de información, así como la adopción de distintas metodologías, como por ejemplo European Foundation for Quality Management (EFQM), SixSigma, Corporate Performance Management (CPM), Balanced Scorecard (BSC), etc.

Uno de los factores clave que está generando mayor confianza en el uso de herramientas de *Business Intelligence* es el aumento de la calidad de la información. Sin calidad en la información de la que disponemos para la toma de decisiones es improbable que se puedan seguir desarrollando proyectos.

Se está produciendo además una tendencia en la reducción del número de herramientas por categorías. Algunas organizaciones disponen de distintas herramientas de distintos fabricantes para cubrir distintas necesidades. Los criterios que les llevaron a la elección fue que cada una de las herramientas soportaba mejor sus necesidades en el momento de la elección. Pero esta diversidad de herramientas provoca: costes de mantenimiento, coste de formación, costes de soporte, dificultades en la negociación con los distintos proveedores, soporte a diversidad de sistemas, etc. Dicha circunstancia está motivando que dentro de cada una de las organizaciones se consolide una solución, en lugar de utilizar varias.

Tradicionalmente, las herramientas de *Business Intelligence* se utilizaban en planificaciones estratégicas a medio y largo plazo: partiendo de lo que había sucedido, se utilizaba la información para poder planificar. En la actualidad, las herramientas de *Business Intelligence* se utilizan cada vez más para gestionar el día a día, el corto plazo: las tareas más operacionales. Ello implica que las cargas de la información sean mucho más frecuentes, llegando en algunos casos a lo que se ha denominado “*Real time*”, o tiempo real.

En muchas organizaciones siguen teniendo un uso privilegiado las hojas de cálculo, destacando el líder del mercado Microsoft Excel. Las tendencias indican que se seguirán usando las hojas de cálculo, pero no como repositorios de información, sino tan sólo como herramientas de acceso y visualización de la información residente en los *datawarehouses*.

Otra tendencia es la externalización del *Business Intelligence*: Aunque de una forma incipiente, comienzan a aparecer experiencias de externalización del *Business Intelligence* fuera de las organizaciones. El principal inhibidor de esta tendencia es que las organizaciones son reticentes a externalizar la información de la que disponen.

Los movimientos en el mercado

Dentro del mercado de las herramientas de *Business Intelligence* se están produciendo distintos movimientos:

- Los fabricantes de soluciones ERP están comercializando sus propios productos (por ejemplo: Business Warehouse de SAP, o su solución de *Business Intelligence* para SAP Business One).
- Los fabricantes de BI se están especializando en soluciones concretas, por ejemplo: Soluciones CPM (por ejemplo Cognos con Cognos Planning).
- Los fabricantes de motores de BBDD tienen sus propias soluciones de *Business Intelligence* (por ejemplo: Oracle con sus productos y los de Siebel Analytics, y Microsoft con SQL 2005).
- Disminución de los precios de algunas soluciones.
- Aparición de las soluciones *Open Source*.
- Futuras compras entre fabricantes.

En el mercado de las Tecnologías de la Información y de la Comunicación siempre se debe estar atento a los cambios que se producen.

Existen algunas consultoras independientes que evalúan continuamente la evolución de los productos: Gartner, Forrester, etc., a las que podemos dirigirnos cuando tengamos que decidir en qué herramienta invertir.

10.7. – Conclusiones

En el apartado partimos de la definición teórica de *Business Intelligence* para poder mostrar la amplitud del término y a partir de ella ver sus posibilidades de uso en las organizaciones. Una vez descritos todos los componentes, les hemos propuesto distintas metodologías para que les ayuden en el desarrollo de los proyectos, y a continuación hemos propuesto los usos más habituales de *Business Intelligence* y las nuevas tendencias en este campo.

En sus inicios, *Business Intelligence* se aplicaba en las organizaciones a nivel táctico, es decir: analizo, tomo la decisión y establezco las políticas a aplicar. La segunda ola de *Business Intelligence* subió al nivel de la estrategia, como una herramienta que ayuda a la planificación estratégica. Hoy hablamos de *Business Intelligence* operativa, que es la que está ligada a la toma de decisiones en el día a día, con el objetivo de ser más ágiles en dicho proceso.

De las experiencias de los proyectos se desprenden distintos factores críticos de éxito:

- La importancia de tener un patrocinador del proyecto, si es posible, al máximo nivel de responsabilidad o dirección de la organización. El patrocinio por su parte nos ayudará a conseguir proyectos de mayor alcance y más ligados con la estrategia de la organización, lo que nos facilitará probablemente un mayor retorno de la inversión y un mayor apoyo durante la ejecución de los mismos.
- A lo largo del apartado hemos insistido mucho en la utilización de una metodología. Este aspecto es clave para conseguir el éxito en el proyecto. La improvisación no es una buena compañera de los proyectos, y menos de los de sistemas de información para la toma de decisiones.
- El equipo multidisciplinario formado por miembros tanto del área de negocio como de la tecnológica. Compartir los proyectos facilita la comprensión de las distintas necesidades y mejora sin duda las relaciones entre ellos.
- La participación de los usuarios en el proyecto es fundamental, ya que ellos serán los que consigan los éxitos con el uso de las soluciones de *Business Intelligence*. Es fundamental que se registren los éxitos

obtenidos con el uso de las soluciones. El poder compartirlos nos asegurará la continuidad y los recursos necesarios para seguir avanzando.

- La definición de unos objetivos alcanzables y alineados con los de la organización. Los mayores éxitos de las organizaciones que compiten mediante el uso de *Business Intelligence* se obtienen cuando somos capaces de aportar información sobre áreas estratégicas de la organización. Normalmente, estas áreas están relacionadas con los clientes: el nivel de servicio, tiempo del ciclo, optimización de costes, selección de personal, etc.
- Debemos establecer un seguimiento del proyecto que nos permita evaluar su nivel de avance y de obtención de resultados. Cuando las organizaciones se acercan a la madurez en el uso de estas tecnologías dejan de plantearse proyectos individuales y los gestionan como una forma de competir.
- La mayoría de las empresas que compiten con *Business Intelligence* no han ido invirtiendo de forma progresiva: estas inversiones, en la mayoría de los casos, se han generado por la aparición de nuevas necesidades y por el propio aprendizaje de las organizaciones.
- La evaluación continua de los resultados del proyecto nos permite mostrar cuáles han sido los obtenidos: al comunicarlos podemos generar el interés en nuevas áreas de análisis, creando nuevos modelos interdepartamentales que sin duda mejoran los resultados de la organización.
- El uso de las soluciones de *Business Intelligence* nos mostrará resultados que nos obligarán a tomar decisiones, por lo que es necesario que estemos preparados para ello.
- La tecnología debe ser coherente con nuestra organización y nuestros usuarios.

En conclusión lo deseable con la ayuda de Business Intelligence es conseguir organizaciones más eficientes, más eficaces y más competitivas, para que con ello puedan contribuir a la creación de puestos de trabajo y al desarrollo de una sociedad más humana, más sostenible, más respetuosa con el medio ambiente y más responsable socialmente.

Conclusiones

11.- CONCLUSIONES

Tras el estudio de este proyecto es sencillo constatar la importancia del software libre en la actualidad, como una ciencia más, puesta en manos de la sociedad, para su estudio, adaptación e incluso mejora de la misma para el beneficio común, en una época tecnológica que algunas han vaticinado como de piratería, resurge el modelo de software libre, el modelo de comunidad, el modelo de bien común.

El software libre como herramienta empresarial a todos los niveles, da oportunidad a las empresas de dotar de la tecnología software adecuada para el manejo de la información, de manera que podamos cubrir todas las áreas funcionales sin previo coste de licencias software.

Las herramientas ERP de software libre no difieren de lo anteriormente citado, la descarga de paquetes ERPs completos para su posterior uso y modificación a todos los niveles, con el fin de adaptarla a cada empresa y no al contrario (que es lo que viene sucediendo con gran parte de herramientas de software propietario), adquiere mayor significación si consideramos la disminución del peligro del software obsoleto, que en este caso disminuye de manera drástica, debido a que somos poseedores de todas las herramientas necesarias para que esto no ocurra.

La elección a partir de los criterios nombrados con anterioridad, nos ayuda a escoger, dependiendo de nuestras necesidades, de la herramienta que mejor se adapte a nuestra empresa, todo ello sin coste alguno de licencias.

En contraposición, hay que ser conscientes de la sociedad actual en la que vivimos, maximizar el beneficio económico es la finalidad fundamental que una empresa de carácter privado tiene, en el tema que hemos analizado contiene similitudes constatables. De entre las herramientas ERPs de software libre que hemos analizado, es constatable que versiones renovadas y con multitud de nuevas funcionalidades pierden el concepto inicial de software libre, ya que es obligado el previo pago para su correspondiente uso, otros paquetes, siguen fieles al dogma del software libre y de comunidad, no adoptando la opción anteriormente citada.

Tras el transcurso del proyecto mi credulidad acerca del software libre, de alguna forma a sido cambiante, en un principio la creencia de que cualquier empresa era capaz de soportar toda su infraestructura de sistemas, la veía como una realidad actual, es más, grandes empresas lo constatan, pero también es cierto que esas grandes empresas poseen departamentos de sistemas informáticos con un nivel y unos recursos que son complicados de ver en la sociedad española, independientemente del

beneficio que saquen con ello. Para una pyme es impensable un gasto “más allá de lo necesario” en un departamento de sistemas informáticos, con lo cual la dependencia con empresas relacionadas para gozar de un soporte adecuado, así como de una adecuación de la aplicación a su negocio, se hace fundamental. Es aquí donde comienzan las dudas, el software libre, es un concepto distinto del que muchos creen, si te regalan un coche, no significa que vayas a tener transporte toda tu vida, el software libre conlleva un coste, que a priori será menor que el de un software propietario.

La elección de un software adecuado y teniendo en cuenta la importancia del mismo no es una decisión que se deba tomar a la ligera, tener la información adecuada es lo que he tratado de mostrar en este proyecto, qué empresas están detrás de cada ERP, que seguridad de continuidad tenemos, tras nuestra decisión de adoptar esa herramienta en particular, qué nos ofrece y que nos ofrecerá. Son factores tan importantes a tener en cuenta que pasarlos por alto nos podría llevar al abandono del proyecto con todo lo que conlleva.

Con todo lo anterior la elección de un ERP adecuado es lograr una aplicación global que contiene o debiera contener toda la información de la empresa. La importancia del uso de la información de manera adecuada, así como mejora de los procesos de negocio, la mejora en la relación con los clientes y la representación de la información importante para la toma de decisiones adecuadas, son puntos tratados con anterioridad a tener muy en cuenta.

Anexo paquetes ERP

12. – Anexo paquetes ERP

Abanq

En este breve documento veremos el funcionamiento general de la aplicación en cuanto a módulos, áreas y formularios.

Organización de los módulos

La configuración está estructurada en los siguientes elementos:

- Áreas. Representan grandes agrupaciones funcionales (facturación, contabilidad, producción,...).
- Módulos. Cada área, a su vez, está dividida en módulos, que cubren una determinada faceta del área a la que pertenecen (por ejemplo, en el área Facturación, los módulos tesorería, almacén...).
- Acciones. Las acciones determinan las posibles operaciones que el usuario puede realizar en un determinado módulo (por ejemplo en el módulo almacén del área Facturación, están las acciones de gestión de artículos, gestión de stocks, etc.). Generalmente, accedemos a las acciones desde la ventana principal de cada módulo.

Dependencias. Los módulos están integrados unos con otros. Por ejemplo, al usar el módulo facturación del área Facturación para dar de alta un pedido a cliente, podemos seleccionar dicho cliente de una lista que reside en el módulo principal del área de Facturación. Esta integración, necesaria para reaprovechar datos y funcionalidad existentes, implica que algunos módulos no pueden ser instalados sin haber instalado previamente aquellos de los que dependen. Abanq controla estas dependencias y avisa al usuario cuando alguna de ellas no se cumple.

Modo general de funcionamiento

Abanq muestra la información al usuario siguiendo un esquema maestro - detalle. La interfaz de usuario se estructura de la siguiente forma:

Ventana de inicio

El la ventana de inicio por defecto de Abanq, y la que da acceso a las áreas. Cada área despliega sus módulos. También accedemos a las opciones generales del programa (Configuración).

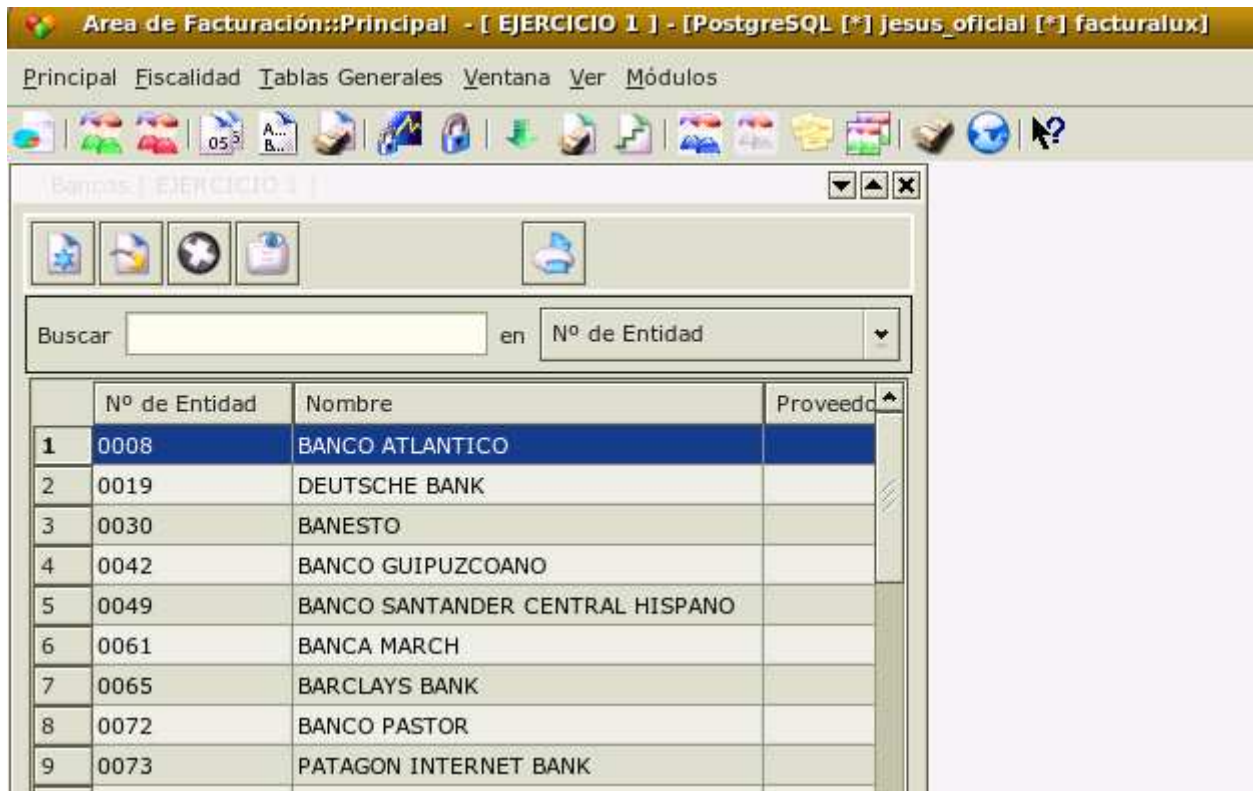


Ventana de inicio

Ventana principal del módulo

Cada módulo tiene una ventana en la que se ofrece al usuario el conjunto de acciones disponibles. Estas acciones son accesibles desde la barra de menú o la barra de herramientas de la ventana.

En cada una de las ventanas principales disponemos de un menú Módulos que permite cambiar a otras áreas y módulos.



Ventana principal del módulo de facturación. Vemos los menús y barra de herramientas

Formulario maestro

En primer lugar, accederemos siempre a un formulario maestro desde la opción de la página principal del módulo. En este formulario maestro se nos ofrece una lista con los registros que existen para la acción seleccionada. Desde aquí podemos seleccionar el registro sobre el que actuaremos y la operación a realizar (creación, modificación, borrado, copia, etc.).

Operaciones con un formulario maestro:

- **Buscar.** En la casilla buscar podemos teclear el texto a buscar. Abanq lo buscará dentro de la primera de las columnas de la tabla. Se puede utilizar el carácter comodín %. Veamos algunos ejemplos suponiendo que la primera columna es el número de entidad bancaria en el formulario maestro de bancos. Si tecleamos %8 estamos buscando todos los bancos cuyo número de entidad termina por 8. Si tecleamos 0%2 obtenemos todas las entidades que comienzan por 0 y terminan por 2, y así sucesivamente.
- **Ordenar.** El formulario maestro siempre ordena de menor a mayor por la primera de las columnas mostradas. Podemos cambiar la primera de las columnas mediante el control desplegable que aparece a la derecha del cuadro de búsqueda.

	Nº de Entidad	Nombre	Proveedo
1	0008	BANCO ATLANTICO	
2	0019	DEUTSCHE BANK	
3	0030	BANESTO	
4	0042	BANCO GUIPUZCOANO	
5	0049	BANCO SANTANDER CENTRAL HISPANO	
6	0061	BANCA MARCH	
7	0065	BARCLAYS BANK	
8	0072	BANCO PASTOR	
9	0073	PATAGON INTERNET BANK	
10	0075	BANCO POPULAR	
11	0081	BANCO DE SABADELL	
12	0093	BANCO DE VALENCIA	

Formulario maestro de bancos, dentro del módulo principal de facturación

Formulario de edición

Una vez seleccionados el registro y la operación a realizar, Abanq nos muestra un formulario de detalle, en el que podemos actuar sobre cada uno de los campos que componen el registro seleccionado. Es posible anidar esta estructura de forma que en un formulario de detalle se incluya una lista de registros que abra, a su vez, un segundo formulario de detalle.

Formulario de edición de bancos

Teclas de acceso rápido

Las teclas de acceso rápido facilitan y agilizan enormemente el trabajo diario. Veamos las más importantes:

Desde cualquier formulario

- Cerrar una ventana. Esc. Para cerrar una ventana tanto de módulo como de formularios maestro o de edición, basta pulsar la tecla escape (Esc).

Desde un formulario maestro

- Crear un nuevo registro. A. Cuando el foco está situado en una tabla, dentro de un formulario maestro o de edición, la pulsación de la tecla A equivale al botón Añadir registro, abriendo un formulario de introducción de un nuevo registro.
- Editar un registro. M/Intro. Situado el cursor sobre un registro de una tabla en un formulario maestro o de edición, la pulsación de las teclas M o Intro equivale al botón Modificar registro, abriendo el registro seleccionado en modo edición.

- Ver un registro en sólo lectura. V. Situado el cursor sobre un registro de una tabla en un formulario maestro o de edición, la pulsación de la tecla V equivale al botón Ver registro, abriendo el registro seleccionado en modo de sólo lectura.
- Eliminar un registro. E/Supr. Situado el cursor sobre un registro de una tabla en un formulario maestro o de edición, la pulsación de las teclas E o Suprimir (Supr) equivale al botón Eliminar registro, borrando el registro seleccionado, previa confirmación del usuario.
- Copiar un registro. C. Algunas tablas contienen este botón que permite hacer un duplicado del registro activo. No obstante el resultado depende de la tabla y es posible que sólo se copien los datos generales y no los registros relacionados si los hay. Situado el cursor sobre un registro de una tabla en un formulario maestro o de edición, la pulsación de la tecla C equivale al botón Copiar registro, creando un nuevo registro a partir del registro seleccionado.

Desde un formulario de edición

- Saltar de campo. Tab/Intro. Al pulsar tabulador (Tab) o Intro saltamos al siguiente campo. El orden de tabulación está predefinido para cada formulario. Intro equivale a Tab salvo en los campos de texto largo en los que tiene su función normal de salto de línea.
- Campos relacionados. F2/+. En un formulario de edición, son aquellos cuyo valor ha de seleccionarse escogiendo uno de los registros de otra tabla (tabla relacionada); por ejemplo, en una cuenta bancaria podemos seleccionar el campo relacionado Entidad escogiendo un registro de la tabla de bancos. Estos campos aparecen en el formulario de edición con el icono de búsqueda (una lupa). Situado el cursor en uno de estos campos se puede abrir la tabla relacionada pulsando F2 o la tecla +.
- Navegar por los registros. F5-F8. Dentro del formulario de edición, es posible avanzar por los registros cuando estamos en modo de edición o de vista (sólo lectura). Las teclas son F5 (primer registro), F6 (registro anterior), F7 (registro siguiente), F8 (último registro)
- Guardar un registro y cerrar el formulario. F10. Dentro del formulario de edición, en modo nuevo registro o edición, pulsando F10 se guarda el registro.
- Guardar un registro e insertar otro nuevo. F9. Dentro del formulario de edición, en modo nuevo registro o edición, pulsando F9 se guarda el registro y se abre otro nuevo en modo nuevo registro.

Guía de Instalación de Abanq y gestión de módulos

En esta guía veremos cómo instalar Abanq y gestionar sus módulos

Descarga e instalación de la aplicación base

En primer lugar debemos dirigirnos a la zona de descargas de www.Abanq.org y obtener la aplicación base -ejecutable- adecuada a nuestra plataforma.

- Instalación para Windows / Mac OS X. Ejecutamos directamente el programa de instalación y seguimos las instrucciones como en cualquier otro programa de instalación para Windows o Mac OS X
- Instalación para GNU / Linux. Abrimos una consola como usuario raíz (root), y desde el directorio donde hemos guardado el paquete, lo ejecutamos:

```
[root@mimaquina midir]# sh ./Abanq-lite-X.X.x86.Linux.bin.run
```

Bases de Datos

Como hemos visto al inicio de este tema, la principal característica de Abanq y una de las claves de su flexibilidad es el hecho de que tanto los datos de usuario como los módulos se almacenan en una base de datos.

Al iniciar la aplicación, el primer cuadro de diálogo pide los datos para la conexión.



Si es la primera vez que iniciamos Abanq, introduciremos el nuevo nombre para la base de datos, a nuestra elección. Rellenaremos el resto del formulario tal como aparece en la figura. Si la base de datos no existe, Abanq preguntará si deseamos crear una nueva base de datos. Si respondemos afirmativamente, se nos avisará de la creación de la base de datos y arrancará la aplicación.

Es importante recordar que cada nueva base de datos no contendrá ningún módulo instalado salvo el de sistema. El resto de módulos deberán ser cargados como se explica posteriormente.

Es perfectamente posible crear una nueva base de datos sólomente indicando un nuevo nombre para la base de datos en el diálogo al iniciar Abanq. Esto no influye en absoluto sobre otras bases de datos previas; podemos perfectamente trabajar con varias bases de datos, cada una con sus módulos y datos de usuario, desde una única instalación del cliente Abanq.

Podríamos, por ejemplo, tener una base de datos para cada empresa, en caso de querer gestionar más de una, o crear una base de datos de pruebas sobre la que realizar un test sobre las funcionalidades de nuevos módulos, antes de instalarlos sobre nuestra base de datos habitual.

Más información sobre PostgreSQL

Instalación de módulos

Nota Importante. En este documento explicaremos cómo cargar, actualizar y desinstalar los módulos sobre la aplicación base de Abanq. Es importante tener en cuenta que la versión actual para Windows incluye ya incorporados los módulos públicos, por lo que no es necesario seguir los pasos aquí descritos a la hora de instalar Abanq por primera vez. Sí es interesante conocer el funcionamiento del módulo de sistema para posteriores instalaciones o actualizaciones de los módulos.

Antes de poder instalar los módulos es necesario obtenerlos en Abanq.org, donde hay módulos públicos gratuitos y muchos otros módulos especiales. Una vez descargados, descomprimimos el paquete en una carpeta de nuestro disco duro, desde la que a continuación cargaremos los módulos sobre Abanq.

1. Cómo funciona el módulo de administración

El módulo de administración se integra en el área de Sistema. Se crea automáticamente cuando instalamos Abanq, desde él podemos cargar, borrar o recargar todos los demás módulos. Está incluido en un área con el mismo nombre y es el único módulo que no podemos desinstalar. La ventana principal de este módulo está compuesta de las siguientes opciones:

- **Cargar Módulos.** Desde esta opción se realiza la carga de módulos que aún no existen o que han sido modificados.
- **Áreas.** Se abrirá una ventana en la que aparecerá el nombre de todas aquellas áreas que componen nuestra configuración, desde este mismo lugar podremos crear, borrar o modificar cualquiera de ellas.
- **Módulos.** Desde esta opción podremos acceder a todos los módulos que componen nuestras áreas, teniendo acceso a todos los ficheros de que está compuesto cada uno de ellos.

2. Cargando los módulos

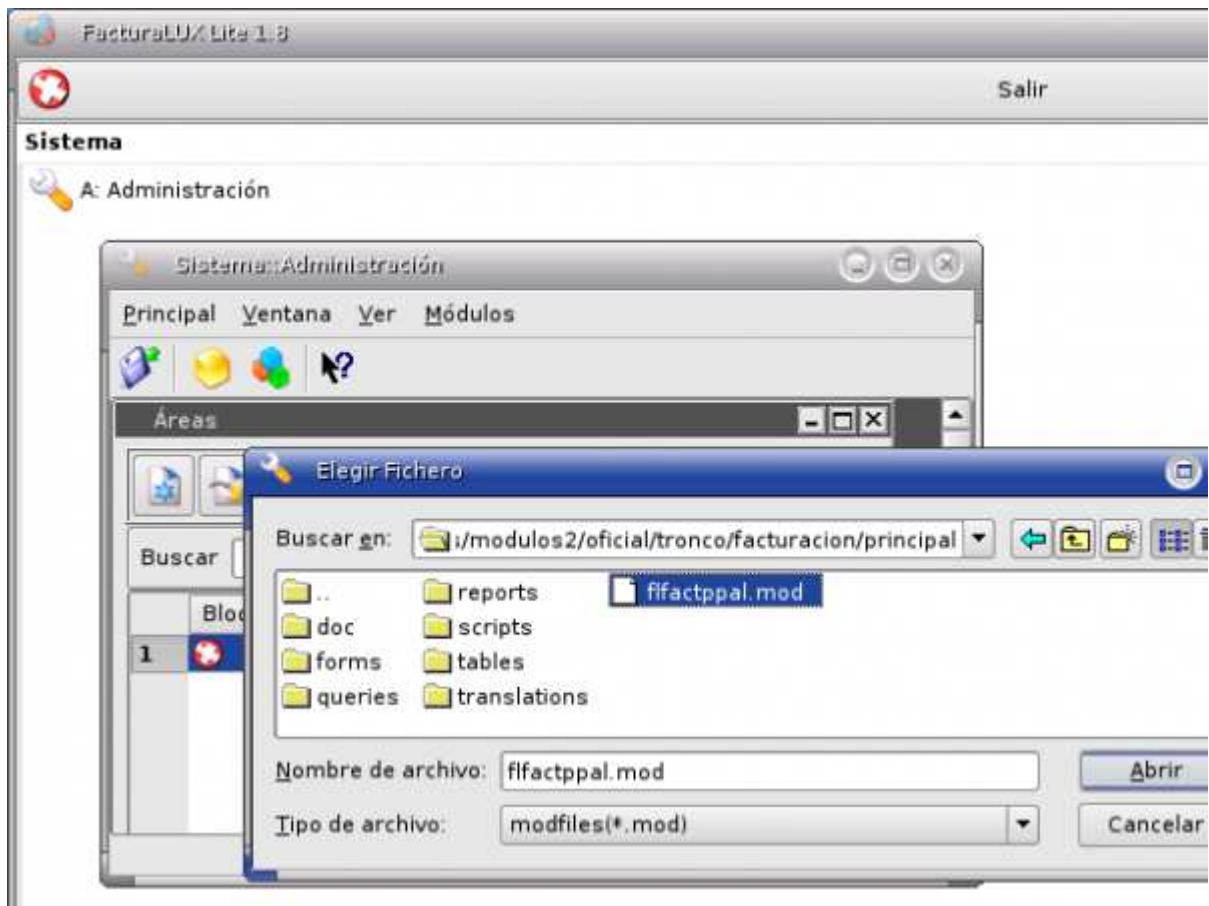
Arrancamos Abanq y seleccionamos el área Administración. Pulsamos el botón Cargar Módulo y seleccionamos el fichero *.mod correspondiente al módulo que deseamos cargar.

Si todavía no existe el área correspondiente al módulo que estamos instalando, Abanq nos preguntará si queremos crearla en este momento.

Contestamos Sí, e introducimos la descripción del área.

Abanq cargará los ficheros que componen el módulo, mostrando su icono en el menú del área correspondiente.

Con esto hemos terminado la instalación del módulo. Podemos probarlo pulsando sobre el icono.



Desinstalación de módulos

Seleccionamos el área Sistema y el módulo Administración. Pulsamos el botón Modulos y seleccionamos el fichero correspondiente al módulo que deseamos desinstalar, pulsamos el botón Eliminar Registro.

La desinstalación de un módulo implica que la funcionalidad que dicho módulo proporciona deja de estar disponible para el usuario, aunque

los datos (tanto de usuario como de definición del módulo) se mantienen en la base de datos.

Recarga de nuevas versiones

Seleccionamos el área Sistema y el módulo Administración. Pulsamos el botón Cargar Módulo y seleccionamos el fichero .mod correspondiente al módulo que deseamos recargar.

Puesto que el modulo ya existe, debemos confirmar que deseamos recargar el módulo. A continuación Abanq cargará automáticamente todos aquellos ficheros que componen el módulo y tengan alguna modificación.

Cuando cambiemos de módulo Abanq se reiniciará para cargar los nuevos datos.

Por seguridad, es recomendable hacer un backup de los datos antes de recargar un módulo.

Editar ficheros desde el módulo de sistema (para programadores)

Cuando cargamos un módulo desde disco utilizando el área de sistema, todos los ficheros del módulo son almacenados en la base de datos. En el área de sistema, la acción Módulos permite gestionar estos ficheros residentes en la base de datos. Al abrir un módulo de la lista, la ventana resultante nos mostrará un listado de los ficheros que componen dicho módulo. Si abrimos uno de ellos, nos aparecerá la un formulario.

El botón Editar fichero va a lanzar la aplicación adecuada para editar el archivo según su extensión: si se trata de una tabla abrirá un editor de textos, por ejemplo.

Procedimiento para realizar modificaciones (para programadores)

Hemos visto cómo Abanq incorpora varios editores que son accesibles desde el módulo de sistema para modificar los ficheros contenidos en la base de datos. Dependiendo del tipo de fichero, Abanq abrirá el editor adecuado.

Este sistema es útil a la hora de mostrar la flexibilidad y facilidad de modificación del programa, o para realizar pequeños cambios sobre la marcha. Sin embargo, a la hora de realizar modificaciones más serias, o de entrar en un desarrollo real, el procedimiento recomendado es el siguiente:

1. Abrir directamente los ficheros a modificar en el directorio correspondiente y con el editor adecuado: las tablas, scripts y consultas en un editor de texto, los formularios con QT Designer, cuyo ejecutable es designer, instalado en el mismo directorio que el ejecutable de Abanq.

2. Realizar las modificaciones oportunas
3. Volver a Abanq y recargar el módulo
4. Probar las modificaciones realizadas
5. Repetir los pasos 2 , 3 y 4 hasta finalizar las modificaciones

Volcado a disco de los módulos

Según hemos visto, una vez cargado un módulo, éste pasa a la base de datos para ser operativo. Es importante recordar que aunque la estructura de ficheros de los módulos reside en disco, es de la base de datos de donde Abanq toma los módulos previamente cargados.

Abanq integra varias herramientas para editar los módulos, tales como un procesador de textos. Para usar éstas herramientas integradas debemos abrir uno de los ficheros que integran un módulo desde el área de sistema.

No obstante, si vamos a realizar modificaciones importantes en los módulos, resulta más eficiente trabajar sobre los ficheros de los módulos en disco y, una vez modificados, recargarlos a la base de datos desde Abanq.

Si, por ejemplo, deseamos insertar nuevos campos en una tabla, podemos abrir el archivo .mtd correspondiente dentro del directorio tables del módulo correspondiente con nuestro editor de texto favorito. Una vez modificado el fichero para insertar los nuevos campos, recargaremos el módulo tal como hemos visto en apartados anteriores.

Es importante notar que si ya hemos realizado cambios desde el área de sistema, estos cambios residen en la base de datos pero no han sido trasladados a los ficheros en disco, por tanto si recargamos un módulo desde el disco podemos perder los cambios realizados. Para evitar esto existe la posibilidad de volcar a disco un módulo.

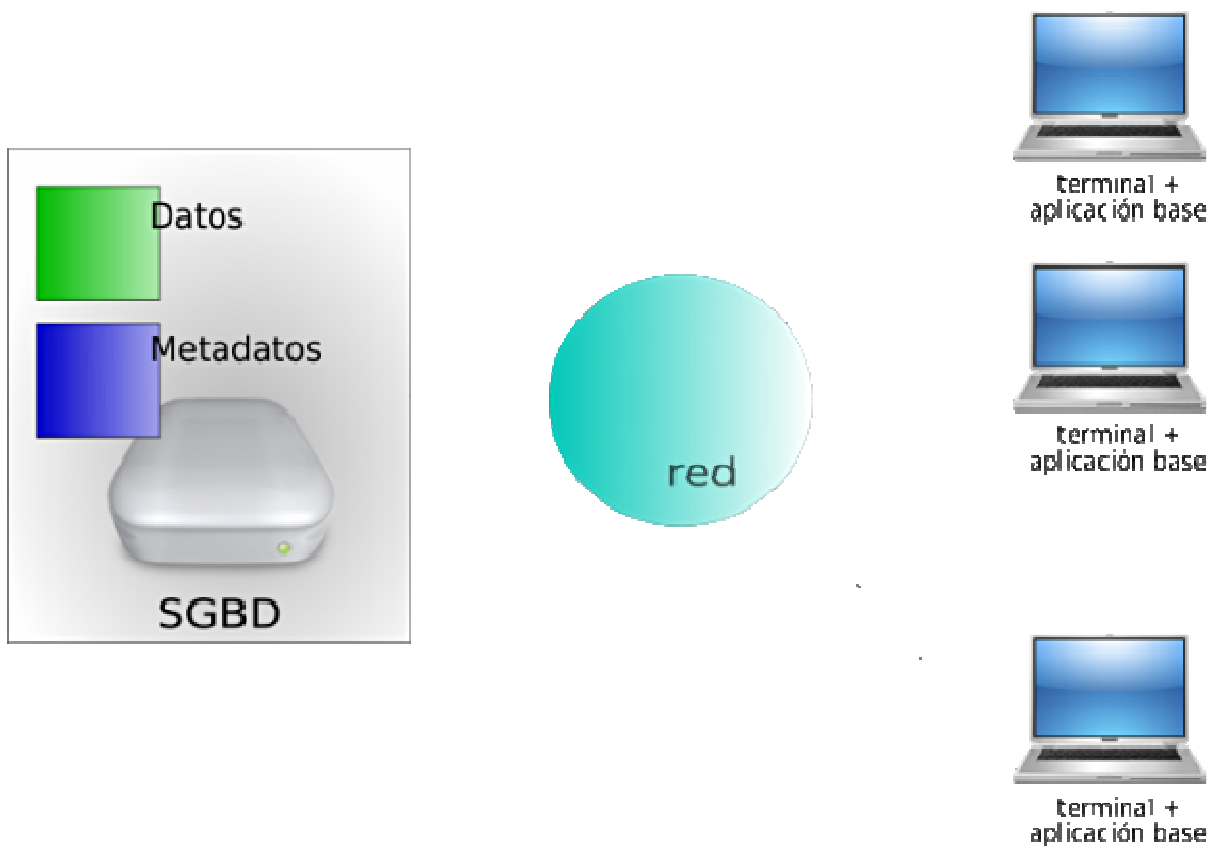
Para ello abriremos, dentro del área de sistema y la acción módulos, el módulo que deseamos volcar. Pulsando el botón indicado en la figura se creará una copia en disco de los ficheros residentes en la base de datos.

Introducción a la estructura de Abanq

En este breve artículo vamos a tratar de clarificar la estructura de Abanq, paso previo muy importante antes de comenzar la programación y personalización de los módulos.

Arquitectura del Sistema

En la figura se observa el esquema general de la arquitectura Abanq. Vemos cómo todo se almacena en la base de datos y sólo el servidor puede acceder directamente a ella, sirviendo a los clientes los datos y los módulos de aplicación, y gestionando el control de acceso a los usuarios.



Estructura A3D

Cientes: Los clientes son las máquinas que se encuentran conectadas directamente al SGBD (sistema gestor de base de datos) pudiendo acceder a la base de datos. En cada terminal o máquina cliente se ejecuta el software que denominamos aplicación base.

SGBD: El gestor de base de datos se encarga de almacenar y mantener dos tipos de información (y aquí está la clave):

Datos: En esta zona de la base de datos se almacenan los datos concretos que la aplicación maneja y que tienen sentido para el usuario (datos de clientes, facturas, etc.) . Son los datos "tradicionales".

Módulos de metadatos: Los módulos contienen la información necesaria para implementar las aplicaciones de usuario: formularios, definiciones de tablas y campos, código de los scripts que realizan los procesos, formato y definición de los informes.

Los metadatos residen en la base de datos, pero previamente deben ser cargados desde el directorio en disco en el que han sido alojados tras su descarga.

La estructura de directorios en los módulos presenta cuatro niveles:

- Nivel 1. Directorio raíz (ejemplo: directorio modulos)
- Nivel 2. Área (ejemplo: directorio facturacion)
- Nivel 3. Módulo (ejemplo: directorio almacen)
- Nivel 4. Metadatos (ejemplo: directorio tables)

En el nivel 4 tendremos varios directorios, uno por cada tipo de metadatos:

- tables. Definiciones de las tablas. Cada tabla se define en un archivo de extensión mtd
- forms. Definiciones de los formularios. Cada formulario se define en un archivo de extensión ui
- scripts. Definiciones de los scripts. Cada script se define en un archivo de extensión qs
- queries. Definiciones de las consultas. Cada consulta se define en un archivo de extensión qry
- reports. Definiciones de los informes. Cada informe se define en un archivo de extensión kut
- translations. Listados de traducciones. Cada listado de traducciones para un determinado idioma se define en un archivo de extensión ts

Comparando Abanq con el sistema de navegación Web

Desde el punto de vista de un programador o usuario avanzado, el SGBD funciona como un servidor de páginas web, mientras que la aplicación base hace las veces de un navegador. La aplicación base no es más que un intérprete de los datos que recibe del SGBD. Cuando la aplicación base se conecta al SGBD, descarga del mismo tanto los datos como los metadatos. En Abanq los formularios y la funcionalidad residen en el servidor de la base de datos, no en la aplicación base, al igual que las páginas web no residen en el navegador.

Siguiendo con la analogía web, podemos comparar los scripts de Abanq con scripts de Javascript que son descargados al navegador y

ejecutados en el ordenador del internauta; los formularios podrían ser tablas o formularios HTML, también aparecen en el navegador pero proceden así mismo del servidor.

Sabemos que nuestro navegador puede conectarse a un número ilimitado de sitios web; igualmente la aplicación base de Abanq puede escoger la base de datos a la que se conecta en el momento del arranque.

Sabemos también que un navegador web depende del sistema operativo sobre el que se instala: Mozilla Firefox para Windows o Linux, Safari para MacOSX, etc. Sin embargo cualquiera de estos navegadores puede conectarse a Google.com

De igual modo, varias aplicaciones base Abanq para distintas plataformas pueden acceder a una misma base de datos central y no sólo compartir los datos, también los informes, formularios y funcionalidades. Cuando se requiere una actualización, basta con actualizar una vez en la base de datos. Cuando una aplicación base cliente se conecte a la misma, automáticamente aparecerán las últimas tablas, informes o scripts cargados.

¿Cómo es posible esta portabilidad? la respuesta es de nuevo análoga al sistema web: la aplicación base recibe los metadatos en formato de texto plano -igual que el HTML o el código javascript- y los interpreta en tiempo real.

Tutorial. Programación en Abanq (I). Primer contacto

Este es el primero de una serie de tutoriales orientados a programadores y usuarios avanzados acerca de programación sobre Abanq. En este tutorial veremos cómo realizar una personalización básica y muy sencilla, pero que nos va a dar una idea de la flexibilidad de la aplicación y la facilidad con que se pueden realizar cambios sobre la estructura de los datos y el aspecto de los formularios.

Requerimientos

Antes de comenzar a trabajar:

- Descargar e instalar la aplicación base de Abanq (recomendamos la versión más reciente) instalada
- Descargar los módulos públicos. Para este tutorial bastarán los módulos del área de Facturación
- Arrancar Abanq con una nueva base de datos y cargar los módulos

Algunos conceptos previos: el área de Sistema

Abanq no es sólo un software de gestión, incluye además un entorno de desarrollo que permite realizar cambios y personalizaciones desde lo más básico a lo más avanzado.

Desde el área de sistema no sólo podemos cargar los módulos, también podemos modificar los ficheros de tablas, formularios, informes, etc que forman parte de un módulo.

Para ello abriremos el módulo de Administración dentro del área de sistema. Pulsamos en el menú Principal -> Módulos. Veremos un listado de los módulos instalados. Si abrimos, por ejemplo, el módulo flfactppal (principal de facturación) accedemos al listado de ficheros. Algunos ejemplos: clientes.mtd es la tabla de clientes; clientes.ui es el formulario de clientes, etc.

Los principales tipos de ficheros que maneja Abanq son:

- tablas (extensión mtd)
- formularios (extensión ui)
- scripts (extensión qs)
- plantillas de informes (extensión kut)
- consultas sql para informes (extensión qry)

Desde el listado de ficheros de un módulo podemos abrir un fichero para ver su contenido (siempre textual). Si pulsamos el botón Editar fichero, Abanq reconoce automáticamente el tipo de fichero y abre el editor adecuado.

Algunos ejemplos: para las tablas se abrirá un editor de texto, para los formularios el editor de formularios QDesigner. Todas estas herramientas son incorporadas durante la instalación de Abanq.

Cambios básicos en tablas y formularios

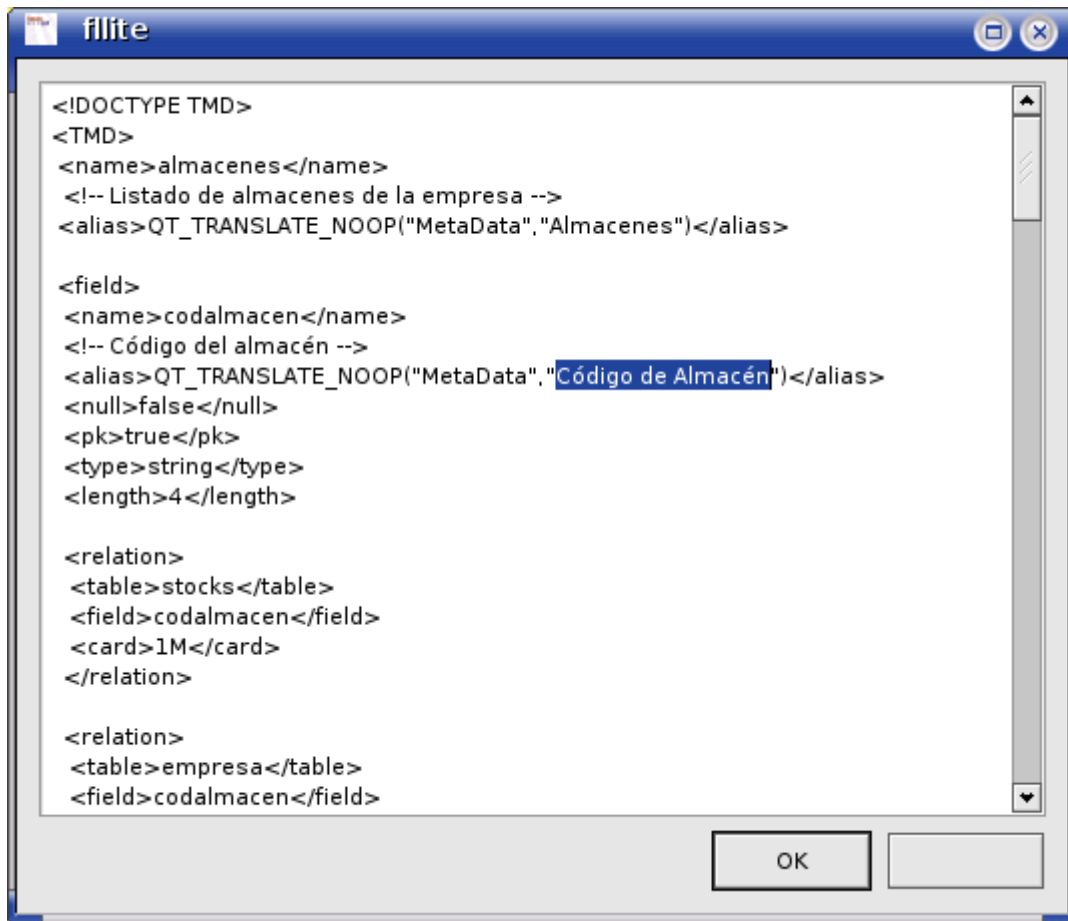
Vamos a utilizar las herramientas que incorpora Abanq para realizar algunos cambios sencillos en tablas y formularios de los módulos previamente cargados

1. Cambio de propiedades de un campo

Cambio de alias. El alias de un campo es el nombre que aparece en los formularios y las tablas maestras. Para los almacenes vamos a modificar el alias del campo ("Código") cambiándolo por "Código de Almacén". En primer lugar abrimos el módulo almacén en el área de facturación. En el menú Almacén -> Almacenes mostramos el listado de almacenes de nuestra base de datos. Podemos ver que el primer campo tiene el alias Código.

Los alias de los datos se especifican en las tablas.

Desde el módulo de sistema: administración, abrimos el módulo flfactalma (almacén), y a continuación la tabla Almacenes (almacenes.mtd). En el campo codalmacen cambiamos la propiedad alias.



Nuevo alias de un campo

Aceptamos todos los formularios. Podemos verificar el cambio abriendo de nuevo el formulario de almacenes y comprobando el alias nuevo.

2. Cambio de la longitud máxima de un campo.

Para las familias de artículos, el campo Código tiene una longitud máxima de 4 caracteres. Vamos a ampliar esta longitud hasta 6 caracteres. Dentro del módulo Almacén abrimos la tabla Familias (familias.mtd) y en el campo codigo cambiamos la propiedad length de 4 a 6:


```

<!DOCTYPE TMD>
<TMD>
  <name>familias</name>
  <!-- Listado de familias de artículos -->
  <alias>QT_TRANSLATE_NOOP("MetaData","Familias")</alias>

  <field>
    <name>codfamilia</name>
    <!-- Código de la familia -->
    <alias>QT_TRANSLATE_NOOP("MetaData","Código")</alias>
    <null>>false</null>
    <pk>>true</pk>
    <type>string</type>
    <length>6</length>

    <relation>
      <table>articulos</table>
      <field>codfamilia</field>
      <card>1M</card>
    </relation>
  </field>

  <field>
    <name>descripcion</name>
    <alias>QT_TRANSLATE_NOOP("MetaData","Descripción")</alias>
    <null>>false</null>
    <pk>>false</pk>
    <type>string</type>
    <length>100</length>
  </field>
</TMD>

```

Nueva longitud máxima de un campo

Podemos verificar el cambio abriendo el formulario de familias y comprobando que efectivamente el código admite ahora hasta 6 caracteres.

3. Cambios en el diseño de los formularios

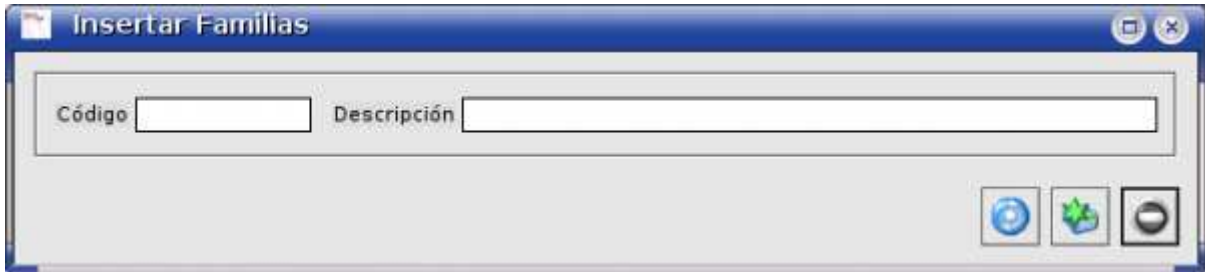
A la hora de trabajar con formularios vamos a utilizar la herramienta QT Designer. Tal como vimos, cuando editamos un fichero con extensión .ui en el módulo de sistema, el editor que aparece es QT Designer.

Algunos aspectos importantes acerca de QT Designer:

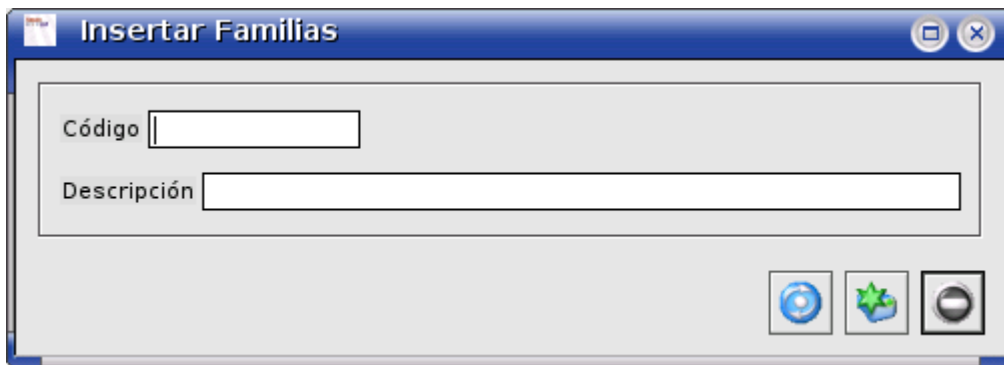
- Los componentes de un formulario pueden cambiarse de posición pulsando sobre ellos con el ratón y arrastrando
- Los componentes pueden agruparse en layouts, utilizando los botones correspondientes (menú Window / Toolbars / Layout)

- Todas las propiedades de los componentes están en la paleta de propiedades (menú window / views / Property Editor)

Vamos a utilizar este editor para cambiar el aspecto del formulario de familias:



Formulario de familias antes del cambio



Formulario de familias después del cambio

Para ello abriremos el fichero familias.ui dentro del módulo de Almacén y procederemos a editarlo según las figuras anteriores.

Cambios en el modelo de datos y formularios

Continuando con la personalización de tablas y formularios, vamos a realizar algunas modificaciones importantes en el formulario y tabla de países (módulo Principal de Facturación, menú Tablas Generales -> Países)

El objetivo del ejercicio es ampliar la información sobre los países que almacenamos en nuestra base de datos. Vamos a añadir los siguientes campos:

- Zona comercial. Un valor a elegir entre Europa, EEUU, Asia, Latinoamérica
- Divisa oficial. Se podrá obtener de la tabla Divisas
- Capital
- Habitantes
- Renta per cápita

El formulario actual de países tiene este aspecto:

Formulario de países antes de los cambios

Nuestro objetivo es crear un formulario de dos pestañas (General y Datos) con el aspecto siguiente:

Pestaña General del nuevo formulario de países

Pestaña Datos del nuevo formulario de países

1. Modificar el modelo de datos

Debemos añadir los nuevos campos a la tabla de países. Abrimos la tabla e insertamos el código siguiente:

```
<field>

  <name>zonacomercial</name>

  <null>>true</null>

  <pk>>false</pk>

  <optionslist>Europa,EEUU,Asia,Latinoamérica</optionslist>

  <default>Europa</default>

  <type>string</type>

  <length>20</length>

</field>

<field>

  <name>coddivisa</name>

  <alias>QT_TRANSLATE_NOOP("MetaData","Divisa oficial")</alias>

  <null>>true</null>

  <pk>>false</pk>

  <type>string</type>

  <length>3</length>

  <relation>

    <table>divisas</table>

    <field>coddivisa</field>

    <card>M1</card>

  </relation>
```

```
</field>
```

```
<field>
```

```
<name>capital</name>
```

```
<alias>QT_TRANSLATE_NOOP("MetaData","Capital")</alias>
```

```
<null>>true</null>
```

```
<pk>>false</pk>
```

```
<type>string</type>
```

```
<length>40</length>
```

```
</field>
```

```
<field>
```

```
<name>habitantes</name>
```

```
<alias>QT_TRANSLATE_NOOP("MetaData","Habitantes")</alias>
```

```
<null>>true</null>
```

```
<pk>>false</pk>
```

```
<type>double</type>
```

```
<partI>10</partI>
```

```
<partD>0</partD>
```

```
</field>
```

```
<field>
```

```
<name>rentapercapita</name>
```

```
<alias>QT_TRANSLATE_NOOP("MetaData","Renta per Cápita")</alias>
```

```
<null>>true</null>
```

```
<pk>>false</pk>
```

```
<type>double</type>
```

```
<partI>6</partI>
```

```
<partD>0</partD>
```

```
</field>
```

Características comunes de los nuevos campos

- Hemos establecido la propiedad null a true en todos los campos. Con ello permitiremos que dichos campos permanezcan vacíos.
- La propiedad pk (clave primaria) forzosamente ha de ser false porque no puede haber más de un campo clave primaria en la tabla, en este caso es el campo codpais.
- La propiedad alias establece el nombre del campo de cara a la interfaz de usuario; es el texto que aparece en los formularios y las cabeceras de campo de las tablas. Siempre pondremos QT_TRANSLATE_NOOP("MetaData", "alias del campo"). Esta función es necesaria para realizar las traducciones de textos a distintos idiomas.

Características específicas de los campos:

- zonacomercial. La propiedad optionslist permite establecer la lista de valores que aparecerán en el desplegable del formulario. Los valores se separan por comas. La propiedad default establecida a Europa indica que éste es el valor por defecto para los nuevos registros. En la propiedad type vemos que se trata de un string -cadena de caracteres-, y en la propiedad length que su longitud máxima es de 20 caracteres.
- coddivisa. Este campo almacena el código de la divisa del país. Vemos que está relacionado con la tabla divisas (propiedad table) mediante el campo coddivisa (propiedad field) de dicha tabla. El valor M1 de la propiedad card establece que puede haber varios países (M) para cada divisa (1).
- capital. Se trata de un campo sencillo de tipo string y 40 caracteres de longitud máxima
- habitantes. Este es un campo numérico que debe almacenar números grandes. Hemos optado por el tipo double. Las propiedades partI y partD indican la longitud de la parte izquierda (entera) y derecha (decimal) del número respectivamente.
- rentapercapita. Similar a habitantes

Para terminar, debemos modificar la tabla divisas para incluir la relación establecida con el campo coddivisa de la tabla de países. El código es el siguiente:

```
<relation>
```

```
<table>países</table>
```

```
<field>coddivisa</field>
```

```
<card>1M</card>
```

```
</relation>
```

Fijémonos en que ahora la propiedad card toma el valor contrario (1M: una divisa, varios países)

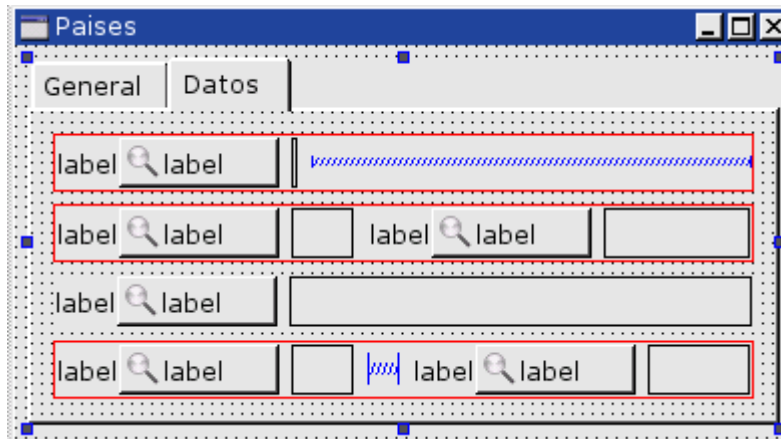
2. Modificar el formulario

Vamos a abrir el formulario de países en QT Designer para modificarlo según las figuras 7 y 8

Una vez abierta la aplicación, vamos a abrir las paletas de herramientas (menú Window / Views / Toolbox) y propiedades (menú Window / Views / Property Editor). De la paleta de herramientas seleccionaremos los componentes a insertar en el formulario. En la paleta de propiedades editaremos las mismas.

Pasos:

- Antes de comenzar debemos romper los layouts
- Las pestañas forman parte de un control tipo TabWidget. Insertamos el control desde la paleta de herramientas y establecemos los nombres de ambas pestañas (General y Datos). Esto puede hacerse pulsando el botón derecho del ratón sobre el control y seleccionando Edit Page Title
- Movemos los campos actuales del formulario hasta la pestaña General
- Activamos la pestaña Datos
- En la paleta de herramientas seleccionamos Database y FLFieldDB para insertar el primer campo, Zona Comercial.
- Establecemos la propiedad name del nuevo campo a fdbZonaComercial.
- Establecemos la propiedad fieldName del nuevo campo a zonacomercial. Este es el nombre del campo en la tabla.
- Repetimos los pasos anteriores para el resto de campos.
- De los campos anteriores, utilizamos uno que no está en la tabla de países. Es el que muestra la descripción de la divisa. Abanq permite mostrar este valor que se encuentra en otra tabla -la tabla divisas- siempre que exista una relación entre ambas tablas. Para este campo las propiedades son fieldName con valor descripcion, tableName con valor divisas, foreignField con valor coddivisa y fielRelation con valor coddivisa.
- Damos formato al formulario mediante controles spacer y layouts, y modificando las propiedades de tamaño de los campos:



Modificando el formulario de países en QT Designer



Tutorial. Programación en Abanq (II). Acciones

En el capítulo anterior de esta serie de artículos sobre programación en Abanq vimos cómo realizar pequeñas modificaciones sobre módulos ya existentes. Básicamente eran modificaciones sobre tablas y formularios en las que añadíamos campos o modificábamos sus propiedades.

En el presente artículo describiremos la forma de realizar lo que Abanq denomina acciones completas. Esto quiere decir que aprenderemos a crear nuestras propias tablas y formularios, y a relacionarlos con otros elementos ya existentes.

Además, dotaremos a nuestros formularios de funcionalidades añadidas mediante la programación de scripts, veremos la estructura del lenguaje QSA y usaremos las clases más importantes que Abanq pone a disposición de los programadores.

Para terminar, confeccionaremos un informe sencillo para obtener un resumen bien presentado de los datos contenidos en las nuevas tablas.

Antes de comenzar

Realizaremos todos los ejemplos prácticos sobre el módulo Principal del área Facturación. Por ello es necesario acceder a una base de datos con este módulo cargado, tal y como hicimos en el artículo anterior.

Si queremos disponer de una base de datos nueva para realizar estos ejemplos, basta con especificar el nuevo nombre de base de datos al arrancar Abanq, y cargar a continuación el módulo Principal de Facturación.

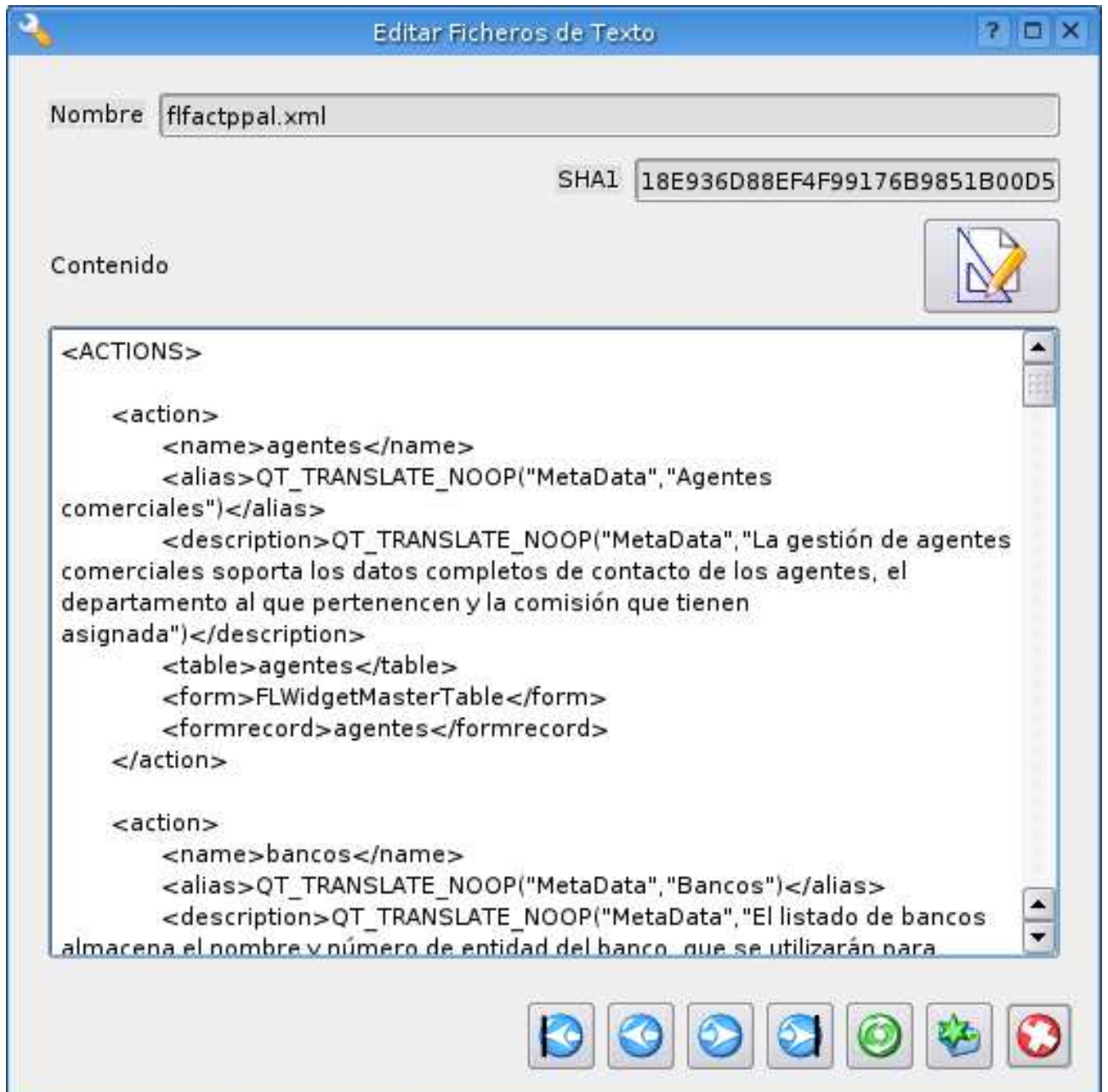
Las acciones en Abanq

Como vimos en el artículo anterior, el motor de Abanq lee e interpreta los metadatos (información sobre tablas, formularios, scripts, etc.) que el Sistema Gestor de Base de Datos (SGDB) le proporciona. Estos metadatos se organizan en lo que llamamos módulos. Cada módulo agrupa un conjunto de metadatos que implementan una funcionalidad concreta (facturación, almacén, etc.).

Vamos a ver ahora la estructura interna de estos módulos. El fichero de metadatos que define esta estructura es el fichero de acciones, y su nombre es `id_modulo.xml`, donde `id_modulo` es el identificador que cada módulo tiene asignado en la tabla Módulos del módulo de Administración.

Tomaremos como ejemplo el módulo Principal del área de Facturación. Su código es `ffactppal`, por tanto el fichero de acciones será `ffactppal.xml`. Lo visualizaremos de la forma ya descrita en el artículo anterior (Área de Sistema -> Módulo de Administración -> Módulos ->

Editar flfactppal -> Seleccionar flfactppal.xml -> Ver registro). Aparecerá una ventana similar a la de la figura siguiente:



Fichero de acciones flfactppal.xml

Vemos que el fichero de acciones está compuesto por nodos <action>. Cada uno de estos nodos define una acción. Una acción es una unidad funcional concreta, que agrupa una serie de elementos necesarios para su correcto funcionamiento. Cada acción puede contener los nombres de una tabla, un formulario maestro, un formulario de edición, un script de formulario maestro y un script de formulario edición.

Por ejemplo, la acción clientes agrupa las funcionalidades de la gestión de clientes: la tabla donde se almacenan sus datos, los formularios utilizados para acceder a dichos datos, los scripts que gestionan los formularios, etc.

Las etiquetas que conforman cada acción son:

- <name> Nombre de la acción.
- <alias> Alias o título de la acción
- <description> Descripción de la funcionalidad acción
- <table> Nombre de la tabla asociada a la acción.
- <form> Nombre del formulario maestro asociado a la acción.
- <formrecord> Nombre del formulario de edición asociado a la acción.
- <scriptform> Nombre del script asociado al formulario maestro.
- <scriptformrecord> Nombre del script asociado al formulario edición.

Veremos qué significa cada una de estas etiquetas a medida que progresemos en el desarrollo de nuestro ejemplo.

Creando nuestra acción

Vamos a suponer que nos es necesario llevar un control de los empleados de nuestra empresa. Para ello es necesario recoger los datos personales de cada uno de ellos. Cada empleado está asociado a un departamento de la empresa, y queremos poder emitir informes con un listado de los empleados de alta agrupados por departamento. Para realizar esta ampliación, el primer paso será crear la acción empleados (la acción departamentos ya existe en el módulo principal). Editaremos el fichero de acciones `ffactppal.xml`, añadiendo un nuevo nodo <action> tal y como aparece en la figura siguiente (recuerda que para editar el fichero debes pulsar Editar Registro y seguidamente Editar Fichero).

Acción empleados en flfactppal.xml

En la nueva acción indicamos que la tabla de empleados debe definirse en el fichero empleados.mtd, y que los formularios maestro y de edición (veremos qué significan estos términos más adelante) son `i_master.ui` y `empleados.ui`. Por ahora no indicaremos los nombres de los scripts. Una vez guardados los cambios (Aceptar -> Aceptar cambios del fichero -> Aceptar cambios del módulo) ya tenemos nuestra acción creada.

Para terminar este apartado, vamos a crear un acceso desde la ventana principal del módulo de Facturación, de manera que podamos acceder a la gestión de empleados desde una opción de menú o desde un botón de la barra de herramientas.

Las ventanas principales de cada módulo son formularios cuyo nombre sigue el esquema `id_modulo.ui`. En nuestro caso, deberemos editar el formulario `flfactppal.ui`. Una vez abierto el formulario mediante `QtDesigner`, incluiremos una nueva opción Empleados en el menú Tablas Generales.



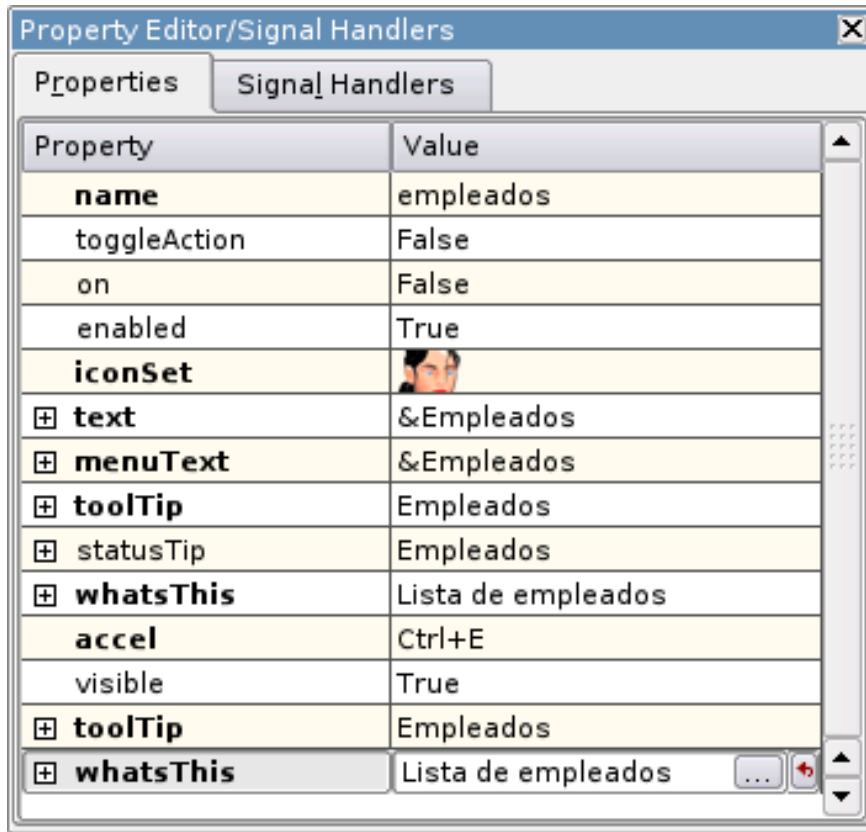
Menú Tablas Generales de flfactppal.ui

Al crear esta nueva opción y pulsar Intro hemos creado una nueva acción en la ventana principal del módulo. Para editar esta acción debemos abrir el editor de acciones (opción de menú Window -> Views -> Action Editor). Vemos que se ha creado una acción cuyo nombre por defecto es `tablas_generalesEmpleadosAction`.



Nueva acción creada en el Action Editor

Cambiaremos este nombre y el resto de propiedades de la acción en la ventana de propiedades (Property Editor) como muestra la figura a continuación.



Property Editor de QtDesigner

Como podemos ver, hemos añadido un icono y especificado las teclas de acceso rápido a la acción. Para incluir un botón en la barra de herramientas arrastraremos el icono desde el Action Editor hasta la posición de la barra en la que deseemos ubicar el botón.

Cada acción de la ventana principal del módulo debe corresponderse con una acción del fichero de acciones. Esta correspondencia se establece haciendo coincidir la propiedad name de la acción de la ventana con la etiqueta <name> del correspondiente nodo action del fichero de acciones. En nuestro caso este valor es empleados.

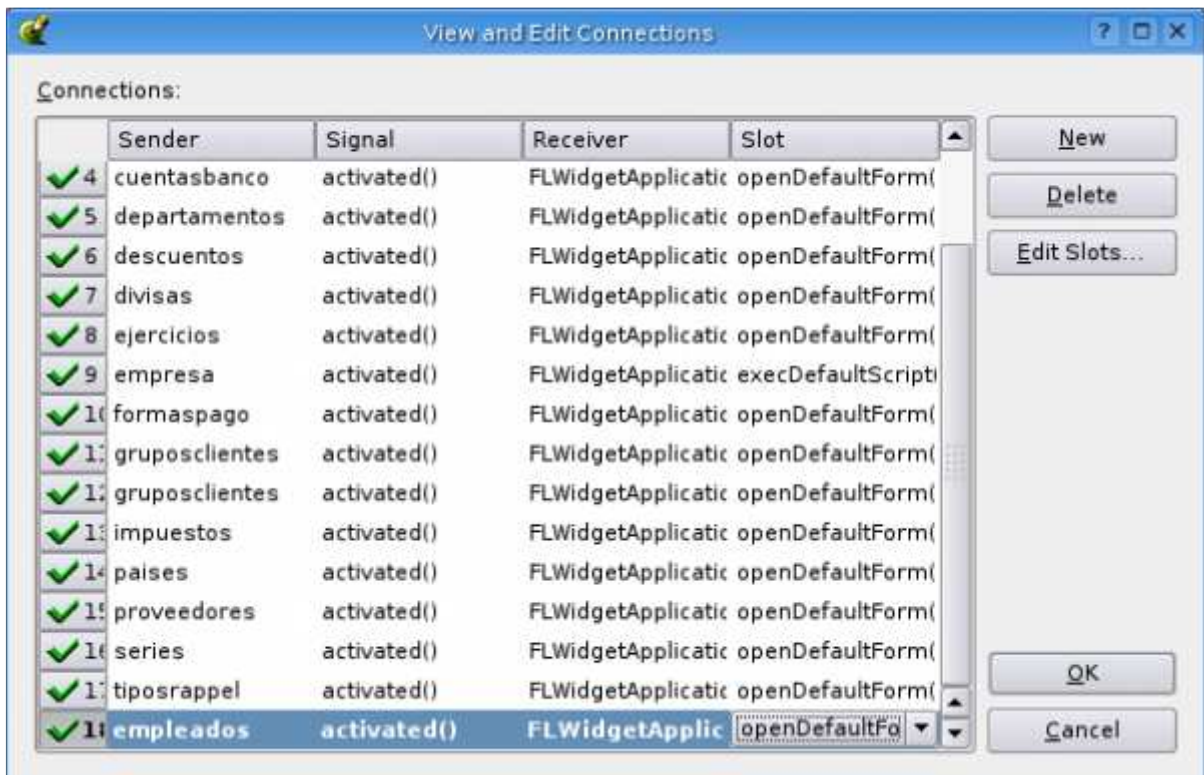
Nos falta por último determinar qué sucederá cuando se seleccione la acción empleados en la ventana principal del módulo. Lo que haremos será abrir el formulario por defecto asociado a la acción. Para ello, en el Action Editor, seleccionamos la acción empleados y pulsamos el botón de conexiones:

En la ventana View and Edit Connections nos aparece la lista de conexiones establecidas. La última entrada de la lista nos propone conectar la acción empleados con el objeto FLWidgetApplication.

Como veremos más adelante, es muy común en la arquitectura de Abanq establecer conexiones entre objetos. Estas conexiones determinan que

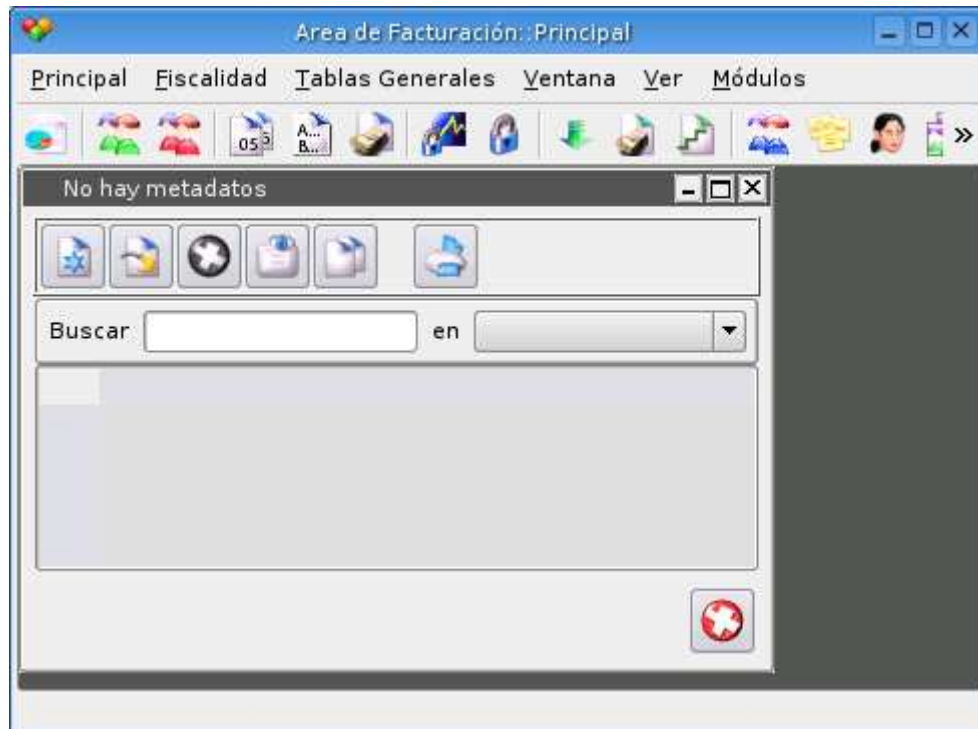
cuando un objeto -el emisor- envíe una determinada señal, otro objeto -el receptor- ejecutará un determinado método o slot.

En nuestro caso, deseamos que cuando el objeto acción empleados se active (emita la señal activated), el objeto ventana principal de la aplicación Abanq (FLWidgetApplication) muestre el formulario por defecto de la acción (slot openDefaultForm). La conexión debe quedar por tanto tal y como describe la figura siguiente.



Ventana de conexiones

Una vez establecida la conexión pulsamos Ok y guardamos los cambios en `ffactppal.ui`. Ya podemos probar nuestra acción. Si accedemos al módulo Principal del área de Facturación y pulsamos el botón o la opción de menú Empleados, Abanq nos mostrará una ventana como la de la figura siguiente.



Formulario de la acción empleados

El mensaje 'No hay metadatos' hace referencia a que no hemos definido todavía la tabla empleados. En el siguiente punto veremos cómo hacer esto.

Creando las tablas

Las tablas se definen dentro del directorio tables de los módulos, y tienen la extensión mtd.

El nombre del fichero de la tabla, según nuestra nomenclatura, debe ser empleados.mtd. Insertamos un registro en el módulo, igual que hicimos con la creación del fichero de acciones, para la nueva tabla con el nombre anterior y el contenido siguiente:

```
<!DOCTYPE TMD> <TMD>
```

```
<name>empleados</name>
```

```
<alias>QT_TRANSLATE_NOOP("MetaData","Empleados")</alias>
```

```
<field>
```

```
<name>codempleado</name>
```

```
<alias>QT_TRANSLATE_NOOP("MetaData","Código")</alias>
```



```

<null>>false</null>
<pk>>true</pk>
<type>string</type>
<length>18</length>
</field>

```

```

<field>
  <name>coddepartamento</name>
  <alias>QT_TRANSLATE_NOOP("MetaData","Departamento")</alias>
  <null>>false</null>
  <pk>>false</pk>
  <type>string</type>
  <length>6</length>

```

```

<relation>
  <table>departamentos</table>
  <field>coddepartamento</field>
  <card>M1</card>
</relation>

```

```

</field>

```

```

<field>
  <name>nombrecompleto</name>
  <alias>QT_TRANSLATE_NOOP("MetaData","Nombre
Completo")</alias>

```

```

<null>>false</null>

<pk>>false</pk>    <!--Código de departamento-->

<type>string</type>

<length>150</length>

</field>

<field>

  <name>titulacion</name>

  <alias>QT_TRANSLATE_NOOP("MetaData","Titulación")</alias>

  <null>>false</null>

  <pk>>false</pk>

  <type>string</type>

  <length>20</length>

  <optionslist>Ninguna,Ed.    Primaria,Ed.    Secundaria,FP,Título
Universitario</optionslist>

  <default>OFICINA</default>

</field>

<field>

  <name>fechaalta</name>

  <alias>QT_TRANSLATE_NOOP("MetaData","Fecha de alta")</alias>

  <null>>false</null>

  <pk>>false</pk>

  <type>date</type>

</field>

```

```

<field>
  <name>debaja</name>
  <alias>QT_TRANSLATE_NOOP("MetaData","De baja")</alias>
  <null>>false</null>
  <pk>>false</pk>
  <type>bool</type>
  <default>>false</default>
</field>

```

```

<field>
  <name>sueldobruto</name>
  <alias>QT_TRANSLATE_NOOP("MetaData","Bruto")</alias>
  <null>>true</null>
  <pk>>false</pk>
  <type>double</type>
  <partI>4</partI>
  <partD>2</partD>
</field>

```

```

<field>
  <name>impuestos</name>
  <alias>QT_TRANSLATE_NOOP("MetaData","% Impuestos")</alias>
  <null>>true</null>
  <pk>>false</pk>

```

```

    <type>double</type>

    <partI>2</partI>

    <partD>2</partD>
</field>

<field>

    <name>sueldoneto</name>

    <alias>QT_TRANSLATE_NOOP("MetaData","Neto")</alias>

    <null>true</null>

    <pk>false</pk>

    <type>double</type>

    <partI>4</partI>

    <partD>2</partD>
</field>

<field>

    <name>causabaja</name>

    <alias>QT_TRANSLATE_NOOP("MetaData","Causa de la
baja")</alias>

    <null>true</null>

    <pk>false</pk>

    <type>stringlist</type>
</field>
</TMD>

```

El campo seccion lo hemos definido como una lista de opciones con varias secciones de la empresa a modo de ejemplo. Es muy recomendable

reparar la estructura xml de esta tabla y comprender bien todos los campos y etiquetas. Podemos repasar las especificaciones de las tablas.

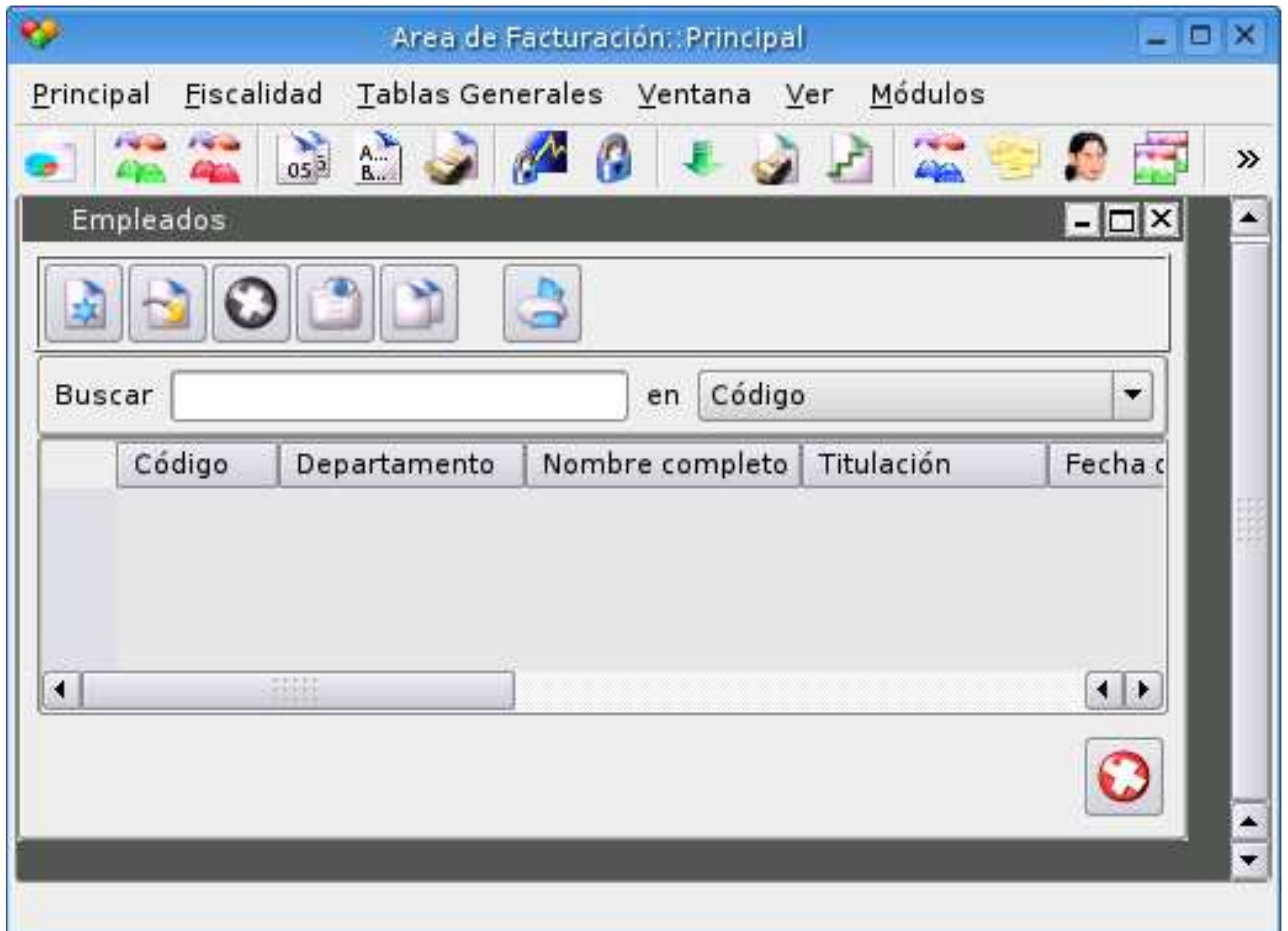
Pulsamos Aceptar cambios y cerrar formulario.

Creando los formularios

Abanq usa dos tipos principales de formularios: El formulario maestro y el formulario de edición. Veamos cómo es y cómo se crea cada uno de ellos.

El formulario maestro es el que muestra en una lista un subconjunto de los registros de la tabla, ofreciendo al usuario la posibilidad de realizar ciertas operaciones (crear, modificar, borrar, etc.) sobre el registro seleccionado. Sirve además para localizar un determinado registro realizando una búsqueda por cualquiera de los campos de la tabla. El formulario maestro asociado a una acción está determinado por el valor de la etiqueta `<form>` del nodo `<action>`.

En el caso de la acción de empleados, hemos nombrado a este formulario `i_master`. Éste no es un nombre al azar, sino el de un formulario predeterminado, ya incluido en Abanq. De hecho, podemos volver a pulsar sobre la acción de empleados y veremos que ya nos aparece un formulario Empleados con los campos de la tabla



Formulario maestro de empleados

En el caso de que quisiéramos crear un formulario maestro personalizado, procederíamos de forma similar a como se describe a continuación para el formulario de edición.

El formulario de edición es el que ofrece al usuario la posibilidad de visualizar, crear o modificar los datos de un determinado registro de la tabla. El formulario de edición asociado a una acción está determinado por el valor de la etiqueta <formrecord> del nodo <action>.

Vamos a crear el formulario de edición correspondiente a la acción empleados. Lo habíamos llamado empleados, luego el fichero deberá llamarse empleados.ui. Siguiendo el mismo procedimiento que para crear la tabla empleados.mtd, crearemos un nuevo registro en el módulo flfactppal. Al tener el fichero la extensión .ui, Abanq lanzará QtDesigner en lugar del editor de textos. En el cuadro de diálogo New File seleccionaremos el tipo Widget.

Nuestro formulario de edición deberá ser similar al de la siguiente figura.

The screenshot shows a window titled 'Empleados' with three main sections:

- Datos personales:** Contains fields 1, 2, and 3. Field 1 is a date field, field 2 is a text field, and field 3 is a text field.
- Departamento:** Contains fields 4 and 5. Field 4 is a dropdown menu, and field 5 is a text field.
- Condiciones del contrato:** Contains fields 6, 7, 8, 9, 10, and 11. Fields 6, 7, 8, 9, and 10 are date fields, and field 11 is a text field.

Formulario empleados.ui

Hemos agrupado los controles FLFieldDB en tres controles groupBox para mayor claridad, aunque esto no es necesario.

Las principales propiedades de cada campo son las siguientes:

- | | name | fieldName | tableName | foreignField | fieldRelation |
|----|--------------------|-----------|-----------------|-----------------|-----------------|
| 1: | fdbCodEmpleado | | codempleado | | |
| 2: | fdbTitulacion | | titulacion | | |
| 3: | fdbNombreCompleto | | nombrecompleto | | |
| 4: | fdbCodDepartamento | | coddepartamento | | |
| 5: | fdbNomDepartamento | nombre | departamentos | coddepartamento | coddepartamento |
| 6: | fdbFechaAlta | | fechaalta | | |

- 7: fdbDeBaja debaja
- 8: fdbSueldoBruto sueldobruto
- 9: fdbImpuestos impuestos
- 10: fdbSueldoNeto sueldoneto
- 11: fdbCausaBaja causabaja

Hemos optado por establecer los nombres de los controles como fdb + NombreCampo. Aunque la propiedad name puede tomar cualquier valor, en este ejemplo es recomendable mantener los de la tabla anterior, para que los scripts que crearemos a continuación no tengan que ser retocados.

En el campo 5 vamos a mostrar el nombre del departamento al que pertenece el empleado. Como este campo no pertenece a la tabla de empleados sino a la de departamentos, debemos establecer el resto de propiedades tal y como ya describimos en el artículo anterior.

Una vez guardado el formulario ya podemos probarlo. Si abrimos la acción empleados aparecerá el formulario maestro (i_master.ui). Si pulsamos ahora sobre el botón Insertar Registro, se abrirá nuestro formulario empleados.ui con el aspecto de la siguiente figura.

Formulario de edición de empleados

Podemos apreciar cómo dependiendo del tipo de campo, el control `FLFieldDB` correspondiente toma el aspecto adecuado para mostrar su valor.

Llegados a este punto ya podríamos comenzar a trabajar con esta ventana. Los botones de aceptar y cancelar, así como las validaciones de datos de cada campo están plenamente operativos, de forma que si establecemos todos los campos requeridos (marcados como `<null>false</null>` en `empleados.mtd`) y pulsamos aceptar habremos creado nuestro primer empleado.

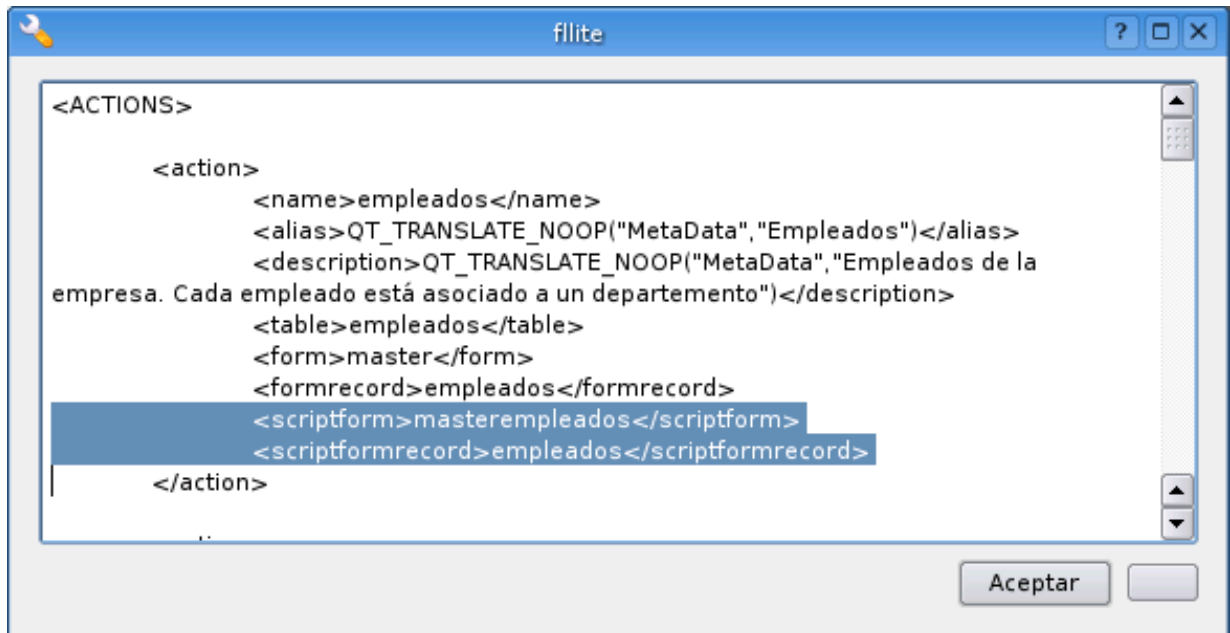
Creando los scripts

Vamos a dotar a nuestros formularios de una mayor funcionalidad asociándoles un script. No vamos a hacer una descripción demasiado formal del lenguaje QSA usado en Abanq, simplemente diremos que es muy similar a JavaScript e incluiremos comentarios en el código de los ejemplos que aclaren su funcionamiento.

Algo que sí es importante recalcar es que, al margen de las clases y funciones que QSA ofrece al programador de scripts, el motor de Abanq también publica una serie de clases que nos permiten acceder a ciertos objetos internos del motor. Como veremos, esto da una gran potencia a los scripts, ya que permite hacer muchas operaciones que de otra manera sólo podrían conseguirse recompilando el código C++ del motor.

Podemos consultar la documentación de QSA en <http://doc.trolltech.com/ksa-1.2/index.html>, y la de la interfaz de objetos de Abanq en el apartado de documentación de esta web.

Lo primero será añadir las referencias a los scripts en el correspondiente nodo `<action>` del fichero de acciones.



Añadiendo las referencias a los scripts en la acción empleados

Hemos asociado al formulario maestro el script `masterempleados.qs`, y al formulario de edición el script `empleados.qs`.

Crearemos primero el script asociado al formulario de edición, `empleados.qs`. A continuación mostramos el código completo del scrip que contiene algunas de las principales funciones que son llamadas automáticamente por el motor de Abanq en ciertos momentos de la ejecución del formulario.

El sistema de clases y herencias escapa al ámbito de este tutorial y se verá con posterioridad.

```

/*****
****

```

empleados.qs - description

begin : lun jul 1 2007

copyright : (C) 2007 by InfoSiAL S.L.

email : mail@infosial.com

```

*****/

```

```

/*****
****

```

```

* *

* This program is free software; you can redistribute it and/or modify *
* it under the terms of the GNU General Public License as published by *
* the Free Software Foundation; either version 2 of the License, or *
* (at your option) any later version. *

* *

*****
*****/

/** @file */

/** @class_declaracion interna */

////////////////////////////////////

/// DECLARACION //////////////////////////////////////

////////////////////////////////////

////////////////////////////////////

/// INTERNA //////////////////////////////////////

class interna {

var ctx:Object;

function interna( context ) { this.ctx = context; }

function init() { this.ctx.interna_init(); }

function validateForm() { return this.ctx.interna_validateForm(); }

    function        calculateField(fN:String):String        {        return
this.ctx.interna_calculateField(fN); }

}

/// INTERNA //////////////////////////////////////

////////////////////////////////////

/** @class_declaracion oficial */

```

```

////////////////////////////////////
//// OFICIAL //////////////////////////////////////

class oficial extends interna {
    function oficial( context ) { interna( context ); }
    function bufferChanged(fN:String) {
        return this.ctx.oficial_bufferChanged(fN);
    }
}

//// OFICIAL //////////////////////////////////////

////////////////////////////////////

/** @class_declaration head */

////////////////////////////////////

//// DESARROLLO //////////////////////////////////////

class head extends oficial {
    function head( context ) { oficial ( context ); }
}

//// DESARROLLO //////////////////////////////////////

////////////////////////////////////

/** @class_declaration ifaceCtx */

////////////////////////////////////

//// INTERFACE //////////////////////////////////////

class ifaceCtx extends head {
    function ifaceCtx( context ) { head( context ); }
}

const iface = new ifaceCtx( this );

```

```

/// INTERFACE //////////////////////////////////////
/// //////////////////////////////////////
/// @class_definition interna */
/// //////////////////////////////////////
/// DEFINICION //////////////////////////////////////
/// //////////////////////////////////////
/// //////////////////////////////////////
/// INTERNA //////////////////////////////////////

/* Función que se llama al iniciar el formulario

Conecta la señal bufferchanged (cambio en el buffer,cambio en un campo)
con el slot o función bufferChanged

Si se la llama en modo alta inhabilitará el campo 'Causa de la baja'

*/

function interna_init()
{
    var cursor:FLSqlCursor = this.cursor(); // Objeto FLSqlCursor asociado al
formulario

    connect(this.cursor(),          "bufferChanged(QString)",          this,
"iface.bufferChanged");

    if (cursor.modeAccess() == cursor.Insert)

        this.child("fdbCausaBaja").setDisabled(true);
}

/* Función que calcula el valor de un campo. En este caso el sueldo neto
cuando a partir del bruto y los impuestos

fN: Nombre del campo a calcular

Resultado: Valor del campo

```

```

*/

function interna_calculateField(fN)
{
    var cursor:FLSqlCursor = this.cursor(); // Objeto FLSqlCursor asociado al
    formulario

    var valor:String = "";

    switch(fN) {
        // El sueldo neto será el sueldo bruto tras descontarle los impuestos
        case "sueldoneto":
            var sueldoBruto = parseFloat(cursor.valueBuffer("sueldobruto"));
            var impuestos = parseFloat(cursor.valueBuffer("impuestos"));
            valor = sueldoBruto * (100 - impuestos) / 100;

            break;
        }

    return valor;
}

/* Función que se llama al pulsar el botón aceptar y que decide si los datos
del formulario son válidos.

Si los datos no son válidos, los datos no se guardarán y el formulario de
edición no se cerrará.

Resultado: true si los datos son válidos, false en caso contrario

*/

function validateForm()
{

```

```

var cursor = this.cursor();

var util:FLUtil = new FLUtil();

// La fecha de alta no puede superar la fecha actual

var hoy = new Date();

var fechaAlta = cursor.valueBuffer("fechaalta");

if (util.daysTo(fechaAlta, hoy) < 0) {

    MessageBox.warning(util.translate("scripts", "La fecha de alta no
puede ser superior a la fecha actual"),

        MessageBox.Ok, MessageBox.NoButton);

    return false;

}

return true;

}

//// INTERNA //////////////////////////////////////

////////////////////////////////////

/** @class_definition oficial */

////////////////////////////////////

//// OFICIAL //////////////////////////////////////

function oficial_bufferChanged(fN:String)

{

    switch (fN) {

        case "sueldobruto":

        case "impuestos":

```

```

        this.child("fdbSueldoNeto").setValue(this.iface.calculateField("sueldo
neto"));

        break;
    }
}

//// OFICIAL //////////////////////////////////////

////////////////////////////////////

/** @class_definition head */

////////////////////////////////////

//// DESARROLLO //////////////////////////////////////

//// DESARROLLO //////////////////////////////////////

////////////////////////////////////

```

Una vez creado el fichero podemos probar el script ejecutando el formulario y comprobar que cada una de estas cuatro funciones funciona correctamente.

Vamos a mejorar un poco más nuestro script. Por un lado, vamos a habilitar nuestro control Causa de la baja cuando el usuario active el check De baja. Para ello ampliaremos la función `bufferChanged` para que realice esto. El código será:

...

```

        case "debaja" : // Si se activa el control, el campo Causa de la baja se
habilitará

            if (cursor.valueBuffer("debaja") == true)

                form.child("fdbCausaBaja").setDisabled(false);

            else {

                form.child("fdbCausaBaja").setValue("");

                form.child("fdbCausaBaja").setDisabled(true);

            }

```



```
break;
```

```
...
```

Ya tenemos plenamente operativo nuestro primer script. Vamos ahora con el script asociado al formulario maestro, masterempleados.qs. Este script conectará el botón Imprimir del formulario con la función que lanzará el informe de empleados que realizaremos en el siguiente apartado. Su código es el siguiente:

```
/*
****

```

```
masterempleados.qs - description
```

```
-----
```

```
begin : lun jul 1 2007
```

```
copyright : (C) 2007 by InfoSiAL S.L.
```

```
email : mail@infosial.com
```

```
*****
*****/
```

```
/*
****

```

```
* *
```

```
* This program is free software; you can redistribute it and/or modify *
```

```
* it under the terms of the GNU General Public License as published by *
```

```
* the Free Software Foundation; either version 2 of the License, or *
```

```
* (at your option) any later version. *
```

```
* *
```

```
*****
*****/
```

```
/** @file */
```

```
/** @class_declaracion interna */
```

```
////////////////////////////////////
```

```

//// DECLARACION //////////////////////////////////////////////////////////////////////
//
//
//
//// INTERNA //////////////////////////////////////////////////////////////////////
class interna {
    var ctx:Object;

    function interna( context ) { this.ctx = context; }

    function init() { this.ctx.interna_init(); }
}

//// INTERNA //////////////////////////////////////////////////////////////////////
//
//
/** @class_declaration oficial */
//
//// OFICIAL //////////////////////////////////////////////////////////////////////
class oficial extends interna {
    var btnImprimir:Object;

    function oficial( context ) { interna( context ); }

    function imprimir() {
        return this.ctx.oficial_imprimir();
    }
}

//// OFICIAL //////////////////////////////////////////////////////////////////////
//
//
/** @class_declaration head */
//

```

```

//// DESARROLLO //////////////////////////////////////
class head extends oficial {
function head( context ) { oficial ( context ); }
}

//// DESARROLLO //////////////////////////////////////
////////////////////////////////////
/** @class_declaration ifaceCtx */
////////////////////////////////////

//// INTERFACE //////////////////////////////////////
class ifaceCtx extends head {
function ifaceCtx( context ) { head( context ); }
}

const iface = new ifaceCtx( this );

//// INTERFACE //////////////////////////////////////
////////////////////////////////////
/** @class_definition interna */
////////////////////////////////////

//// DEFINICION //////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

//// INTERNA //////////////////////////////////////
function interna_init()
{
    connect(this.child("toolButtonPrint"), "clicked()", this, "iface.imprimir");
}

```

```

//// INTERNA //////////////////////////////////////
////////////////////////////////////
/** @class_definition oficial */
////////////////////////////////////
//// OFICIAL //////////////////////////////////////

function oficial_imprimir()
{
    var util = new FLUtil();

    var consulta = new FLSqlQuery("empleados"); //Objeto de consulta de
empleados

    if (!consulta.exec()) {

        MessageBox.critical(util.translate("scripts", "Falló la consulta"),
            MessageBox.Ok, MessageBox.NoButton);

        return;
    }

    var rptViewer = new FLReportViewer(); //Objeto visor de informes

    rptViewer.setReportTemplate("empleados"); // Establece la plantilla del
informe

    rptViewer.setReportData(consulta); // Establece los datos del informe

    rptViewer.renderReport(); // Mezcla plantilla y datos

    rptViewer.exec(); // Muestra el informe
}

//// OFICIAL //////////////////////////////////////
////////////////////////////////////
/** @class_definition head */

```

```

////////////////////////////////////
/// DESARROLLO //////////////////////////////////////
/// DESARROLLO //////////////////////////////////////
////////////////////////////////////

```

Si tratamos de ejecutar la función imprimir la aplicación fallará, dado que la consulta empleados.qry todavía no está creada.

Creando los informes

Para finalizar, vamos a crear un pequeño informe en el que recogeremos los datos de nuestros empleados organizados por departamentos. Para ello crearemos primero una consulta que extraiga dichos datos.

Crearemos el fichero empleados.qry con el siguiente contenido:

```

<!DOCTYPE QRY>

<QRY>

  <name>empleados</name>

  <tables>empleados,departamentos</tables>

  <group>

    <level>0</level>

    <field>departamentos.coddepartamento</field>

  </group>

  <select>

    departamentos.coddepartamento, departamentos.nombre,

    empleados.codempleado, empleados.nombrecompleto,

    empleados.sueldobruto, empleados.fechaalta

  </select>

  <from>

    departamentos INNER JOIN empleados

```

```
ON departamentos.coddepartamento = empleados.coddepartamento
```

```
</from>
```

```
<where>
```

```
empleados.debaja = false
```

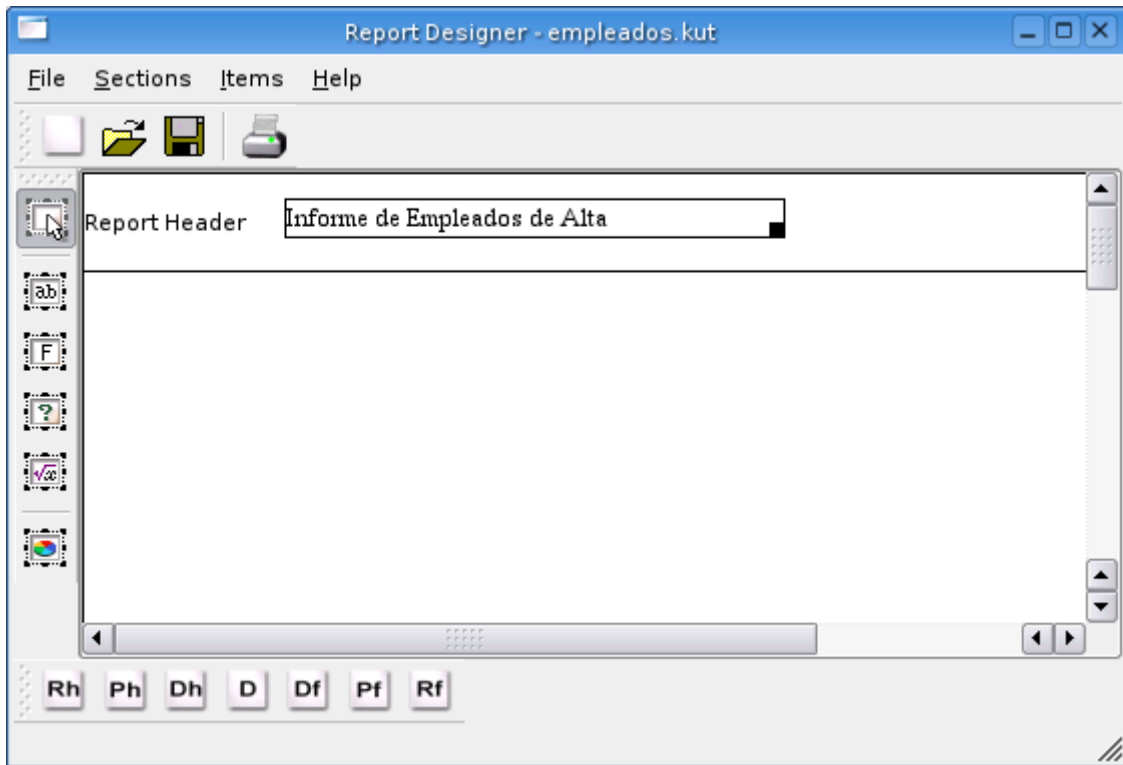
```
</where>
```

```
</QRY>
```

Una vez creada la consulta, el botón imprimir nos debe mostrar el visor de informes vacío. Esto es así porque todavía nos falta por crear el último elemento: La plantilla del informe. En la plantilla especificaremos la posición y formato de los campos, así como algunos campos especiales y de resumen.

Crearemos el informe empleados.kut. El editor que Abanq nos propone para este tipo de archivos es KuDesigner. Estableceremos el nombre de la plantilla como empleados y seleccionaremos el tamaño, orientación y márgenes del papel. Pusaremos Ok para acceder a la ventana de edición del informe.

Lo primero que haremos es crear el encabezado del informe. Para ello seleccionaremos Sections -> Report Header. Nos aparecerá una primera zona en el informe destinada al encabezado. Añadiremos una etiqueta para mostrar el título del informe. Seleccionaremos el icono Label de la barra de herramientas de la izquierda y pulsaremos sobre el punto del encabezado en el que deseemos ubicar la etiqueta. Pulsando con el botón derecho del ratón sobre la etiqueta creada editaremos sus propiedades.



Encabezado del informe de empleados

Por ahora nos preocuparemos únicamente de la estructura del informe, dejando la apariencia para más tarde. Las únicas propiedades que hemos cambiado son: Text = Informe de Empleados de Alta y Width = 250.

De la misma forma que hemos creado el encabezado de informe crearemos el detalle de nivel 0, que contendrá los datos de los departamentos. Para ello crearemos una sección Detail de nivel (level) 0. Una vez creada añadiremos dos campos Label y dos campos Field. La propiedad Field de los campos Field será departamentos.coddepartamento para el primero y departamentos.nombre para el segundo (Figura 15).

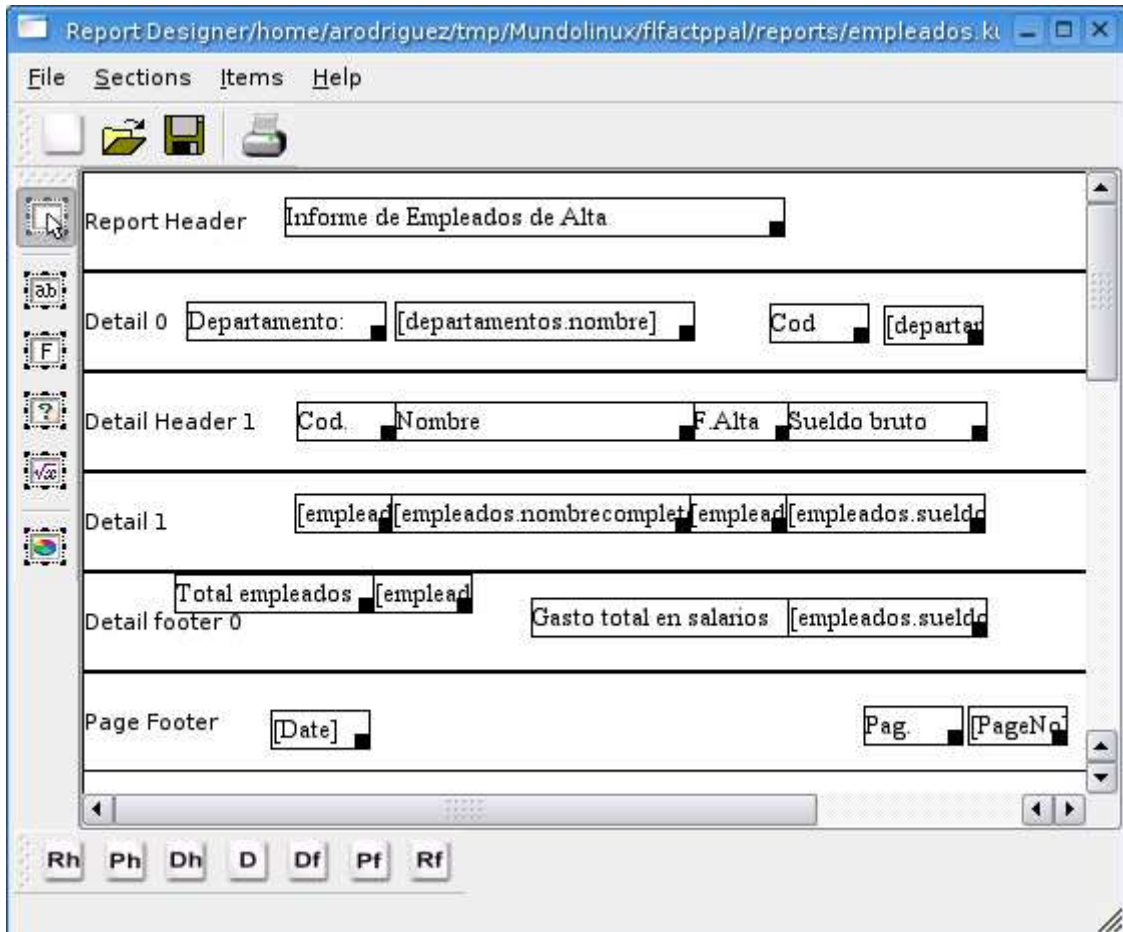
Crearemos ahora el encabezado de nivel 1 (Detail Header). En esta sección incluiremos una serie de campos Label concatenados, que actuará como cabecera para la lista de empleados de cada departamento.

La siguiente sección es el detalle de nivel 1 (Detail), con la información de los empleados (campos Field) en el mismo orden que en el encabezado de nivel 1. Modificaremos la propiedad DataType para los campos empleados.fechaalta y empleados.sueldobruto a los valores 3 (fecha) y 4 (moneda) respectivamente.

Introduciremos un pie de detalle de nivel 0 (Detail Footer) que mostrará un total de los empleados y del gasto en salarios de la empresa. Para ello introduciremos dos campos calculados, el primero con Field = empleados.codempleado y CalculationType = 0 (cuenta) y el segundo con Field = empleados.salariobruto y CalculationType = 1 (suma).

Por último cerraremos el informe con un pie de página (Page Footer) en el que mostraremos la fecha de generación del informe y el número de página. Cada uno de estos datos se consigue insertando un campo de tipo Special, la fecha con Type = 0 y el número de página con Type = 1.

El informe debe presentar un aspecto similar al de la siguiente figura.



Plantilla del informe de empleados

Una vez guardado ya podemos probar el informe. El visor de informes debe mostrarnos algo parecido a lo siguiente.

Visor de informes

Menú

Informe de Empleados de Alta

Departamento: Compras Cod: 000001

Cod.	Nombre	F. Alta	Sueldo bruto
000001	Francisco Javier Peláez Jimenez	02.08.200	2.000,00
000002	Luis Miguel Parra Martínez	17.07.200	1.800,00

Departamento: Ventas Cod: 000002

Cod.	Nombre	F. Alta	Sueldo bruto
000003	Juan Pelotas Campanas	01.07.200	2.200,00

Total empleados: 3

Gasto total en salarios: 6.000,00

Informe de empleados

Bien, nuestro informe funciona, pero todavía no estamos en condiciones de ir enseñándolo por ahí. Ahora debemos darle una buena presentación. Como este trabajo es cuestión de gustos, dejaremos que cada uno dé al informe la apariencia que prefiera.

Podemos obtener más información sobre las distintas propiedades de las secciones y campos de los informes en el apartado Documentación de <http://www.Abanq.org>

Como ejemplo de presentación podemos ver el informe que muestra la siguiente figura.

Visor de informes

Menú

Informe de Empleados de Alta

Departamento: 000001 - Compras

Cod.	Nombre	F.Alta	Sueldo bruto
000001	Francisco Javier Peláez Jimeno	02-08-2005	2.000,00
000002	Luis Miguel Parra Martínez	17-07-2005	1.800,00

Departamento: 000002 - Ventas

Cod.	Nombre	F.Alta	Sueldo bruto
000003	Juan Pelotas Campanas	01-07-2005	2.200,00

Total empleados: 3 Gasto total en salarios: 6.000,00

Ejemplo de formato de informe

ADEMPIERE

1 Vista General

1.1 Introducción a ADempiere Business Solution

ADempiere Business Solution es una sofisticada solución de negocios Open Source que se posiciona como una fuerte alternativa a los productos propietarios. La mayoría de las soluciones ERP disponibles en el mercado actualmente, proporcionan similar funcionalidad, y muchas organizaciones evalúan soluciones basándose en las capacidades funcionales medidas en un instante de tiempo en particular. Este enfoque es común, pero no es la metodología más apropiada para evaluar y seleccionar una solución de negocio a largo plazo. Este enfoque puede conducir a diferentes resultados cuando un producto es evaluado en diferentes momentos, a raíz de nuevas versiones que pueden ser liberadas al mercado.

El ciclo de vida de una solución ERP se estima generalmente en diez o incluso más años, y durante este período de tiempo la tecnología y los requisitos del negocio cambian. Así como la capacidad funcional de un producto es importante, es también muy importante tener en cuenta la tecnología en la que dicho producto está basado y la posibilidad que éste brinda para ser modificado y adaptado a las necesidades de la organización, las cuales se van renovando a medida que las reglas del negocio van cambiando. También es crítico asegurarse que los cambios esenciales realizados no comprometan la posibilidad de migrar a futuras versiones del producto, y preserven la integridad de las modificaciones específicas efectuadas para su negocio.

El verdadero poder de ADempiere queda demostrado cuando, siendo funcionalmente rico, tiene la posibilidad de incorporar los cambios específicos de su negocio y preservarlos en la liberación de nuevas versiones.

1.2 Fortalezas de ADempiere

1.2.1 Flexibilidad

- 1) ADempiere adopta estándares abiertos, lo cual permite:
 - La estandarización, estabilidad e interoperabilidad de sistemas
 - Descripciones de datos y comportamientos claros, públicos y visibles
- 2) Independencia de Hardware y Sistemas Operativos

1.2.2 Viabilidad a largo Plazo

- 1) ADempiere se protege de la obsolescencia, cumpliendo con los estándares de la industria y utilizando un conjunto de herramientas que sostienen estos estándares:
 - a) Permite a ADempiere cambiar los componentes fundamentales.

- b) Asegura la disponibilidad de una gran base de desarrolladores, quienes conocen las herramientas utilizadas.
- 2) La disponibilidad del código fuente reduce los riesgos de la no-disponibilidad de soporte a largo plazo.
 - 3) Sumamente escalable para sostener un crecimiento orgánico o explosivo producido, por ejemplo, por una adquisición.
 - 4) No es dependiente de la viabilidad en el largo plazo de la organización responsable por el desarrollo del producto. Por ejemplo, Peoplesoft ha adquirido recientemente JD Edwards, causando una significativa incertidumbre en los usuarios finales de los productos de JD Edwards. Del mismo modo, Oracle ha tomado Peoplesoft con la revelada intención de convertir a los usuarios finales de Peoplesoft al producto Oracle Financials. La viabilidad continuada del software Open Source NO está sujeta a la supervivencia de ninguna organización en particular.

1.2.3 Bajo Costo de Propiedad (TCO)

- 1) Sin cargos por Licencias de Software (sujeto a la elección de la base de datos).
 - 2) Bajo incremento del costo a medida que la cantidad de usuarios crece.
 - 3) No tiene que pagar por las actualizaciones anuales.
 - 4) No requiere adoptar costosos, y frecuentemente no garantizados, ciclos de actualización.
- Bajos costos de contratos de soporte.

1.3 Fortalezas del Open Source

Algunas de las ventajas que puede obtener con la utilización de una solución Open Source como ADempiere son:

1.3.1 Reducción de la dependencia de un solo proveedor del producto

- 1) Minimiza el riesgo de tecnología propietaria.
- 2) Elimina la dependencia de un proveedor que provea las licencias.

1.3.2 Auto dependencia

- 1) Proceso flexible en el desarrollo, con mayor enfoque en las necesidades específicas del negocio.
- 2) Mayor grado de participación y entendimiento entre el proveedor y el usuario final.
- 3) Independencia tecnológica.
- 4) Mejor receptividad para direccionar las necesidades locales y las oportunidades de negocios identificadas.
- 5) Las prioridades de desarrollo son manejadas por el usuario NO por el proveedor.

1.3.3 Amplio rango de opciones de soporte

- 1) Soporte comercial brindado por muchas organizaciones.
- 2) Soporte gratuito, disponible en:
 - a) Comunidad de Desarrolladores
 - b) Listas de correo
 - c) Archivos
 - d) Base de datos de soporte

Las experiencias de soporte son generalmente más responsables que con las aplicaciones propietarias.

1.3.4 Técnicamente Superior

- 1) Los productos Open Source están más alineados con los estándares abiertos que los productos propietarios, alcanzando así un mayor grado de interoperabilidad.
- 2) La revisión permanente por parte de la comunidad de desarrolladores, lleva a productos generalmente de una calidad superior.

1.4 Soporte de ADempiere

Una solución Open Source, muchas veces es asociada con un menor costo a lo largo de todo su ciclo de vida, pero también es percibida con un menor nivel de soporte y un alto riesgo, comparada con un sistema propietario. Este no es el caso justamente.

El nivel de soporte proporcionado por organizaciones de Open Source, puede ser considerablemente superior que el proporcionado por un revendedor que distribuye aplicaciones de software propietarias. El primero motivo es que el código fuente está disponible, y por lo tanto puede ser modificado para resolver el problema localmente, a diferencia de los productos propietarios donde el código fuente normalmente no está al alcance de la organización que brinda el soporte; éstos dependen de su desarrollador para proporcionar una corrección. Y esto generalmente se hace en una nueva versión, unos seis a doce meses más tarde.

Adicionalmente, es posible obtener soporte entre la comunidad de desarrolladores, partners y usuarios del software, los cuales responden a las consultas realizadas en los foros, muchas veces en cuestión de horas e inclusive de minutos de realizado el requerimiento.

Además del soporte, la mayoría de las organizaciones buscan obtener “garantías” de que el software adquirido está libre de defectos, o en caso de existir alguno, el mismo se solucionará rápidamente. La historia reciente y la experiencia indican que comprar un software a un proveedor no es garantía de libertad de errores.

La realidad es que en soluciones de Open Source, la lista de errores es conocida y el código fuente está disponible para la organización de soporte, lo cual le permite corregir cualquier error que surja. Este no el caso del software propietario, donde generalmente los errores no se publican y el

código fuente no está disponible para las organizaciones que lo distribuyen y dan soporte.

1.5 ADempiere – Requerimientos de Infraestructura & Hardware

1.5.1 Infraestructura de Red y Hardware

ADempiere tiene la capacidad de operar en una variada gama de redes y sistemas operativos. Esta flexibilidad le da al usuario la libertad de escoger el hardware y sistemas operativos que mejor se adapten a sus necesidades individuales.

1.5.2 Sistemas Operativos

ADempiere puede correr sobre un amplio rango de sistemas operativos, tales como Unix, Windows, Linux y Mac OS X, permitiendo al usuario elegir desde una amplia gama de sistemas operativos abiertos, hasta los sistemas propietarios ofrecidos por los proveedores tradicionales.

1.5.3 Servidor de Aplicaciones

ADempiere utiliza el servidor de aplicaciones Jboss, por el cual no hay que abonar ningún cargo.

1.6 Licencias del Software

NO existen cargos para el uso del software ADempiere.

1.6.1 Licencias

1) Licencias de productos intermedios: No existen licencias o CALs requeridas para correr ADempiere.

Todos los productos utilizados por ADempiere son productos abiertos de la industria estándar, los cuales están libres de cargos por licencias.

2) Licencias de Base de Datos: los usuarios de ADempiere puede elegir entre adquirir su propia licencia de Oracle o seleccionar otras bases de datos, algunas de las cuales son Open Source (Oracle XE, PostgreSQL) u otros productos comerciales ofrecidos sin costo o a un bajo precio.

1.6.2 Gastos Recurrentes

1) Soporte de Hardware: los costos dependerán de la elección de hardware realizada por el usuario. La elección sobre que tipo de hardware utilizará ADempiere, dependerá de los requerimientos individuales del usuario y muchas veces, de las relaciones de éste con sus proveedores de hardware habituales.

2) Licencia de Mantenimiento de Base de Datos: vea los comentarios referidos antes en la sección de Licencias.

3) Mantenimiento (upgrades) del Software de Aplicación: los usuarios de ADempiere tienen la posibilidad de descargar sin costo alguno todos los cambios y mejoras del producto y efectuar las migraciones de la base de datos utilizando recursos propios. El contrato de soporte de ADempiere, también incluye la migración de la base de datos y soporte para actualizar a versiones posteriores de la aplicación.

4) Soporte del Software de Aplicación: el contrato de soporte puede ser adquirido con las organizaciones que dan soporte a ADempiere, o efectuarlo el mismo usuario con recursos propios, muchas veces utilizando los foros de soporte de ADempiere, los cuales son de acceso público y abierto.

5) Extensiones y Modificaciones: ADempiere ha sido diseñado para facilitar las extensiones o modificaciones, realizadas por o para un usuario de ADempiere. La incorporación de un Diccionario de Datos Activo (Active Data Dictionary) posibilita la modificación del diccionario, que puede ser efectuado muchas veces por personas que no tengan conocimientos de codificación y sin depender de proveedores externos.

También pueden ser efectuadas, si el usuario lo desea, por organizaciones como OPENBIZ, con un cargo básico.

1.6.3 Términos de la Licencia

Muchos software Open Source están licenciados bajo los términos de la GNU Public License. Esta licencia requiere que las modificaciones efectuadas al producto (distintas que las realizadas para uso interno) deban ser retornadas a la comunidad Open Source. El sistema ADempiere Business Solution está licenciado bajo los términos de la General Public License (GPL). Esta licencia permite a los usuarios a desarrollar funcionalidades adicionales y utilizarlas internamente ó inclusive licenciarla mediante un cargo a terceras partes, sin la obligación de retornar la mejora a la comunidad Open Source.

Los términos de la licencia de ADempiere se encuentran detallados en <http://www.ADempiere.org/license.html>

2.- ADempiere Business Solution

Generalidades

ADempiere brinda una funcionalidad completa, fácil de usar y de primer nivel para empresas del rango medio. A diferencia de los sistemas tradicionales, ADempiere está organizado en procesos de negocios y no en módulos. Se suministra como un sistema unitario, integrado y completo, en lugar de una serie de módulos acoplados con transferencia de datos entre ellos. De esta manera el usuario obtiene una vista unificada del negocio, con procesos que involucran a toda la organización y no solo a unos cuantos departamentos o unidades tratados como islas. Con ADempiere tiene todos los módulos en uno. Esta integración se aplica tanto al CRM

(Administración de Relación con el Cliente), el Web Store (tienda Web), como a la información del ERP tradicional.

2.1 Organización de ADempiere – Procesos de Negocios

El diseño de ADempiere permite manejar los procesos de negocios, en lugar de los departamentos tradicionales; actualmente, y especialmente en el caso de las empresas medianas, los empleados frecuentemente realizan el proceso de negocio entero o procesos relacionados entre sí.

Procesos de ADempiere ERP & CRM						
Módulo Tradicional	Cotización a Ingresos	Requisición a Pago	Admin. de Clientes	Admin. de Socios	Abastecimiento	Análisis de Resultados
Contabilidad General						*
Cuentas a Pagar		*			*	*
Cuentas a Cobrar	*		*			*
Orden de Compra		*	*		*	*
Orden de Ventas	*		*		*	*
Inventario	*	*	*		*	*
Activo Fijo						*
CRM	*		*	*	*	

La tabla anterior muestra cómo se ven los procesos de ADempiere respecto a los módulos encontrados en los sistemas propietarios tradicionales.

2.2 Conceptos de ADempiere

ADempiere proporciona servicios a múltiples *clientes*. Cada uno de ellos es una entidad, tal como una compañía padre o de máximo nivel equivalente. Cada *cliente* entonces tiene múltiples subsidiarias, departamentos, divisiones, llamadas *organizaciones*.

Se permiten efectuar transacciones entre las *organizaciones*. Por ejemplo, un pago por una organización de un gasto para otra organización resultará automáticamente en una transacción interorganización en ambas, además de las entradas por el pago y el gasto.

Cada entidad externa con la cual la organización efectúa transacciones de negocio se denominan *socios de negocios*. Por ejemplo, clientes y proveedores son *socios de negocios*. Los empleados también son tratados como *socios de negocios*.

Cada transacción está asociada con un *documento*. Por ejemplo, facturas de venta, recibo de materiales, documentos de entregas, pagos a proveedores o recibos de clientes. Cada *documento* tiene predefinido un número de *documento* automático y es almacenado bajo ese número. También es posible adjuntar imágenes para cada *documento*. Además, para cada *documento* el usuario puede definir las consecuencias contables causadas por el procesamiento del mismo.

2.3 Proceso de Cotización a Ingresos

Cubre los procesos de negocios utilizados para la creación de cotizaciones, administración de órdenes de venta, facturación y recepción de dinero por cobranzas. Esta funcionalidad se integra con la Administración de la Cadena de Suministro (SCM) y con la Administración de Relaciones con el Cliente (CRM) de ADempiere. En sistemas tradicionales, esta funcionalidad se encuentra en los módulos de órdenes de venta y cuentas a cobrar.

2.4 Proceso de Requerimiento a Pagos

Cubre el proceso de negocio utilizado para la creación de órdenes de compra, procesamiento de facturas de proveedores y pagos efectuados. Se integra con la Administración de la Cadena de Suministro (SCM). Esta funcionalidad se encuentra generalmente en los módulos de compras y cuentas a pagar.

2.5 Administración de Ítems Pendientes

Cubre el proceso de negocio utilizado para la creación de órdenes de compra, procesamiento de facturas de proveedores y pagos efectuados. Se integra con la Administración de la Cadena de Suministro (SCM). Esta funcionalidad se encuentra generalmente en los módulos de compras y cuentas a pagar.

2.6 Administración de Relaciones con el Cliente (CRM)

Es un módulo integrado que provee una vista lógica de todas las actividades relacionadas con clientes y prospectos. En contraste con los sistemas de CRM tradicionales, no existe la necesidad de efectuar procesos batch ni sincronizaciones con la funcionalidad del backoffice.

2.7 Administración de Relaciones de Socios

La Administración de Relaciones de Socio liga diferentes clientes uno al otro, posibilitando manejar la distribución principal, pedidos de servicio, y gastos de marketing. Ello facilita la provisión de servicios compartidos (centralizados) de una entidad organizacional a otras entidades organizacionales. Esta funcionalidad posibilita a las organizaciones que son propietarias de partes u otras organizaciones enteras (tales como operaciones de franquicias), a proveer servicios centralizados para las operaciones remotas.

2.8 Administración de la Cadena de Suministro (Abastecimiento)

Cubre todas las actividades de administración de materiales, incluyendo recepciones, entregas, movimientos y administración y procesamiento de tomas de stock.

2.9 Análisis de Resultados

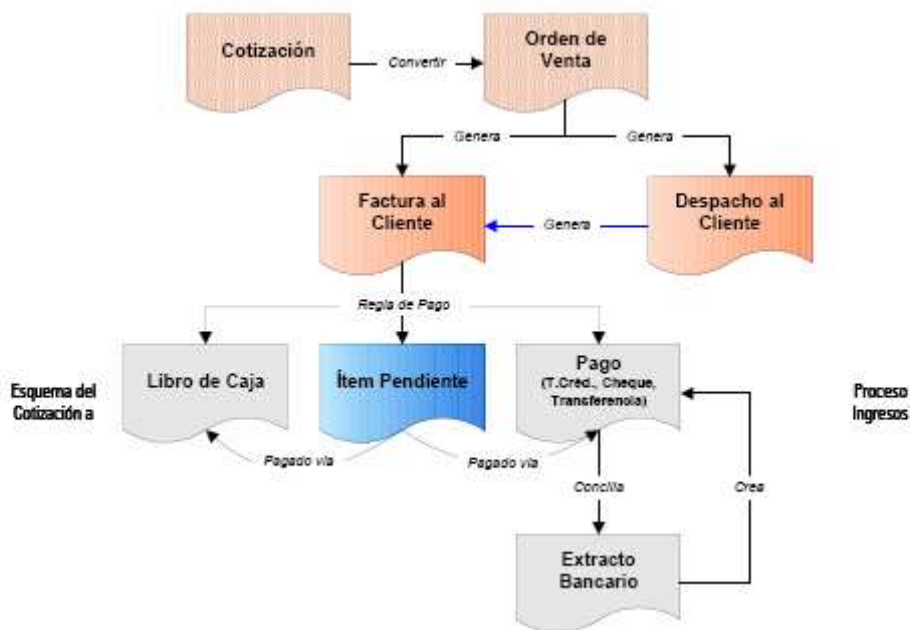
Cubre el costeo y dimensiones contables de la aplicación. Esta funcionalidad generalmente se encuentra en los módulos de Reportes y Contabilidad General, como también en los módulos que generan entradas contables.

2.10 Web Store y Autoservicio

El Web Store de ADempiere, permite a una organización mantener y operar mediante la Web. La información disponible en el Web store es compartida con la aplicación estándar, sin requerirse sincronización ni integraciones adicionales. Los componentes del Web store pueden ser customizados para adecuarse al look-and-feel del sitio Web existente. Proporciona además la posibilidad funcional de un auto servicio para permitir a los Socios de Negocio ver sus propias transacciones online con un apropiado nivel de seguridad.

3 Proceso de Cotización a Ingresos

Cubre el proceso de negocios requerido para la creación de una cotización para un prospecto o cliente, administración de órdenes de venta, facturación y cobranzas. En los sistemas tradicionales esta funcionalidad se encuentra generalmente en los módulos llamados procesamiento de órdenes de venta y cuentas a cobrar.



3.1 Cotizaciones

ADempiere permite la creación e impresión de cotizaciones a clientes basadas en listas de precios generales o específicos por cliente. Las cotizaciones pueden ser efectuadas de manera tal que reserven inventario inclusive. Pueden ser modificadas en cualquier momento y ser convertidas automáticamente en órdenes de venta sin necesidad de ingresar datos adicionales.

3.2 Orden de Venta

Una orden de venta es el documento de control perfecto. Desde una orden de venta se pueden generar de manera automática documentos de entregas y facturas. Adicionalmente, es posible generar automáticamente Órdenes de Compra a Proveedor para los ítems de una orden de venta y que se efectúe la entrega directamente al cliente si corresponde.

Los diferentes tipos de órdenes de venta, causan diferentes comportamientos en el proceso de negocio.

ADempiere maneja los siguientes tipos de orden de venta:

- Orden Estándar: crea la orden y reserva inventario; después genera el despacho y la factura correspondiente. Este tipo de orden se utiliza cuando la entrega de productos se efectúa en base a disponibilidad. La factura puede ser generada inmediatamente o después del despacho.
 - Orden POS (Point of Sale): en un solo paso crea la orden, genera el despacho, factura y recibe el pago (efectivo, cheque, tarjeta de crédito, transferencia). Generalmente se utiliza para ventas en mostrador o con entrega inmediata, con clientes anónimos.
 - Orden a Crédito: crea la orden, genera el despacho y la factura y opcionalmente puede recibir el pago. Se utiliza para clientes identificados, que pueden tener crédito asignado o no.
 - Orden de Depósito o Bodega: crea la orden y despacha el producto. La factura se genera posteriormente. Se utiliza generalmente cuando se realizan facturas que agrupan diversas órdenes. Las facturas se pueden generar manualmente o basándose en reglas de facturación (por Ej. Semanalmente, del 1 al 15, mensualmente, etc.).
 - Orden Prepaga: crea la orden y una factura pro-forma; envía el despacho y genera la factura definitiva una vez recibido el dinero correspondiente.
 - Autorización de Devolución de Material (RMA – Return Material Authorization): recibe un ítem, previamente enviado, y crea una Nota de Crédito.
- Es posible convertir cotizaciones a cualquier tipo de orden e inclusive cambiar de un tipo de orden a otro.

3.3 Despachos

En base a los detalles tomados de la orden de venta, se pueden generar uno o más despachos, inmediatamente o automáticamente cuando existe inventario disponible.

ADempiere puede ser configurado para permitir que los despachos sean efectuados desde documentación de despacho o, alternativamente, requerir la confirmación explícita de tomar y/o despachar previo a la generación de la factura. Las confirmaciones se pueden utilizar para manejar los movimientos de inventario entre áreas que permitan un depósito más disciplinado.

3.4 Factura a Clientes

En base a acuerdos con los clientes, las facturas pueden ser generadas:

- Inmediatamente después de cada despacho,
 - Cuando la orden se entrega de manera completa, o
 - Basadas en un calendario de facturación predefinido para el cliente.
- Por ejemplo, una factura resumen, que contiene todos los despachos realizados en los días previos, semanas o meses.

3.5 Recibos (Cobros)

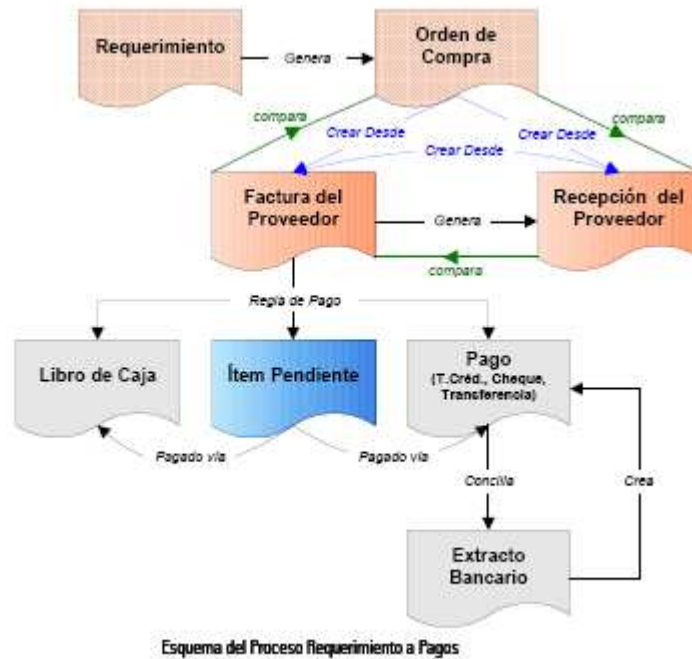
Cuando se recibe una orden de venta o factura, las reglas de pago permiten flexibilidad en la generación automática de recibos:

- Por transacciones en efectivo, se genera automáticamente una entrada en el Libro de Caja.
- Por transacciones con tarjeta de crédito, cheque y débito directo se genera una entrada automática contra la cuenta bancaria correspondiente. Actualmente ADempiere soporta los procesadores de pago de VeriSign PayFlowPro y se planean agregar otros procesadores en futuros releases del software.

Los Ítems Abiertos son actualizados ingresando un pago (por Ej. recibiendo un cheque o creando un débito directo), creando una entrada en el Libro de Caja (por Ej. una factura cobrada por caja chica) o durante el proceso de conciliación bancaria (por Ej. transferencias bancarias).

4 Proceso de Requerimiento a Pagos

Cubre el proceso de negocios necesario para la creación de pedidos, ordenes de compra, recepción de mercadería, facturas de proveedores y el procesamiento de pagos. Esta funcionalidad está integrada con la Administración de Abastecimiento (SCM). Los módulos de Compras y Cuentas a Pagar contienen esta funcionalidad en los sistemas tradicionales.



4.1 Requerimientos (Pedidos)

Los requerimientos pueden ser tomados automáticamente desde los Reportes de Reabastecimiento de Material o ingresarse de manera manual. ADempiere proporciona un Workflow, incluido en el producto estándar, por el cual los pedidos por montos superiores a los \$1.000 deban ser aprobados. Automáticamente se genera un reporte de las aprobaciones efectuadas, desde el cual se pueden generar de manera manual las órdenes de compra correspondientes. Este Workflow se entrega como una muestra de funcionamiento del Workflow de documentos.

4.2 Orden de Compra

Las órdenes de compra se pueden generar, y consolidar si lo requiere, desde los reportes de reabastecimiento de material, desde las requisiciones aprobadas, o pueden ser generadas manualmente. Las órdenes de compra pueden ser transmitidas vía EDI, por e-mail o por fax.

4.3 Recepción de Material

Son procesadas creando un registro de recepción de material. Estos registros son entonces comparados con las órdenes de compra o facturas del proveedor. Es posible generarlas automáticamente desde órdenes de compra o facturas de proveedor para aliviar el reingreso de datos manuales.

4.4 Facturas de Compra

Pueden ser ingresadas manualmente, en base a la factura del proveedor, o ser creadas automáticamente desde órdenes de compra o recepciones de material, en cuyos casos son comparadas con éstos. La recepción de material puede ser creada desde la factura de proveedor, cuando ambas llegan al mismo tiempo.

4.5 Pagos

ADempiere permite generar pagos basados en términos de pago (30 días, contado, etc.), permitiendo además la incorporación de descuentos automáticos. Los pagos pueden ser efectuados mediante transacciones de débito directo (transferencias ó ACH) o imprimir los cheques en formularios preimpresos otorgados por el banco. También puede registrar los pagos efectuados mediante tarjetas de crédito.

4.6 Conciliación Bancaria

Pueden ser ingresadas o cargadas automáticamente, dependiendo de su banco. Puede conciliar pagos en tránsito, ingresar cargos o registrar débitos directos por pagos efectuados.

5 Proceso de Administración de Ítems Pendientes

Este proceso automatiza la entrada y asignación de dinero recibido de los clientes y los pagos efectuados a proveedores. También provee la conciliación bancaria y libros de caja, teniendo en cuenta los pagos en tránsito, cargos bancarios y la creación de pagos por transferencias directas.

5.1 Reglas de Pago para Cuentas a Pagar

Los pagos son creados de manera automática, basados en el conjunto de reglas de pago establecidos para la factura del proveedor. Estas reglas pueden cambiarse en cualquier momento, para reflejar el método efectivamente utilizado, en caso de utilizar uno alternativo. ADempiere soporta las siguientes reglas de pago:

- Efectivo: se crea una entrada automáticamente en el Libro de Caja en ese día.
- Cuenta corriente: es el término de pago por defecto para el Socio de Negocio, salvo que se especifique otro diferente.
- Tarjeta de Crédito: las transacciones por tarjeta de crédito pueden ser procesadas online. Las facturas son marcadas como pagadas y el cargo se mantiene como pagos sin conciliar. Este método puede requerir un procesador externo para las tarjetas de crédito.
- Cheques: luego de seleccionar el banco apropiado, se pueden ingresar los cheques al sistema. Las facturas son marcadas como pagadas y el cheque se mantiene en el sistema como pago sin conciliar.

5.2 Asignaciones

La asignación liga el pago, o múltiples pagos, a las facturas (o múltiples facturas) o acredita Notas de Crédito y registra los descuentos en pagos y cancelaciones de cuentas por cobrar. El usuario selecciona los documentos correspondientes e ingresa o confirma las diferencias como pagos parciales, descuentos o cancelaciones.

5.3 Conciliación Bancaria

Pueden ser ingresadas manualmente o cargadas automáticamente de manera electrónica, provista por la institución financiera. ADempiere posibilita la conciliación de pagos en tránsito y cargos bancarios o la creación de transferencias de débito directas.

5.4 Libro de Caja

Todas las facturas pagadas y/o cobradas por caja chica son ingresadas de manera automática al libro de caja. Para ello se crea un diario de caja por día y organización. El diario de caja es utilizado también para:

- Gastos generales, para las cuentas definidas en el libro de caja.

- Ingresos generales, para las cuentas definidas en el libro de caja.
- Diferencias de caja chica, para las cuentas definidas en el libro de caja.
- Cargos
- Transferencias desde o hacia una cuenta bancaria.

5.5 Cargos

Son utilizados en ADempiere para permitir procesar costos o ganancias no relacionados con productos, tales como cargos por transportes, cargos bancarios e intereses. Los cargos son ligados a cuentas de la contabilidad general, y varios tipos de cargos pueden apuntar a la misma cuenta contable. Por ejemplo, los cargos “Resma de Papel” y “Cartuchos de Impresora” pueden apuntar ambos a la cuenta “Gastos de Impresión”.

Un cargo puede referir tanto a un gasto como a un ingreso. Así por ejemplo el cargo “Transporte” puede ser acreditado a una cuenta de ganancia si es ingresado en una factura de venta, o a una cuenta de gasto si aparece en una factura de proveedor. El sistema determinará el tipo, en base al contexto en el que se ingrese. Un cargo en una factura a cliente es una ganancia, mientras que en una factura de proveedor es un gasto. Por otro lado, un cargo con signo positivo en el Libro de Caja será una ganancia, mientras que si es con signo negativo será un costo.

Los cargos pueden ser definidos para debitar o acreditar en diferentes cuentas contables, de acuerdo con un porcentaje de división preestablecido. El usuario puede también definir el tratamiento de impuesto para los cargos.

El importe de los cargos puede estar predefinido, para ganar velocidad y seguridad en su registración.

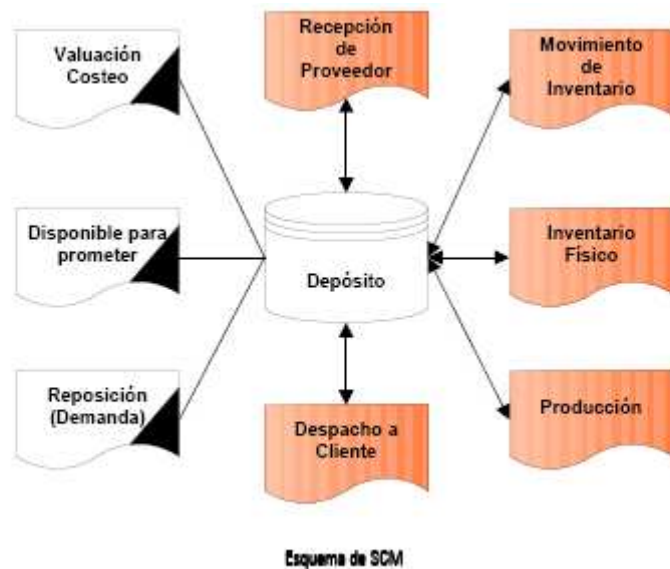
Por otro lado, el importe de un cargo no tiene asociada una moneda específica, sino que ella está determinada en base a la moneda del documento.

Algunas otras características del proceso de administración de ítems abiertos son:

- Invertir asignaciones
- Análisis de deudas por antigüedad
- Procesamiento online de Tarjetas de Crédito y Cheques electrónicos
- Procesador de pagos mediante VeriSign PayFlowPro (no disponible en Argentina)
- Recordatorio de Deudas (es el proceso de recordar a los clientes la deuda mantenida. Puede comenzar con simples recordatorios hasta notas más firmes conforme las deudas sean más antiguas).

6 Administración de la Cadena de Suministro

Este proceso cubre todas las actividades de administración de materiales, incluyendo recepción, despachos, movimientos y balances de stock, dentro de una compañía y sus sucursales, y entre proveedores y clientes.



El Catálogo de Productos lista los productos y servicios con la lista de materiales y sustitutos opcionales. El sistema le permite importar y actualizar precios de compra desde sus proveedores.

Los productos se organizan en categorías y jerarquías, y pueden ser buscados también en base a atributos que se aplican sobre un determinado número de productos, por ejemplo “todas las camisas amarillas de manga corta”.

ADempiere soporta múltiples listas de precios para todos los ítems comprados y vendidos. La funcionalidad del precio de lista de compra, permite un control simple de los descuentos desde el proveedor y el sistema posibilita la existencia de listas de precio de ventas generales o específicas por cliente.

6.1 Control de Depósitos

ADempiere soporta las siguientes características en el manejo avanzado de depósitos:

- Múltiples almacenes físicos y cada uno de ellos ser descompuesto en múltiples almacenes lógicos, como ser recepción, control de calidad y testeo, almacenamiento y entrega.
- Almacenar en cada almacén en una ubicación referenciada por 3 ejes (pasillo, cajón y nivel) definido por el usuario.

- Múltiples unidades de Medida (por ejemplo almacenar en cajas y vender en unidades).
- Prioridades de salida, para asegurarse que salen de una ubicación con una secuencia preestablecida.
- Prioridades de usuario para despachos o recepción.
- Los movimientos de inventario entre ubicaciones o almacenes pueden configurarse para que se efectúen con la documentación adecuada y el manejo de stock “en tránsito”.
- La toma y los ajustes de inventario pueden ser procesados en paralelo con las actividades de venta.
- El stock utilizado para propósitos internos puede ser fácilmente descontado para registrar el decrecimiento de stock y las consecuencias financieras de la contabilidad general.

6.2 Administración de Materiales

La documentación de entrega, puede ser creada en forma serial (batch) o individualmente una por orden. Los bienes recibidos de los proveedores pueden ser comparados directamente con la orden de compra o la factura del proveedor.

El sistema permite tener un “disponible para prometer”, calculado teniendo en cuenta las reservaciones para despachos a realizar a clientes y las recepciones esperadas del proveedor.

Las Listas de Reabastecimiento de Material, son creadas basadas en reglas de reabastecimiento de inventario. Los pedidos y órdenes de compra pueden ser generados automáticamente desde el Reporte de Reabastecimiento de Material.

ADempiere además permite:

- Seguimiento de Lotes/Series y manejo de números de serie.
- Listado de número de parte del proveedor y otros atributos.
- BOM o desmontaje.
- Fusión de productos.
- Cantidades de stock negativas.
- Administración de Activos.

6.3 Listas de Materiales (BOM)

Una Lista de Materiales puede contener uno o más Productos, Servicios o inclusive otros BOM. No existe un límite en cuanto a la cantidad de elementos ni niveles que pueda contener un BOM. La limitación está solamente en que un BOM tiene que ser “no-circular”, es decir que no puede tener referencias a él mismo ni a sus partes (ejemplo de estos casos se encuentran en las recetas de industrias químicas).

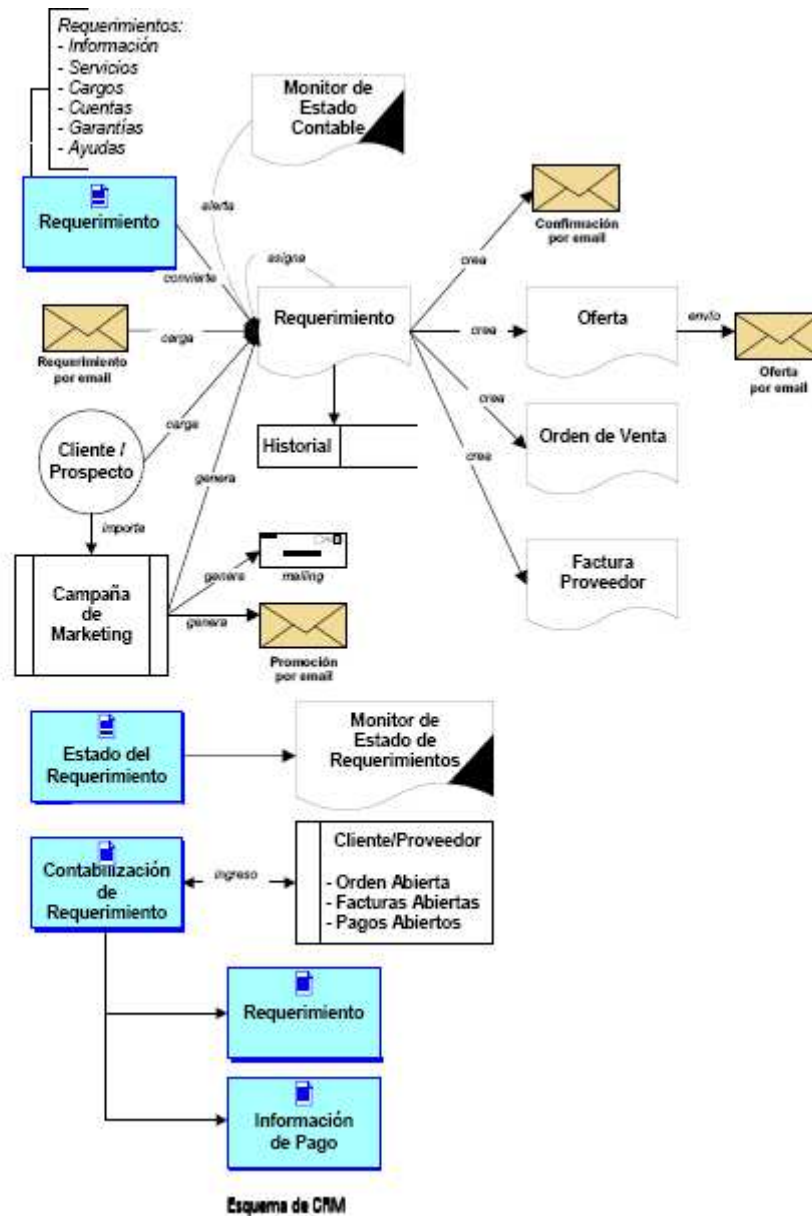
ADempiere maneja dos tipos de BOM:

- **Almacenado:** si se indica que el BOM es almacenado, es tratado como si fuera un producto normal en términos de disponibilidad. Para crearlo, es necesario “ensamblarlo” (o desensamblarlo) mediante “Producción”. La disponibilidad representa la cantidad que existe en stock, no lo que se podría producir. Si el precio establecido es 0.00, entonces es calculado de manera dinámica (por ejemplo, sumando las partes individuales) y para este caso se requiere que el BOM y todos sus componentes estén en la lista de precios seleccionada. Normalmente se imprime solo la información del BOM, pero para las facturas, entregas y listas tiene la opción de imprimir el detalle (allí se imprimen las cantidades también).
- **No almacenado:** generalmente se utilizan por una conveniencia en el ingreso de datos.

Cuando se procesa la orden o la factura, se generan las líneas de los productos involucrados. La cantidad disponible en un BOM no almacenado se calcula dinámicamente en base a cada ítem y representa lo que podría estar disponible. Para este tipo de BOMs, el precio es siempre la suma de precios de los ítems individuales que lo componen.

7 (CRM) Administración de Relaciones con el Cliente

El CRM en ADempiere no es un módulo independiente, sino una vista lógica de todas las actividades relacionadas con los clientes o potenciales clientes (llamados prospectos).



En ADempiere las funciones de CRM son una parte integral del proceso de negocio, por lo tanto no se requieren procesos batch ni de sincronización, como es habitual en los sistemas de CRM tradicionales.

Puede administrar la creación, distribución y seguimiento del cliente, proveedor y los pedidos generados internamente, para asegurar un tiempo de respuesta oportuno, crecimiento de acuerdo a procesos y tiempos definidos. ADempiere soporta los siguientes tipos de requerimientos en el área de CRM:

- Información: requerimiento no estructurado originado desde la Web o vía email.
- Servicios: requerimientos estructurados para realizar un servicio en un lugar y fecha determinados.
- Cargos: requerimiento estructurado para reembolso de costos.

- Cuenta: requerimiento estructurado relacionado con una orden, factura, despacho o pago relativo a un proveedor o cliente en particular.
- Garantía: requerimiento estructurado relacionado con un problema con un servicio o producto.
- Ayuda: requerimiento estructurado de servicios a clientes.

Dependiendo del tipo de requerimiento que se trate, este puede ser convertido automáticamente a un documento (por Ej. una oferta, orden o factura). Es posible enviar manual o automáticamente un e-mail de confirmación con un número de seguimiento y, utilizando ese número, el autor del requerimiento puede actualizar información en el mismo. Los requerimientos pueden ser asignados a usuarios del sistema, para que tome acciones o realice el seguimiento.

Los requerimientos pueden ser generados también en base al estado de la cuenta (por Ej. fecha de la última venta, pago vencido, etc.) para el seguimiento por parte de la fuerza de ventas o de atención al cliente.

7.1 Administración de Campañas de Marketing

La retención de clientes es una misión crucial para cualquier compañía; se calcula que retener un cliente cuesta 6 veces menos que conseguir uno nuevo. ADempiere soporta esto mediante la creación de mailing o requerimientos para facilitar el seguimiento de la fuerza de ventas (o tele-ventas). Los criterios para las campañas de retención podrían ser última venta, volumen de ventas, productos comprados u otros motivadores. Para atraer nuevos clientes se pueden importar perspectivas desde mailing o requerimientos.

La eficiencia de estas campañas de marketing puede ser medida por la ganancia o beneficio bruto generada por cada una, ligando cada documento (por Ej. factura u orden) a cada campaña en el momento que se genera el documento. Esta información está disponible dentro de ADempiere, para reportes y análisis.

7.2 Análisis de Ganancias de Cliente

Los reportes de ganancias y beneficios brutos de clientes específicos o grupos de clientes en un determinado período de tiempo, pueden obtenerse utilizando la posibilidad de generar reportes que provee ADempiere, o utilizando generadores de reportes de terceros y/o visores OLAP.

7.3 Autoservicio para pedidos online

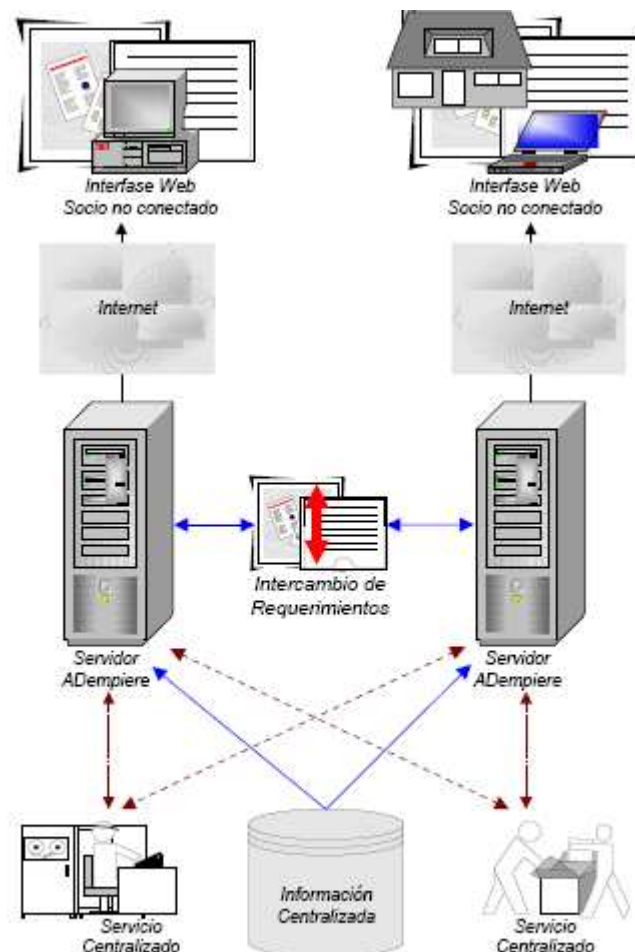
ADempiere permite el acceso vía Web de Socios de Negocios autorizados (por Ej. clientes, proveedores o empleados), con el propósito de ver o consultar información relevante para ese socio de negocios, utilizando para ello un browser Web. La información puede incluir saldos de cuentas,

facturas o cosas por el estilo, o iniciar el seguimiento y efectuar pagos sobre ítems abiertos. Esta funcionalidad de auto servicio, puede ser utilizada también para permitirle a los clientes registrarse a fin de recibir material de marketing seleccionando áreas de interés o para descargar archivos con datos seguros, por ejemplo listas de teléfonos donde solicitar soporte.

8 Administración de Socios

La Administración de Socios vincula diferentes clientes entre sí, permitiendo manejar prioridades de distribución, gastos de marketing, o proporcionar servicios centralizados.

Básicamente la Administración de Socios proporciona la funcionalidad de CRM a través de los clientes de ADempiere, intercambiando automáticamente los requerimientos con Socios de Negocios que están conectados a ADempiere. Aquellos Socios que no están conectados, pueden hacer seguimiento de información mediante la interfase Web.



Esquema de Administración de Socios

ADempiere puede ser utilizado para crear guías y distribuir las a los Socios de Negocios. El sistema puede utilizarse también para seguir y

monitorear progresos y resultados. También le permite a los Socios de Negocio crear facturas por cargos directamente por gastos en actividades de marketing.

ADempiere facilita la administración y provisión de servicios compartidos (por Ej. contabilidad, despachos, help desk, etc.) para los Socios de Negocios tales como franquicias. Como proveedor de servicios, el usuario solo tiene acceso a la información que necesita para sus tareas, a través de múltiples clientes y organizaciones.

El sistema puede mantener datos centralizados, tales como productos, listas de precios, información contable para todos sus socios. Estos pueden agregar entidades adicionales, pero no pueden modificar los elementos que son mantenidos centralmente, por una cuestión de consistencia y seguridad. La combinación de información mantenida centralmente y localmente, posibilita la administración de una “red de organizaciones”; un típico ejemplo de ello son las operaciones de franquicias o aquellas organizaciones que proveen funciones centralizadas para asociados independientes.

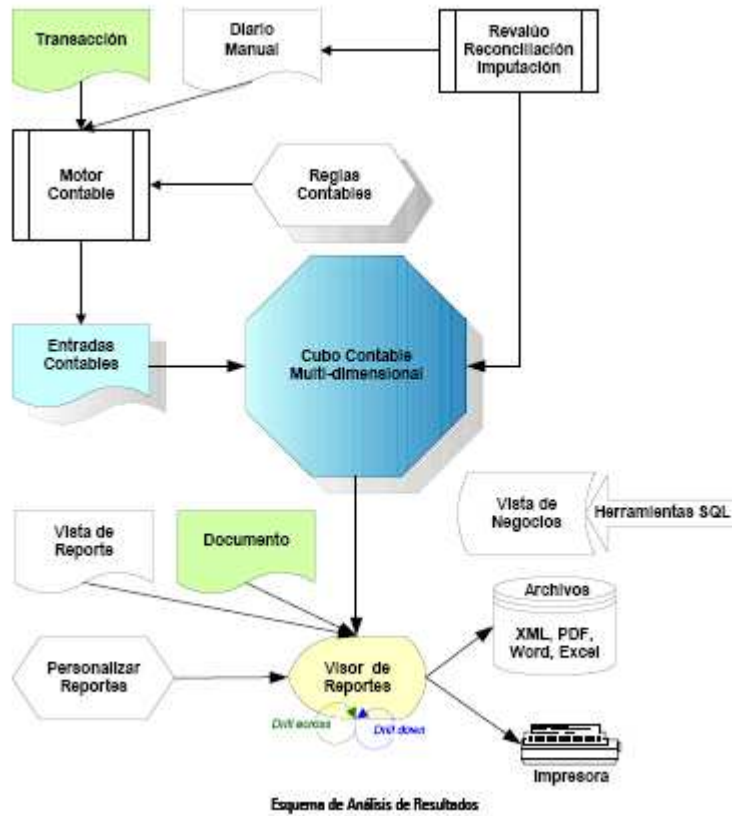
8.1 Contador de Documentos

ADempiere proporciona una funcionalidad que permite a organizaciones independientes pero relacionadas, a generar automáticamente un documento en otra organización. Esta funcionalidad reduce considerablemente el esfuerzo implicado en el doble procesamiento de las transacciones (una vez en cada organización) y asegura que los impuestos sean manejados de manera correcta entre personas jurídicas separadas. Por ejemplo, un franquiciado puede colocar una orden de compra en el concesionario y la orden de venta será creada automáticamente en el libro de contabilidad de este último. Cuando el concesionario despacha al franquiciado, el recibo de material será creado en la contabilidad del franquiciado y así cada transacción que ha sido configurada automáticamente creará un contador de documentos. Además permite que sean registrados diferentes costos en las cuentas del franquiciante y el franquiciado.

ADempiere permite configurar de manera más sofisticada la situación descripta, permitiendo por ejemplo que el receptor de la mercadería “confirme” la recepción, permitiendo la administración de bienes en tránsito y asegurar que no pueda emitirse una factura automática, hasta que el receptor de los bienes no haya confirmado su recepción.

9 Análisis de Resultados

Esta funcionalidad cubre el costeo y dimensiones contables de la aplicación y se encuentra generalmente en los módulos de Reportes y Contabilidad general en los sistemas tradicionales.



9.1 Reglas Contables

Las entradas contables son generadas automáticamente en base a reglas que son aplicadas a documentos de transacciones y son definidas por el sistema, pudiendo ser extendidas por el usuario si lo desea. Estas reglas, definen los códigos de cuentas para cada grupo de transacciones generadas por un documento contable, permitiendo que la mayoría de las transacciones sean ingresadas al sistema sin que los usuarios deban conocer los números de cuentas a imputar. El sistema también permite el ingreso manual para generar imputaciones adicionales (actual, presupuesto y estadística).

9.2 Reportes Integrados, Data Warehousing y OLAP

Los reportes pueden ser creados para cada tipo de documento en el sistema. El usuario puede definir el layout, secuencia, formato y totalizar cualquier reporte y poner este a disposición de cualquier usuario del sistema u organización, configurando la seguridad de manera apropiada.

Las facilidades de reportes permiten desde navegar dentro de los mismos (por ejemplo, desde una orden de compra ir a socio de negocio,

reglas de pago, etc.). Para las entradas multidimensionales, el usuario puede seleccionar la dimensión que utilizará en el reporte.

Por otro lado, toda la información disponible para generar reportes, puede ser exportada a una gran variedad de formatos, para su utilización en hojas de cálculo y procesadores de texto. Es posible generar nuevos reportes o extender los existentes, permitiendo ver información más detallada cuando y donde sea necesario.

9.3 Diarios Manuales

La mayoría de las transacciones contables son generadas como consecuencia del procesamiento de Documentos, permitiendo grabar las transacciones individuales en múltiples esquemas contables. Los diarios manuales, permiten crear entradas contables para un esquema contable en particular. ADempiere soporta la auto reversión de entradas en el diario y proporciona la funcionalidad de documentos recurrentes, que permite procesar cualquier documento basado en transacciones.

9.4 Distribuciones de Contabilidad General

El sistema facilita la creación de reglas definidas por el usuario que permiten que importes debitados o acreditados en el sistema por cualquier documento sean distribuidas sobre múltiples cuentas contables.

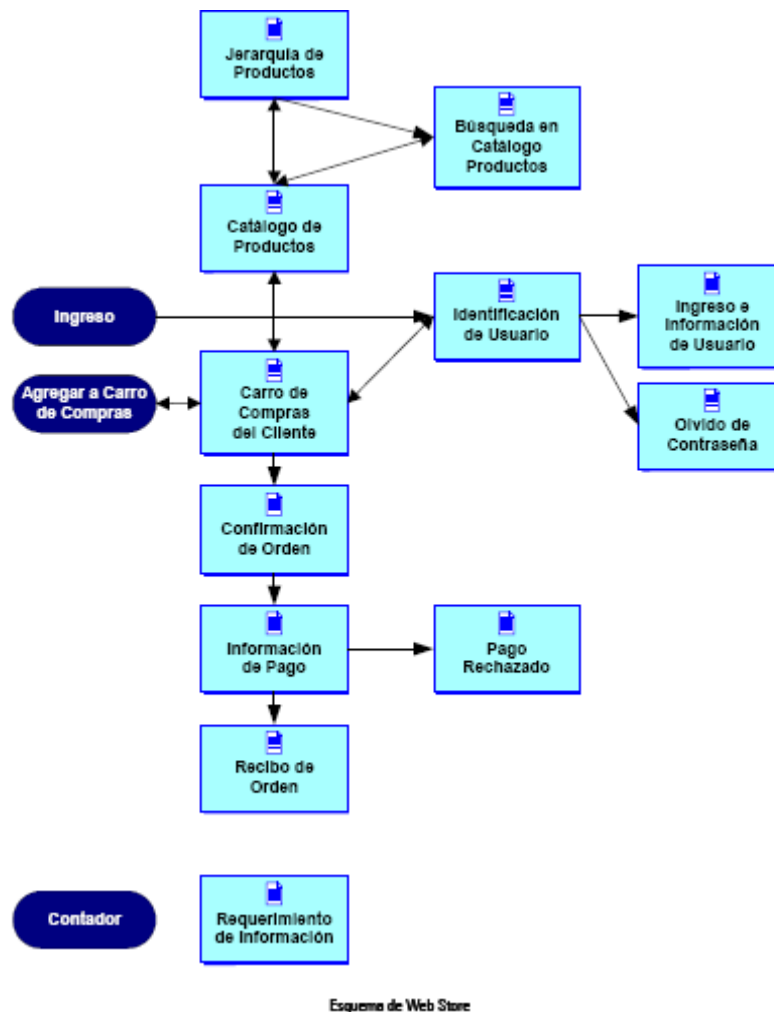
9.5 Características Adicionales

Otras características provistas por el proceso de Análisis de Performance son:

- Soporte de múltiples esquemas de cuentas (por Ej. para propósitos legales o internos).
- Soporte de entradas multi-dimensionales (por Ej. organización, producto, socio de negocios, proyectos, etc.).
- Soporte para mult- costeos (por Ej. Estándar, promedio, FIFO).
- Soporte multi-monetario (por Ej. comprar en dólares y vender en pesos).
- Asiento automático mediante el servidor de aplicaciones.
- Seguimientos de auditoría.
- Reportes de Performance para Acuerdos de Nivel de Servicios (SLA - Service Level Agreement).

10 Web Store

El Web Store de ADempiere le permite a la empresa mantener y operar mediante la Web. La información está compartida con la aplicación estándar, sin necesidad de sincronizaciones o integración adicionales entre el Web store y la aplicación. Los componentes del Web store pueden ser customizados para lograr el look-and-feel necesario, por ejemplo para que tenga el mismo estilo del sitio Web de la empresa.



Además el Web store tiene varios niveles de seguridad para permitir a los socios de negocio ver sus propias transacciones online.

10.1 Catálogo de Productos Online

Los usuarios tienen la posibilidad de ver y buscar en el catálogo de productos. Es posible almacenar y mostrar imágenes y especificaciones para cada producto sobre el Web store a elección del usuario. Por otro lado, los productos pueden estar restringidos de acuerdo a los derechos de acceso del socio de negocio y una vez que el usuario se ha identificado – logeado – en el Web store, los productos son clasificados y sus precios son mostrados acorde a la lista de precios específica del cliente.

También es posible definir jerarquías que permitan limitar la selección de productos y efectuar las búsquedas de productos en base a Categorías y/o Atributos.

10.2 Transacciones de Venta Online

ADempiere permite a los usuarios agregar ítems al Carrito de Compras Virtual, mediante el catálogo de productos o a través de un formulario de pedido. Estos ítems pueden ser modificados y/o borrados del carrito de compras. Es obligatorio que el ingreso sea mediante acceso seguro para recuperar información almacenada del cliente.

La información del pago es ingresada o confirmada. Actualmente se soporta el procesador de pagos de VeriSign, pero en el futuro están previstos otros más. Luego de recibir la confirmación del pago, se crea la orden y se muestra el recibo, junto con el código de autorización recibido del Gateway de pago. Opcionalmente es posible enviar un email con el recibo correspondiente. Por otro lado se envía un email a la persona responsable de las Ordenes Web para notificarlo de la orden y a efectos de su procesamiento posterior.

10.3 Componentes Soportados

Administración de Usuarios: se puede almacenar información de los usuarios y habilitar cookies para permitir la detección e ingreso automáticos.

Contador: el sistema monitorea los pedidos Web y recopila información para analizar la actividad, tal como usuario – si se ha logeado -, fecha, host del usuario, etc.

Pedidos o Requerimientos: un pedido de un usuario Web puede ser remitido a uno o más direcciones de mail.

También se puede enviar una confirmación al usuario que efectuó el requerimiento. El Requerimiento o Pedido es la información base para el CRM.

11 Fabricación

ADempiere Business Solution proporciona elementos básicos para soportar el proceso de fabricación, manejando la funcionalidad BOM (Bills of Material) o Cuenta de Materiales. Se puede “producir” un producto BOM y los productos terminados serán agregados al inventario, actualizando también la lista de materias primas y servicios utilizados.

Está prevista, para un próximo release, la integración de un módulo más completo para aquellos usuarios que requieran de soluciones más complejas.

Este módulo, ya desarrollado y en proceso de testeo, cubre los siguientes tópicos relacionados con la fabricación:

- Administración de Recursos
- Workflow de Fabricación
- Fórmulas y BOM
- Administración de Ordenes de fabricación
- Pronóstico y Planeamiento de Requerimiento de materiales
- Capacidad de Planificación de Requerimientos
- Administración de Costos

12 Vista Técnica

12.1 Arquitectura Técnica

ADempiere proporciona CRM, Administración de Relaciones de Socios, SCM, ERP y OLAP totalmente integrados. La aplicación ha sido diseñada para ser alojada en la Web y permite múltiples opciones de implementación. El Diccionario de Datos Activo, en el cual está basada la aplicación, asegura una funcionalidad estable con un look-and-feel consistente.

ADempiere está diseñado para cambiar según las necesidades del negocio. En cualquier momento, aún en producción, los usuarios del sistema pueden cambiar la estructura de la información contable y de negocios, ajustándola a nuevas necesidades sin impactos negativos.

ADempiere proporciona múltiples vistas de la información basadas en el detalle de las transacciones.

Esta estructura permite máxima flexibilidad y fácil integración, ya que por ser solo vistas de la información almacenada en la base de datos, pueden ser cambiadas rápidamente.

12.2 Tecnología

ADempiere es una aplicación cliente-servidor escrita enteramente en Java, que soporta el procesamiento de grandes volúmenes de información y una interfase gráfica de usuario de alta performance.

Está prevista en un futuro la migración hacia una arquitectura de 3 capas.

12.2.1 Servidor de Aplicaciones

El servidor de aplicaciones está implementado en Java, con la tecnología J2EE, utilizando la infraestructura del servidor de aplicaciones JBoss. Este servidor puede estar corriendo de manera stand alone o en el mismo equipo que el servidor de la base de datos. Para la administración del servidor se utiliza JMX (Java Management Extensions). El acceso a la base de datos se realiza mediante el protocolo JDBC (Java Database Connectivity). Está planificado que futuros releases de ADempiere soporten otros servidores de aplicaciones que cumplan con las especificaciones de J2EE (por Ej. IBM Websphere, Oracle Application Server, etc.). Además del estándar HTTP, se utiliza el protocolo SSL para la implementación de la funcionalidad Web Store.

12.2.2 Aplicación Cliente

Los componentes de la aplicación Cliente están escritos enteramente en Java, diseñados para utilizar las capacidades que brindan las PC's actualmente. La aplicación Java o cliente Java Applet es la elección ideal para altos volúmenes de datos y proporciona una interfase gráfica de usuario de alta performance. Se comunica vía thin JDBC (Java Database Connectivity) con la base de datos y mediante RMI (Remote Method Invocation) con el servidor de aplicaciones. El cliente puede acceder a los servidores a través de Internet o de una Intranet.

En aquellos casos donde la instalación o descarga de la aplicación no sea posible (por Ej. para la funcionalidad de auto servicio de proveedores, clientes o empleados), es posible utilizar un cliente HTML. Este está implementado mediante Java Servlets y Java Server Pages almacenadas en Servidores de Servlet. Si bien este cliente proporciona mucha funcionalidad, es menor a la soportada por el cliente Java Applet.

12.2.3 Base de Datos

La funcionalidad de los procedimientos de PL/SQL fueron movidos hacia el motor de Workflow de ADempiere. Para ello se extendió el motor de persistencia de ADempiere lo cual posibilitó la eliminación de los Triggers, y todas las funciones PL/SQL fueron convertidas a SQLJ (Java corriendo en la base de datos).

ADempiere genera las sentencias SQL y las analiza sintácticamente por seguridad. La capa de independencia de la base de datos convierte los SQL a la notación correspondiente de la base de datos.

El programa de configuración empaqueta las librerías requeridas para la instalación de clientes y servidores de aplicaciones. Este enfoque elimina la necesidad de portar ADempiere a otros motores de base de datos y permite que las nuevas versiones estén disponibles para otras plataformas simultáneamente.

Los requerimientos de ADempiere para la base de datos son:

- Que soporte completamente del estándar ANSI SQL 99 (CASE, todos los tipos de JOIN, etc.)
- Que soporte vistas y vistas sobre vistas.
- Que soporte Funciones Definidas por el Usuario (preferentemente a través de SQLJ)
- Que soporte vistas en línea (por Ej. SELECT ... FROM (SELECT xx FROM yy)...))
- Que soporte JDBC 3.0 (especialmente Row Set)

12.2.4 Criterio para la selección de la Base de Datos

La base de datos es crucial para cualquier aplicación ERP ó CRM y los usuarios deberían seleccionar su base de datos en base a los siguientes criterios:

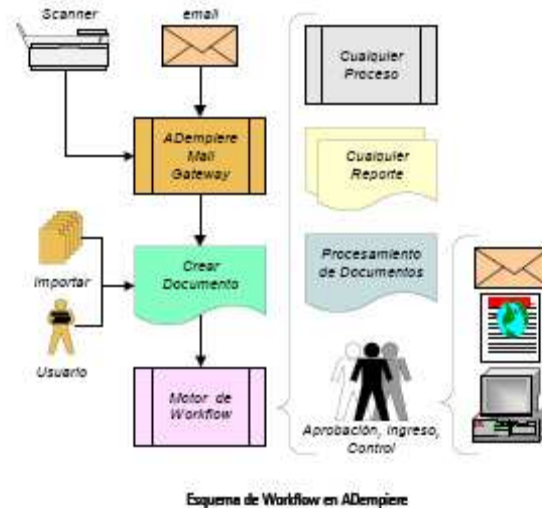
- Auto administración: tiene la base de datos capacidades de auto-tunning y auto-extensión?
- Estabilidad: puede la base de datos correr sin mantenimiento ni caídas por años y tolera fallas en los programas y sistemas operativos con recuperación automática?
- Disponibilidad: es posible correr la base de datos 24/7 (por Ej. si tiene un Web Store), hace cold/hot backups y provee fail over automático?
- Performance y Escalabilidad: incluye wizards de performance, índices, utiliza CPU's adicionales y RAID de hardware y tiene la posibilidad de cluster.

La selección de la base de datos es importante, ya que el usuario podría tener problemas con ella y suponer erróneamente que son problemas de ADempiere.

12.3 Workflow y Administración de Procesos de Negocios

Generalmente un Workflow se define como “pasos que involucran gente”, mientras que Administración de Proceso de Negocios es definido como “Workflow y actividades del sistema”.

ADempiere soporta el Business Process Management (BPM) y está basado en los estándares Workflow Management Coalition y OMG. En este documento se utilizará el término Workflow para incluir las capacidades BPM.



A diferencia de otras aplicaciones ERP y CRM, el Workflow no está “por encima” de la aplicación, sino que ADempiere está basado en Workflow. El motor de Workflow de ADempiere es el corazón del administrador de transacciones, razón por la cual todos los procesos en ADempiere son activados por Workflows y son fáciles de extender y modificar. Al estar los Workflows completamente integrados son fáciles de mantener y proveen mucho más funcionalidad que los Workflows externos o agregados que ofrecen algunos otros proveedores de ERP y CRM.

12.3.1 Tipos de Workflows

ADempiere ofrece tres tipos de Workflows:

- General: proporcionan guías e instrucciones paso a paso para cumplir una tarea. Por ejemplo, los wizards de configuración. Este tipo de Workflow lo inicia el usuario desde el menú.
- Procesador de Documentos: controlan los pasos de procesamiento de todos los documentos y se inician automáticamente cuando se procesa un documento. Pueden ser extendidos, por ejemplo para solicitar autorización en una orden de compra si el importe de la misma supera un cierto valor.
- Por Valor de documentos: es iniciado automáticamente cuando una entidad cumple con una condición especificada por el usuario. Por ejemplo, requerir un proceso de aprobación para definir el crédito de un cliente nuevo.

12.3.2 Acciones de Nodos y Transiciones

Un Nodo en el Workflow de ADempiere puede tener las siguientes acciones:

- Proceso Automático: Cualquier acción en Proceso, Reporte, Tares, Workflow, Documento.
- Acción de Usuario: Cualquier pantalla o formulario donde un usuario necesita confirmar la realización.

- Establecer una Variable: Cualquier Columna a Constante o Variable.
- Selección de Usuario: Cualquier selección, por ejemplo aprobar o selección en una Lista.
- Wait (Espera): puede ser utilizado para iniciar, finalizar, etc.

La transición entre los nodos puede, opcionalmente, tener condiciones y además se permite el procesamiento paralelo mediante múltiples transiciones de un nodo. Esto permite modelar escenarios complejos a través de la funcionalidad que proporcionan los Workflows de ADempiere.

12.3.3 Aprobaciones (Personas Responsables)

El usuario puede definir una jerarquía para aprobaciones o utilizar la de la organización. La persona responsable de un Workflow puede ser un usuario específico o el invocador, un grupo (rol) o el supervisor de una organización. Pueden también existir diferentes responsables para cada nodo/paso del Workflow.

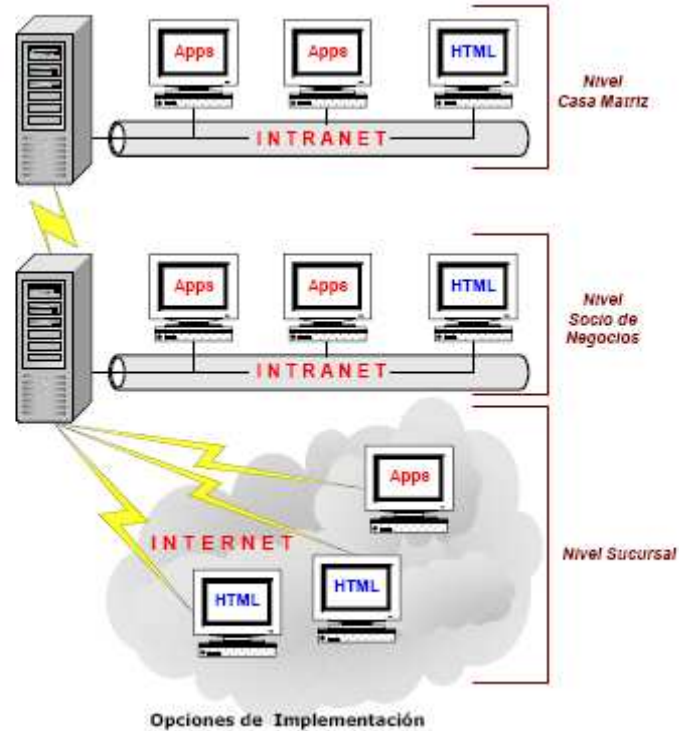
12.3.4 Prioridad, Avances, Alertas

Es posible administrar las prioridades dinámicamente, lo cual permite que sea usado para ruteos de Call Centers y soporte basado en las prioridades de cliente. Además los usuarios pueden definir reglas de avance por inactividad y enviar alertas a los responsables del Workflow y/o supervisor.

12.4 Opciones de Implementación

ADempiere tiene los siguientes componentes principales:

- Cliente
 - Aplicación Java
 - Applet Java
 - Basado en HTML
- Servidor de Servlet para aplicación basada en HTML
- Servidor de Aplicaciones
- Servidor de Base de Datos



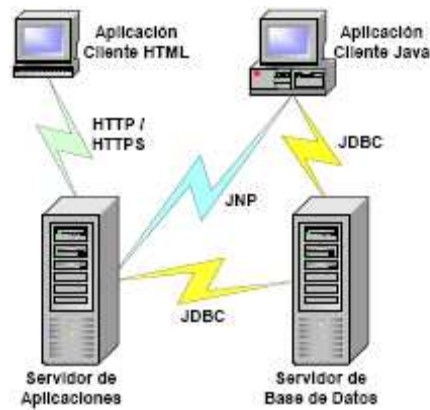
Todos los componentes pueden ser implementados en cualquier plataforma que soporte Java:

Windows (NT, 2000, XP), Unix, Linux, Mac, etc.

Se soportan una gran variedad de configuraciones; cuando el ancho de banda lo permite, es posible instalar el la aplicación cliente Java. Se pueden obtener accesos seguros utilizando herramientas estilo Terminal Services, ya sean soluciones propietarias u open source.

Para utilizar el cliente HTML, se necesita un Servidor de JSP y Java Servlet, y para implementar la funcionalidad del Web store se requiere el protocolo SSL, adicional al estándar http.

El servidor de aplicaciones JBoss puede ser instalado de manera stand alone o en el mismo servidor de la base de datos; se utiliza JMX (Java Management Extensions) para la administración del servidor. El servidor de la base de datos almacena los datos y la lógica de la aplicación, y se accede a él mediante el protocolo estándar JDBC.



13 Arquitectura de la Aplicación

Debido a que las aplicaciones de negocios cambian constantemente, es necesario utilizar nuevas tecnologías y siempre existe la necesidad de proveer soporte a funcionalidades adicionales. Las aplicaciones deben también soportar la incorporación de nuevas funcionalidades específicas para el cliente, aunque muchas veces no sean adecuadas para la integración con la funcionalidad central de la aplicación (por Ej. personalizaciones y ciertas extensiones).

Si bien es sabido que los requisitos para las aplicaciones cambian constantemente, son pocas las que están diseñadas para resistir cambios y agregados. Estas aplicaciones de negocios pueden tener una larga expectativa de vida y tender a proporcionar mayor funcionalidad en el tiempo, por lo cual es muy importante proporcionar un buen armazón que permita administrar este crecimiento de complejidad. En caso contrario, si no están diseñadas para soportarlo, se volverán inestables al añadir funcionalidad extra a la aplicación base.

ADempiere utiliza los siguientes principios de diseño a fin de crear una arquitectura que sea sustentable:

- Arquitectura MVC de Smalltalk (desconectado del Model-View-Controller)
- Desconexión Asíncrona de procesos vía mensajes.
- Motor de Reglas Explícito, para implementar la lógica compleja.
- Transacciones seguras de fallas y recuperación.

ADempiere tiene una Object Architecture (comparada con Object-Oriented, Object-like o las arquitecturas tradicionales), en la cual cada Objeto es tan independiente de otros Objetos como sea posible, incluyendo el desacoplado de las transacciones.

Las primeras versiones de la arquitectura de ADempiere se diseñaron a mediados de los 80 utilizando Smalltalk, uno de los primeros lenguajes y ambientes verdaderamente orientado a objetos. Otras raíces de la

arquitectura están basadas en el proyecto “Next Generation” de ADV/Org, que era muy similar al proyecto original R/3 de SAP.

13.1 Interfase de Usuario Inteligente

La interfase de usuario de la aplicación y las pantallas HTML son generadas en tiempo de ejecución, basada en reglas del Diccionario de la Aplicación. Como resultado se obtiene una interfase de usuario consistente, que permite navegar rápidamente en áreas de la aplicación que no son familiares. Esta metodología de generación de la interfase de usuario permite un rápido desarrollo y el sistema resultante es mucho más estable que en otras aplicaciones. Este método también permite que el layout de las pantallas pueda ser modificado o extendido y que se puedan generar nuevas pantallas, creadas por el administrador del sistema, sin necesidad de modificar el código; los usuario automáticamente ven las nuevas pantallas la próxima vez que ingresen a la aplicación.

El Diccionario de Datos sabe de las estructuras y dependencias, permitiendo al usuario el acceso directo mediante zoom desde una lista a la ventana del dato, donde puede actualizarlo o ingresar nueva información. Esto permite que, por ejemplo, un usuario pueda ingresar un nuevo cliente mientras carga una orden, sin salir de la ventana original.

Los usuarios pueden Consultar registros. Esto reduce el número de registros en una ventana, y le permiten ingresar uno o más criterios de selección en una ventana.

Por otro lado, un usuario con los permisos adecuados, puede personalizar los layout de las ventanas y puede acomodar ventanas para una situación y cliente específico. Todos los usuarios pueden establecer valores por defecto en los campos de sus pantallas, a fin de evitar la selección de valores utilizados comúnmente.

13.2 Reportes Inteligentes

En muchas otras aplicaciones, los Reportes son entidades separadas o agregadas. Los reportes de ADempiere están basados en el diccionario de datos. Al tener acceso a la definición desde el visor de reportes, es posible navegar dentro de un reporte desde una entidad referenciada en él, hacia otros reportes. Los links son generados automáticamente y son señalados mediante un subrayado en el mismo reporte. La navegación está sujeta a las definiciones de acceso y seguridad que fueran configuradas oportunamente.

Las Vistas de Negocios están diseñadas para los usuarios finales y permiten acceder a la información utilizando herramientas estándar de SQL, sin necesidad de crear joins de tablas con SQL. La mayoría de las Vistas de Negocio son generadas en base al Diccionario de la Aplicación.

Las salidas de los reportes pueden ser vistas en pantalla antes de enviarlas a una impresora o generar archivos en diferentes formatos (por Ej. Excel, HTML, Word y PDF).

13.2.1 Drill-down

Cuando utiliza drill-down se genera un nuevo reporte basado en la entidad seleccionada. Por ejemplo, en el reporte de una orden es posible navegar a las líneas de la misma haciendo un doble clic sobre la cabecera de la orden.

Adicionalmente el drill-down está disponible con las transacciones. Por ejemplo:

- Reportes donde un número mostrado es la sumatoria de otros números.
- Navegar desde un monto totalizado mensualmente hacia las transacciones originales.

13.2.2 Drill-across

Permite crear un nuevo reporte al usuario donde se utiliza una entidad específica. Por ejemplo, en un reporte de producto, un usuario puede seleccionar una línea específica (producto); de allí navegar hacia el detalle de una orden o factura, que muestre solamente las líneas donde aparece dicho producto.

13.2.3 Tipos de Reportes

ADempiere proporciona tres tipos de reportes:

- Reportes por listas desde cada ventana.
- Reportes Financieros.
- Vistas OLAP (utilizando la herramienta OLAP de Oracle u otras herramientas OLAP de terceras partes).

Las listas están basadas en información de las ventanas y es posible generar múltiples reportes para cada ventana en el sistema. Cualquiera de esos reportes pueden ser iniciados desde dentro de una ventana en particular o, alternativamente, colocarlos en el menú, incluyendo parámetros definidos por el administrador del sistema.

Los visores OLAP proporcionan diferentes dimensiones (como cuentas, productos, clientes) que serán LAP de terceros que seleccione el usuario. Los datos pueden ser almacenados también en datawarehouses de terceros que elija el usuario.

13.2.4 Personalización de Reportes

ADempiere diferencia la “vista” del “modelo”. La aplicación provee un número de vistas estándar predefinidas, pero es posible que el usuario cree vistas adicionales de los datos utilizando sentencias Select SQL provistas por el mismo. A diferencia de otras aplicaciones, el usuario no necesita resolver referencias a claves foráneas (que requeriría conocer el modelo de datos) o preocuparse por la seguridad de los datos, ya que ADempiere resuelve esos temas de manera automática.

Generalmente, la gente tiene diferentes preferencias en cuanto a la forma en que cada reporte debería mostrarse. Por ello ADempiere permite que el usuario defina los reportes a nivel del Sistema, Cliente, Organización o inclusive Usuario:

- Columnas del reporte
- Orden de las columnas
- Orden dentro del reporte
- Cabecera de las columnas
- Sumas, conteos de cantidad, mínimo, máximo, desviación, media y varianza (para las columnas numéricas).
- Agrupación

El lenguaje del reporte se encuentra basado en el lenguaje que el usuario escogió en el momento de ingresar a la aplicación y cada usuario puede tener uno diferente.

La selección de datos se hace mediante los parámetros del reporte ingresados cuando se inicia el reporte, o mediante el panel de Consulta avanzado, lo cual permite al usuario ingresar un criterio en un estilo “consulta por ejemplo” (“query by example”) extendido.

13.3 Seguridad ante Fallas

Generalmente las aplicaciones son diseñadas para ser seguras a fallas, lo cual asume que todos los trabajos y los datos son ingresados de manera correcta y consistente. En caso de fallas, los expertos deberán buscar las causas y verificar los daños producidos. El usuario normalmente nota el problema tiempo después que ha ocurrido y la realidad es que las aplicaciones algunas veces fallan.

En contraste, ADempiere ha sido diseñado para ser seguro ante fallas. Cada transacción puede ser repetida y regenerada. Muchas de las fallas que se producen son identificadas por el sistema y el usuario puede intentar reparar el problema. En caso de no ser posible la recuperación, el error es aislado y el resto del sistema continúa trabajando. El diseño de transacciones desacopladas de ADempiere permite esta posibilidad.

Cada transacción realiza solamente una tarea, por lo que es simple de estabilizar y aislar el impacto ante una falla, facilitando además su identificación.

La comunicación entre las transacciones individuales está basada en mensajes, permitiendo lotes asincrónicos de transacciones. Así mismo, es más fácil de implementar funcionalidad adicional, y el costo de agregarla en ADempiere es mucho menor que en otras aplicaciones.

El usuario puede continuar trabajando con restricciones menores si la transacción principal (por Ej. Un ajuste de inventario) es exitosa. Las transacciones restantes pueden ser generadas posteriormente, cuando el problema haya sido solucionado.

El sistema regularmente verifica si una transacción está completa; en caso de no estarlo, o de no ser consistente, da una falla y el administrador y el usuario son informados de ello mediante un mensaje.

A medida que las aplicaciones se van haciendo más complejas, la posibilidad de errores crece exponencialmente. ADempiere proporciona un marco o framework de validación, y si eso falla, aísla el problema asegurando una alta disponibilidad de las funciones centrales.

13.4 Seguridad del Sistema

ADempiere proporciona una infraestructura de seguridad completa y flexible para cumplir con las necesidades del usuario, y soporta función, seguridad de datos, como así también auditoria.

La función de seguridad está basada en Roles de Usuario, la cual controla el acceso a Ventanas, Reportes y Procesos.

Por otro lado, la seguridad de los datos está basada en Cliente y Organización, y es mantenida a nivel del contexto de seguridad de la base de datos. Este es un nivel adicional de seguridad posterior al login normal de usuario de la base de datos. Antes de acceder a cualquier dato, el usuario debe identificarse mediante un store procedure con un nombre de usuario, contraseña, rol y opcionalmente la preferencia de lenguaje. Todas las contraseñas se almacenan de manera encriptada.

La funcionalidad de auditoria incluye registro de accesos (que funciones/datos fueron utilizados), registro de cambios (que datos fueron cambiados, que valor tenían y cuales se establecieron; incluyendo datos borrados), como así también archivos (documentos y reportes generados). Si bien, el tema de seguridad es complejo para desarrollar, se brindan algunos aspectos relacionados con el tema, que muestran la flexibilidad y poder que proporciona ADempiere.

13.4.1 Roles

Definen el primer nivel de seguridad en ADempiere. Los usuarios ingresan a la aplicación con un Rol específico. Si bien un usuario puede tener muchos roles, el acceso a ADempiere se obtiene basado en el Rol que se escogió al momento de ingresar.

Los roles definen la Organización, Ventanas, Procesos, Formularios, Workflows y Tareas (en adelante llamadas entidades) a las que el usuario puede acceder. El usuario no ve ítems de menú a los que no tiene acceso; no es que los tiene deshabilitados, sino que sencillamente no los puede visualizar.

Los Roles también definen las acciones que el usuario puede efectuar en las entidades a las que tiene acceso.

13.4.2 Control de Roles

La definición de Rol permite que una serie de acciones pueda ser habilitada o deshabilitada para cada Rol en particular:

- **Mostrar Contabilidad:** le permite al Rol acceder a los Tabs y Ventanas con información contable. En caso de estar deshabilitado, los usuarios con este rol no podrán ver ni modificar información contable.
- **Reportes:** permite al Rol el acceso a reportes.
- **Exportar:** permite la exportación de datos. Para permitir la exportación, el Rol debe tener habilitado Reportes también.
- **Bloqueo Personal:** le permite al Rol bloquear registros para que no puedan ser accedidos por otro Rol.
- **Acceso Personal:** le permite al Rol acceder a registros que han sido bloqueados.
- **Solo Lectura:** controla que el Rol tenga permitido hacer modificaciones a los registros.
- **Entidad Dependiente:** controla si el acceso debe estar restringido para otras pantallas y procesos que usan ese registro; por ejemplo permitir que alguien que trate con Términos de Pago pueda ver Ordenes, Facturas, etc., donde se utilice algún Término de Pago.
- **Sobrescribir Precio Límite:** controla la posibilidad de sobrescribir los precios límites cuando se ingresan órdenes o facturas.
- **Mantener Log de Cambios:** determina si el sistema debe mantener un registro de los cambios efectuados por los usuarios de este Rol.
- **Acceso a Todas las Organizaciones:** controla el acceso a las Organizaciones. Si no está habilitado, es posible restringir el acceso a una organización asignada para un usuario específico.

Nivel de Preferencias: controla la posibilidad de los usuarios del rol de establecer preferencias a nivel de Cliente, Organización, Ventana o Usuario.

13.4.3 Acceso a Datos por Rol

Es el segundo nivel de seguridad de ADempiere. Para un determinado Rol y privilegios, es posible además establecer el acceso a tablas, columnas o registros específicos. Por Ejemplo:

- Que determinados usuarios solo puedan crear Ordenes de Venta con el Término de Pago Inmediato; así, no podrán seleccionar, por ejemplo, el Término de Pago Crédito.
- Prevenir que ciertos usuarios puedan utilizar determinadas cuentas contables en el Diario o ver información de esas cuentas.

13.4.4 Bloqueo Personal

Cuando un Rol tiene habilitado el Bloqueo aparece un icono de bloqueo en la barra de herramientas.

El bloqueo en posición abierta, indica que el registro está abierto a todos los usuarios, mientras que en posición cerrada significa que solo está abierto para el usuario que lo bloqueó y para aquellos que tengan habilitado Acceso Personal.

14 Estructura de la Información

ADempiere posee una avanzada estructura de la información, la cual permite realizar cambios estructurales en cualquier momento, si la limitación del viejo sistema de “Conjunto de Libros” utilizado por muchos competidores.

14.1 Los Multi's de ADempiere

En las aplicaciones que han ido evolucionando desde soluciones personalizadas, el diseño inicial de ellas no incluía la mayoría de la funcionalidad “Multi”. La misma fue agregada por encima de ellas en etapas posteriores, lo cual resulta en aplicaciones difíciles de extender y mantener, con una gran sobrecarga de trabajo (Como ejemplos podemos destacar el “Multi Reporting Currencies and Accounting Engine” de Oracle, o el “Extended GL” de SAP).

ADempiere en cambio, está diseñado con toda la funcionalidad Multi desde sus orígenes. Puede activarla o desactivarla si ya no la necesita más, en cualquier momento.

Este diseño resulta en una aplicación más fácil de mantener y extender, y con mayor estabilidad. Los beneficios obtenidos de ello son menores costos de implementación y mantenimiento, además de mayor funcionalidad.

14.1.1 Multi-Organización y Centros de Servicios

Esta característica posibilita que diferentes entidades organizacionales puedan compartir datos o asegurar que los datos privados no sean accesibles desde otras entidades. Seguridad y datos compartidos son un prerequisite indispensable para contar con la funcionalidad de centralización, outsourcing u operación de centros de servicios.

Muchas aplicaciones han procurado agregar esta característica, pero frecuentemente el concepto de datos compartidos y privados, es implementado replicando datos, con la sobrecarga y problemas de sincronización que ello trae aparejado.

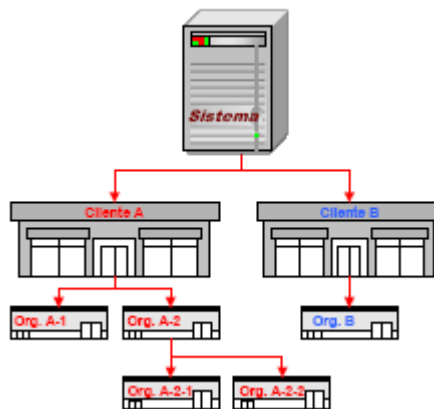
En su lugar, ADempiere fue diseñado para mantener diferentes organizaciones y soporta tres niveles de entidades:



· Sistema: Los datos a nivel de Sistema, son generalmente información de infraestructura, pero también podrían incluir socios de negocios, productos, esquemas contables y otros a lo largo de todo el sistema. El nivel de Sistema

es equivalente a la instalación de la base de datos y ADempiere provee herramientas que permiten la sincronización entre diferentes sistemas.

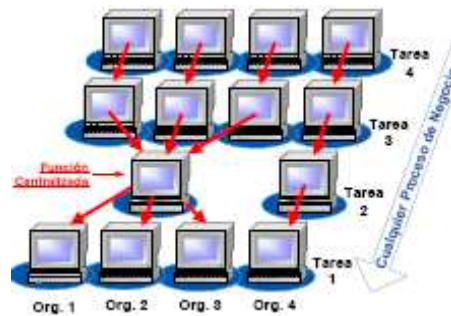
- Cliente: definen información y estructuras contables para una entidad – o grupo de entidades – como así también socios de negocios comunes, productos, etc.
- Organización (y su jerarquía asociada): es el nivel de transacciones; los niveles Sistema y Cliente no pueden efectuar transacciones, se necesita al menos tener una Organización para ello. Las Organizaciones pueden tener también sus propias estructuras de datos e información, sin que ellas estén compartidas con otras Organizaciones. También se pueden establecer jerarquías de organizaciones, en las cuales las organizaciones hijas pueden tener acceso a datos de su organización padre.



Los datos de cada nivel pueden ser ingresados o modificados solamente si el rol de usuario tiene privilegios para hacerlo en ese nivel. El usuario también puede ver y utilizar datos de niveles más altos en la jerarquía, pero no puede alterarlos.

ADempiere también le permite reorganizar la estructura de su organización o fusionar entidades.

Los Centros de Servicio son organizaciones virtuales que ejecutan transacciones por otras organizaciones. Como ejemplo de ellos podemos citar los servicios de compras centralizadas. Para ello los roles pueden ser configurados de manera tal que permitan a departamentos centrales u organizaciones externas acceder al área funcional, con acceso únicamente a la información necesaria. Los Centros de Servicio pueden acceder a múltiples organizaciones sin cambiar los roles, inclusive si las organizaciones tienen diferente información y estructuras contables.



14.1.2 Multi-Moneda

A medida que el comercio se va globalizando, la funcionalidad multi-monetaria va adquiriendo mayor importancia. Mientras que la mayoría de las aplicaciones soportan esta característica con significativas restricciones, ADempiere le permite olvidarse de ellas.

Las características de Multi-moneda de ADempiere incluyen:

Transacciones Multi-monetarias

- Posibilidad de realizar transacciones en una moneda diferente a la moneda contable.
- Posibilidad de revalorizar las transacciones.
- Cuentas bancarias en una moneda diferente a la contable.

Reportes Multi-monetarios

- Posibilidad de traducir transacciones o balances con el propósito de generar reportes.

Contabilidad Multi-monetaria

- Posibilidad de contabilizar transacciones paralelas en diferentes monedas.

Muchas aplicaciones que dicen soportar el esquema multi-monetario, solo soportan transacciones en múltiples monedas, y no cuentas bancarias en moneda extranjera (que son balanceadas en moneda extranjera en lugar de la moneda contable). En contraste, ADempiere soporta todos los aspectos de la funcionalidad multi-monetaria (por ejemplo listas de precio, moneda preferida del cliente, etc.) sin necesidad de copiar ni replicar transacciones. Una transacción puede ser contabilizada en una o muchas monedas. Iniciar, cambiar y discontinuar una moneda es fácil, debido a falta de una moneda de contabilidad primaria. Todas las monedas están a un mismo nivel.

14.1.3 Multi-Contabilidad

En muchas entidades se requiere la contabilización en múltiples y paralelos estándares contables, siendo una combinación de:

- Contabilización de Acumulación y basada en efectivo.
- Diferentes estándares contables (por Ej. US GAAP, UK SAP, etc.).
- Diferentes métodos de costeo de inventario (por Ej. Estándar, Promedio, FIFO, etc.).
- Diferentes monedas.

Generalmente se define un Conjunto de Libros, como un conjunto de transacciones con el mismo Plan de Cuentas, Calendario, Moneda Contable, Estándar de Contabilidad y Método de Costeo. Muchas veces esto es suficiente para pasar de un Conjunto a otro Conjunto de Libros y para muchas aplicaciones esta es la única opción disponible. Pero existen situaciones donde ello no es suficiente, por ejemplo si se producen diferencias de redondeo o de cambio inaceptables, o si se provoca un gran esfuerzo manual para hacerlo, o se pierde la posibilidad de efectuar rastreos de auditoria, o se producen retrasos considerables en la obtención de resultados. Algunas aplicaciones replican los datos de las transacciones a fin de permitir diferentes dimensiones contables.

ADempiere fue diseñado para soportar los requerimientos de multi-contabilidad. El concepto de Conjunto de Libros, que es utilizado por la mayoría de las aplicaciones, fue mejorado con el concepto de Esquema Contable.

Un Esquema Contable es la combinación de:

- Conjunto de Cuentas.
- Contabilidad basada en acumulación o efectivo.
- Estándar contable.
- Método de costeo.
- Moneda contable

En contraste con el Conjunto de Libros, el Calendario no forma parte directa del Esquema Contable, ya que podrían existir múltiples calendarios por esquema. El calendario se reduce solamente a una función de soporte de transacciones (abrir/cerrar períodos, asientos resumen, definición de asignaciones y facilitar el ingreso).

ADempiere, en contraste con la mayoría de las otras aplicaciones, diferencia las transacciones de las consecuencias contables resultantes, obteniendo así los siguientes beneficios:

- Los datos de las transacciones no están replicados.
- Un esquema contable puede ser agregado o discontinuado en cualquier momento.
- Puede generar información contable de transacciones históricas.
- Cualquier atributo puede ser modificado o reemplazado (y opcionalmente regenerar la contabilización)
- Los esquemas contables son fáciles de extender y mantener.
- El sistema es tolerante a fallos, pues puede ser corregido y regenerado.

Si las reglas contables predefinidas no fueran suficientes, es posible extenderlas mediante programación.

14.1.4 Multi-Impuestos

ADempiere soporta impuestos de venta y de valor agregado, incluyendo impuestos múltiples, por ejemplo impuestos estatales y provinciales. El motor de impuestos calcula automáticamente el impuesto correcto, su monto y fecha, basándose en la fecha de la transacción, categoría del producto, desde donde y hacia donde se efectúa la entrega y desde donde y hacia donde se emite la factura (por ejemplo ventas al exterior podrían estar exentas de impuesto).

14.1.5 Multi-Costeo

La utilización de diferentes métodos de costeo (Estándar, Promedio, Actual) puede reflejar distintos resultados financieros. ADempiere le permite configurar diferentes Esquemas Contables para reflejar distintos métodos de costeo. Puede también utilizar más de un método de costeo, por ejemplo por cuestiones legales contables, o para la toma de decisiones.

ADempiere mantiene información para los siguientes métodos:

- Costeo Estándar
- Costeo Actual (Última Orden de Compra, Última Factura, LIFO, FIFO)
- Promedio (Orden de Compra o Factura)

Los costos se pueden registrar en tres niveles: Cliente, Organización o por Lotes (Batch). El método de costeo se define para cada Esquema Contable. También puede especificar un método diferente de costeo, o nivel de costos, para una Categoría de Producto. Esto le permite disponer de una gran flexibilidad para el análisis financiero. Además puede cambiar el método de costeo en cualquier momento. Los costos se mantienen en su moneda contable.

14.1.6 Multi-Lenguaje

ADempiere provee la traducción de los siguientes elementos del sistema al lenguaje que usted requiera:

- Pantallas
- Reportes
- Mensajes
- Datos almacenados
- Transacciones

Muchas aplicaciones le permiten traducir pantallas, reportes y mensajes, pero solo unos pocos la traducción de datos y aún menos, la traducción de transacciones.

Además tiene la opción de cambiar el lenguaje del usuario del sistema y, finalmente, es posible que la emisión de los documentos sean realizados en el lenguaje de su cliente o proveedor, independientemente del lenguaje que usted utiliza en la aplicación. Los formatos de fechas y/o direcciones son también reemplazados con los del país de destino.

Al estar la traducción basada en diccionario, es mucho más consistente que otras aplicaciones que tienen herramientas que permiten traducir los distintos elementos.

14.2 Cambios en la Estructura de la Información

Muchas veces, después que una aplicación se encuentra en producción o inclusive durante la fase de implementación, los usuarios requieren cambios en la estructura de la información, debido a que encuentran que cierta información no es necesaria o que requieren de otra distinta para la toma de decisiones. O también debido a cambios en el negocio que pueden requerir cierta información adicional a la planificada inicialmente.

Muchas aplicaciones no permiten efectuar estos cambios, o lo permiten a costa de una nueva implementación desde cero del sistema. ADempiere sin embargo, permite agregar, modificar o eliminar dimensiones de información en cualquier momento, preservando la estructura OLAP (OnLine Analysis Processing) automáticamente.

15 Personalización e Interfases Externas

15.1 Diccionario de Datos

A diferencia de la mayoría de las aplicaciones, donde los desarrolladores deben diseñar, codificar y probar cada pantalla, ADempiere utiliza un concepto más avanzado de Diccionario de Datos Central, llamado Repositorio de Información, que permite facilitar esta tarea.

El Diccionario de Datos de ADempiere, alojado en la capa de metadatos, sabe como acceder a los datos y como se relaciona la información. Contiene definiciones de entidades de datos (tipos, validaciones, etc.), como se muestran (títulos sobre pantallas y reportes, ayudas, posición relativa con respecto a otros datos, etc.) y las reglas para mostrarlos. También se almacenan aquí las reglas de seguridad y acceso. Este diccionario es “activo”, significando con ello que es utilizado en tiempo de ejecución y es sensible al contexto.

Es extensible por el usuario y puede incluir reglas e información especificada por el usuario. También les permite a usuarios autorizados, agregar nuevas tablas, nuevas pantallas y datos adicionales sobre pantallas ya existentes en la aplicación. Toda esta información agregada está automáticamente disponible en los listados y reportes.

15.2 Personalización

Además de la posibilidad de personalizar las Interfases de Usuario, Reportes y Extensiones, ADempiere proporciona capacidades de personalización adicionales:

- Preferencias Default o elecciones preseleccionadas
 - o Preferencias de Login: Organización, Lenguaje, Fecha de Transacciones e Impresora.
 - o Preferencias definidas por el Usuario, tales como tipos de transacciones específicas.
- Personalización de la Barra de Menú, permitiendo guardar cualquier entrada en la barra (Ventanas, Procesos, Reportes) como un acceso rápido.
- La Terminología puede ser cambiada. Por ejemplo si los usuarios en lugar de Productos utilizan Ítems
 - o Artículos, o a la Organización la denominan Sucursal, etc.
- Los Textos de Ayuda pueden ser modificados y extendidos por el usuario para proporcionar sugerencias y ayudas específicas.

Las personalizaciones son definibles a diferentes niveles:

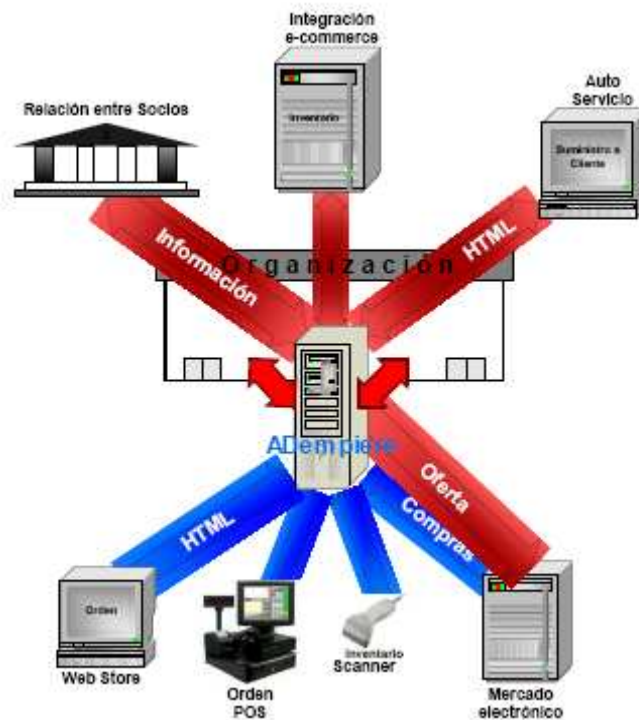
- Sistema o implementación
- Ventana, si es apropiado (por Ej. para preferencias)
- Cliente

- Organización
- Usuario Específico

Los niveles más específicos tienen preponderancia sobre los más bajos. Los cambios efectuados a nivel del sistema pueden ser guardados y definirse como personalizaciones si se requieren reaplicar luego de la instalación.

15.3 Integración Funcional

ADempiere integra completamente las funcionalidades ERP, CRM y Procesamiento Analítico, asegurando que las diferentes áreas funcionales dispongan de toda la información requerida para la toma de decisiones, sin necesidad de derivar información puesto que toda está basada en las transacciones originales.



Muchas aplicaciones no proporcionan esta integración funcional. En ADempiere ingresar o ver información de proyectos en las transacciones no requiere de pasos adicionales. Cuando ocurren excepciones, como ser proveedores debiendo dinero o clientes requiriendo fondos, ADempiere las trata sin necesidad de procesamientos adicionales.

15.4 Interfases

15.4.1 Vistas de Negocios

En caso de no ser suficientes las facilidades de reportes de ADempiere, se pueden utilizar herramientas de terceras partes, basadas en

SQL. ADempiere proporciona vistas de negocios, las cuales resuelven todas las referencias a claves foráneas y están “listas para usar”. No hay necesidad de conocer el modelo de datos, ni de desarrollar y mantener catálogos para utilizar las herramientas de terceros.

15.4.2 Exportación de Datos

ADempiere exporta todos sus datos de reportes a los siguientes formatos:

- Excel
- HTML
- XML
- Archivos planos de texto
- PDF
- PS
- Word
- Cubos OLAP

15.4.3 Importación de Datos

ADempiere importa datos desde XML, formatos de registros fijos, etc. Ya trae incorporados formatos predefinidos, pero el usuario puede definir sus propios formatos. ADempiere proporciona las interfases de acuerdo al OAGIS (Open Applications Group Integration Specification).

15.5 Extensiones

Además del diccionario interno de la aplicación, ADempiere tiene también la posibilidad de extender la aplicación utilizando Java Business API's. A diferencia de otras aplicaciones, las extensiones de clientes son posibles en ambientes hosteados y son preservadas durante las actualizaciones del producto a nuevas versiones.

15.5.1 Estructura de la Información

En caso que la estructura provista no sea suficiente, el usuario puede agregar campos a cualquier registro, con reglas de presentación y validación. El ingreso de datos puede ser obligatorio si se cumplen ciertas condiciones y la validación de datos puede estar basada en listas, tablas o funciones tales como callouts.

15.5.2 Scripting

Los scripting de ADempiere permiten al usuario extender la funcionalidad utilizando sintaxis Java. También puede ser utilizado para la conversión de datos al importarlos. ADempiere utiliza el lenguaje de scripting BeanShell.

15.5.3 Call-Out

Las extensiones funcionales son implementadas utilizando la tecnología de “callout”. Los clientes pueden proporcionar funcionalidad adicional en Java o inclusive funcionalidad nativa en C, por ejemplo para validaciones adicionales o alimentación de datos. Los callouts pueden ser invocados antes o después del ingreso de datos en cualquier campo. ADempiere asegura que los callouts no permitan la caída o corrupción del sistema.

15.5.4 Reglas

Los usuarios avanzados pueden extender y en ciertas áreas modificar las reglas base. Las reglas están organizadas en paquetes, asegurando que se mantiene la integridad de las transacciones. Las extensiones podrían utilizarse para generar entradas estadísticas o para necesidades de reportes especiales.

Actualmente, las reglas se utilizan para la creación de transacciones contables y para precios.

15.6 Comercio Electrónico (e-Commerce)

XML: OAGIS y OFX

ADempiere provee interfaces de acuerdo a OAGIS (Open Applications Group Integration Specification), las cuales están implementadas en XML o mediante Tablas de Interfaces.

El Open Applications Group es un consorcio sin fines de lucro, enfocado en las mejores prácticas y procesos basados en contenido XML para Integración de Aplicaciones y eBusiness. Es la mayor publicación en el mundo sobre contenido basado en XML para interoperabilidad entre software de negocios. ADempiere también soporta transacciones OFX (Open Financial Exchange), una especificación unificada para el intercambio electrónico de datos financieros entre negocios, consumidores e instituciones financieras a través de internet.

OFX soporta un amplio rango de actividades financieras incluyendo las bancarias o pagos de cuentas de consumidores y pequeñas empresas.

16 Otras Características

16.1 Imágenes y Adjuntos

ADempiere permite adjuntar múltiples notas, archivos externos (por Ej. Excel) o imágenes a cualquier registro del sistema. Si un registro tiene un adjunto, entonces se informa mediante un icono en la barra de herramientas.

16.2 Alertas

Un usuario puede definir situaciones de excepción para recibir notificaciones preactivas. Esta notificación puede ser una simple nota o corre un proceso o reporte. Por ejemplo, una lista de clientes con un cierto porcentaje sobre el límite de crédito, nuevos clientes, etc.

Por otro lado, en un segundo plano o background, ADempiere proporciona automáticamente reportes de problemas resultantes del monitoreo de consistencia y detección de fallas en el sistema.

16.3 Planificador

ADempiere proporciona un planificador integrado para reportes y procesos, y el resultado se envía mediante email a un usuario en particular o a una lista de mailing. Esto permite una fácil distribución de reportes.

16.4 Integración de e-mail

Emails estándar (y multi lenguajes) son generados por el sistema, para notificaciones tales como requerimientos de aprobación o alertas. También pueden ser utilizados para crear cartas a clientes o confirmaciones automáticas. Si ocurre un error, el usuario también puede enviar el mensaje de error con información de soporte generada por el sistema hacia una persona de soporte interna o al soporte de ADempiere directamente.

16.5 Ayuda

ADempiere proporciona un sistema de ayuda multinivel integrado y personalizable. Cada tarea, reporte, ventana y tab, tiene información general, cada campo tiene un “tool-tip” hint y un texto de ayuda.

Si la ayuda no es suficiente, el usuario puede hacer un “zoom” desde la ventana de ayuda hacia el sistema de ayuda online de ADempiere, para obtener información actualizada, consejos y secciones de Preguntas Frecuentes (FAQ).

Compiere ERP & CRM.

Compiere ERP & CRM es una sofisticada solución de negocios Open Source que se ha posicionado como una fuerte alternativa a los productos propietarios, es decir a aquellas aplicaciones que son desarrolladas y comercializadas por una organización comercial.

La mayoría de las soluciones ERP disponibles en el mercado actualmente proporcionan similar funcionalidad y muchas organizaciones evalúan estas soluciones basándose en las capacidades funcionales medidas en un instante de tiempo en particular. Este enfoque es común pero no es la metodología más apropiada para evaluar y seleccionar una solución de negocio a largo plazo. Este enfoque puede conducir a diferentes resultados cuando un producto es evaluado en diferentes momentos, a raíz de nuevas versiones que pueden ser liberadas al mercado.

El ciclo de vida de una solución ERP se estima generalmente en diez o incluso más años, y durante este período de tiempo la tecnología y los requisitos del negocio cambian. Así como la capacidad funcional de un producto es importante, es también muy importante tener en cuenta la tecnología en la que dicho producto está basado y la posibilidad que éste brinda para ser modificado y adaptado a las necesidades de la organización, las cuales se van renovando a medida que las reglas del negocio van cambiando. También es crítico asegurarse de que los cambios esenciales realizados no comprometan la posibilidad de migrar a futuras versiones del producto, y preserven la integridad de las modificaciones específicas efectuadas para su negocio.

El verdadero poder de Compiere queda demostrado cuando, siendo funcionalmente rico, tiene la posibilidad de incorporar los cambios específicos de su negocio y preservarlos en la liberación de nuevas versiones.

Fortalezas de Compiere

Flexibilidad

1) Compiere adopta estándares abiertos, lo cual permite:

- La estandarización, estabilidad e interoperabilidad de sistemas;
- Descripciones de datos y comportamientos claros, públicos y visibles.

2) Independencia de Hardware y Sistemas Operativos.

Viabilidad a largo plazo

1) Compiere se protege de la obsolescencia, cumpliendo con los estándares de la industria y utilizando un conjunto de herramientas que sostienen estos estándares:

- a) permite a Compiere cambiar los componentes fundamentales;
- b) asegura la disponibilidad de una gran base de desarrolladores que conocen las herramientas utilizadas.

2) La disponibilidad del código fuente reduce los riesgos de la no-disponibilidad de soporte a largo plazo.

3) Es sumamente escalable lo que permite sostener un crecimiento orgánico o explosivo producido, por ejemplo, por una adquisición.

4) No es dependiente de la viabilidad en el largo plazo de la organización responsable por el desarrollo del producto. Por ejemplo, Peoplesoft ha adquirido recientemente JD Edwards, causando una significativa incertidumbre en los usuarios finales de los productos de JD Edwards. Del mismo modo, Oracle ha tomado Peoplesoft con la revelada intención de convertir a los usuarios finales de Peoplesoft al producto Oracle Financials. La viabilidad continuada del software Open Source NO está sujeta a la supervivencia de ninguna organización en particular.

Bajo Costo de Propiedad (TCO)

1) Sin cargos por licencias de software (sujeto a la elección de la base de datos).

2) Bajo incremento del costo a medida que la cantidad de usuarios crece.

3) No tiene que pagar por las actualizaciones (upgrades) anuales.

4) No requiere adoptar costosos -y frecuentemente no garantizados- ciclos de actualización.

5) Bajos costos de contratos de soporte.

Ventajas del Open Source

Algunas de las ventajas que puede obtener con la utilización de una solución Open Source como Compiere son:

Reducción de la dependencia de un solo proveedor del producto

1) Minimiza el riesgo de tecnología propietaria.

2) Elimina la dependencia de un proveedor que provea las licencias.

Auto Dependencia

1) Proceso flexible en el desarrollo, con mayor enfoque en las necesidades específicas del negocio.

2) Mayor grado de participación y entendimiento entre el proveedor y el usuario final.

3) Independencia tecnológica.

- 4) Mejor receptividad para direccionar las necesidades locales y las oportunidades de negocios identificadas.
- 5) Las prioridades de desarrollo son manejadas por el usuario y NO por el proveedor.

Amplio rango de opciones de soporte

- 1) Soporte comercial brindado por múltiples organizaciones.
- 2) Soporte gratuito, disponible en:
 - a) comunidad de desarrolladores,
 - b) listas de correo,
 - c) archivos,
 - d) base de datos de soporte.
- 3) Las experiencias de soporte son generalmente más responsables que con las aplicaciones propietarias.

Técnicamente Superior

- 1) Los productos Open Source están más alineados con los estándares abiertos que los productos propietarios, alcanzando así un mayor grado de interoperabilidad.

La revisión permanente por parte de la comunidad de desarrolladores, lleva a productos generalmente de una calidad superior.

Soporte de Compiere

Una solución Open Source, muchas veces es asociada con un menor costo a lo largo de todo su ciclo de vida, pero también es percibida con un menor nivel de soporte y un alto riesgo, comparada con un sistema propietario. Este no es justamente el caso.

El nivel de soporte proporcionado por organizaciones de Open Source, puede ser considerablemente superior al proporcionado por un revendedor que distribuye aplicaciones de software propietarias. El primer motivo es que el código fuente está disponible, y por lo tanto puede ser modificado para resolver el problema localmente, a diferencia de los productos propietarios donde el código fuente normalmente no está al alcance de la organización que brinda el soporte; éstos dependen de su desarrollador para proporcionar una corrección. Y esto generalmente se hace en una nueva versión, unos seis a doce meses más tarde.

Adicionalmente, es posible obtener soporte entre la comunidad de desarrolladores, partners y usuarios del software, los cuales responden a las consultas realizadas en los foros, muchas veces en cuestión de horas e inclusive de minutos de realizado el requerimiento.

Además del soporte, la mayoría de las organizaciones busca obtener “garantías” de que el software adquirido está libre de defectos, o en caso de existir alguno, el mismo se solucionará rápidamente. La historia reciente y la experiencia indican que comprar un software a un proveedor no es garantía de libertad de errores. La realidad es que en soluciones de Open Source, la lista de errores es conocida y el código fuente está disponible para la organización de soporte, lo cual le permite corregir cualquier error que surja. Este no es el caso del software propietario, donde generalmente los errores no se publican y el código fuente no está disponible para las organizaciones que lo distribuyen y dan soporte.

Compiere Requerimientos de Infraestructura y Hardware

Infraestructura de red y hardware

Compiere tiene la capacidad de operar en una variada gama de redes y sistemas operativos. Esta flexibilidad le da al usuario la libertad de escoger el hardware y sistema operativo que mejor se adapten a sus necesidades individuales.

Sistemas Operativos

Compiere puede correr sobre un amplio rango de sistemas operativos, tales como Unix, Windows, Linux y Mac OS X, permitiendo al usuario elegir desde una amplia gama de sistemas operativos abiertos, hasta los sistemas propietarios ofrecidos por los proveedores tradicionales.

Servidores de Aplicaciones

Compiere utiliza el servidor de aplicaciones Jboss, por el cual no hay que abonar ningún cargo. Actualmente está en plan de desarrollo que Compiere corra también sobre IBM Websphere y sobre Oracle Application Server (OAS).

Para aquellas organizaciones que elijan utilizar OAS, éste puede ser licenciado por Oracle o utilizarse sin cargo adicional si se cuenta con un contrato de soporte con un Partner Certificado de Compiere.

Licencias del Software

NO existen cargos para el uso del software Compiere.

Licencias

1) Licencias de productos intermedios: no existen licencias o CALs requeridas para correr Compiere. Todos los productos utilizados por Compiere son productos abiertos de la industria estándar, los cuales están libres de cargos por licencias.

2) Licencias de Base de Datos: los usuarios de Compiere puede elegir entre adquirir su propia licencia de Oracle o adquirir las licencias pagando un cargo anual de soporte de Compiere, el cual proporciona un bajo costo de la base de datos Oracle como una aplicación embebida, bajo los términos del contrato negociado entre Oracle y Compiere Inc. en Febrero de 2005. Los usuarios pueden seleccionar otras bases de datos, algunas de las cuales son Open Source, u otros productos comerciales ofrecidos sin costo o a un bajo precio.

Gastos Recurrentes

1) Soporte de Hardware: los costos dependerán de la elección de hardware realizada por el usuario. La elección sobre el tipo de hardware que utilizará Compiere, dependerá de los requerimientos individuales del usuario y muchas veces, de las relaciones de éste con sus proveedores de hardware habituales.

2) Licencia de Mantenimiento de Base de Datos: vea los comentarios referidos antes en la sección de Licencias.

3) Mantenimiento (upgrades) del Software de Aplicación: los usuarios de Compiere tienen la posibilidad de descargar sin costo alguno todos los cambios y mejoras del producto y efectuar las migraciones de la base de datos utilizando recursos propios. El contrato de soporte de Compiere, también incluye la migración de la base de datos y soporte para actualizar a versiones posteriores de la aplicación.

4) Soporte del Software de Aplicación: el contrato de soporte puede ser adquirido con las organizaciones que dan soporte a Compiere, o efectuarlo el mismo usuario con recursos propios, muchas veces utilizando los foros de soporte de Compiere, los cuales son de acceso público y abierto.

5) Extensiones y Modificaciones: Compiere ha sido diseñado para facilitar las extensiones o modificaciones, realizadas por o para un usuario de Compiere. La incorporación de un Diccionario de Datos Activo (Active Data Dictionary) posibilita la modificación del diccionario, que puede ser efectuado muchas veces por personas que no tengan conocimientos de codificación y sin depender de proveedores externos.

También pueden ser efectuadas, si el usuario lo desea, por organizaciones como OPENBIZ, con un cargo básico.

Términos de la Licencia

Muchos softwares Open Source están licenciados bajo los términos de la GNU Public License. Esta licencia requiere que las modificaciones efectuadas al producto (distintas que las realizadas para uso interno) deban ser retornadas a la comunidad Open Source. El sistema Compiere ERP & CRM está licenciado bajo los términos de la Mozilla Public License. Esta licencia permite a los usuarios desarrollar funcionalidades adicionales y utilizarlas internamente ó inclusive licenciarla mediante un cargo a terceras partes, sin la obligación de retornar la mejora a la comunidad Open Source.

Los términos de la licencia de Compiere se encuentran detallados en <http://www.compiere.org/license.html>

Compiere ERP & CRM – Generalidades

Compiere brinda una funcionalidad completa, fácil de usar y de primer nivel para empresas del rango medio. A diferencia de los sistemas tradicionales, Compiere está organizado en procesos de negocios y no en módulos. Se suministra como un sistema unitario, integrado y completo, en lugar de una serie de módulos acoplados con transferencia de datos entre ellos. De esta manera el usuario obtiene una vista unificada del negocio, con procesos que involucran a toda la organización y no solo a unos cuantos departamentos o unidades tratados como islas. Con Compiere tiene todos los módulos en uno. Esta integración se aplica tanto al CRM (Administración de Relación con el Cliente), el Web Store (tienda web), como a la información del ERP tradicional.

¿Por qué Compiere está organizado reflejando el Proceso de Negocio?

El diseño de Compiere permite manejar los procesos de negocios, en lugar de los departamentos tradicionales; actualmente, y especialmente en el caso de las empresas medianas, los empleados frecuentemente realizan el proceso de negocio entero o procesos relacionados entre sí.

Conceptos de Compiere

Compiere proporciona servicios a múltiples clientes. Cada uno de ellos es una entidad, tal como una compañía padre o de máximo nivel equivalente. Cada cliente entonces tiene múltiples subsidiarias, departamentos, divisiones, llamadas organizaciones.

Se permiten efectuar transacciones entre las organizaciones. Por ejemplo, un pago por una organización de un gasto para otra organización resultará automáticamente en una transacción inter-organización en ambas, además de las entradas por el pago y el gasto.

Cada entidad externa con la cual la organización efectúa transacciones de negocio se denominan socios de negocios. Por ejemplo, clientes y proveedores son socios de negocios. Los empleados también son tratados como socios de negocios.

Cada transacción está asociada con un documento. Por ejemplo, facturas de venta, recibo de materiales, documentos de entregas, pagos a proveedores o recibos de clientes. Cada documento tiene predefinido un número de documento automático y es almacenado bajo ese número. También es posible adjuntar imágenes para cada documento. Además, para cada documento el usuario puede definir las consecuencias contables causadas por el procesamiento del mismo.

Procesos de Negocio de Compiere

A continuación describiremos brevemente los Procesos de Negocio que maneja Compiere. Una descripción más detallada de los mismos puede obtenerla en otro documento redactado a tal efecto.

Cotización a Ingresos

Cubre los procesos de negocios utilizados para la creación de cotizaciones, administración de ordenes de venta, facturación y recepción de dinero por pagos en efectivo (cobranzas en efectivo). Esta funcionalidad se integra con la Administración de la Cadena de Suministro (SCM) y con la Administración de Relaciones con el Cliente (CRM) de Compiere. En sistemas tradicionales, esta funcionalidad se encuentra en los módulos de ordenes de venta y cuentas a cobrar.

Requisición a Pago

Cubre el proceso de negocio utilizado para la creación de órdenes de compra, procesamiento de facturas de proveedores y pagos efectuados. Se integra con la Administración de la Cadena de Suministro (SCM). Esta funcionalidad se encuentra generalmente en los módulos de compras y cuentas a pagar.

Administración de Items Abiertos

Automatiza los procesos asociados con la entrada y asignación de dinero en efectivo recibido de los clientes y los pagos efectuados a los proveedores. Aquí puede también efectuar la conciliación bancaria y libros de caja. Al momento de la conciliación, Compiere provee funciones que le permiten la conciliación de pagos en tránsito y cargos bancarios o la creación de pagos por transferencias directas.

Administración de Relaciones con el Cliente (CRM)

Es un módulo integrado que provee una vista lógica de todas las actividades relacionadas con clientes y prospectos. En contraste con los sistemas de CRM tradicionales, no existe la necesidad de efectuar procesos batch ni sincronizaciones con la funcionalidad del backoffice.

Administración de Relaciones de Socios

La Administración de Relaciones de Socio liga diferentes clientes uno al otro, posibilitando manejar la distribución principal, pedidos de servicio, y gastos de marketing. Ello facilita la provisión de servicios compartidos

(centralizados) de una entidad organizacional a otras entidades organizacionales. Esta funcionalidad posibilita a las organizaciones que son propietarias de partes u otras organizaciones enteras (tales como operaciones de franquicias), a proveer servicios centralizados para las operaciones remotas.

Administración de la Cadena de Suministro (Abastecimiento)

Cubre todas las actividades de administración de materiales, incluyendo recepciones, entregas, movimientos y administración y procesamiento de tomas de stock.

Análisis de Resultados

Cubre el costeo y dimensiones contables de la aplicación. Esta funcionalidad generalmente se encuentra en los módulos de Reportes y Contabilidad General, como también en los módulos que generan entradas contables.

Web Store y Auto-Servicio de Socio de Negocio

El Web Store de Compiere, permite a una organización mantener y operar mediante la web. La información disponible en el web store es compartida con la aplicación estándar, sin requerirse sincronización ni integraciones adicionales.

Los componentes del web store pueden ser customizados para adecuarse al look-and-feel del sitio web existente. Proporciona además la posibilidad funcional de un auto servicio para permitir a los Socios de Negocio ver sus propias transacciones online con un apropiado nivel de seguridad.

Los Multi's de Compiere

En las aplicaciones que han ido evolucionando desde soluciones personalizadas, el diseño inicial de ellas no incluía la mayoría de la funcionalidad "Multi". La misma fue agregada encima de ellas en etapas posteriores, lo cual resulta en aplicaciones difíciles de extender y mantener, con una gran sobrecarga de trabajo (Como ejemplos podemos destacar el "Multi Reporting Currencies and Accounting Engine" de Oracle, o el "Extended GL" de SAP).

Compiere en cambio, está diseñado con toda la funcionalidad Multi desde sus orígenes. Puede activarla o desactivarla si ya no la necesita más, en cualquier momento.

Este diseño da como resultado una aplicación mas fácil de mantener y extender y también con mayor estabilidad. Los beneficios medibles son menores costos de implementación y mantenimiento, además de mayor funcionalidad.

Multi-Organización

Esta característica posibilita que diferentes entidades organizacionales puedan compartir datos o asegurar que los datos privados no sean accesibles desde otras entidades.

Seguridad y datos compartidos son un prerequisite indispensable para contar con la funcionalidad de centralización, outsourcing u operación de centros de servicios.

Muchas aplicaciones han procurado agregar esta característica, pero frecuentemente el concepto de datos compartidos y privados, es implementado replicando datos, con la sobrecarga y problemas de sincronización que ello trae aparejado.

En su lugar, Compiere fue diseñado para mantener diferentes organizaciones y soporta tres niveles de entidades:

- Sistema: Los datos a nivel de Sistema, son generalmente información de infraestructura, pero también podrían incluir socios de negocios, productos, esquemas contables y otros a lo largo de todo el sistema. El nivel de Sistema es equivalente a la instalación de la base de datos y Compiere provee herramientas que permiten la sincronización entre diferentes sistemas.

- Cliente: definen información y estructuras contables para una entidad – o grupo de entidades – como así también socios de negocios comunes, productos, etc.

- Organización (y su jerarquía asociada): es el nivel de transacciones; los niveles Sistema y Cliente no pueden efectuar transacciones, se necesita al menos tener una Organización para ello. Las Organizaciones pueden tener también sus propias estructuras de datos e información, sin que ellas estén compartidas con otras Organizaciones. También se pueden establecer jerarquías de organizaciones, en las cuales las organizaciones hijas pueden tener acceso a datos de su organización padre.

Los datos de cada nivel pueden ser ingresados o modificados solamente si el rol de usuario tiene privilegios para hacerlo en ese nivel. El usuario también puede ver y utilizar datos de niveles más altos en la jerarquía, pero no puede cambiarlos.

Compiere también le permite reorganizar la estructura de su organización o fusionar entidades.

Los Centros de Servicio son organizaciones virtuales que ejecutan transacciones por otras organizaciones. Como ejemplo de ellos podemos citar

los servicios de compras centralizadas. Para ello los roles pueden ser configurados de manera tal que permitan a departamentos centrales u organizaciones externas acceder al área funcional, con acceso únicamente a la información necesaria. Los Centros de Servicio pueden acceder a múltiples organizaciones sin cambiar los roles, inclusive si las organizaciones tienen diferente información y estructuras contables.

Multi-Moneda

A medida que el comercio se va globalizando, la funcionalidad multi-monetaria va adquiriendo mayor importancia. Mientras que la mayoría de las aplicaciones soportan esta característica con significativas restricciones, Compiere le permite olvidarse de ellas.

Las características de Multi-moneda de Compiere incluyen:

Transacciones Multi-monetarias

- Posibilidad de realizar transacciones en una moneda diferente a la moneda contable.
- Posibilidad de revalorizar las transacciones.
- Cuentas bancarias en una moneda diferente a la contable.

Reportes Multi-monetarios

- Posibilidad de traducir transacciones o balances con el propósito de generar reportes.

Contabilidad Multi-monetaria

- Posibilidad de contabilizar transacciones paralelas en diferentes monedas.

Muchas aplicaciones que dicen soportar el esquema multi-monetario, solo soportan transacciones en múltiples monedas, y no cuentas bancarias en moneda extranjera (que son balanceadas en moneda extranjera en lugar de la moneda contable).

En contraste, Compiere soporta todos los aspectos de la funcionalidad multi-monetaria (por ejemplo listas de precio, moneda preferida del cliente, etc.) sin necesidad de copiar ni replicar transacciones. Una transacción puede ser contabilizada en una o muchas monedas. Iniciar, cambiar y discontinuar una moneda es fácil, debido a falta de una moneda de contabilidad primaria. Todas las monedas están a un mismo nivel.

Multi-Contabilidad

En muchas entidades se requiere la contabilización en múltiples y paralelos estándares contables, siendo una combinación de:

- Contabilización de Acumulación y basada en efectivo.
- Diferentes estándares contables (por ej. US GAAP, UK SAP, etc.).

- Diferentes métodos de costeo de inventario (por ej. Estándar, Promedio, FIFO, etc.).
- Diferentes monedas.

Generalmente se define un Conjunto de Libros, como un conjunto de transacciones con el mismo Plan de Cuentas, Calendario, Moneda Contable, Estándar de Contabilidad y Método de Costeo. Muchas veces esto es suficiente para pasar de un Conjunto a otro Conjunto de Libros y para muchas aplicaciones esta es la única opción disponible. Pero existen situaciones donde ello no es suficiente, por ejemplo si se producen diferencias de redondeo o de cambio inaceptables, se provoca un gran esfuerzo manual para hacerlo, se pierde la posibilidad de efectuar rastreos de auditoria, o se producen retrasos considerables en la obtención de resultados. Algunas aplicaciones replican los datos de las transacciones a fin de permitir diferentes dimensiones contables.

Compiere fue diseñado para soportar los requerimientos de multi-contabilidad. El concepto de Conjunto de Libros, que es utilizado por la mayoría de las aplicaciones, fue mejorado con el concepto de Esquema Contable.

Un Esquema Contable es la combinación de:

- Conjunto de Cuentas.
- Contabilidad basada en acumulación o efectivo.
- Estándar contable.
- Método de costeo.
- Moneda contable

En contraste con el Conjunto de Libros, el Calendario no forma parte directa del Esquema Contable, ya que podrían existir múltiples calendarios por esquema. El calendario se reduce solamente a una función de soporte de transacciones (abrir/cerrar períodos, asientos resumen, definición de asignaciones y facilitar el ingreso).

A diferencia de la mayoría de las otras aplicaciones, Compiere diferencia las transacciones de las consecuencias contables resultantes, obteniendo así los siguientes beneficios:

- Los datos de las transacciones no están replicados.
- Un esquema contable puede ser agregado o discontinuado en cualquier momento.
- Puede generar información contable de transacciones históricas.
- Cualquier atributo puede ser modificado o reemplazado (y opcionalmente regenerar la contabilización)
- Los esquemas contables son fáciles de extender y mantener.
- El sistema es tolerante a fallos, pues puede ser corregido y regenerado.

Es posible extender, mediante programación, las reglas contables, si las predefinidas no son suficientes.

Multi-Impuestos

Compiere soporta impuestos de venta y de valor agregado, incluyendo impuestos múltiples, por ejemplo impuestos estatales y provinciales. El motor de impuestos calcula automáticamente el impuesto correcto, su monto y fecha, basándose en la fecha de la transacción, categoría del producto, desde donde y hacia donde se efectúa la entrega y desde donde y hacia donde se emite la factura (por ejemplo ventas al exterior podrían estar exentas de impuesto).

Multi-Costeo

La utilización de diferentes métodos de costeo (Estándar, Promedio, Actual) puede reflejar distintos resultados financieros. Compiere le permite configurar diferentes Esquemas Contables para reflejar distintos métodos de costeo. Le permite utilizar más de un método de costeo, por ejemplo por cuestiones legales contables o para la toma de decisiones.

Compiere mantiene información para los siguientes métodos:

- Costeo Estándar
- Costeo Actual (Ultima Orden de Compra, Ultima Factura, LIFO, FIFO)
- Promedio (Orden de Compra o Factura)

Los costos se pueden registrar en tres niveles: Cliente, Organización o por Lotes o Batch. El método de costeo se define para cada Esquema Contable. También puede especificar un método diferente de costeo o nivel de costos para una Categoría de Producto. Esto le permite disponer de una gran flexibilidad para el análisis financiero.

Además puede cambiar el método de costeo en cualquier momento. Los costos se mantienen en su moneda contable.

Multi-Lenguaje

Compiere provee la traducción de los siguientes elementos del sistema al lenguaje que usted requiera:

- Pantallas
- Reportes
- Mensajes
- Datos almacenados
- Transacciones

Muchas aplicaciones le permiten traducir pantallas, reportes y mensajes, pero solo unos pocos la traducción de datos y aún menos, la traducción de transacciones.

Además tiene la opción de cambiar el lenguaje del usuario del sistema y finalmente es posible que la emisión de los documentos sean realizados en el lenguaje de su cliente o proveedor, independientemente del lenguaje que usted utiliza en la aplicación. Los formatos de fechas y/o direcciones son también reemplazados con los del país de destino.

Al estar la traducción basada en diccionario, es mucho más consistente que otras aplicaciones que tienen herramientas para efectuar dichas traducciones de los distintos elementos.

Cambios en la Estructura de la Información

Muchas veces, después que una aplicación se encuentra en producción o inclusive durante la fase de implementación, los usuarios requieren cambios en la estructura de la información, debido a que encuentran que cierta información no es necesaria o que requieren de otra distinta para la toma de decisiones. O también debido a cambios en el negocio que pueden requerir cierta información adicional a la planificada inicialmente.

Muchas aplicaciones no permiten efectuar estos cambios, o lo permiten a costa de una nueva implementación desde cero del sistema. Compiere sin embargo, permite agregar, modificar o eliminar dimensiones de información en cualquier momento, preservando la estructura OLAP (OnLine Analysis Processing) automáticamente.

Customizaciones e Interfaces Externas

Diccionario de Datos

A diferencia de la mayoría de las aplicaciones, donde los desarrolladores deben diseñar, codificar y probar cada pantalla, Compiere utiliza un concepto más avanzado de Diccionario de Datos Central, llamado Repositorio de Información, que permite facilitar esta tarea.

El Diccionario de Datos de Compiere, alojado en la capa de metadatos, sabe como acceder a los datos y como se relaciona la información. Contiene definiciones de entidades de datos (tipos, validaciones, etc.), como se muestran (títulos sobre pantallas y reportes, ayudas, posición relativa con respecto a otros datos, etc.) y las reglas para mostrarlos. También se almacenan aquí las reglas de seguridad y acceso. Este diccionario es “activo”, significando con ello que es utilizado en tiempo de ejecución y es sensible al contexto. Es extensible por el usuario y puede incluir reglas e información especificada por el usuario. Permite a usuarios autorizados, agregar nuevas tablas, nuevas pantallas y datos adicionales sobre pantallas ya existentes en la aplicación. Toda esta información agregada está automáticamente disponible en los listados y reportes.

Vistas de Negocios

En caso de no ser suficientes las facilidades de reportes de Compiere, se pueden utilizar herramientas de terceras partes, basadas en SQL.

Exportación de Datos

Compiere exporta todos sus datos de reportes a los siguientes formatos:

- Excel
- HTML
- XML
- Archivos planos de texto
- PDF
- PS
- Word
- Cubos OLAP

Importación de Datos

Compiere importa datos desde XML, formatos de registros fijos, etc. Ya trae incorporados formatos predefinidos, pero el usuario puede definir sus propios formatos. Compiere proporciona las interfases de acuerdo al OAGIS (Open Applications Group Integration Specification).

Extensiones

Además del diccionario interno de la aplicación, Compiere tiene también la posibilidad de extender la aplicación. Estas extensiones son preservadas durante las actualizaciones del producto a nuevas versiones.

Call-out

Las extensiones funcionales son implementadas mediante la tecnología de “callout”. Los clientes pueden agregar funcionalidad adicional en Java o C, a fin de efectuar validaciones adicionales. Los Callouts pueden ser invocados antes o después que un dato es ingresado. Compiere asegura que los callouts no permitan la caída o corrupción del sistema.

Workflow

Compiere soporta el BPM (Business Process Management) y está basado en los estándares de la Workflow Management Coalition y la OMG.

En contraste con otros ERP y CRM, los Workflow no están “sobre” la aplicación; Compiere está basado en Workflows. El motor de Workflow de Compiere es el corazón del administrador de transacciones. Esto significa que todos los procesos en Compiere, son automáticamente manejados por

workflows y son fáciles de extender y modificar. Al estar totalmente integrados a Compiere, son fáciles de mantener y pueden proporcionar mucha mayor funcionalidad que los workflows externos o como ad-on que son ofrecidos por otros proveedores de ERP y CRM.

Compiere provee tres tipos de Workflows:

- **General:** sirven como guía para la realización de alguna tarea. Por ejemplo, los wizards de configuración. Estos workflows son iniciados por el usuario desde el menú.
- **Procesamiento de Documentos:** que controlan los pasos al procesar un documento y son iniciados de manera automática. Estos pueden ser extendidos, como por ejemplo en una Orden de Compra, para requerir aprobación si el importe excede un cierto valor.
- **Valor de Documentos:** se inicia automáticamente cuando una entidad cumple con una cierta condición especificada por el usuario. Por ejemplo requerir aprobación para establecer el crédito de un cliente nuevo.

Los pasos de un Workflow pueden determinar ciertas acciones, como por ejemplo iniciar Procesos Automáticos, una acción de usuario, asignar variables, esperar una selección del usuario, etc.

Compiere provee la administración dinámica de prioridades, posibilitando utilizar el Workflow para ruteos de Call Centers o soporte basado en prioridades del cliente. Es posible definir reglas escalables por inactividad, que envíen alertas a los responsables del Workflow y/o supervisor. El responsable puede ser una persona, un grupo o el supervisor de una organización y además diferentes pasos del Workflow pueden tener distintos responsables asignados.

Seguridad

La seguridad está basada en Roles de Usuario, la cual controla el acceso a Pantallas, Reportes y Procesos. La seguridad de los Datos para los Clientes y Organizaciones es mantenida a nivel del contexto de seguridad de la base de datos. Esto es un nivel adicional de seguridad posterior al login normal de la base de datos. Antes de acceder a cualquier dato, el usuario debe identificarse con un nombre de usuario, contraseña, rol de usuario y opcionalmente su preferencia de lenguaje. Todas las contraseñas son almacenadas en forma encriptada.

Roles: definen el primer nivel de seguridad en Compiere. El usuario se identifica en Compiere con un Rol específico y ve las organizaciones, pantallas, procesos, formularios, workflows y tareas a las que el Usuario puede acceder. El usuario no ve los ítems de menú a los que no puede acceder y el Rol controla una serie de acciones que son habilitadas o deshabilitadas para cada Rol en particular (Por ejemplo ver información

contable, poder exportar, poder ejecutar Reportes, poder actualizar información, acceso a organizaciones, etc.).

Roles para Acceso a Datos: es el segundo nivel de seguridad que dispone Compiere. La seguridad de acceso para un determinado Rol, puede ser refinada adicionalmente definiendo los accesos a tablas, columnas o registros específicos. Por ejemplo, un usuario específico que solo pueda crear Ordenes de Venta con la condición de pago inmediata; para este caso no dispondrá de la posibilidad de seleccionar cuenta corriente. Deshabilitar el acceso a un usuario a determinadas cuentas contables, en cuyo caso ellas no podrán ser utilizadas por ese usuario, ni podrá ver los balances para esas cuentas.

OASISERP

FEDETICAM, es la FEDERACIÓN DE EMPRESAS DE TECNOLOGÍAS DE LA INFORMACIÓN DE CASTILLA LA MANCHA. En FEDETICAM, se engloban las APETI (Asociaciones Provinciales de Empresas de Tecnologías de la Información).

En FEDETICAM están incluidas empresas que comercializan hardware, que desarrollan software, que imparten formación en TIC's, y también empresas especializadas en comunicaciones.

OASIS está desarrollado totalmente con herramientas de software libre.

Las ventajas que proporciona el Software Libre frente al Software Propietario:

- 1.- Libertad para modificar el software según las necesidades.
- 2.- Seguridad, ya que se dispone del código fuente del programa y ello permite la revisión del mismo. De esa forma se puede mejorar al detectar posibles fallos de seguridad en el software.
- 3.- Confiabilidad, pues al disponer libremente del código fuente, este es revisado por muchos usuarios mejorando la calidad del mismo.
- 4.- Portabilidad, ya que al disponer del código fuente es mucho más sencillo adaptar los programas para su funcionamiento en diferentes arquitecturas (de ordenadores). En cambio, el Software Propietario solo se puede utilizar en aquellas arquitecturas para las que se diseñó.
- 5.- Precio, pues al no tener restricciones en la distribución del software junto al código fuente esto hace que el costo sea muy bajo, e incluso cero. En cambio, el Software Propietario se caracteriza por el pago de licencias de uso por cada copia del programa, lo que encarece notablemente su utilización.
- 6.- Al no depender de una empresa propietaria y desarrolladora única, el programa permanece vivo y no depende de los avatares del tiempo con respecto a dicha empresa (puede desaparecer y dejar a los usuarios de su programa 'colgados').

La Planificación de Recursos Empresariales o ERP (Enterprise Resource Planning), es un conjunto de sistemas de información para la gestión de una empresa, que integra sus diferentes departamentos.

Hay tres características que distinguen a un ERP de otros sistemas informáticos de gestión. El ERP es un sistema integral, modular y adaptable.

De esta manera, a través de aplicaciones ERP, en vez de estar los programas trabajando de forma independiente unos de otros y sin tener una

conexión entre sí, trabajan de una forma integrada que permite la interconexión de todos ellos.

Los ERP permiten controlar los diferentes procesos de las empresas entendiendo que todos los departamentos se relacionan entre sí. Si la empresa no utiliza un ERP, necesitará tener varios programas que gestionen sus procesos con la desventaja de que al no estar integrados, la información se duplicará y se producirán errores en la información como consecuencia de las diferentes capturas de datos.

Otra diferencia con respecto a otras aplicaciones informáticas es que el ERP se divide internamente en módulos, lo que permite que se vayan instalando en función de las necesidades y requerimientos del cliente. Los ERP trabajan tratando a la empresa como un conjunto de departamentos interrelacionados por la información que comparten.

Los ERP están creados para adaptarse a las necesidades de cada empresa. Esto se logra en el momento de la configuración o parametrización de los procesos de acuerdo a las necesidades de cada caso concreto.

OASIS es un ERP que está desarrollado en PHP-GTK2, Base de Datos PostgreSQL y Ágata Report. Se ha desarrollado completamente con filosofía de software libre.

OASIS puede funcionar en red (intranet) o monopuesto. Al tratarse de software libre, está exenta de pagos por royalties o licencias de uso. El producto se distribuye con las fuentes, habiendo sido verificado y testeado.

Cubre las necesidades de las pymes al disponer de un software libre para Gestión (sin costes), fácil de usar, y compatible con cualquier Sistema Operativo (Linux, Windows). **OASIS** permite llevar la Gestión completa de una empresa.

OASIS es multiusuario, multiempresa, multisucursal, multimarca y multialmacén. Es altamente parametrizable y extremadamente flexible, siendo posible disponer de configuraciones personalizadas, incluso para cada usuario.

OASIS permite trabajar con diferentes códigos de barras, tanto en impresión como en lectura. Por ejemplo, en el módulo de Fichajes de Operarios/Mecánicos de SAT es posible realizar fichajes a través de lectores de códigos de barras.

OASIS dispone de exportación de datos de forma directa, a través de ficheros en formato CSV, lo cual hace al mismo una herramienta ideal para el intercambio de información con otras aplicaciones de tipo ofimático o de gestión diferida (asesorías fiscales, etc).

Está adaptado a las normas NIC/NIIF (Normas Internacionales de Contabilidad) de reciente entrada en vigor. Permite generar y validar documentos firmados electrónicamente compatibles con la AEAT española.

OASIS adopta un compromiso firme con el ahorro de papel y por un consumo sostenible apoyando con ello al Medio Ambiente, ya que todos los informes son visualizados previamente antes de ser impresos. Será el propio usuario el que decida si es o no necesaria su impresión a papel. Todos los documentos se pueden imprimir a diferentes formatos, tales como TXT, CSV, PDF, HTML.

OASIS dispone de autoarchivo de los documentos electrónicos emitidos y recibidos.

OASIS cumple la LOPD (Ley de Protección de Datos), controlando los accesos al sistema mediante usuarios y permisos.

Este ERP dispone de un completo manual de usuario y ayuda en línea siendo el mismo sensible al contexto y accesible desde cualquier opción de la Aplicación.

1. - Módulos.

Los módulos principales que componen el ERP son:

- Parámetros de la Aplicación.
- Compras.
- Ventas.
- Gestión de Almacén.
- Servicio Técnico y Postventa. S.A.T.
- Preventa y Promoción Comercial.
- Gestión de Cobros y Pagos. Previsión de Tesorería. Cartera.
- Contabilidad e Informes Financieros.
- Impuestos.
- Gestión de la Aplicación.

OASIS al estar diseñado y pensado totalmente de forma modular permite adaptar, desarrollar e incorporar nuevos módulos de forma sencilla y ágil. Todos los módulos están interrelacionados entre sí.

La modularización es transparente al usuario final de manera que éste solo ve aquellos módulos y opciones que tiene instalados y establecido el correspondiente permiso sobre los mismos.

Módulo de parámetros de la aplicación. Los parámetros de la aplicación permiten definir y modificar de forma sencilla los procesos y la

interrelación entre los diferentes módulos, así como la definición de la forma de trabajo particular de cada tipo de empresa.

El módulo de parámetros permite configurar desde aspectos referentes al entorno de trabajo (iconos, menús favoritos, preferencias, colores, etc), hasta cualquier elemento que conforman los diferentes módulos.

El módulo de parámetros a su vez se subdivide en parámetros generales, parámetros de almacén, parámetros para la gestión de cobros y pagos, y parámetros de servicio técnico y postventa.

En el módulo de parámetros se definen aspectos tales como trabajar con múltiples sucursales y/o almacenes, marcas, familias, tipos de documento, series de numeración, países, poblaciones, entidades financieras, etc.

Módulo de compras.

- Contempla la gestión de proveedores.
- Trabaja con diversos tipos de documento. Pedidos, Albaranes y Facturas.
- Asocia diferentes formatos a cada tipo de documento.
- Recepción de mercancías.
- Conformar facturas con los albaranes de compra.
- Enlaza automáticamente con el módulo de contabilidad y la cartera de pagos/ tesorería.
- Contempla el tratamiento de la recepción y verificación de Facturas Electrónicas, mediante firma con certificado reconocido por la AEAT.
- Gestiona los números de serie de los artículos.
- Contempla las fechas de finalización de garantía/caducidad.
- Existe un histórico de compras para su consulta en cualquier momento.
- Permite obtener diferentes informes de análisis de compras por proveedor, familia, etc.
- Total trazabilidad de documentos entre módulos.

Módulo de ventas.

- Contempla la gestión de los clientes.
- Control de límites de crédito, diferentes domicilios, formas de pago, etc.
- Trabaja con diversos tipos de documento. Ofertas/Presupuestos, Pedidos, Albaranes y Facturas, totalmente parametrizables.
- Asocia diferentes formatos a cada tipo de documento.
- Contempla el tratamiento de la Facturación Electrónica, mediante firma con certificado reconocido por la AEAT.
- Utiliza impresión de facturas en modo totalmente gráfico, con anagrama de la empresa en formato bmp, jpg, etc. Definible.
- Pensando en la diferente tipología de los clientes, conviven las impresiones en formato texto y en formato gráfico.
- Imprime información en códigos de barras.
- Posibilita diferentes tipos de líneas de facturación que son definibles por el usuario. Artículos, Mano de Obra, Servicios subcontratados, Portes, Recargos, Textos descriptivos, etc.
- Permite definir y trabajar con kits o conjuntos para facturación.
- Contempla diferentes descuentos por categorías de clientes.
- Existe un histórico de ventas y de facturación donde quedan guardados absolutamente todos los pedidos realizados, pudiéndose volver a realizar cualquier reimpresión del documento.
- Enlaza automáticamente con el módulo de contabilidad y la cartera de cobros/tesorería.
- Permite obtener diferentes informes de análisis de ventas por cliente, familia, producto, etc.
- Total trazabilidad entre documentos entre módulos.

Módulo de Gestión de Almacén.

- Trabaja con diferentes almacenes por empresa / sucursal.

- Emite informes de artículos, separados por ventas, compras, familias, grupos de descuentos, clientes, precios, existencias, tipos de pedidos, etc.
- Incorpora cuatro precios por cada artículo (p.v.p., coste, coste promedio y última compra).
- Emite informes de artículos diseñados por el propio usuario.
- Obtiene informes de inventario valorado a cualquier fecha.
- Realiza el pedido de reposición de artículos a proveedores.
- El cambio de precios, se realiza de forma totalmente automática, a partir de la tarifa de cada fabricante o por un incremento en porcentaje.
- Gestiona por cada artículo, entre otros datos, los máximos, mínimos, cantidades de embalaje, existencias negativas, obsolescencia, etc.
- Realiza inventarios de artículos a día fijo o permanente, emitiéndose un listado de recuento ordenado por artículo o ubicación.
- Las valoraciones del almacén se realizan en el instante en que se desee, emitiéndose por pantalla y/o impresora en formato resumido o detallado por almacén y/o familia.
- Contempla la clasificación ABC de los artículos en importes y en unidades, de acuerdo a los consumos de cada una de las referencias y a unos rangos desde/hasta que se solicitan al operador por la pantalla. El análisis ABC de los artículos se realiza a partir del fichero histórico de movimientos.
- Habilita un módulo de agenda de Almacén para atención de venta telefónica a clientes.
- Gestiona los números de Serie de los Artículos (Compras y Ventas).
- Realiza pedidos a proveedores con enlace automático al módulo de compras.
- Emisión de etiquetas de artículos con impresión de códigos de barra.
- Histórico de movimientos de almacén con trazabilidad entre diferentes módulos.

Módulo de Servicio Técnico y Postventa. S.A.T.

- Permite definir y usar un catálogo de operaciones de Mano de Obra (tarifario), por marca y/o modelo.

- Controla campañas técnicas.
- Gestiona las Agendas de Servicio y de Citas previa para el S.A.T.
- Controla automáticamente si se ha producido o no una visita programada y en consecuencia se pueden obtener estadísticas de visitas realizadas, no realizadas, etc.
- Trabaja con calendarios y turnos para el personal.
- Controla las horas de presencia y los rendimientos de los Operarios/Mecánicos de SAT.
- Informes de Fichajes de operarios.
- Realiza diferentes informes de Rendimientos y Productividad.
- Gestiona el seguimiento de Acciones Técnicas y Mantenimiento del S.AT.

Módulo de Preventa y Promoción Comercial.

- Seguimiento de Acciones Comerciales.
- Seguimiento de Acciones Técnicas.
- Emite partes de trabajo.
- Trazabilidad entre documentos.

Módulo de Gestión de Cobros y Pagos. Previsión de Tesorería.

- Realiza gestión de remesas.
- Obtención de ficheros en los formatos AEB (asociación española de banca) 19, 58, 34 y 68.
- Impresión de documentos en formato Recibo, Talón y Pagaré.
- Incluye el módulo de tesorería con previsiones de cobro y pago.
- La introducción de las previsiones puede ser automática, o manual.
- Emite informes de previsión de tesorería por fechas, bancos y conceptos.
- Totalmente integrado con los módulos de compras, ventas y contabilidad.

- Trazabilidad de documentos entre módulos.

Módulo de Contabilidad.

- Integrado con el resto de áreas y departamentos de la empresa.
- Adaptado al nuevo plan general de contabilidad.
- Total flexibilidad en la definición de la estructura de cuentas.
- Permite generar los asientos contables de forma automática a partir de los módulos de Gestión de compras y ventas, así como la cartera de cobros y pagos / Tesorería.
- Dispone de asientos automáticos predefinidos.
- Los asientos son totalmente parametrizables por el usuario, pudiendo definir los enlaces contables de una forma sencilla y rápida.
- Está preparado para la migración de datos desde diferentes Aplicativos estándar del mercado.
- Gestiona Presupuestos.
- Balances presupuestarios.
- Dispone de Punteo automático o manual de movimientos.
- Los informes financieros son definibles por el propio usuario.
- La aplicación se distribuye con los informes oficiales indicados en el nuevo Plan General de Contabilidad, tanto para empresas pequeñas, pymes o Grandes empresas.
- No son necesarios cierres mensuales ni actualizaciones periódicas.
- Emisión de balances y cuentas de resultados a la fecha deseada.
- Permite el alta automática de las cuentas desde la captura de asientos contables.
- Posibilidad de generar asientos automáticos, repetitivos y duplicar asientos.

- Búsqueda de apuntes por importes, conceptos, etc.
- Exporta los datos contables en formato csv, para su tratamiento con herramientas ofimáticas.
- Informa de la antigüedad del saldo de una cuenta, a una fecha de referencia, para saber el importe que “me deben o debo “ y desde “cuando me lo deben o lo debo”.
- Genera los ficheros correspondientes para la presentación telemática de los libros registros de contabilidad general para su registro (registro mercantil).
- Generación automática de asientos de cierre.
- Posibilidad de reapertura de ejercicio contable cerrado.
- Posibilidad de gestionar movimientos contables procedentes de otros aplicativos.

Módulo de Impuestos.

- Obtiene el modelo 347 de operaciones anuales.
- Obtiene las liquidaciones de IVA mensuales o trimestrales.
- Genera los ficheros correspondientes para la presentación telemática de los libros registros de IVA para su presentación en el registro mercantil.
- Permite gestionar los libros registro de facturas emitidas y recibidas, así como su impresión.

Módulo de Gestión de la Aplicación.

- Permite personalizar y definir las opciones y menús que componen la Aplicación.
- Permite definir Grupos y Usuarios.
- Gestiona el acceso de los diferentes usuarios.
- Control los permisos y accesos desde los menús.
- Configura el entorno de trabajo de la aplicación.
- Realiza copias de seguridad.

- Consulta de Log de Acceso. Es posible consultar quien, cuando y donde se ha accedido.
- Importación directa desde otros Aplicativos.
- Gestión de empresas. Creación, baja, copiado, etc.
- Otras funcionalidades.

1.2. Requisitos.

La puesta en marcha de **OASIS** no supone ningún gasto en adquisición de Software. Tanto el propio desarrollo como las herramientas adicionales utilizadas (bases de datos, generadores de informes, etc), son de libre distribución al estar acogidas a las licencias GNU/GPL y son incluidas dentro del pack de instalación.

La licencia GNU/GLP se puede consultar en los siguientes enlaces:

<http://www.es.gnu.org/modules/content/index.php?id=8>
<http://www.gnu.org/licenses/licenses.es.html>

Los requerimientos son mínimos:

- PC (en local) con Sistema Operativo Linux o Windows XP
- Equipo Servidor (sólo si se instala en red) con S.O. Linux o Windows.
- PostgreSQL (versión 8.3 o superior)
- PHP (versión 5.2.9)
- PHP-GTK (versión 2.12.9)
- Agata Report (7.5 o superior)
- Certificado Electrónico (solo si se desea disponer de firma electrónica de documentos).
- Codificación de página UTF-8.

OPENBRAVO

Openbravo permite a las empresas gestionar de forma integral sus operaciones diarias, optimizar los procesos operativos, aumentar la satisfacción de los clientes y, en definitiva, incrementar su rentabilidad.

¿ POR QUÉ OPENBRAVO ?

1 Aplicación web nativa

Openbravo ha sido diseñado de **manera nativa en entorno web**, habilitando un acceso universal y proporcionando **seguridad de acceso y sencillez** a través de un navegador convencional.

2 Completamente funcional

Openbravo incluye un amplio abanico de funcionalidades propias de las soluciones **ERP extendidas** e integra la gestión avanzada de clientes o **CRM** (Customer Relationship Management) y **BI** (Business Intelligence).

3 Realmente integrado, realmente un ERP

Openbravo ha sido **diseñado como un ERP estándar**: las diferentes funcionalidades comparten la misma arquitectura, filosofía, reglas e interfaz de usuario, integrándose entre ellas de manera natural.

4 Software libre

Openbravo se **distribuye bajo una licencia de tipo Open Source, sin ningún coste** por uso y proporcionando **acceso al código fuente y libertad para modificarlo** según las necesidades, con total **independencia del proveedor**.

5 Arquitectura revolucionaria

Openbravo se ha diseñado sobre la base de una **arquitectura revolucionaria** que resulta en una manera más eficiente de desarrollar aplicaciones.

6 Adaptado a sus necesidades

Tras la definición de los requerimientos y procesos de negocio, **en pocas semanas es posible disponer de un sistema completamente operativo y adaptado a sus necesidades**.

ARQUITECTURA REVOLUCIONARIA

Openbravo se ha construido usando estándares abiertos y combinando de forma revolucionaria las metodologías de programación MVC y MDD en el motor WAD creado especialmente para este propósito.

Componentes

:: Entorno operativo

Compuesto, entre otras, por conocidas aplicaciones como Apache Tomcat y una base de datos PostgreSQL™ u Oracle®. Éstas pueden instalarse sobre GNU/Linux o Microsoft® Windows®.

:: Framework MVC

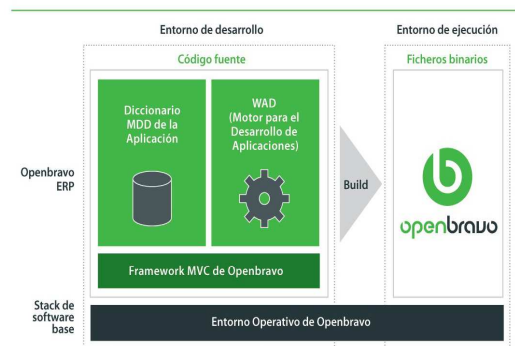
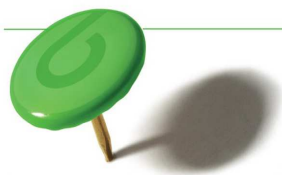
Un conjunto de sólidas utilidades de programación desarrolladas por Openbravo que facilitan el desarrollo de aplicaciones web MVC.

:: Diccionario MDD de la aplicación

Almacena los metadatos que describen los elementos de la aplicación y su comportamiento.

:: WAD (Motor para el Desarrollo de Aplicaciones)

Desarrollado por Openbravo, genera automáticamente el código ejecutable a partir del diccionario MDD. Los ficheros generados por WAD se ajustan al modelo MVC.



¿Por qué es revolucionario?

:: MVC (Modelo, Vista, Controlador)

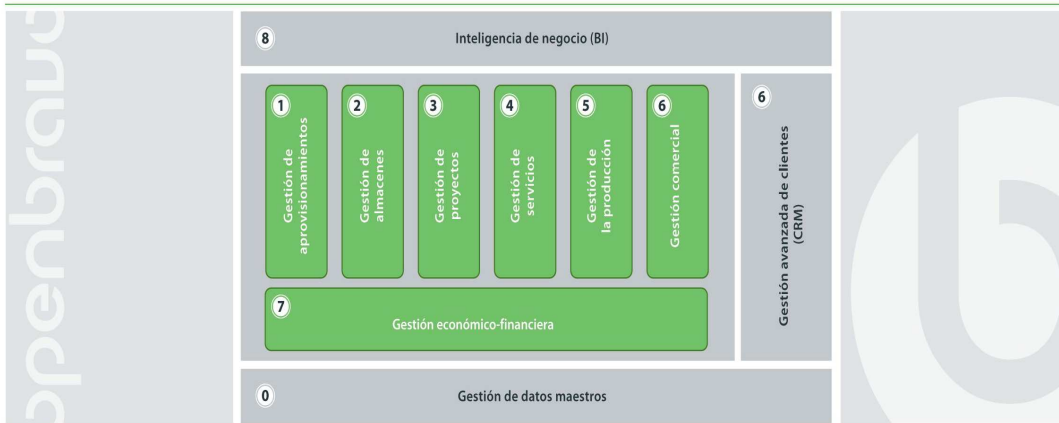
Es una metodología que ha demostrado su valía en el desarrollo de aplicaciones web, separando las capas de datos, interfaz gráfico y lógica de negocio. Esta separación de los componentes en ficheros distintos, resulta en un código más estructurado, facilitando su desarrollo y mantenimiento.

:: MDD (Desarrollo Dirigido por Modelos)

Es una metodología de desarrollo que se basa en metadatos almacenados en un diccionario para modelar el comportamiento de la aplicación. Esto conlleva una drástica reducción en la codificación manual y, por consiguiente, en errores de desarrollo.

COMPLETAMENTE FUNCIONAL

Openbravo incluye las funcionalidades necesarias para dar soporte a todas las áreas de la pyme:



0 Datos maestros

Tarifas, pedidos de compra, recepción de mercancías, registro y contabilización de facturas de proveedores, planificación de necesidades de compra, etc.

La correcta gestión de los datos maestros del negocio (productos, clientes, proveedores, etc.) constituye un aspecto fundamental para garantizar la coherencia y trazabilidad de los procesos.

1 Gestión de aprovisionamientos

Tarifas, pedidos de compra, recepción de mercancías, registro y contabilización de facturas de proveedores, planificación de necesidades de compra, etc.

Garantiza la integridad, trazabilidad, homogeneidad de todo el proceso y la integración con el resto de áreas de la aplicación.

2 Gestión de almacenes

Almacenes y ubicaciones, unidades de almacén, lotes, números de serie, bultos, etiquetas, entradas, salidas, movimientos entre almacenes, inventarios, valoración de existencias, transportes, etc.

Permite que las existencias estén actualizadas, localizadas y correctamente valoradas, garantizando la trazabilidad regulada en un gran número de sectores.

3 Gestión de proyectos

Proyectos, fases, presupuestos, gastos, compras asociadas, etc.

Facilita el control de los mismos en los aspectos relativos al presupuesto, las fases, los costes y las compras asociadas a cada proyecto individual.

4 Gestión de servicios

Recursos, servicios, gastos, gastos refacturables, facturación de servicios, nivel de servicio, etc.

Permite su definición, gestión de los recursos, el control de todas las actividades, así como la monitorización detallada de los gastos incurridos.

5 Gestión de la producción

Estructura de planta, planes de producción, BOM's, MRP, órdenes de fabricación, partes de trabajo, incidencias de trabajo, partes de mantenimiento, etc.

Proporciona información acerca de la estructura productiva de cada organización y de los datos relevantes para la producción.

6 Gestión comercial y gestión avanzada de clientes (CRM)

Pedidos de venta, tarifas, albaranes, facturación, comisiones, CRM, etc.

Diseñada con el objetivo de permitir la máxima flexibilidad y agilidad en la ejecución, garantizando la trazabilidad del proceso y permitiendo la integración con sistemas de captura de pedidos externos (tipo PDA o PdV).

7 Gestión económico-financiera

Plan de cuentas, cuentas contables, impuestos, contabilidad general, cuentas a pagar, cuentas a cobrar, contabilidad bancaria, balance, cuenta de resultados, activos fijos, etc.

Orientada a la minimización de la introducción manual de datos actuando como recolector automático de todas las actividades generadas en el resto de áreas de gestión.

8 Inteligencia de negocio (BI)

Reporting, análisis multidimensional (OLAP), cuadros de mando predefinidos.

Facilita toda la información relevante para la toma de decisiones mediante la monitorización de una serie de indicadores clave para la organización.

Encuentre información más detallada en www.openbravo.com



Openbravo es miembro fundador de la Open Solutions Alliance.
Copyright © 2007 Openbravo SL. La marca y el logotipo Openbravo son marcas registradas de Openbravo SL. Otros productos mencionados son marcas registradas propiedad de sus respectivos titulares.

- Arquitectura tecnológica
- Cobertura funcional
- Plan de desarrollo
- Casos de éxito
- Descargas y demo online

OPENERP

1. INTRODUCCIÓN

El Open ERP (Enterprise Resource Planning, ERP por sus siglas en inglés) es un sistema planeador de recursos empresariales que permite realizar una gestión integrada de los recursos. Entre sus características están la contabilidad analítica, contabilidad financiera, gestión de almacenes/inventario, gestión de ventas y compras, automatización de tareas, campañas de marketing, ayuda técnica (Helpdesk), y punto de venta, dentro de la construcción misma del software se hace uso intensivo de flujos de trabajo que se puede integrar con los módulos haciendo la modificación de aprobación y en general de cualquier proceso adaptable.

Este Tutorial básico de Open ERP provee una introducción técnica y al mismo tiempo completa de cómo implementar Open ERP en tu computador y aprender a construir diferentes tipos de módulos dependiendo de necesidades específicas. El tutorial incluye desde la instalación de Open ERP hasta la personalización de los diferentes módulos.

Para este tutorial es importante tener conocimiento previos de Python para la construcción de los objetos que componen un módulo, también es necesario un poco de XML para construir las vistas de los objetos, los flujos de trabajo y los diferentes procesos de negocio que componen a los módulos.

Es necesario recordar que este tutorial es de uso básico, por lo tanto es necesario comprometerse un poco y tener voluntad para experimentar. Pero no hay de qué preocuparse: aprender a implementar Open ERP es de lo más divertido y proporciona una gran satisfacción cuando se da con la solución correcta.

2. QUE ES OPEN ERP

Open ERP es un sistema planeador de recursos empresariales o ERP que cubre las necesidades de las áreas de contabilidad, ventas, compras, y almacén e inventario, entre otras. Open ERP soporta múltiples monedas, múltiples compañías y múltiples contabilidades; además incorpora funcionalidades de gestión de documentos para agilizar la colaboración entre departamentos y equipos en la empresa; y permite trabajar remotamente mediante una interfaz web desde una computadora conectada a Internet. Tiene componentes separados en esquema Cliente-servidor. Dispone de interfaces XML-RPC, y SOAP. Anteriormente se le conoció como TinyErp.

Open ERP es multiplataforma, funciona sobre Linux y Windows, y la interfaz de usuario está construida sobre la biblioteca grafica Gtk+, también hay una alternativa construida sobre Qt. Adicionalmente Open ERP tiene un cliente para ambiente Web llamado Etiny que fue construido sobre el framework para desarrollo de aplicaciones web TurboGears.

Emplea a Postgresql como Sistema manejador de bases de datos y ha sido programado con Python, lo cual permite que su adecuación e implantación sea limpia teniendo un esquema de arquitectura menor que otras soluciones.

Dentro de la construcción misma del software se hace uso intensivo de flujos de trabajo que se puede integrar con los módulos haciendo fácil la modificación de y en general de cualquier proceso adaptable.

2.1. Ventajas de OpenERP

Completo

En estos momentos existen más de 300 módulos específicos para distintos sectores de actividad.

Potente

OpenERP añade en la mayor parte de sus áreas herramientas de análisis y generación de reportes, con lo que la gestión y visualización de la información se simplifica.

Flexible

Las modificaciones y adaptaciones de código a las necesidades de las empresas se pueden realizar en forma ágil. Por ejemplo: flujos de trabajo (workflows) editables; reportes personalizados; control de productos y vistas.

Libre

Es un sistema basado en estándares, abierto y ampliamente soportado. Existe una importante comunidad de desarrolladores que están constantemente fortaleciendo el proyecto (amplia documentación, foros, cvs, mailing, listas, etc.).

Accesible

Open ERP se suministra bajo licencia GPL, por lo que no se abonan licencias de adquisición. Ud. sólo paga por los costos de integración y adaptación a las necesidades de su empresa.

Avanzado técnicamente

- Usa doble entrada en la gestión de inventarios.

- Soporta múltiples vistas de la contabilidad.
- Está preparado para conformar normas ISO9001.
- Funciona con bases de datos de objetos.
- Utiliza flujos de trabajos flexibles y dinámicos.
- Soporta plataformas heterogéneas: Linux, Windows.
- Utiliza un esquema de servidor distribuido.

2.2. Integración con otros software

Todos los informes de Open ERP se generan en PDF para una perfecta impresión. También se pueden generar archivos en Word o Excel que después pueden modificarse antes de ser enviados a un cliente por carta, mail o fax en forma automática.

Open ERP se integra con los siguientes softwares comerciales:

- Visualización bajo Adobe Reader (PDF).
- Importación/Exportación de Microsoft Office u OpenOffice.
- Exportación a Excel (or CSV).
- Google maps: servidor de aplicaciones de mapas en la web (gratis).

También existen publicados conectores con software libre como:

- OpenOffice: La suite ofimática de código abierto desarrollada por Sun Microsystems/Oracle.
- Mozilla Thunderbird: Cliente de correo electrónico de los creadores de Mozilla Firefox.
- Jasper Reports (iReport): herramienta de creación de informes Java libre.
- Magento: aplicación de comercio electrónico online.
- Oscommerce: otra aplicación de comercio electrónico online.
- Joomla: gestor de contenidos (integración parcial a través de xml-rpc).

- Spree: software de comercio electrónico para Ruby on Rails.
- Dia UML: software de diagramas para implementación de módulos.

3. INSTALACIÓN

La instalación es total mente automática bajo todos los sistemas operativos windows, solo se debe ingresar a la página oficial (www.openerp.com) y descargar el paquete AllInOne y ejecutar la instalación, es preferible realizar la instalación en el sistema de archivos.

Para la conexión del cliente al servidor Open ERP se debe iniciar la aplicación, si es la primera vez que se ejecuta el cliente puede aparecer primero una encuesta sobre el interés en el uso de Open ERP, este paso se puede obviar, después aparecerá una ventana pequeña para iniciar la conexión la cual está conectada por el puerto 8069 con el protocolo XML-RPC o por el puerto 8070 con el protocolo NET-RPC los cuales son valores por defecto. Si en el campo Base de datos de la ventana de conexión se informa que *¡No se puede conectar al servidor!*, significa que los parámetros del Servidor son incorrectos. Se debe hacer clic en el botón Cambiar y modificar los parámetros de Servidor, Puerto y Protocolo de conexión. En una configuración por defecto donde cliente y servidor están en la misma máquina estos serían los valores típicos:

- **Servidor:** localhost
- **Puerto:** 8070
- **Protocolo de conexión:** NET-RPC (faster)

O también estos:

- **Servidor:** localhost
- **Puerto:** 8069
- **Protocolo de conexión:** XML-RPC

OPENXPERTYA

Las empresas tienen la necesidad de unas herramientas flexibles que se adapten plenamente a su forma de trabajo e integren correctamente tanto los procesos de negocio como la información, garantizando así su renovación y ampliación a medida que la empresa y sus necesidades cambien o crezcan. Se necesita **incrementar la eficiencia empresarial** mediante procedimientos de trabajo óptimos, integrando la cadena de valor con todos sus componentes, internos (unidades de negocio) y externos (clientes, proveedores, administraciones, etc.). **Rebajar los costes** es imprescindible, las nuevas tecnologías informáticas y de comunicaciones que les ofrecemos les permitirán aumentar en eficacia y ahorro de recursos. Una moderna gestión de una empresa debe contar en todo momento con la información de negocio necesaria como soporte a la toma de decisiones. Disponer del dato preciso, en el momento que realmente se necesita y en el formato más adecuado, será vital para actuar conociendo nuestra realidad y las necesidades de los clientes.

openXpertya es la respuesta a estas necesidades, no es un simple programa de gestión o facturación, se trata de un **ERP** proactivo e inteligente, adaptable plenamente al modo de trabajo de cualquier tipo de negocio y con todas las herramientas necesarias para todas las áreas ya incluidas, y dispuestas para su uso en cualquier momento. Esto le permite usarlas en la medida que las necesidades de la empresa vayan aumentando sin incremento alguno de costes ni de tiempo o usarlas desde el primer día.

En el entorno actual, los sistemas de la información **ERP** se han convertido en una pieza clave para la competitividad y la consecución de los objetivos de empresa. *openXpertya* es una aplicación **ERP** especialmente diseñada para adaptarse a las necesidades específicas de cada una de las empresas. Todo ello sin renunciar a las ventajas del software estándar y con una visión de total integración de procesos. *openXpertya* está concebido como una solución global, integradora y abierta a la colaboración con terceros. Cuenta con Internet como parte natural de la solución; sus colaboradores, clientes y proveedores pueden estar involucrados en los circuitos de la información que usted determine, desde cualquier lugar y en cualquier momento. *openXpertya* es nuestra apuesta por una solución de **Código Abierto ERP + CRM** para las **PYMES, franquicias y empresas de distribución** y en esta información queremos dárselo a conocer.

¿QUÉ HACE OPENXPERTYA?

Administra mejor sus procesos de negocio.

openXpertya es una aplicación de gestión integral de la empresa (**ERP**) basada en los *procesos de negocio* en lugar del tradicional y desfasado sistema de gestión por departamentos o áreas. Es una aplicación de entorno *cliente-servidor* con capacidades multiempresa, multiusuario, multiejercicio, multiubicación y con posibilidad de bases de datos distribuida (no centralizada si es necesario para empresas descentralizadas con múltiples sedes). *openXpertya* puede llevar el 100% de las transacciones de la gestión de su empresa. La dirección debe concentrarse en mejorar sus propuestas de valor y ventajas competitivas, *openXpertya* se encarga de los detalles operativos.

La aplicación se basa en un sistema por niveles y áreas de trabajo de cada empleado o colaborador de la empresa, basadas íntegramente en los procesos de negocio que realiza la empresa en general y cada uno de los trabajadores en concreto. Así, *openXpertya* está diseñado para cambiar a medida que los negocios evolucionan; en cualquier momento, los usuarios, incluso en un sistema en producción en tiempo real, pueden cambiar de una manera extremadamente fácil y visual la estructura de la información, ajustándola a las nuevas necesidades. En contraste, un sistema tradicional de gestión corporativa está basado en estructuras fijas, de difícil cambio sin modificar el código del programa y desde luego con grandes costes para la empresa y los usuarios finales (tiempo). *openXpertya* proporciona también múltiples vistas de la información de la empresa con detalle de las transacciones actuales en tiempo real, esta estructura permite una máxima flexibilidad e integración con información suplemental externa (listas de precios de proveedores, datos de *rappels* de proveedores que se traspasan proporcionalmente a clientes, etc). Y dado que este sistema está basado en vistas, puede ser fácilmente modificado para actualizar posibles cambios futuros sin tocar una sola línea de código e incluso directamente por el usuario final.

openXpertya integra *Gestión de relaciones con clientes; CRM [Customer Relationship Management]*, *Gestión de relaciones con socios o proveedores; PRM [Partner Relations Management]*, *Gestión de la cadena de suministros SCM [Supply Chain Management]*, *Gestión integral de todos los recursos; ERP [Enterprise Resource Planning]* y *Procesamiento de Análisis en Línea; OLAP [Online Analysis Processing; EDI [Electronic Data Interchange], intercambio electrónico de datos.*

La aplicación fue diseñada para su utilización desde cliente ligero (incluye servidor de páginas web para su utilización remota desde navegadores web) multiplataforma (es posible su instalación y uso en cualquier tipo de máquinas o sistema operativo que soporte Java y un sistema de gestión de base de datos compatible; desde Unix, Linux, FreeBSD, MacOS, Windows, sistemas AS/400, mainframes, etc) y permite las opciones más flexibles de acceso y desarrollo (clientes java remotos con tecnología GSM/GPRS por ejemplo).

La aplicación está basada enteramente en un *Diccionario de Datos Activo (ADD Active Data Dictionary)* que asegura la funcionalidad, estabilidad y consistencia de un sistema distribuido multilinguaje, multimonedado, multiempresa, multiejerccio y multiubicación de desarrollo y gestión global.

Una moderna gestión de una empresa debe contar en todo momento con la información de negocio necesaria como soporte a la toma de decisiones. Disponer del dato preciso, en el momento que realmente se necesita y en el formato más adecuado, será vital para actuar conociendo nuestra realidad y las necesidades de los clientes.

MÓDULOS DE OPENXPERTYA

Lista Detallada de las principales funcionalidades.

Los *procesos de negocio* influyen en el diseño de *openXpertya* más que los departamentos tradicionales. En el mundo actual, y más en organizaciones de tamaño medio, el personal cubre a menudo procesos de negocio completos, e incluso múltiples procesos relacionados. En contraste, los sistemas tradicionales son usualmente dependientes de la estructura contable, lo que resulta en carencias informativas que son cubiertas por información derivada o costosos e ineficientes puentes a procesos externos (por ejemplo con puentes a la aplicación externa de *B2B Business to Business* o distribución directa a través de la web en lugar del canal tradicional).

En *openXpertya* todos los procesos de negocio son realizados en tiempo real y directamente sobre la base de datos centralizada, sin tiempos muertos ni errores sobre el *stock* existente en todos y cada uno de los almacenes o establecimientos.

Módulos Principales: Contabilidad General, Pagos a proveedores, Pagos de clientes, Stock e Inventarios, Pedidos de Compras, Pedidos de Ventas, Costes de personal (logística), Gestión central del Sistema.

Varios módulos Avanzados: Gestión transparente con Códigos de barras, Lista de Materiales (productos compuestos), Gestión Avanzada (toma de decisiones estratégicas), Gestión y relaciones de contactos (clientes y proveedores), Activos fijos, Terminales Punto de Venta y Cajas, Contratos de Servicio, Soporte de *Call Center*, Módulo de facturación por tiempo, Gestión de Gastos de Viajes, Gestión automática de almacenes, colas LIFO y FIFO, Ordenes de reparación y garantías, Seguimiento y facturación de material en cesión o alquiler.

Módulos Estándar de la Industria: Cristal Reports, FRx, F9, Intercambio electrónico de datos (*EDI, XML*), Gestión por flujo de trabajo (*WorkFlow*). Exportación de datos a cubos OLAP.

Submódulos enlazados adicionales: Conciliación bancaria, Contabilidad de cajas, Planificación de flujos de caja, Gestión de contratos, gestión de monedas, Explorador de datos, Gestión de Depósitos directos, *Rappels* y comisiones a vendedores y empleados, Asistente para la importación de datos, Multicompañía (con conciliación de grupos o sociedades), Multidivisa (con compras o ventas en otras divisas y tasas de cambio automáticas), Análisis multidimensional, Gestión de promociones, Seguimiento de materiales por número de serie y enlace con códigos de barras unidimensionales y multidimensionales, Planificación de ventas y Campañas, Costes estándar. Multiidioma (con asignación de idiomas por usuario y soporte actual de lengua Castellana, Gallega, Catalana y Bable entre otras), Gestión de la Cadena de valor de distribuidores, etc. Otros módulos de producción por fases, y soporte de integración fuerte con Microsoft Office para la generación de documentación de marketing.

UNA ERP PARA PYMES

openXpertya tiene características y ventajas claramente insuperables.

openXpertya proporciona una solución integral de gestión para **empresas pequeñas y medianas** de entre 3 y 5000 empleados y con un movimiento anual entre los 2.000,00 € y los 2.000M de €, con presencia en el mercado global, gestionando todas las áreas desde la administración de clientes, cadenas de abastecimiento y contabilidad; que quieran migrar a una solución de instalación y personalización rápida.

Algunos ejemplos del uso de *openXpertya* pueden ser los sectores de distribución, franquicias y agrupaciones o proveedores

ASP: Cadenas de distribución:

Las operaciones de las Cadenas de Distribución necesitan o requieren disponer de una solución de software para sus proveedores y clientes en su cadena de abastecimiento, que usualmente son PyMEs. *openXpertya* provee la funcionalidad de administración de relaciones con asociados y puntos de integración de la cadena de abastecimiento, así como enlace de compras mediante intercambio automático de documentos (**EDI**) y ventas directas a la cadena de valor a través del soporte Web (**B2B**) e incluso la posibilidad de distribución directa (**B2C**), y de gestión remota por parte de empleados (**B2E**) todo ello con actualización de stocks y precios en tiempo real. Esto hace de *openXpertya* una solución idónea para las empresas de distribución y servicios. Siempre tendrá una visión total de 360 grados de las actividades relacionadas con sus productos y clientes.

Sistemas de Franquicia:

Las franquicias disponen de una red de múltiples establecimientos, almacenes y una logística compleja. *openXpertya* gestiona la cadena de suministros dentro de la propia organización tanto como la gestión entre los franquiciados y la central. Los miembros de la cadena de distribución pueden ser organizaciones independientes o parte de la propia organización central, pero la gestión debe ser integral en todo momento, aunque los franquiciados y las relaciones en la franquicia determinen que datos comparten y cuales no. Además, la solución de **TPV** integrada mejora la productividad de cara al público y proporciona toda la información manejada directamente desde la central.

La solución ASP (Proveedor de soluciones alojadas):

openXpertya fue diseñado desde su concepción para entornos alojados (*en remoto*) de alto volumen. En contraste con las aplicaciones tradicionales, *openXpertya* provee funcionalidad de administración **ASP** para agregar o incluso mover clientes entre bases de datos. Esto permite una solución centralizada para grupos sectoriales de empresas minoristas que pretendan compartir información a nivel de producto, mediante la posibilidad de actualización directa de referencias y precios desde la central.

VENTAJAS DE OPENXPERTYA

openXpertya tiene significativos beneficios de producto sobre su competencia.

Algunos beneficios vienen automáticamente con el entorno de Software Libre, pero *openXpertya* tiene además, significativos beneficios en el producto sobre su tradicional competencia.

Enlace a otras aplicaciones de software libre del mismo equipo

El equipo de desarrolladores de *openXpertya*, dispone de una gama de productos de software para las empresas, abarcando todas las necesidades en los procesos de negocios. De esta manera, nuestros productos están concebidos como módulos de una macro solución empresarial, que pueden ser ensamblados de una manera fácil y rápida. Dentro de esta gama podemos destacar el sistema *Satpel*. *Satpel* permite enviar datos desde el propio ERP (*sistema de gestión*) a las estanterías en tiempo real, mostrando los precios en pantallas **LCD**. Una solución que optimizara el rendimiento de su negocio y le permitirá enormes ahorros de costes y garantizara una consistencia eficaz en los precios entre la estantería y el punto de venta.

Rápida Implementación sin decisiones finales

Implementar *openXpertya* puede ser cuestión de horas, no semanas ni meses. Del mismo modo que nuestra competencia, nosotros utilizamos

plantillas para adecuar el software a sus necesidades, pero sin restringir funcionalidades. *openXpertya* no requiere de ningún tipo de decisiones irreversibles en tiempo de implementación.

Cualquier cosa puede ser modificada más tarde, incluyendo su plan de cuentas, calendario de ejercicio o incluso la moneda en que contabiliza o los principios contables. En *openXpertya*, esos cambios no requieren de ninguna intervención o habilidad técnica específica. Esto nos permite entrar en producción rápidamente. Usted puede adaptar el sistema a los cambios de entorno del negocio siempre que lo necesite.

Interfaz HTML "extendida" e Interfase Gráfica de Usuario "enriquecida"

openXpertya ofrece dos interfaces de usuario en paralelo: Un interfaz de navegador Web basada en DHTML diseñada para acceder a su aplicación desde cualquier lugar en Internet y una Aplicación Gráfica con una interfaz enriquecida (que permite una entrada rápida de datos - sin necesidad de ratón). Ambas interfaces de usuario son generadas utilizando las mismas reglas y con la misma apariencia, aprovechando al máximo las ventajas de cada plataforma.

Ningún otro ERP de nuestros competidores soporta en su totalidad ambas interfaces de usuario ni fue diseñado para ello. Para el caso de oficinas remotas, en sitios de cliente o vía dispositivos inalámbricos, seguramente se desea acceder a la aplicación de forma rápida y eficiente. *openXpertya* proporciona un cliente ligero DHTML con bajos requerimientos. La Aplicación Cliente es utilizada en producción con sólo 128kb de ancho de banda mientras que el acceso mediante navegador es posible con sólo 56kb.

Interface original en castellano y en varias lenguas.

Implementaciones reales en grandes empresas, estudios sectoriales y abundante comunidad de usuarios (más de 3500 instalaciones reportadas a Enero del 2007).

Más de cuatro años inmersos en el desarrollo, crecimiento y mejora del que ya está considerado por observadores independientes como el **ERP de software libre número uno del mercado español y latinoamericano**, tanto en la extensión de la red de soporte, como en número usuarios del sistema, en número de implantaciones finales reales, y en el tamaño de las mismas en muchos casos.

No depende de una sola empresa.

Según cinco estudios independientes, realizados por diversas organismos locales dependientes de varios Gobiernos Autonómicos Locales españoles y diversas Universidades de España, Argentina y Mexico,

openXpertya es el ERP de software libre en español con la mayor red de soporte profesional y está situada en el grupo de cabeza de los de mayor madurez.

LA FACILIDAD DE USO DE OPENXPERTYA

Interfaz de Usuario Inteligente (Personalización)

Las aplicaciones complejas tienen una larga curva de aprendizaje ya que se necesita conocer lo que significan las diferentes opciones. **openXpertya** utiliza un enfoque diferente utilizando una Personalización sensible al contexto. *¡Sólo se ve lo que se necesita ver!* Por Ejemplo, si no utiliza moneda extranjera, no necesita ver la opción -pero existen situaciones en las cuales Usted necesita realizar transacciones en moneda extranjera. Otro ejemplo son los diferentes procesos y campos requeridos para las diferentes opciones de pago. Usted sólo ve la información requerida en una situación y contexto específico. Nuestra gente de marketing llama a esto "*funcionalidad reservada*", existe pero no molesta.

Toda la información ingresada es inmediatamente validada, se incluyen sugerencias para el ingreso de datos. La consistente interfaz de usuario es generada basada en reglas, resultando en una aplicación extraordinariamente estable. Sin codificación manual, **openXpertya** provee controles cruzados de validación de campos y soporta dependencias de campos (por ejemplo: la dirección depende del cliente seleccionado).

Se pueden reorganizar los formularios de ingreso, ocultar campos, cambiar los colores y la tipografía utilizada tanto como modificar títulos, etiquetas y agregar ayuda y sugerencias específicas para clientes. Los usuarios avanzados pueden agregar campos y crear reglas de validación. La interfase dinámica de usuario de **openXpertya** hace que la aplicación resulte fácil de usar y reduce la curva de aprendizaje. La funcionalidad está disponible en ambas, el cliente ligero y la Aplicación Java.

Flujo de tareas y guía Inteligente (Workflow):

openXpertya incluye un sistema de flujo de tareas definible y automatizado. Un flujo de tareas es una serie de pasos guiados en la cual podría trabajar más de un usuario final. Cada paso puede ser el comienzo de un proceso o transacción, la ejecución de un informe o una entrada de información en pantalla. El sistema provee por defecto flujos predefinidos para muchas tareas estándar. Los flujos de tareas pueden ser procesos de cierre de mes, de facturación por etapas, cálculo de *rappels* u objetivos, etc. En todos los lugares donde un flujo de tareas es posible, existe un asistente (la guía inteligente) que le permite automatizar paso a paso las necesidades más complejas de sus procesos de negocio.

Alertas y Programación (Workflow):

En *openXpertya*, cualquier usuario puede definir situaciones de excepción para recibir notificaciones proactivas, La notificación puede ser una simple nota o la ejecución de un proceso o un reporte. Como ejemplo, una lista de clientes con un cierto porcentaje de sobreutilización de crédito, deuda pendiente antigua, nuevo cliente, inventario inmovilizado, etc. En segundo plano, el Sistema proporciona automáticamente un esquema proactivo de informes de problemas, obtenido a partir de una monitorización constante y detección de errores o posibles problemas predefinidos y personalizables.

PRINCIPALES MÓDULOS DE OPENXPERTYA

- Configuración
- Entidades comerciales
- Artículos, stock y logística. Almacén caótico.
- Compras (automatización)
- Ventas (automatización)
- Cobros y Pagos
- Contabilidad
- Proyectos (costes y balances por proyecto)
- CRM (seguimiento de clientes)
- Comercio electrónico y portal de empleados
- Producción por fases con trazabilidad.

CONFIGURACIÓN

- Flujos de trabajo definibles.
- Menús personalizables en niveles.
- Procesador de tareas automatizadas.
- Organizaciones, grupos y compañías.
- Control de accesos, roles y permisos.
- Menús adaptados a roles de usuario.
- Importación de datos de todo tipo.
- Alertas personalizadas y programables.

ENTIDADES COMERCIALES

- Gestión unificada de proveedores, clientes, empleados, etc.
- Asignación individual de condiciones y acuerdos comerciales en función del tipo de entidad.
- Múltiples esquemas de condiciones de compras y ventas en función de proveedor, cliente, producto, familia, etc.
- Información accesible en todo momento.

ARTÍCULOS Y ALMACENES

- Varios niveles de jerarquía posibles.
- Toda la información es obtenible en tiempo real tanto localmente como a distancia.
- Múltiples tarifas de compras y ventas en función de eventos internos o externos.
- Fácil generación de tarifas o listas de precios y descuentos automatizados o no. Varios niveles de esquemas de descuentos.
- Información de gestión de riesgo comercial y seguimiento de crédito.

COMPRAS

- Aviso de pedido coordinado por el CRM.
- Pedido, albarán, envío, factura y muchos más documentos configurables directamente.
- Toda la información y generación de informes en tiempo real.
- Seguimiento de mercados de compras.
- Automatización de pedidos en función de stock, ventas, eventos externos, etc.

VENTAS

- Petición de material. Escandallos
- Pedido, albarán, envío, paquete, factura y muchos más documentos configurables directamente y generados de forma manual o automática.
- Toda la información y generación de informes personalizables en tiempo real.
- Venta directa automática desde proveedor sin pasar por almacén.
- Automatización de pedidos de ventas en función de stock, cliente, autorización, rappels de ventas y acumulados, etc.
- Informes multinivel personalizables.

COBROS Y PAGOS

- Informes de vencimientos y de pagos parciales, pendientes, en tiempo real.
- Pagos y cobros automatizados.
- Conciliación bancaria semiautomática a partir de ficheros emitidos por el banco.
- Remesas bancarias integradas. Envío de pagos y cobros al banco.
- Generación y aceptación de ficheros de las Normas de la Asociación Española de Banca y del CEMLA.

CONTABILIDAD

- Asientos y contabilización automática y manual.
- Infinitos informes personalizables.
- Control de accesos a nivel de asientos y de cuentas.
- Presentación telemática de documentación fiscal y contable.

- Adaptación de las nuevas normas NIC-NIIF.
- Varios tipos de contabilidad simultánea.

PROYECTOS

- Gestión de recursos y asignación de costes por proyectos o recursos.
- División del proyecto en fases o periodos.
- Control de pagos parciales y repartos según fases del proyecto.
- Seguimiento de acciones y/o resultados. Contabilidad analítica.
- Informes según actividad, cliente, etc.

CRM

- Mensajería interna directa con seguimientos.
- Mensajería externa con enlace a correo electrónico.
- Mailing por áreas de interés.
- Suscripción a áreas de interés o boletines de información periódica
- Acceso vía web a noticias, comunicados, ofertas, etc.

Comercio electrónico y portal de empleados

- Menú y aspecto modificable en función del tipo de usuario que abre sesión.
- Integración con el portal general de empresa.
- Soporte B2B para distribución a través de comercio electrónico en tiempo real.
- Soporte B2C y varias pasarelas de pago disponibles para venta al menor.
- Soporte B2E y portal de empleados con servicios sociales privados.
- Personalización y diversas posibilidades de imagen de las webs
- Toda la información en tiempo real (stock, etc.).
- Generación de ficheros estándar EDI

Bibliografía

13. - BIBLIOGRAFÍA

- Adam, Frederic y Sammon, David (2004). The Enterprise Resource Planning Decade: Lessons Learned and Issues for the Future. Idea Group Publishing.
- Administración de los sistemas de información: organización y tecnología. Kenneth C. Laudon, Jane Price Laudon
- Administración de organizaciones y gestión integrada de los sistemas y tecnologías de la información. Rafael Bernal Montañés, Lorenzo Ros Mcdonell, Gonzalo Grau Gadea, Ricardo Chalmeta Rosaleñ
- Administración de sistemas de información. Effy Oz
- Accesibilidad y usabilidad de contenidos digitales. Por una sociedad de la información y el conocimiento no excluyente. Lilia Fernández Aquino
- An AHP-based approach to ERP system selection. Chun-Chin Wei, Chen-Fu Chien, Mao-Jiun J. Wang
- Análisis y Diseño de Sistemas de Información. Kendall & Kendall. (2004) Prentice Hall.
- An ERP Life-cycle-based research agenda. José M. Esteves, Joan A. Pastor, 1999.
- Auditoría de los sistemas de información. Rafael Bernal Montañés, Óscar Coltell Simon
- Auditoría de tecnologías y sistemas de información. Mario Piattini Velthuis, Emilio del Peso Navarro, Mar del Peso
- BUSINESS INTELLIGENCE: COMPETIR CON INFORMACIÓN. Josep Lluís Cano
- Business Intelligence for the Enterprise BIERE, M, IBM Press, 2003.
- Data Warehouse Project Management. ADELMAN, S., AND L. MOSS. Boston, MA: Addison Wesley, 2000.
- Dirección y gestión de los sistemas de información en la empresa: una visión integradora. ESIC

- El futuro tecnológico de las Terminales Marítimas de Vehículos: La integración de sus sistemas de información. Departamento de Ciència i Enginyeria nàutiques.
- ERP: guía práctica para la selección e implantación. : ERP: enterprise resource planning o sistema de planificación de recursos empresariales. Luis Muñiz, Gestión 2000
- ERPIs Dead - Long Live ERP II. BOND B, GENOVESE Y., MIKLOVIC D, WOOD N., ZRIMSEK B y RAYNER N (2000)
- Guia molinux para PYMES. CESLCAM
- Introducción a BPM para PYMES. Kiran Garimella, Michael Lees, Bruce Williams
- La empresa y sus sistemas de información. Universidad Politécnica de Valencia
- Las aplicaciones informáticas y las empresas: soluciones ERP. Juan José Guarch Bertolín, Llanos Cuenca González
- Los sistemas ERP en la práctica. José Vicente Tomás Miquel, Manuel Expósito Langa, Josep Capó Vicedo
- Metodología de planificación y desarrollo de sistemas de información MÉTRICA Versión 2. MAP,1990.
- Sistemas de Información Gerencial. Laudon K., Laudon Jane (2008) Pearson Education.
- Sistemas de información: herramientas prácticas para la gestión empresarial. Álvaro Gómez Vieites, Ra-ma
- Sistemas de información para la dirección. Manfredo Monforte, Pirámide
- Sistemas de información y nuevas tecnologías: Influencias de las nuevas tecnologías en la estructura organizativa de la empresa cántabra. María Elena Garcia Ruiz
- Software libre para una sociedad libre. Richard M.Stallman

Bibliografía web.

- <http://aahidalgo.blogspot.com/2009/04/erp-en-el-mundo-del-opensource.html>
- <http://abanq.org/documentacion/>
- <http://civicom.eu/cuestore/ERP/ERP-Libres.htm>
- <http://creativecommons.org/licenses/by-sa/3.0/>
- http://en.wikipedia.org/wiki/Customer_Relationship_Management
- <http://evaluandoerp.blogspot.com/>
- http://io.us.es/cio2006/docs/000142_final.pdf
- <http://juliopari.com/2010/04/15/adempiere-erpcrm-para-la-empresa-pyme/>
- <http://muypymes.com/formacion/tecnologia/1377-aprende-a-elegir-el-mejor-erp-para-tu-pyme.html>
- <http://pegasus.javeriana.edu.co/~CIS0830IS18/swf/index.html>
- <http://redalyc.uaemex.mx>
- <http://solucioneslustrum.wordpress.com/2009/05/11/erp-de-software-libre/>
- <http://www.acis.org.co/fileadmin/Conferencias/PresentacionAdempiereACIS2009.pdf>
- <http://www.adempiere.org/>
- <http://www.adempiere.com/index.php>
- <http://www.aplicacionesempresariales.com/abanq-software-erp-open-source-para-pymes.html>
- <http://www.asturlinux.org/>
- <http://www.aplicacionesempresariales.com/>
- <http://www.compiere.com/>

- <http://www.evaluation-matrix.com>
- <http://www.gestiopolis.com/canales/demarketing/articulos/43/crmmba.htm>
- <http://www.informatica-hoy.com.ar/software-erp/Caracteristicas-fundamentales-del-ERP.php>
- <http://www.informatica-hoy.com.ar/software-erp/Sistemas-ERP-de-software-libre.php>
- <http://www.infosial.com/>
- <http://www.iti.es>
- <http://www.lantek.es>
- <http://www.mckinseyquarterly.com>
- <http://www.oasis-clm.es>
- <http://www.oasis.com.co/>
- <http://www.openbiz.com.ar/Seleccion%20ERP.pdf>
- <http://www.openbravo.com/es/>
- <http://www.openerp.com/>
- <http://www.openerp.tv>
- <http://www.openxpertya.org/OpenERP>
- <http://www.pivotal.co.cr>
- <http://www.proyeccion.tv/blog/2007/02/22/congreso-sobre-herramientas-erp-con-software-libre/>
- <http://www.sbi-technology.com>
- <http://www.somoslibres.org>
- <http://www.tecnolinux.com>
- <http://www.versvs.net/anotacion/que-es-un-erp-enterprise-resource-planning-linux>

- <http://www.wheelwrightsf.com.ar>