

UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA
APLICADA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica Superior
d'Informàtica Aplicada

**CONTROL DE SERVICIOS INFORMATICOS
DEL GRUPO EMPRESARIAL X**

**Proyecto Fin de Carrera
CÓDIGO P.F.C.: DSIC-159**

Presentado por José María Iranio Marín

Dirigido por Manuel Herrero Mas

Valencia, Mayo de 2011

Agradecimientos

En un primer lugar, me gustaría agradecer a Marcos González, por su paciencia, y comprensión, por su tiempo e inestimable ayuda dedicados a este proyecto, así como sus consejos y la confianza que siempre ha depositado en mi.

A mi Codirector Manuel Herrero Blas por su dedicación y por su facilidad a la hora de hacerse entender.

Quiero dedicarles este proyecto muy especialmente a mis padres, José María y Maribel a mi hermana Raquel y a mi pareja Paula por aguantarme todas mis rabietas.

Y a todos a los que no os he nombrado y habéis formado parte de mi vida, gracias.

Tabla de contenidos

Agradecimientos	2
Tabla de contenidos	3
Tabla de ilustraciones	5
1 Introducción	8
2 Especificación de requisitos IEEE.	9
2.1 Introducción.	9
2.1.1 Propósito.	9
2.1.2 Ámbito.	9
2.1.3 Definiciones, Acrónimos y abreviaturas.	9
2.1.4 Referencias.	10
2.1.5 Visión Global.	10
2.2 Descripción General.	11
2.2.1 Perspectiva del producto.	11
2.2.2 Funciones del Producto	11
2.2.3 Características del usuario.	13
2.2.4 Restricciones Generales.	14
2.2.5 Supuestos y Dependencias.	14
2.3 Requerimientos específicos.	14
2.3.1 Requerimientos de interfaces externos.	14
2.3.2 Requisitos funcionales.	15
2.3.3 Requisitos de eficiencia.	21
2.3.4 Restricciones de diseño.	22
2.3.5 Limitaciones hardware.	22
3 Análisis.	23
3.1 Requisitos del Sistema.	23
3.2 Casos de Uso.	24
3.2.1 Plantillas de descripción.	26
3.3 Escenarios y Diagramas de Secuencias	30
3.3.1 Identificación en el sistema.	30
3.3.2 Alta de incidencia.	31
3.3.3 Alta de actuación.	31
3.3.4 Alta de proveedores.	32

4	Diseño.....	34
4.1	Arquitectura De La Aplicación.....	34
4.2	Modelo Relacional.....	35
4.2.1	Introducción.....	35
4.2.2	El Modelo De La Aplicación.....	36
5	Procesos De Desarrollo De Software.....	47
5.1	Modelos de proceso de software.....	49
5.2	Metodologías de desarrollo de software.....	50
5.3	Rational Unified Process (RUP).....	51
5.3.1	Proceso dirigido por Casos de Uso.....	52
5.3.2	Proceso centrado en la arquitectura.....	53
5.4	eXtreme Programming (XP).....	61
5.4.1	Proceso XP.....	63
5.4.2	Prácticas XP.....	66
5.5	Conclusiones.....	71
6	Borland Delphi.....	72
6.1	Herramienta para el Desarrollo y Diseño de Interfaces.....	72
6.1.1	Integrated Development Enviroment (IDE).....	72
6.2	Herramientas de Soporte.....	78
6.2.1	Herramientas de Documentación.....	78
6.2.2	Herramientas de Bases de Datos.....	79
6.2.3	Herramienta de Diseño Gráfico.....	80
6.3	Herramienta de Análisis Dinámico, Intrusiva.....	80
6.3.1	Winsight.....	80
6.3.2	Integrated Debugger.....	81
6.4	Documentación para los Desarrolladores.....	82
6.5	InstallShield.....	83
7	Manual De Usuario.....	84
7.1	Instalación.....	84
7.2	Manual de uso.....	84
7.2.1	Identificación y menú principal.....	84
7.2.2	Enlaces.....	86
7.2.3	Gestión de activos Fijos.....	88
7.2.4	Gestión de incidencias.....	92

7.2.5	Facturación telefónica.....	94
7.2.6	Gestión de Obras.....	97
7.2.7	Gestión de Usuarios.....	98
7.2.8	Gestión de cierres.....	98
7.2.9	Gestión de proveedores.....	99
7.2.10	Gestión de personal.....	102
7.3	Listín telefónico.....	103
8	Conclusiones.....	105

Tabla de ilustraciones

Ilustración 3.1:	Diagrama de casos de uso.....	25
Ilustración 3.2:	Secuencia inicio de sesión.....	31
Ilustración 3.3:	Secuencia alta de incidencia.....	31
Ilustración 3.4:	Secuencia alta de actuación.....	32
Ilustración 3.5:	Secuencia alta de proveedor.....	33
Ilustración 4.1:	Cliente / Servidor.....	34
Ilustración 4.2:	Incidencias, activos y usuarios.....	37
Ilustración 4.3:	Obra, cierres, y proveedores.....	38
Ilustración 4.4:	Telefonia.....	39
Ilustración 5.1:	Proceso de desarrollo de software.....	47
Ilustración 5.2:	Elementos del proceso del software.....	48
Ilustración 5.3:	Relación entre elementos del proceso del software.....	49
Ilustración 5.4:	Los Casos de Uso integran el trabajo.....	52
Ilustración 5.5:	Trazabilidad a partir de los Casos de Uso.....	53
Ilustración 5.6:	Evolución de la arquitectura del sistema.....	54
Ilustración 5.7:	Los modelos se completan, la arquitectura no cambia drásticamente.....	55
Ilustración 5.8:	Los modelos se completan, la arquitectura no cambia drásticamente.....	56
Ilustración 5.9:	Estructura global de RUP.....	57
Ilustración 5.10:	Fases e hitos en RUP.....	58
Ilustración 5.11:	Estructura estática del proceso.....	59
Ilustración 5.12:	Flujos de trabajo en RUP.....	61
Ilustración 5.13:	Flujos de trabajo de RUP.....	70
Ilustración 6.1:	Sección de la pantalla principal.....	73

Ilustración 6.2: Configuración del IDE.....	74
Ilustración 6.3: Inspector de Objetos.	75
Ilustración 6.4: Configuración del IDE.....	76
Ilustración 6.5: Librería de Expertos.	77
Ilustración 6.6: Control de versiones.....	78
Ilustración 6.7: BDE Administrator	80
Ilustración 6.8: Integrated Debugger.....	81
Ilustración 6.9: Documentación de desarrolladores.	82
Ilustración 7.1: Pantalla de inicio y login.	84
Ilustración 7.2: Pantalla principal.	86
Ilustración 7.3: Panel de servicios.	86
Ilustración 7.4: Subir fichero.	87
Ilustración 7.5	88
Ilustración 7.6: Menú servicios de gestión de activos fijos.....	88
Ilustración 7.7: Añadir activo.....	89
Ilustración 7.8: Listar activos.....	90
Ilustración 7.9: Búsquedas personalizadas.....	90
Ilustración 7.10: Búsquedas personalizadas 2.....	91
Ilustración 7.11: Cambio de propietario de activo.	91
Ilustración 7.12: solicitud de activo.	92
Ilustración 7.13: Menú de servicios de gestión de incidencias.....	92
Ilustración 7.14: Alta de incidencias.....	93
Ilustración 7.15: Gestión de actuaciones.	94
Ilustración 7.16: Menú servicios de gestión de telefonía.....	94
Ilustración 7.17: Insertar factura telefónica.	95
Ilustración 7.18: Consulta facturación.	96
Ilustración 7.19: Mantenimiento de oficinas.	97
Ilustración 7.20: Gestión de obras.	97
Ilustración 7.21: Asignar usuarios a obra.....	98
Ilustración 7.22: Gestión de cierres.	99
Ilustración 7.23: Menú servicios gestión de proveedores.	99
Ilustración 7.24: Alta de proveedores.	100
Ilustración 7.25: Asignación de obras a un proveedor.	100
Ilustración 7.26: Asignación de contrata a un proveedor.	101

Ilustración 7.27: Consulta de proveedores.....	101
Ilustración 7.28: Subir documentación de proveedor.....	102
Ilustración 7.29: Menú servicio gestión de personal.	102
Ilustración 7.30: Alta de usuario.....	103
Ilustración 7.31: Asignar superior.	103
Ilustración 7.32: Listín telefónico.....	104

1 Introducción

El presente proyecto nace por la necesidad de ofrecer, una serie de servicios centralizados en una única aplicación, para los distintos departamentos y empresas que conforman un grupo empresarial.

Existe una serie de causas motivadoras para el desarrollo de la aplicación, que vienen derivadas, de que en dicho grupo empresarial, a causa de su diversificación en distintos sectores de negocio, emplean una gran cantidad de software de gestión específico para cada uno; Como es de esperar, al ser cada una de las aplicaciones empleadas de distintos fabricantes, la gran mayoría de este software no se encuentra interrelacionado, por lo que no se puede contemplar una integración y comunicación entre ellas, generando una disgregación de datos, y duplicidad de la información.

También la falta de comunicación entre empresas del grupo y departamentos, sería otro de los factores motivadores que se pretende mejorar.

Es por ello que para solventar estas necesidades globales que afecta al grupo, se presente este proyecto con el objetivo de:

- Centralizar las incidencias informáticas.
- Centralizar la gestión de activos fijos del parque tecnológico de la empresa.
- Centralizar las incidencias de telecomunicaciones.
- Centralizar la facturación del gasto en telecomunicaciones.
- Centralizar la agenda del grupo empresarial.
- Generar un flujo de información en la aprobación de los presupuestos del sector de obra.
- Centralizar los datos y certificados de los distintos Proveedores de obra.
- Notificaciones departamento RRHH a los distintos departamentos.
- Directorio de acceso a otras utilidades u información.

2 Especificación de requisitos IEEE.

2.1 Introducción.

2.1.1 Propósito.

El propósito de la ERS es mostrar al usuario cual va a ser la funcionalidad del proyecto, realizando una descripción general del mismo y especificando los requisitos del mismo. Esta especificación de requisitos está destinada a ser leída por los usuarios o cualquier sujeto que tenga interés en saber como funciona el producto, si bien está principalmente destinada a los primeros, dado que quien vaya a utilizar esta herramienta, debe conocer ampliamente todas sus restricciones de manera que pueda comenzar a usarlo con total corrección.

2.1.2 Ámbito.

El producto que se va a desarrollar se denominará CONTROL DE SERVICIOS INFORMATICOS DEL GRUPO EMPRESARIAL X, realizará el mantenimiento de la información alojado en un servidor MYSQL. Así se pretende que la persona que acceda a la aplicación pueda proceder con la gestión de la información almacenada en el servidor de la base de datos, tanto realizando altas, bajas y actualizaciones sobre aquellos campos que los formularios a los que el usuario pueda acceder según permisos que se le hayan asignado.

Pretendiendo que la información presentada en la aplicación sea útil para los usuarios (empleados de las distintas empresas y departamentos).

2.1.3 Definiciones, Acrónimos y abreviaturas.

➤ Definiciones

- Password: seña secreta que permite el acceso a la aplicación. Constituida por una serie de letras o números, que pueden estar intercaladas.
- RAR: Extensión de fichero, empleado para especificar que se encuentra comprimido.
- MYSQL: Sistema gestor de base de datos.

➤ Acrónimos

- BD: Base de datos.
- IEEE: Institute of Electrical & Electronics Engineers.
- ERS: Especificación de Requisitos Software.
- SW: Software.

- HW: Hardware.
- PC: Personal Computer.
- RAM: Random Access Memory.
- Abreviaturas
 - MS: Microsoft.
 - TCP/IP: Transmission Control Protocol / Internet Protocol. Normas y reglas encargadas de gestionar la comunicación entre dos ordenadores en Internet.
 - API: Application Program Interface. Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.
 - Cliente: Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio. Una estación de trabajo que solicita el contenido de un fichero a un servidor de ficheros es un cliente de este servidor.
 - Host: Ordenador conectado a Internet. Ordenador en general. Literalmente anfitrión.
 - IEEE: The Institute of Electrical and Electronics Engineers. asociación técnico-profesional mundial sin ánimo de lucro formada por profesionales de las nuevas tecnologías, dedicada a la estandarización.

2.1.4 Referencias.

- Guía ANSI/IEEE Std. 830-1998, para la especificación de requisitos software.
- Guía del IEEE para la Especificación de Requerimientos Software. MicroWeb UPV de Ingeniería del Software.
- Diccionario de la Real Academia Española.
- Roger S. Pressman. "Ingeniería del Software. Un enfoque práctico".
- Bases de datos y sistemas relacionales. Delobel, Claude.
- Programación con Delphi. Charle Ojeda, Francisco.

2.1.5 Visión Global.

A continuación nos disponemos a realizar una descripción general del producto software realizado, detallando la perspectiva del programa "CONTROL DE SERVICIOS INFORMATICOS DEL GRUPO EMPRESARIAL X", sus funciones, las características del usuario que lo ha de manejar, las restricciones generales que rodean el producto, y posibles dependencias que puedan hacer que los requisitos del programa cambien.

2.2 Descripción General.

A continuación vamos a ver los factores que afectan al producto y a sus requisitos.

2.2.1 Perspectiva del producto.

El programa "CONTROL DE SERVICIOS INFORMATICOS DEL GRUPO EMPRESARIAL X", funciona bajo el sistema operativo Windows y como tal, debe poseer las características comunes a todos ellos. Además de esto, dado que la aplicación utiliza bases de datos MYSQL donde se centralice la información, es necesario tener un servidor MYSQL, el cual habrá que instalar aparte.

Para el correcto funcionamiento del mismo, no es necesario disponer de un ordenador excesivamente potente, dado que las operaciones que este realizará serán relativamente sencillas. Además de ello, la aplicación hará uso de un servidor MYSQL para obtener datos administrativos y, en determinados casos será el motor de bases de datos el encargado de realizar cálculos de mayor complejidad.

2.2.2 Funciones del Producto

Las funciones que debe realizar el producto pueden resumirse en las siguientes:

- a) Iniciar sesión por parte de usuarios registrados.
- b) Autenticación, autorización y acceso a los distintos servicios y formularios. Todos los empleados que tengan cuenta en el sistema, podrán iniciar sesión en la aplicación con su usuario y contraseña mediante un formulario.
Dependiendo de los permisos asignados al usuario, este tendrá disponibles las áreas del programa para las cuales esté autorizado, ocultando aquellas funcionalidades a las cuales no tenga permiso de acceso.
- c) Gestión de activos fijos. Sección encargada de gestionar el parque informático (ordenadores, impresoras, monitores, servidores, cámaras de fotos, cámaras de vídeo, routers, switches, etc.), al igual que los teléfonos móvil y las tarjetas sim de estos. Cada uno de los activos estará asignado a un responsable. Las tarjetas SIM estarán asociadas a los teléfonos móviles.
- d) Gestión de incidencias informáticas. En él, los usuarios podrán crear incidencias, listarlas o cancelarlas. El departamento de informática podrá gestionarlas, creando actuaciones sobre estas, hasta que una vez finalizada las puedan cerrar. Si se

requiere, la incidencia se pueda vincular a un activo, para de esta forma poder listar un histórico de incidencias que dicho activo ha generado. Los usuarios podrán ver los activos que tiene asignados.

- e) Gestión de incidencias de telefonía y datos. Al igual que la gestión de incidencias informáticas, pero en este caso no será el departamento de informática de gestionar las incidencias si no el administrativo asignado para tal tarea. Este procedimiento tiene sin embargo mayor complejidad al tener que gestionar (recibir y devolver) documentación para tratar las incidencias, así como el flujo de información entre los distintos usuarios del programa
- f) Gestión de facturación telefónica. El proveedor de comunicaciones mensualmente envía una serie de ficheros codificados por cada una de las empresas, que contienen los consumos de cada uno de los teléfonos, con una serie de descuentos, los cuales hay que contabilizar, para facilitar esta tarea a los administrativos de cada una de las empresas, la aplicación procesará unos xls generados por una tercera aplicación de telefónica llamada Factel, centralizando todos los ficheros en una única base de datos, para después poder ser consultados mediante los informes que tiene que generar la aplicación y sacar estadísticas de consumo. También se podrá ver los consumos de cada uno de los empleados, al igual que cada uno de ellos pueda ver su propio consumo.
- g) Gestión de cierres. Al final de cada mes, todos los administrativos de obra tienen que mandar el cierre (facturación de la obra) para ser revisada y aprobada por los jefes de obra. A su vez estos cierres tiene que ser aprobados por el director de construcción. En caso de existir discrepancias, el jefe de obra o si se ha escalado al siguiente nivel el director de construcción, podrá no aprobar dicho cierre, para que se rectifique.
- h) Gestión de proveedores. Tiene que centralizar las certificaciones que tiene cada uno de los proveedores que suministran a las obras. Se podrá asignar a un empleado la responsabilidad de encargarse de actualizar estos certificados solicitándolos al proveedor cuando se acerque a su vencimiento, y ver para que obra trabaja cada uno de los proveedores.

- i) Gestión de personal. Permitirá a miembros del departamento de recursos humanos, informar a los departamentos de informática, calidad y recursos humanos, el alta o baja de un nuevo empleado en la empresa, con los datos pertinentes, para que, desde estos departamentos puedan tomar las medidas oportunas, de creación de cuentas en los sistemas informáticos, y asignación de permisos a los recursos que necesite acceder por su puesto asignado, así como la asignación de activos, si así procediese.
- j) Agenda centralizada. Ya que la base de datos contiene los datos de cada uno de los usuarios (nombres, apellidos, tlf, cargo, email, etc..) esta sección se encargara de ofrecer al usuario una forma de consultarlos en forma de agenda.
- k) Cerrar sesión. Los usuarios validados deberán poder cerrar su sesión, para posibilitar a otros usuarios de la aplicación su inicio de sesión, aplicándose a estos últimos sus propias directivas de seguridad.
- l) Otras utilidades. Como la solicitud de asistencias remotas al departamento de informática.
 - o Subir ficheros a la web, motivados por la existencia de las limitaciones en el tamaño de ficheros en la mayoría de servidores de correo, este servicio tendrá que ofrecer una forma fácil de colgar ficheros en un servidor web.
 - o Directorio de enlaces, a los distintos servicios web del grupo.

2.2.3 Características del usuario

A continuación vamos a ver qué tipo de usuarios van a usar el producto y como afectan estos a las funciones que debe realizar la aplicación.

El nivel de especialización del usuario dentro ya del sistema operativo, o bien dentro del entorno de trabajo de un inventario, ha de ser mínimo. Es decir, no precisa conocimiento experto del sistema operativo con el que trabaje (en este caso, WINDOWS). Sin embargo, es conveniente que tenga un conocimiento acerca de los procesos administrativos, así como algunos procesos concretos acerca de la gestión de obras.

Todo usuario dado de alta en el sistema puede autenticarse, pero según que permisos se le hayan dado al usuario podrá ver más servicios o menos, y dentro de cada servicio

según el perfil de usuario puede realizar unas tareas u otras, aunque esto no hace que dichos usuarios requieran mayores conocimientos acerca de la aplicación. Estos perfiles o permisos vienen dados según las tareas concretas que ese usuario deba modificar en un instante concreto.

2.2.4 Restricciones Generales.

La aplicación esta implementada bajo la plataforma de desarrollo RAD de Borland, Delphi 7. Para el correcto funcionamiento de la aplicación será necesario disponer de un mínimo de un ordenador con Microsoft Windows 2000 o XP instalado en el mismo. Además es necesario un servidor de bases de datos MYSQL. Los ordenadores clientes deberán disponer de una cantidad mínima de memoria de 512G MB de RAM y unas 20MB de espacio en disco duro.

Los ordenadores clientes no requerirán grandes capacidades de procesamiento, si bien es deseable que el servidor de bases de datos sea más potente, a fin de poder dar servicio eficiente a todas las conexiones que se hagan a él.

2.2.5 Supuestos y Dependencias.

El buen funcionamiento del programa dependerá del funcionamiento de MS Windows utilizado, pudiéndose no ejecutar en caso del mal funcionamiento del sistema operativo citado, siendo otro factor importante el funcionamiento de la red.

Por otro lado, la velocidad de procesamiento del servidor puede verse afectado por el número de ordenadores clientes que estén solicitando información de manera simultánea.

2.3 *Requerimientos específicos.*

2.3.1 Requerimientos de interfaces externos.

2.3.1.1 Interfaces de usuario.

Existe una única interface la cual al iniciar la aplicación muestra un formulario de identificación donde se comprueba el usuario y el password introducidos en el formulario. Si la identificación es satisfactoria, se mostraran en el menú los servicios a los que tiene permiso acceder dicho usurario. Junto a la derecha del menú se encuentra el panel en el cual da paso a cada modulo que gestiona los distintos formularios que compone el servicio, que se solicite a través del menú.

2.3.1.2 Interfaces Hardware.

El usuario deberá disponer de un ordenador personal y conexión al servidor de bases de datos.

2.3.1.3 Interfaces Software.

El programa está desarrollado bajo el sistema operativo MS WINDOWS XP Professional.

2.3.2 Requisitos funcionales.

2.3.2.1 Iniciar sesión de usuario registrado.

Introducción: Para que un usuario pueda acceder a los servicios a los que tiene acceso tiene que autenticarse en el sistema.

Entradas: El usuario introduce el nombre de usuario y el password para registrarse.

Proceso: La aplicación comprueba que coincidan el nombre de usuario y el password con los que hay en la base de datos.

Salida: Se desplegara en el menú aquellas opciones a las cuales el usuario tiene permiso acceder, y en el panel principal, se mostrara la pantalla principal común a todos los usuarios, donde se mostrara el listado de activos informáticos, teléfonos móviles, y tarjetas sim que tiene asignado el usuario autenticarse. El listado de incidencias informáticas o de comunicaciones que el usuario tiene abiertas.

2.3.2.2 Lista de enlaces.

Introducción: Tener un listado de ayuda al usuario con enlaces interesantes.

Entradas: Un usuario ya dado de alta en la parte de administración, dispone un desplegable con enlaces a los distintos recursos.

Proceso: Al pinchar sobre la opción del menú enlaces

Salida: Se desplegara una lista básica de enlaces a recursos.

2.3.2.3 Colgar archivo.

Introducción: Permite fácilmente colgar un fichero en un servidor web para permitir su descarga.

Entradas: Se seleccionara el fichero rar que se desee colgar en el servidor web.

Proceso: La aplicación colocara el fichero suministrado en un directorio de un servidor web

Salida: Aparecerá por pantalla el contenido de la tabla que gestiona los usuarios de la

Aplicación, recargada con los datos actualizados.

2.3.2.4 Asistencia remota.

Introducción: La aplicación permite que un usuario solicite asistencia remota al departamento informático.

Entradas: Seleccionar el nombre del informático.

Proceso: Se realizara una petición de escritorio remoto por vnc al usuario seleccionado.

Salida: Ofrecerá al informático la posibilidad de gestionar el equipo remoto que le haya solicitado la asistencia remota.

2.3.2.5 Gestión de activos fijos

2.3.2.5.1 Añadir activo.

Introducción: Permitir al usuario poder dar de alta un activo en el sistema para asignarlo a un usuario, permitiendo tener todos los activos inventariados y sabiendo quien lo tiene.

Entradas: Activo y Usuario. Estas entidades se encuentran definidas en el capítulo entidad y relación.

Procesos: Se añadirá el activo a la base de datos y se asignara a un usuario.

2.3.2.5.2 Consultar activo.

Introducción: Ofrecer al usuario una forma de saber que activos son los que el tiene asignados, y en caso de ser un informático poder consultar quien tiene que activo.

Procesos: Se consultará la tabla de activos cruzándola con el identificador del usuario, o del usuario seleccionado.

Entradas: Id de usuario.

Salida: Listado de activos que tenga el usuario asignado.

2.3.2.5.3 Solicitar activo.

Introducción: Permitirá a un usuario solicitar al departamento de compras un activo.

Entradas: Lista de tipos de activo que quiere solicitar.

Procesos: Una vez realizada la lista de activos y enviada al sistema, este se encargara de enviar una serie de correos electrónicos al departamento de compras y a su jefe del usuario, para evaluar la adquisición de estos.

Salida: Correrios electrónicos a los involucrados.

2.3.2.6 Gestión de facturación telefónica

2.3.2.6.1 Introducir facturas.

Introducción: El objetivo es ofrecer al usuario introducir en la base de datos la facturación pre procesada por un programa llamado Factel.

Entradas: Fecha facturación, Empresa, Fichero xls con la facturación, cantidad facturada, tipo factura, descripción.

Procesos: Se almacenaran los datos en la base de datos.

Salida: Confirmación de que los datos han sido guardados correctamente.

2.3.2.6.2 Consultar facturas.

Introducción: El programa permitirá al usuario listar la facturación de la empresa del mes y tipo seleccionado.

Entradas: Mes, Empresa, tipo.

Procesos: El programa cruzara mediante sentencias sql los datos proporcionados por los usuarios.

Salida: Listado agrupando las líneas con el consumo realizado por los usuarios de la empresa / mes seleccionados. Junto a la agrupación de las líneas por usuario aparecerá la suma del consumo realizado por la agrupación de líneas.

También se mostrara al final de la tabla, un total facturado, y un total consumido. Ya que el operador contratado factura por estimación y no por consumido.

2.3.2.7 Gestión cierres

2.3.2.7.1 Subir cierres.

Introducción: El programa permite al usuario subir la facturación del mes generado por una obra.

Entradas: Código obra, mes, Fichero.

Procesos: Almacenar en un ftp el fichero con el cierre, y guardar el registro en la base de datos para su posterior consulta.

Salida: Confirmación de que los datos han sido guardados correctamente.

o.

2.3.2.7.2 Borrar cierre.

Introducción: Permite al usuario marcar un cierre como borrado o como nulo.

Entradas: Acción borrado, el cierre.

Procesos: Ser realiza un update en la base de datos para marcar el registro que representa el cierre como borrado.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.7.3 Bajar cierre.

Introducción: Permitir al usuario la consulta del cierre.

Entradas: Cierre a descargar.

Procesos: Realiza la descarga tras consultar que fichero tiene asociado el cierre.

Salida: Fichero con el cierre.

2.3.2.7.4 *Invalidar cierre.*

Introducción: Permite al usuario marcar un cierre como borrado o como nulo.

Entradas: Acción de anulación, cierre.

Procesos: Se realiza una actualización en la base de datos para marcar el registro que representa el cierre como anulado.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.8 Gestión Obras

2.3.2.8.1 *Crear obra.*

Introducción: Permite al usuario dar de alta una obra en la base de datos.

Entradas: Código de obra, nombre de obra.

Procesos: Se registra la obra en el sistema.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.8.2 *Borrar obra.*

Introducción: Permite al usuario el borrado de una obra en la base de datos.

Entradas: Código obra.

Procesos: Eliminación de la obra del sistema, si no tiene ni cierres ni proveedores asignados.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.8.3 *Asignar usuario con categoría a una obra.*

Introducción: Permite al usuario asignar un usuario, y la categoría de este a una obra.

Entradas: Usuario, categoría, obra.

Procesos: Una vez que el usuario haya seleccionado mediante un formulario el usuario, categoría, y la obra, este será registrado en la base de datos.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.9 Gestión de incidencias

2.3.2.9.1 *Añadir incidencia.*

Introducción: El programa permitirá al usuario dar de alta una incidencia, y enviar notificaciones por correo electrónico.

Entradas: Tipo de incidencia, Empresa, numero teléfono, urgencia, código de inventario, ubicación, Descripción, usuario.

Procesos: Mediante los datos proporcionados la aplicación insertara en la tabla incidencias el nuevo registro.

Salida: Se mostrara la confirmación de la creación de la incidencia, y se enviaran dos correos, uno al departamento de informática, y otro al creador de la incidencia para que tenga constancia dela creación.

2.3.2.9.2 *Consultar incidencias.*

Introducción: El programa mostrara un listado con las incidencias abiertas por el usuario en el caso de ser un usuario quien las consulte o todas las incidencias en caso de ser un informático.

Entradas: usuario.

Procesos: Consulta y mostrar las incidencias creadas por un usuario o por todos los usuarios.

Salida: Lista de incidencias.

2.3.2.9.3 *Añadir actuación o cierre.*

Introducción: El programa ofrecerá al usuario la forma de añadir actuaciones a las incidencias.

Entradas: Código de incidencia, descripción de actuación, fecha, hora, tipo de actuación

Procesos: Añade una actuación o una actuación de tipo cierre de incidencia, en la tabla de actuaciones.

Salida: En caso de ser una incidencia de tipo cierre se enviara un correo electrónico al usuario que creó la incidencia para notificar la resolución y otro al departamento de informática para que se conozca que dicha incidencia ha sido resuelta.

2.3.2.9.4 *Consultar actuaciones.*

Introducción: El programa permite ver al informático las actuaciones que se han ido añadiendo a una incidencia.

Entradas: código de incidencia.

Procesos: Consultar que actuaciones tiene una incidencia.

Salida: Lista de actuaciones.

2.3.2.10 Listín telefónico.

2.3.2.10.1 *Mostrar listín.*

Introducción: El programa ofrecerá un listín de teléfonos.

Entradas: Nombre, apellidos, Nick.

Procesos: Consultara en la base de datos todos los datos que se puede extraer del usuario que cumpla con las condiciones de entrada.

Salida: Listado de usuario que cumplan la condición, con los campos, Empresa, Nick, nombre, apellidos, número de teléfono, extensión, email, cargo, superior, dirección de skype, etc.....

2.3.2.11 Gestión Proveedores

2.3.2.11.1 Alta proveedor.

Introducción: La aplicación nos permitirá dar de alta proveedores.

Entradas: Nombre, CIF., nombre.

Procesos: Registrar los datos suministrados por el usuario en la base de datos.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.11.2 Añadir documentación del proveedor.

Introducción: La aplicación nos permitirá adjuntar la documentación que demuestre las certificaciones oficiales de la empresa proveedora y su caducidad.

Entradas: código de proveedor, documentación que demuestre certificaciones oficiales, caducidad.

Procesos: Guardar en la base de datos los certificados y asociarlos a un proveedor

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.11.3 Asignar proveedor a una obra.

Introducción: La aplicación permitirá la vinculación de una obra con proveedores.

Entradas: código obra, código proveedor.

Procesos: Relacionar en la base de datos, el proveedor con la obra.

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.11.4 Editar un proveedor.

Introducción: La aplicación permitirá actualizar la documentación de los proveedores.

Entradas: código de proveedor, documentación que demuestre certificaciones oficiales, caducidad.

Procesos: Actualizar en la base de datos los certificados y asociarlos a un proveedor

Salida: Confirmación de que los datos han sido actualizados correctamente.

2.3.2.11.5 Consultar un proveedor.

Introducción: La aplicación permitirá al usuario consultarla documentación asociada a un proveedor.

Entradas: Código de proveedor.

Procesos: Consultara en la base de datos la documentación asociada a un proveedor.

Salida: Listado de la documentación y señalizara aquellos documentos que estén caducadas.

2.3.2.12 Gestión personal.

2.3.2.12.1 Altas.

Introducción: La aplicación permitirá al usuario dar de alta al personal contratado.

Entradas: Nombre, apellidos, categoría, Nick.

Procesos: Crear un nuevo usuario en la base de datos.

Salida: Confirmación de que el usuario a sido creado correctamente, se emitirá un correo electrónico, al departamento de informática, al de comunicaciones, y a su jefe directo para notificar la llegada de un empleado y el cargo que desempeñara, para así poder tomar las medidas adecuadas.

2.3.2.12.2 Bajas.

Introducción: La aplicación permitirá al usuario dar de alta al personal contratado.

Entradas: Nick.

Procesos: Se marcara en la base de datos que dicho usuario ha hecho baja.

Salida: Confirmación de que el usuario a sido eliminado correctamente, se emitirá un correo electrónico, al departamento de informática, al de comunicaciones, y a su jefe directo para notificar la baja de un empleado, para así poder tomar las medidas adecuadas.

2.3.3 Requisitos de eficiencia.

El producto está pensado para un trabajo multiusuario, en el, una gran cantidad de usuarios puedan estar accediendo simultáneamente a los datos mediante una instancia de la aplicación, ya sea para consulta o modificación de información. El rendimiento de la aplicación, depende en gran medida del número de usuarios que estén conectados al servidor de bases de datos. Por otro lado, unas malas características de la red instalada pueden influir negativamente en la eficiencia del mismo, pero con una conexión mínima de 1Mb/s será suficiente.

En el caso de un incremento notable en el número de usuarios de la red, el ordenador servidor puede verse incapaz de devolver los datos solicitados por los clientes en un tiempo deseable, bajando de esta manera la eficiencia de todos los clientes conectados. Se supone además, que el sistema operativo de los ordenadores (clientes y servidor) serán estables, no influyendo por tanto en el rendimiento de la aplicación.

2.3.4 Restricciones de diseño.

Las limitaciones del producto coinciden con las limitaciones del sistema operativo Windows dado que el programa cumple con los estándares establecidos por Microsoft para este producto.

Por otro lado, la aplicación está diseñada para una resolución de de pantalla de 1024 x 768 píxeles, si bien los formularios son autoredimensionables a resoluciones inferiores para una mayor compatibilidad con otros posibles ordenadores.

2.3.5 Limitaciones hardware.

Estas limitaciones vienen determinadas por la calidad de los equipos utilizados y de la red o conexión utilizada.

3 Análisis.

En el presente apartado se presenta un análisis de la aplicación desarrollada, empleando los modelos analíticos más utilizados hoy en día. Gracias a estos modelos, comprobaremos las diversas funcionalidades del sistema, distinguiendo los principales actores que podrán interactuar con la aplicación, así como averiguando cuales son los límites del sistema. Para observar todo esto, haremos uso del modelo de Casos de Uso, del Modelo Conceptual, del Modelo Relacional, así como de los Requisitos del Sistema.

3.1 Requisitos del Sistema.

El objetivo del sistema es el diseño y desarrollo de una aplicación que facilite la gestión de incidencias informáticas y de telecomunicaciones, gestión de activos informáticos y telefonía, gestión facturación telefónica, gestión de cierre de presupuestos de obra, gestión de proveedores, y otras pequeñas tareas, las cuales permita a los empleados realizar tareas específicas que o bien no se contempla en otras aplicaciones, o por generar un flujo de información más adecuado al estar la información centralizada en un servidor de bases de datos. Así como un seguimiento de las mismas, a través de las distintas etapas que estas deberán atravesar. Esta aplicación está diseñada para permitir un trabajo siguiendo el modelo cliente/servidor, gracias a lo cual, la complejidad en cuanto al mantenimiento de datos se centraliza en una sola máquina, motivo por el cual se evitarán posibles condiciones de carrera así como cualquier otro tipo incorrecto de flujo de información. Del mismo modo, el sistema facilitará la copia de seguridad de la base de datos, ofreciendo facilidades para la realización y restauración de la misma.

Los requisitos que debe ofrecer la solución son los siguientes:

- a) Los usuarios del sistema deberán poder gestionar el mantenimiento (altas, bajas, modificaciones, consultas) de incidencias, activos, presupuestos, proveedores, facturas telefónicas, personal, según el perfil o permisos que el usuario tenga para hacerlo, colgar ficheros en un servidor web, ofrecer un directorio de enlaces al usuario y facilitar la solicitud de asistencias remotas al departamento de informática.
- b) Según los permisos asignados al usuario este dispondrá de acceso a los distintos módulos de gestión (incidencias informáticas y de telecomunicaciones, gestión de

activos informáticos y telefonía, gestión facturación telefónica, gestión de cierre de presupuestos de obra, gestión de proveedores) y personal.

- c) A su vez dentro de cada modulo el usuario puede tener permisos específicos para realizar ciertas tareas.

3.2 Casos de Uso.

Los casos de uso describen cómo los actores interactúan con el sistema. El actor, en este caso los usuarios del sistema, genera eventos hacia el sistema que exigen la ejecución de operaciones.

A partir de los casos de uso se identifican los eventos dirigidos hacia el sistema y desde el sistema a los actores. En definitiva se detecta la comunicación dentro del sistema.

Los niveles de usuario a diferencia de otros software no se pueden definir como roles, si no, que cada uno de los usuarios tiene un perfil específico el cual le permite acceder a ciertos módulos. Y dentro de cada modulo, según su categoría profesional (y aquí si se podría hablar de rol) podrá realizar unas tareas u otras.

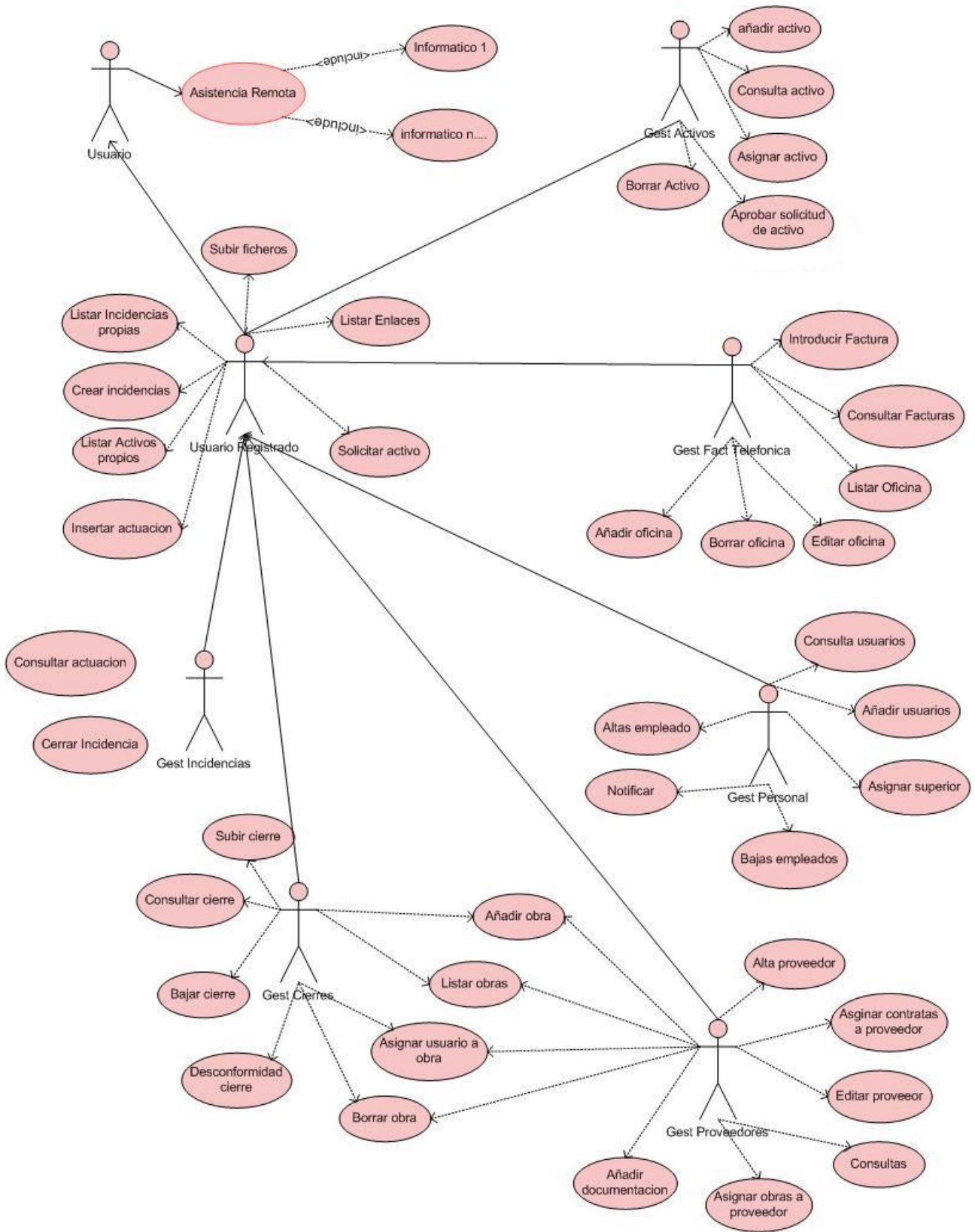


Ilustración 3.1: Diagrama de casos de uso.

3.2.1 Plantillas de descripción.

Las plantillas de especificación son una colección de plantillas diseñadas para facilitar la especificación de artefactos UML. A continuación se muestran algunas de las plantillas de descripción del sistema. No se mostraran todas por la extensión del programa.

Asistencia remota:

Actores	Usuario
Propósito	Que un usuario pueda resolverle un problema en el ordenador de forma remota.
Resumen	Solicitar asistencia remota al departamento informático
Precondiciones	Que el informático tenga el servicio de vnc ejecutándose.
Postcondiciones	Se ejecutara una instancia del cliente de vnc, solicitando una conexión remota al departamento de informática.
Incluye	
Extiende	
Hereda de	

Listar incidencias propias:

Actores	Usuario Registrado
Propósito	Haberse registrado en el sistema. Mostrar las incidencias que tiene sin resolver el usuario.
Resumen	Lista las incidencias sin resolver que el usuario registrado tenga.
Precondiciones	Que tenga alguna incidencia abierta.
Postcondiciones	Se mostraran los listados con las incidencias abiertas.
Incluye	

Extiende	
Hereda de	

Crear incidencias:

Actores	Usuario registrado
Propósito	Permitir al usuario dar de alta una incidencia. Y notificar al departamento de informática que hay un usuario con problemas.
Resumen	Debe registrar en la base de datos la incidencia con los datos que el usuario haya rellenado y enviar las notificaciones.
Precondiciones	Usuario registrado en el sistema.
Postcondiciones	La incidencia quedara registrada en la base de datos y se emitirá un correo electrónico al departamento de informática y al usuario.
Incluye	
Extiende	
Hereda de	

Listar activos propios:

Actores	Usuario Registrado.
Propósito	Haberse registrado en el sistema. Mostrar los activos asignados al usuario.
Resumen	Lista los activos al usuario registrado.
Precondiciones	Que tenga algún activo a su nombre.
Postcondiciones	Se mostraran los listados con los activos asignados..
Incluye	
Extiende	
Hereda de	

Subir ficheros:

Actores	Usuario registrado.
Propósito	Permitir al usuario una forma de compartir ficheros por internet.
Resumen	Debe permitirse subir un fichero a un servidor web para poder compartir dicho fichero por internet.
Precondiciones	El fichero tiene que estar previamente comprimido.
Postcondiciones	Se le dara al usuario un enlace al fichero para su descarga.
Incluye	
Extiende	
Hereda de	

Cerrar sesión:

Actores	Usuario registrado
Propósito	Cerrar la sesión activa
Resumen	Es necesario poder cerrar una sesión para validarse en el sistema con otro nombre de usuario.
Precondiciones	Haberse identificado en el sistema.
Postcondiciones	El sistema muestra la ventana de identificación.
Incluye	
Extiende	
Hereda de	

Insertar actuación:

Actores	Gestor de incidencias.
Propósito	Añadir actuaciones sobre una incidencia
Resumen	Registrar la actuación realizada por un informático en el sistema.
Precondiciones	Se tiene que haber seleccionado previamente una incidencia, a la cual se le añadirá la actuación.
Postcondiciones	Se registrara la actuación en la base de datos vinculada a la incidencia.
Incluye	
Extiende	
Hereda de	

Alta de proveedor:

Actores	Gestor de proveedores.
Propósito	Añadir un proveedor al sistema.
Resumen	Debe permitir al gestor de proveedores añadir un proveedor al base de datos, con la documentación aportada.
Precondiciones	El cif del proveedor tiene que ser correcto.
Postcondiciones	Se registrara el proveedor con los documentos aportados en la base de datos.
Incluye	
Extiende	
Hereda de	

Subir cierre:

Actores	Gestor de cierres.
Propósito	Compartir con el resto de gestores de cierres la facturación de la obra.
Resumen	Debe permitir guardar en la base de datos la facturación de la obra en un mes dado.
Precondiciones	Los documentos que forman el cierre tienen que estar comprimidos.
Postcondiciones	Se registrara el cierre perteneciente a una obra y fecha dada con los documentos aportados en la base de datos.
Incluye	
Extiende	
Hereda de	

3.3 Escenarios y Diagramas de Secuencias

Los diagramas de secuencia muestran la secuencia de mensajes entre objetos durante un escenario concreto. En dichos diagramas, cada objeto representado viene dado por una barra vertical, mientras que el transcurso del tiempo sucede de arriba hacia abajo.

Los diagramas de secuencia son de gran utilidad, dado que nos permiten obtener una visión rápida de las funcionalidades del programa y de qué elementos intervienen en la comunicación.

A continuación se mostrarán los distintos diagramas de secuencia de la aplicación, si bien algunos se han omitido, debido a la sencillez de los mismos.

3.3.1 Identificación en el sistema.

Este escenario sucederá cuando el usuario trate de validarse en la aplicación para poder hacer uso de la misma. El programa mostrará una ventana en la que le solicitará que se identifique por medio de su usuario y contraseña, comprobando que dichos datos sean válidos, aceptando o denegando su acceso en función de estos.

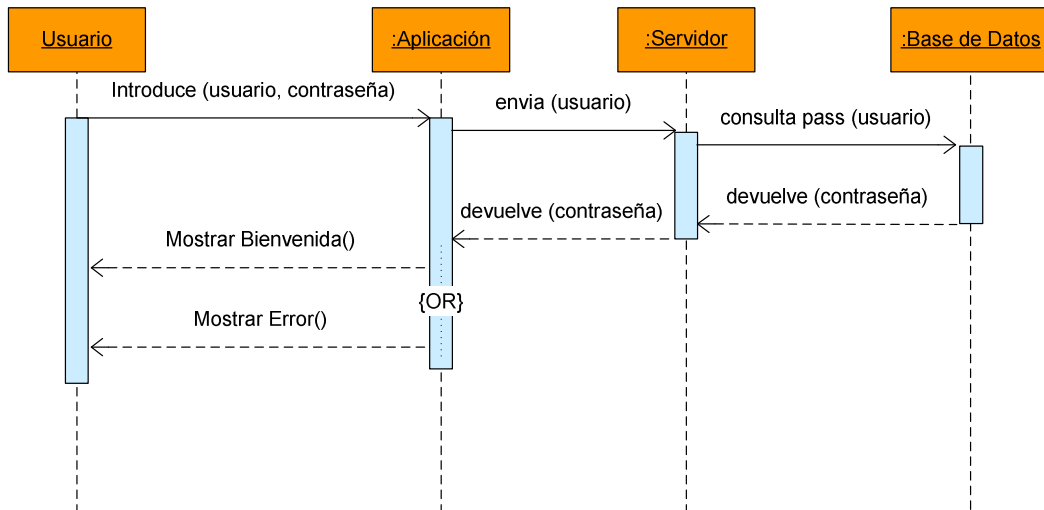


Ilustración 3.2: Secuencia inicio de sesión.

3.3.2 Alta de incidencia.

Este escenario sucederá cuando el usuario trate de dar de alta una incidencia. El programa mostrara una venta en la que se le solicitara unas series de datos que conformaran una incidencia. Una vez rellenados los datos se guardaran en la base de datos. Este diagrama de secuencia viene a ser claro ejemplo para muchos otros diagramas como pudiera ser el alta de un activo, un cierre, etc., donde el campo identificador de la entidad dada de alta es autoincrementar y no hay que verificar si existe previamente.

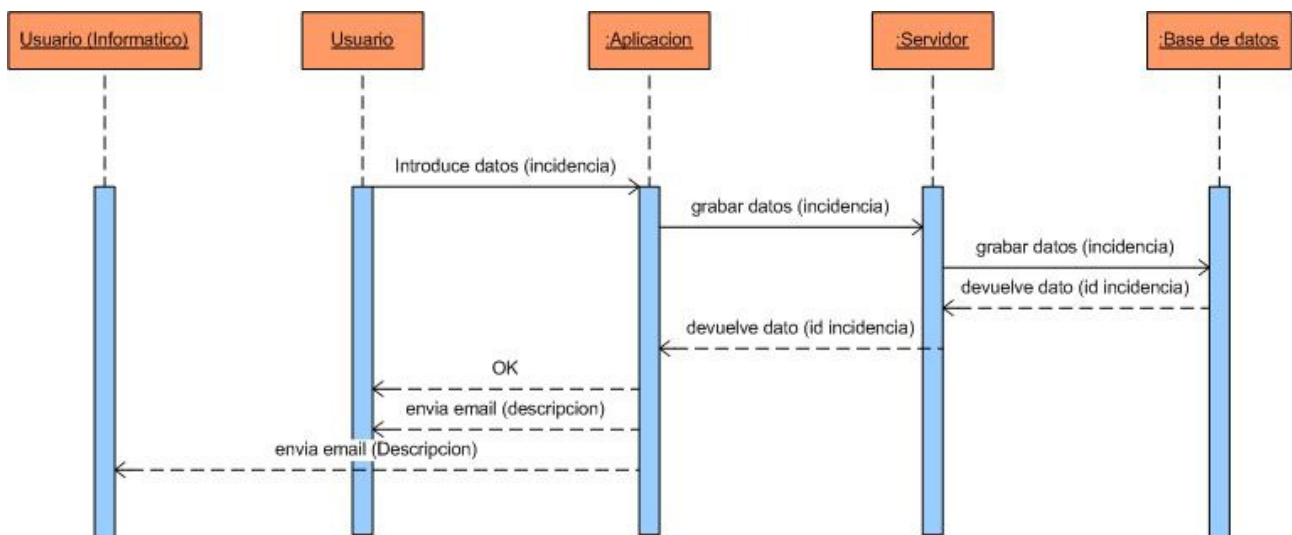


Ilustración 3.3: Secuencia alta de incidencia.

3.3.3 Alta de actuación.

Este escenario sucederá cuando un usuario de tipo informático quiera agregar una actuación a una incidencia.

La siguiente secuencia de datos viene a describir, otro tipo de entrada de datos dependiente de otra entidad, en este caso como una actuación depende de una incidencia, la incidencia tiene que ser previamente seleccionada.

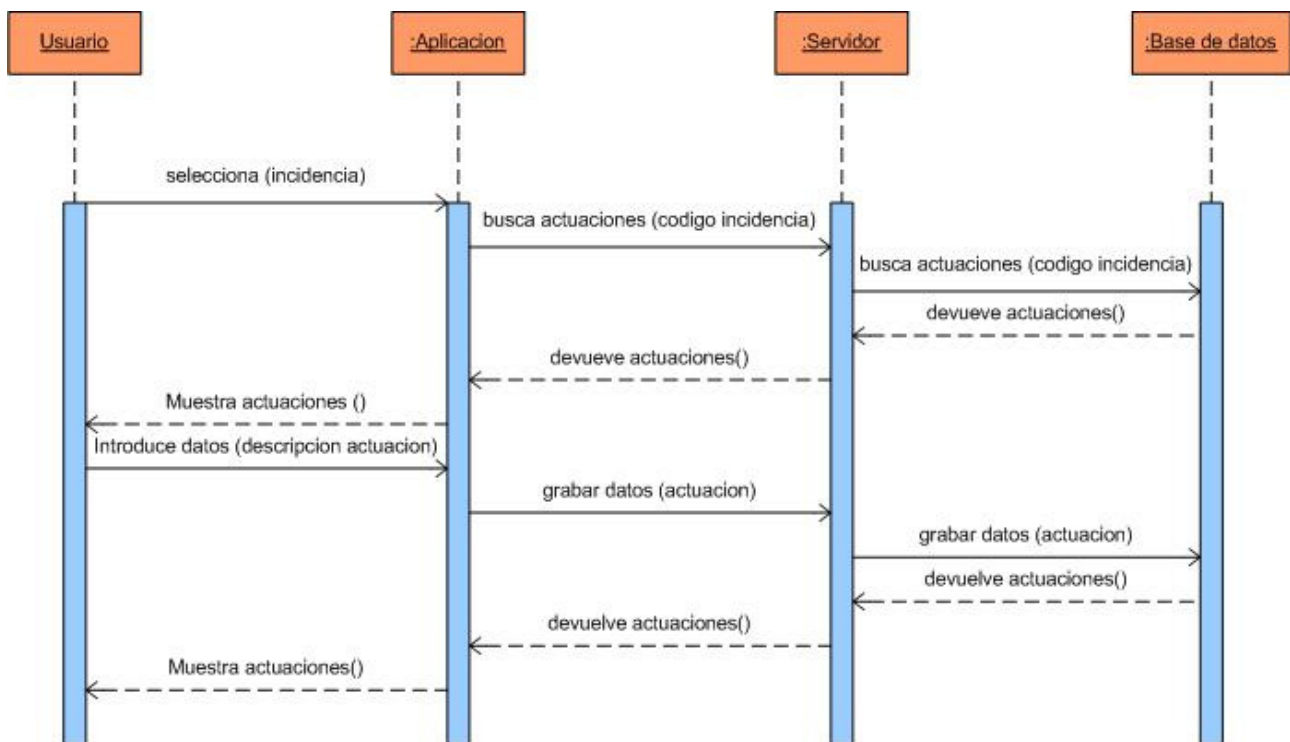


Ilustración 3.4: Secuencia alta de actuación.

3.3.4 Alta de proveedores.

Este otro escenario describe el alta de un proveedor, con sus certificaciones.

Para dar de alta el proveedor hay que verificar previamente si el CIF de la empresa que se quiere registrar no existe en la base de datos.

Como se puede apreciar este es el otro modelo que se puede presentar a la hora de dar de alta una entidad que tenga un campo único.

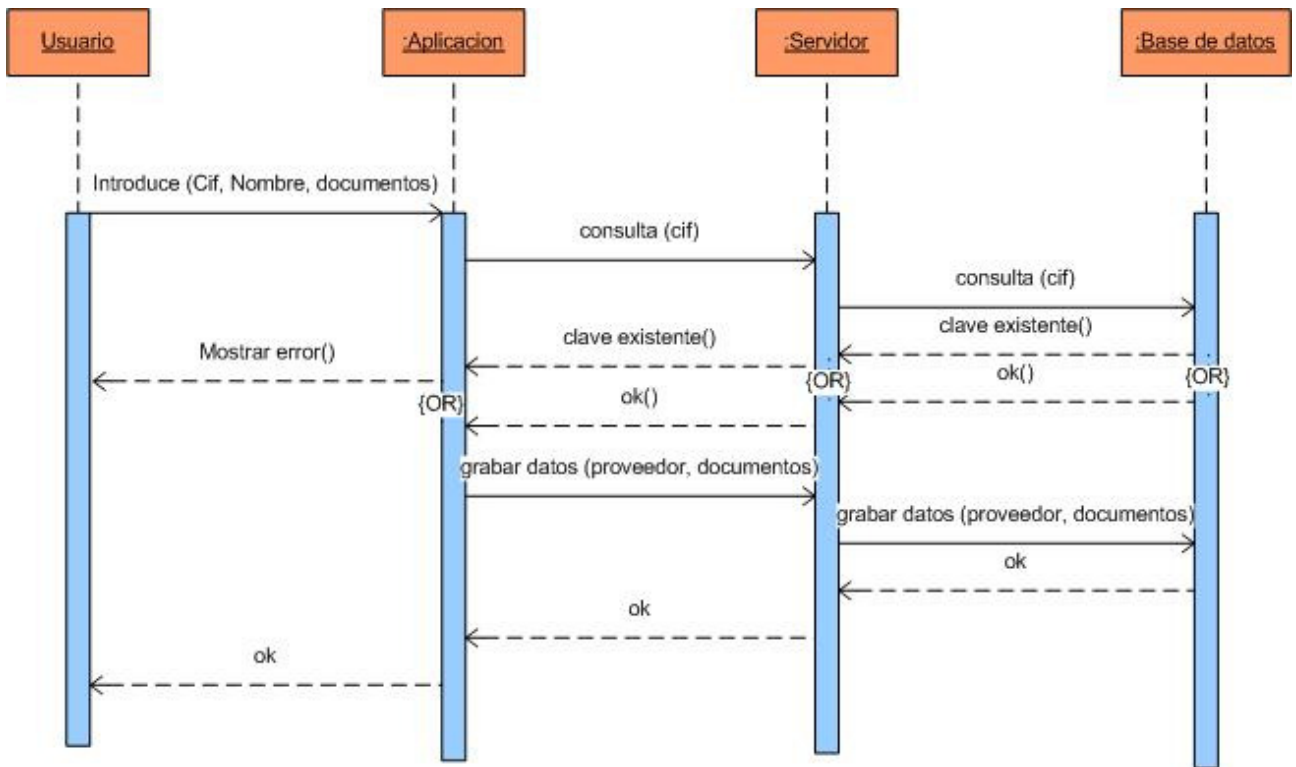


Ilustración 3.5: Secuencia alta de proveedor.

4 Diseño.

4.1 Arquitectura De La Aplicación.

Para el diseño de la aplicación se ha decidido utilizar una arquitectura cliente/servidor, pues al ser el ámbito de la aplicación dirigido a centralizar los datos en un único punto, el cual sea actualizado en tiempo real por múltiples clientes simultáneamente, evitar la redundancia de información, y mantener un flujo de información adecuado, se ha creído conveniente utilizar esta arquitectura concreta.

La arquitectura Cliente/Servidor hace referencia a una relación cooperativa entre dos o más entidades. En la mayoría de los casos, el cliente requiere una serie de recursos y de servicios que el servidor es capaz de proporcionarle. Ambos procesos pueden existir en una máquina o en máquinas separadas que se comunican a través de una red. Por lo general, la mayoría de las soluciones cliente/servidor son arquitecturas de dos capas. Las arquitecturas de dos capas requieren una interfaz de usuario que se instala y corre en un PC o estación de trabajo; y que envía solicitudes a un servidor para ejecutar operaciones complejas. Por ejemplo, una estación de trabajo utilizada como cliente puede ejecutar una aplicación de interfaz de usuario que interroga a un servidor central de bases de datos. En este tipo de arquitecturas, el servidor de la base de datos asume la responsabilidad de gestionar las conexiones, si bien la aplicación cliente debe utilizar los mecanismos necesarios para asegurar una gestión y escritura de datos correcta en la base de datos. La siguiente ilustración muestra el modelo de arquitectura de dos capas:

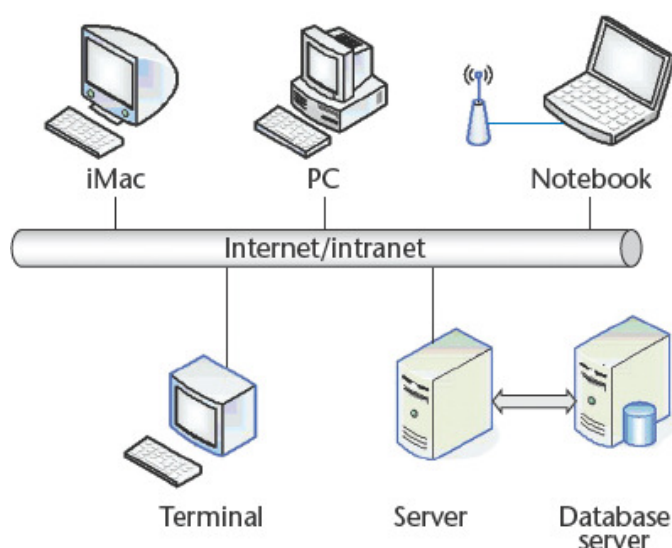


Ilustración 4.1: Cliente / Servidor

4.2 Modelo Relacional.

4.2.1 Introducción.

En 1970, el modo en que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. El modelo relacional representa la segunda generación de los SGBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa simple estructura hay un fundamento teórico importante del que carecen los SGBD de la primera generación, lo que constituye otro punto a su favor.

El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- Estructura de datos.
- Integridad de datos.
- Manejo de datos.

El modelo relacional se basa en el concepto matemático de *relación*, que gráficamente se representa mediante una tabla. Codd, que era un experto matemático, utilizó una terminología perteneciente a las matemáticas, en concreto de la teoría de conjuntos y de la lógica de predicados.

A continuación se definen algunos términos necesarios para comprender el modelo relacional:

- Una relación es una tabla con columnas y filas. Un SGBD sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas. Esta percepción sólo se aplica a la estructura lógica de la base de datos (en el nivel externo y conceptual de la arquitectura de tres niveles ANSI-SPARC). No se aplica a la estructura física de la base de datos, que se puede implementar con distintas estructuras de almacenamiento.
- Un atributo es el nombre de una columna de una relación. En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Una relación se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a los campos o atributos de esos registros. Los atributos pueden aparecer en la relación en cualquier orden.

- Un dominio es el conjunto de valores legales de uno o varios atributos. Los dominios constituyen una poderosa característica del modelo relacional. Cada atributo de una base de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.
- Una tupla es una fila de una relación. Los elementos de una relación son las tuplas o filas de la tabla. Las tuplas de una relación no siguen ningún orden.
- El grado de una relación es el número de atributos que contiene. El grado de una relación no cambia con frecuencia.
- La cardinalidad de una relación es el número de tuplas que contiene. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de las mismas varía constantemente.
- Una base de datos relacional es un conjunto de relaciones normalizadas.

4.2.2 El Modelo De La Aplicación

La base de datos realizada para la aplicación se ha creado para el gestor de bases de datos MySQL Server habiendo elegido como formato para las distintas tablas InnoDB. InnoDB provee a MySQL con el soporte para trabajar con transacciones, además de hacer un mejor bloqueo de registros para las instrucciones SELECT muy parecido al usado por Oracle, con lo que incrementa el rendimiento y la concurrencia en ambientes multiusuario, por otro lado, InnoDB es el único formato que tiene MySQL para soportar llaves foráneas (FOREIGN KEY). Además de todo lo comentado, InnoDB ofrece unos rendimientos superiores a la anterior tecnología de tablas de MySQL (MyISAM).

En primer lugar pasemos a analizar el diagrama entidad-relación, para posteriormente analizar cada una de las tablas, atributos y relaciones entre las tablas. Notar que en el diagrama entidad-relación se ha preferido omitir algunos atributos, para una mayor claridad del mismo.

La herramienta empleada para el diseño de la base de datos ha sido mysql workbench. Como la base de datos es muy extensa voy a mostrar los diagramas en tres partes para su mejor interpretación.

También se ha seccionado en tres apartados el modelo para un mejor entendimiento.

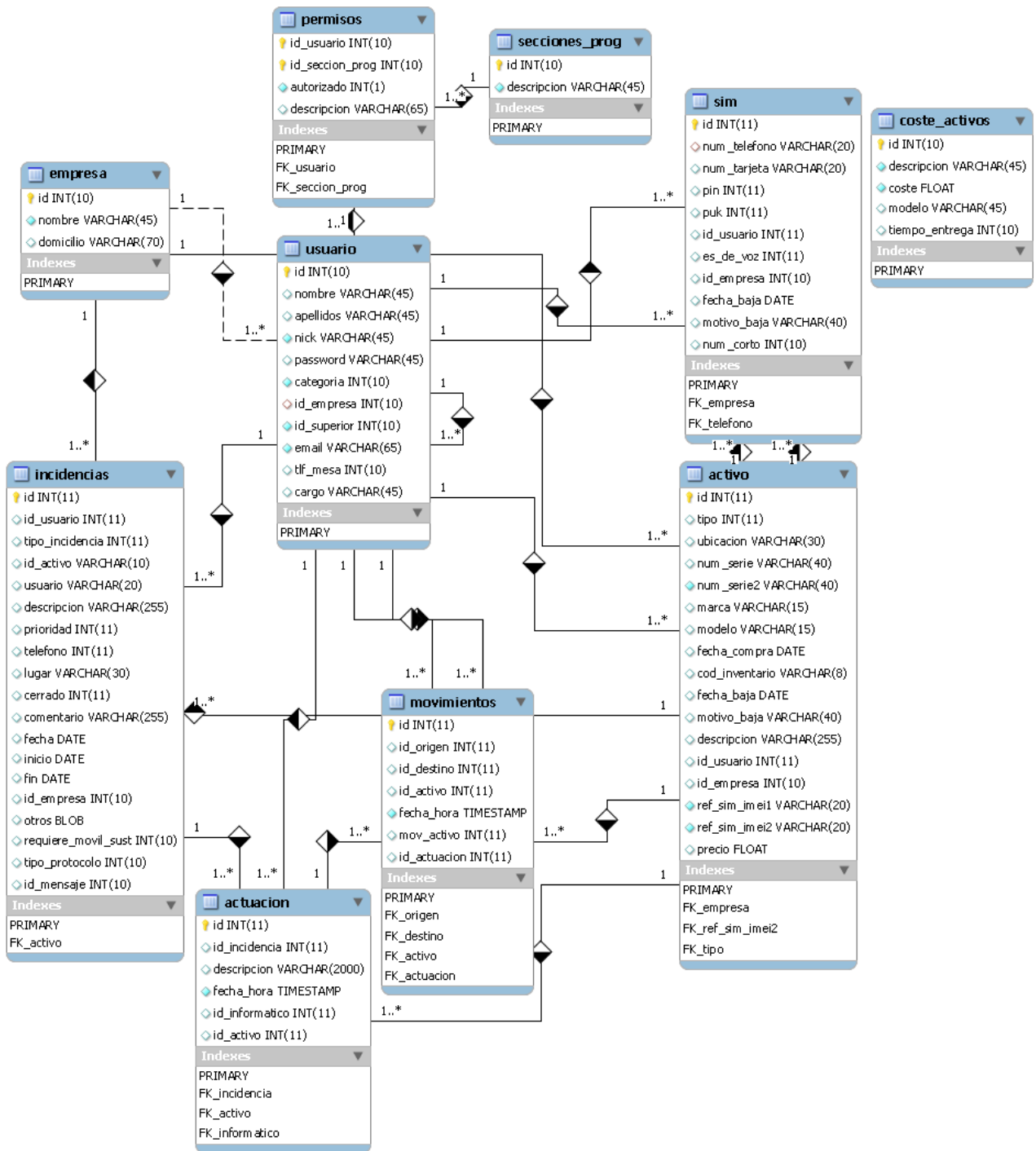


Ilustración 4.2: Incidencias, activos y usuarios.

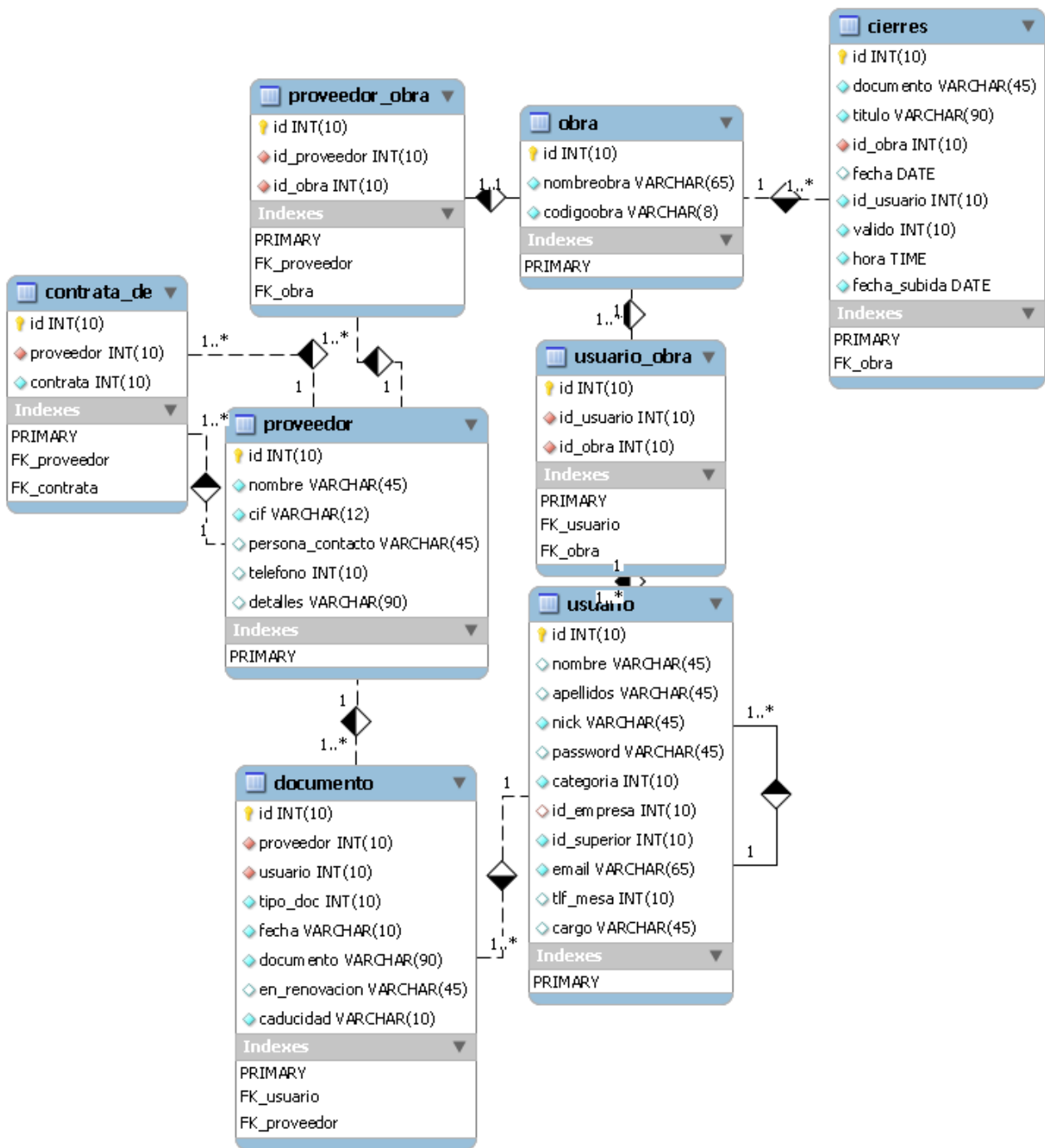


Ilustración 4.3: Obra, cierres, y proveedores.

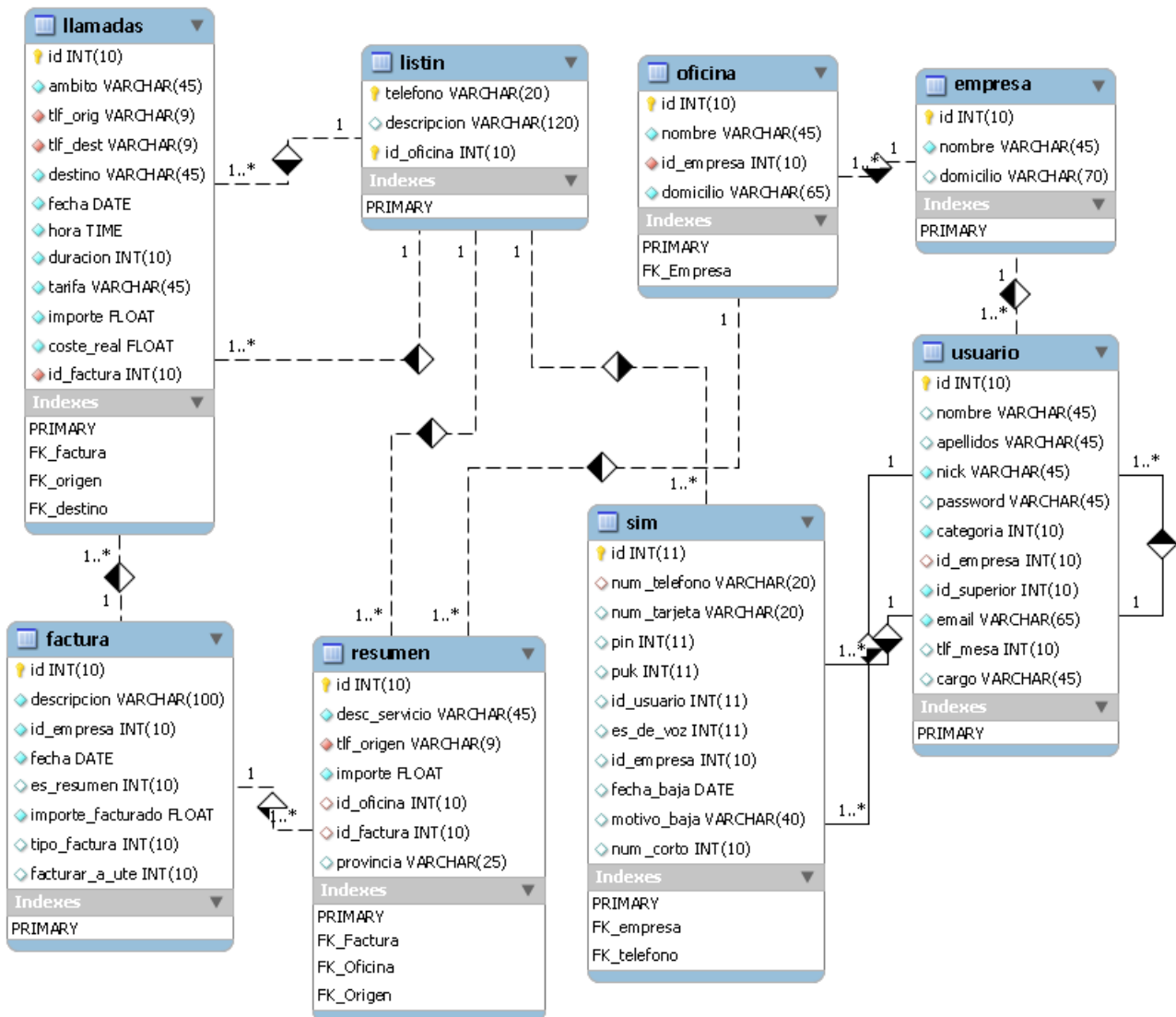


Ilustración 4.4: Telefonía.

Usuario

usuario(id:dom_id,nombre:dom_nombre,apellidos:dom_apellidos,nick:dom_nick,password:dom_password,categoria:dom_categoria,id_empresa:dom_id_empresa,id_superior:dom_id_superior,email:dom_email,tlf_mesa:dom_tlf_mesa,cargo:dom_cargo)

CP:{id}

VNN:{nick, categoria, email, id_superior}

C.Aj:{empresa} referencia a empresa

C.Aj:{superior} referencia a usuario

Listin

listin(telefono:dom_telefono,descripcion:dom_descripcion,id_oficina:dom_id_oficina)

CP:{telefono}

C.Aj:{id_oficina} referencia a oficina

Sim

sim(id:dom_id,num_telefono:dom_num_telefono,num_tarjeta:dom_num_tarjeta,pin:dom_pin,puk:dom_puk,id_usuario:dom_id_usuario,es_de_voz:dom_es_de_voz,id_empresa:dom_id_empresa,fecha_baja:dom_fecha_baja,motivo_baja:dom_motivo_baja,num_corto:dom_num_corto)

CP:{id}

C.Aj:{id_empresa} referencia a empresa

C.Aj:{telefono} referencia a listin

tipo activo

tipo_activo(id:dom_id,nombre:dom_nombre)

CP:{id}

VNN:{nombre}

Activo

activo(id:dom_id,tipo:dom_tipo,ubicacion:dom_ubicacion,num_serie:dom_num_serie,num_serie2:dom_num_serie2,marca:dom_marca,modelo:dom_modelo,fecha_compra:dom_fecha_compra,cod_inventario:dom_cod_inventario,fecha_baja:dom_fecha_baja,motivo_baja:dom_motivo_baja,descripcion:dom_descripcion,id_usuario:dom_id_usuario,id_empresa:dom_id_empresa,ref_sim_imei1:dom_ref_sim_imei1,ref_sim_imei2:dom_ref_sim_imei2,precio:dom_precio)

CP:{id}

VNN:{num_serie2}

C.Aj:{empresa} referencia a empresa

C.Aj:{ref_sim_imei1} referencia a sim

C.Aj:{ref_sim_imei2} referencia a sim

C.Aj:{tipo} referencia a tipo

C.Aj:{usuario} referencia a usuario

Ref_sim_imei1 y ref_sim_imei2 son valores únicos.

Incidencias

incidencias(id:dom_id,id_usuario:dom_id_usuario,tipo_incidencia:dom_tipo_incidencia,id_ activo:dom_id_activo,usuario:dom_usuario,descripcion:dom_descripcion,prioridad:dom_pr ioridad,telefono:dom_telefono,lugar:dom_lugar,cerrado:dom_cerrado,comentario:dom_co mentario,fecha:dom_fecha,inicio:dom_inicio,fin:dom_fin,id_empresa:dom_id_empresa)

CP:{id}

C.Aj:{usuario} referencia a usuario

C.Aj:{activo} referencia a activo

C.Aj:{empresa} referencia a empresa

actuacion

actuacion(id:dom_id,id_incidencia:dom_id_incidencia,descripcion:dom_descripcion,fecha_ hora:dom_fecha_hora,id_informatico:dom_id_informatico,id_activo:dom_id_activo)

CP:{id}

VNN:{fecha_hora}

C.Aj:{incidencia} referencia a incidencia

C.Aj:{activo} referencia a activo

C.Aj:{informatico} referencia a usuario

asign_activo

asign_activo(id:dom_id,id_usuario:dom_id_usuario,id_activo:dom_id_activo,fecha:dom_fe cha)

CP:{id}

VNN:{ fecha }

C.Aj:{ id_usuario } referencia a usuario

C.Aj:{ id_activo } referencia a activo

coste_activos

coste_activos(id:dom_id,descripcion:dom_descripcion,coste:dom_coste,modelo:dom_mod elo,tiempo_entrega:dom_tiempo_entrega)

CP:{id}

C.Aj:{usuario} referencia a usuario

C.Aj:{activo} referencia a activo

factura

factura(id:dom_id,descripcion:dom_descripcion,id_empresa:dom_id_empresa,fecha:dom_fecha,es_resumen:dom_es_resumen,importe_facturado:dom_importe_facturado,tipo_factura:dom_tipo_factura,facturar_a_ute:dom_facturar_a_ute)

CP:{id}

VNN:{descripcion, id_empresa, fecha, importe_facturado}

llamadas

llamadas(id:dom_id,ambito:dom_ambito,tlf_dest:dom_tlf_dest,destino:dom_destino,fecha:dom_fecha,hora:dom_hora,duracion:dom_duracion,tarifa:dom_tarifa,importe:dom_importe,coste_real:dom_coste_real,id_factura:dom_id_factura)

CP:{id}

VNN:{ambito, destino, fecha, hora, duracion, tarifa, importe, coste_real}

C.Aj:{origen} referencia a listin

C.Aj:{destino} referencia a listin

C.Aj:{factura} referencia a factura

Oficina

oficina(id:dom_id,nombre:dom_nombre,id_empresa:dom_id_empresa,domicilio:dom_domicilio)

CP:{id}

VNN:{nombre, domicilio}

C.Aj:{empresa} referencia a empresa

Resumen

resumen(id:dom_id,desc_servicio:dom_desc_servicio,tlf_origen:dom_tlf_origen,importe:dom_importe,id_oficina:dom_id_oficina,id_factura:dom_id_factura,provincia:dom_provincia)

CP:{id}

VNN:{desc_servicio, importe}

C.Aj:{factura} referencia a factura

C.Aj:{Oficina} referencia a oficina

C.Aj:{listin} referencia a listin

Proveedor

proveedor(id:dom_id,nombre:dom_nombre,cif:dom_cif,persona_contacto:dom_persona_contacto,telefono:dom_telefono,detalles:dom_detalles)

CP:{id}

VNN:{nombre,cif}

contrata de

contrata_de(id:dom_id,proveedor:dom_proveedor,contrata:dom_contrata)

CP:{id}

C.Aj:{proveedor} referencia a proveedor

C.Aj:{contrata} referencia a proveedor

documento

documento(id:dom_id,proveedor:dom_proveedor,usuario:dom_usuario,tipo_doc:dom_tipo_doc,fecha:dom_fecha,documento:dom_documento,en_renovacion:dom_en_renovacion,caducidad:dom_caducidad)

CP:{id}

VNN:{tipo_doc,fecha,documento,caducidad}

C.Aj:{usuario} referencia a usuario

C.Aj:{proveedor} referencia a proveedor

obra

obra(id:dom_id,nombre:dom_nombre,obra:dom_obra,codigoobra:dom_codigoobra)

CP:{id}

VNN:{nombreobra,codigoobra}

proveedor obra

proveedor_obra(id:dom_id,id_proveedor:dom_id_proveedor,id_obra:dom_id_obra)

CP:{id}

C.Aj:{id_proveedor} referencia a proveedor

C.Aj:{id_obra} referencia a obra

usuario obra

usuario_obra(id:dom_id,id_usuario:dom_id_usuario,id_obra:dom_id_obra)

CP:{id}

C.Aj:{id_usuario} referencia a usuario

C.Aj:{id_obra} referencia a obra

cierres

cierres(id:dom_id,documento:dom_documento,titulo:dom_titulo,id_obra:dom_id_obra,fecha:dom_fecha,id_usuario:dom_id_usuario,valido:dom_valido,hora:dom_hora,fecha_subida:dom_fecha_subida)

CP:{id}

VNN:{documento, titulo, valido, hora, fecha_subida}

C.Aj:{id_usuario} referencia a proveedor

C.Aj:{id_obra} referencia a proveedor

Dominio	Tipo de Datos	Comentarios
dom_ambito	Alfanumérico	Define el ámbito de la llamada (nacional, internacional, 900, ...)
dom_apellidos	Alfanumérico	Apellidos del usuario.
dom_archivo	Alfanumérico	nombre del fichero a descargar.
dom_authorized	Booleano	Indica si se autoriza o deniega el acceso .
dom_caducidad	Alfanumérico	Tiempo restante para la caducidad del documento de la
dom_cargo	Alfanumérico	Identificado del cargo que desempeña el empleado.
dom_categoria	Numérico	Categoría a la que pertenece el usuario.
dom_cerrado	Booleano	Indica si la incidencia esta cerrada.
dom_cif	Alfanumérico	Cif del proveedor contratado.
dom_cod_inventario	Numérico	
dom_codigoobra	Numérico	
dom_comentario	Alfanumérico	
dom_contrata	Numérico	id del proveedor.
dom_coste	Numérico	Coste estimado de un activo.
dom_coste_real	Numérico	Coste real de un activo.
dom_desc_servicio	Alfanumérico	
dom_descripcion	Alfanumérico	
dom_destino	Alfanumérico	
dom_detalles	Alfanumérico	
dom_documento	Alfanumérico	Enlace al documento.
dom_domicilio	Alfanumérico	
dom_duracion	Numérico	Duración desde que se inicia asta que se finaliza una incidencia.
dom_email	Alfanumérico	
dom_en_renovacion	Alfanumérico	quien ha solicitado el nuevo
dom_es_de_voz	Booleano	Discrimina si una sim es de tipo voz o de tipo datos.
dom_es_resumen	Booleano	Indica si una factura tiene un resumen asignado.

dom_facturar_a_ute	Numérico	Indica si dicha factura hay que facturarla a la UTE
dom_fecha	Fecha	
dom_fecha_baja	Fecha	
dom_fecha_cambio	Fecha	
dom_fecha_compra	Fecha	
dom_fecha_hora	Fecha	
dom_fecha_subida	Fecha	
dom_fin	Fecha	Hora de finalización de incidencia.
dom_hora	Hora	
dom_id	Numérico	
dom_id_activo	Numérico	
dom_id_activo_prest	Numérico	
dom_id_actuacion	Numérico	
dom_id_destino	Numérico	
dom_id_empresa	Numérico	
dom_id_factura	Numérico	
dom_id_incidencia	Numérico	
dom_id_informatico	Numérico	
dom_id_obra	Numérico	
dom_id_oficina	Numérico	
dom_id_origen	Numérico	
dom_id_proveedor	Numérico	
dom_id_seccion_prog	Numérico	
dom_id_superior	Numérico	
dom_id_usuario	Numérico	
dom_importe	Numérico	Importe real
dom_importe_facturado	Numérico	Importe facturado por la empresa proveedora de comunicaciones.
dom_info_adicional	Alfanumérico	
dom_inicio	Fecha	
dom_lugar	Alfanumérico	
dom_lugar_trabajo	Alfanumérico	
dom_marca	Alfanumérico	
dom_modelo	Alfanumérico	
dom_motivo_baja	Alfanumérico	
dom_mov_activo	Numérico	
dom_nick	Alfanumérico	
dom_nombre	Alfanumérico	
dom_nombre_doc	Alfanumérico	
dom_num_corto	Numérico	Numero de teléfono corto asociado a un teléfono móvil.
dom_num_serie	Numérico	numero de serie del activo.
dom_num_serie2	Numérico	Por si tuviera una segunda referencia
dom_num_tarjeta	Numérico	
dom_num_telefono	Numérico	

dom_obra	Alfanumérico	
dom_password	Alfanumérico	
dom_persona_contacto	Alfanumérico	
dom_pin	Numérico	
dom_precio	Numérico	
dom_proveedor	Numérico	
dom_provincia	Alfanumérico	
dom_puk	Numérico	
dom_ref_sim_imei1	Numérico	Código sim del imei1 que tiene el teléfono móvil o enlace digital.
dom_ref_sim_imei2	Numérico	Código sim del imei2 que tiene enlace digital.
dom_tarifa	Numérico	
dom_telefono	Numérico	
dom_tiempo_entrega	Numérico	
dom_tipo	Numérico	Tipo de activo (teléfono, monitor, teclado, impresora, etc....)
dom_tipo_cambio	Numérico	0 alta, 1 baja
dom_tipo_factura	Numérico	Móvil, Fija
dom_tipo_incidencia	Numérico	Indica el tipo de incidencia (Hardware, Software, telefonía, conexión).
dom_titulo	Alfanumérico	
dom_tlf_dest	Numérico	Destino de la llamada
dom_tlf_mesa	Numérico	
dom_tlf_origen	Numérico	Origen de la llamada
dom_ubicacion	Alfanumérico	
dom_usuario	Numérico	
dom_valido	Booleano	Si el cierre ha tenido conformidad o no.

5 Procesos De Desarrollo De Software

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Dicho proceso, en términos globales se muestra en la ilustración 5.1. Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. Aunque un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido.



Ilustración 5.1: Proceso de desarrollo de software

El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software.

A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos:

- Especificación de software: Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- Diseño e Implementación: Se diseña y construye el software de acuerdo a la especificación.
- Validación: El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- Evolución: El software debe evolucionar, para adaptarse a las necesidades de cliente.

Pressman caracteriza un proceso de desarrollo de software como se muestra en la ilustración 5.2.

Los elementos involucrados se describen a continuación:

- Un marco común del proceso, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
- Un conjunto de tareas, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.
- Las actividades de protección, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

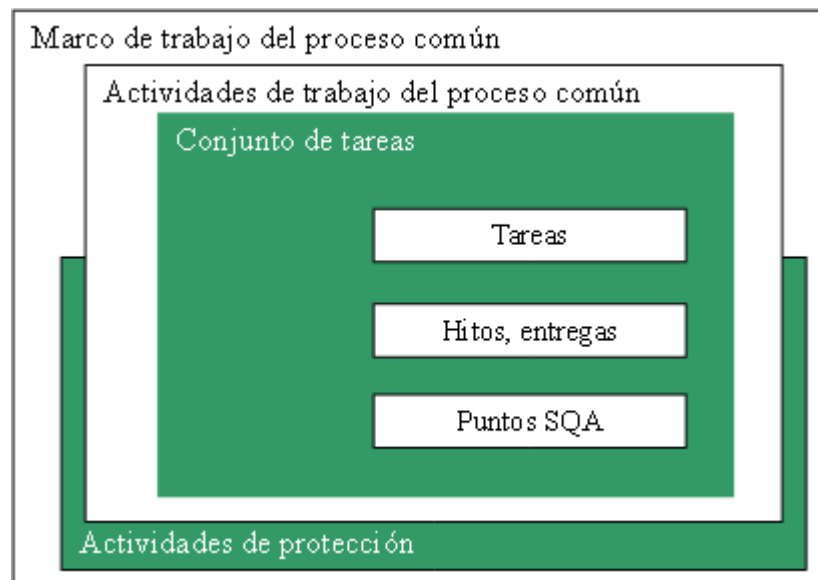


Ilustración 5.2: Elementos del proceso del software.

Otra perspectiva utilizada para determinar los elementos del proceso de desarrollo de software es establecer las relaciones entre elementos que permitan responder *Quién* debe hacer *Qué*, *Cuándo* y *Cómo* debe hacerlo.

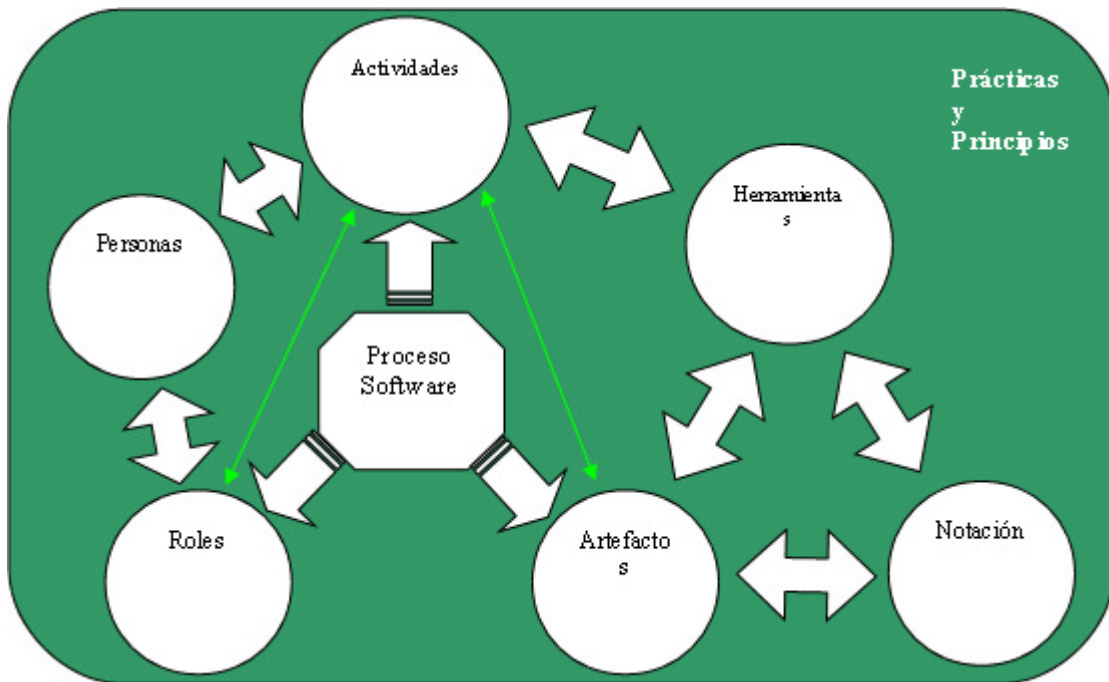


Ilustración 5.3: Relación entre elementos del proceso del software.

En la ilustración 5.3 se muestran los elementos de un proceso de desarrollo de software y sus relaciones. Así las interrogantes se responden de la siguiente forma:

Quién: Las personas participantes en el proyecto de desarrollo desempeñando uno o más roles específicos.

Qué: Un artefacto (pieza de información producida, modificada o usada por el usuario, que define una responsabilidad para un rol y está sujeta a control de versiones) es producido por un rol en una de sus actividades. Los Artefactos se especifican utilizando notaciones específicas. Las herramientas apoyan la elaboración de artefactos soportando ciertas notaciones.

Cómo y Cuándo: Las actividades son una serie de pasos que lleva a cabo un rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos artefactos.

La composición y sincronía de las actividades está basada en un conjunto de Principios y Prácticas. Las Prácticas y Principios enfatizan ciertas actividades y/o la forma como deben realizarse, por ejemplo: desarrollar iterativamente, gestionar requisitos, desarrollo basado en componentes, modelar visualmente, verificar continuamente la calidad, gestionar los cambios, etc.

5.1 Modelos de proceso de software

Sommerville define el modelo de proceso de software como “Una representación simplificada de un proceso de software, representada desde una perspectiva específica. Por su naturaleza los modelos son simplificados, por lo tanto un modelo de procesos del software es una abstracción de un proceso real.”

Los modelos genéricos no son descripciones definitivas de procesos de software; sin embargo, son abstracciones útiles que pueden ser utilizadas para explicar diferentes enfoques del desarrollo de software.

Algunos modelos tradicionales son:

- Codificar y corregir
- Modelo en cascada
- Desarrollo evolutivo
- Desarrollo formal de sistemas
- Desarrollo basado en reutilización
- Desarrollo incremental
- Desarrollo en espiral

¿Cuál es el modelo de proceso más adecuado?

Cada proyecto de software requiere de una forma de particular de abordar el problema. Las propuestas comerciales y académicas actuales promueven procesos iterativos, donde en cada iteración puede utilizarse uno u otro modelo de proceso, considerando un conjunto de criterios (Por ejemplo: grado de definición de requisitos, tamaño del proyecto, riesgos identificados, entre otros).

5.2 Metodologías de desarrollo de software

Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y/o diseño.

La comparación y/o clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. A grandes rasgos, si tomamos como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, podemos clasificar las metodologías en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales (o peyorativamente denominada Metodologías Pesadas, o Peso Pesado). Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo y se dirigen a equipos de desarrollo pequeños haciendo especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso. A continuación se revisan las dos últimas categorías de metodologías.

- Metodologías tradicionales (no ágiles).

Las metodologías tradicionales son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Entre las metodologías pesadas tenemos que destacar Rational Unified Process (RUP desde ahora) por ser uno de los procesos más generales de los existentes actualmente, ya que está pensado para adaptarse a cualquier proyecto.

- Metodologías ágiles.

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento).

Entre las metodologías ágiles con más auge en la actualidad tenemos que destacar Extreme Programming (XP desde ahora).

Por último hablaremos más detalladamente de RUP, como metodología tradicional, y de XP, como metodología ágil.

5.3 Rational Unified Process (RUP).

RUP (Rational Unified Process) es un producto comercial desarrollado y comercializado, por primera vez en el año 1998, por Rational Software Corporation una compañía de IBM. Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

5.3.1 Proceso dirigido por Casos de Uso

Según, los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en la ilustración 5.4.

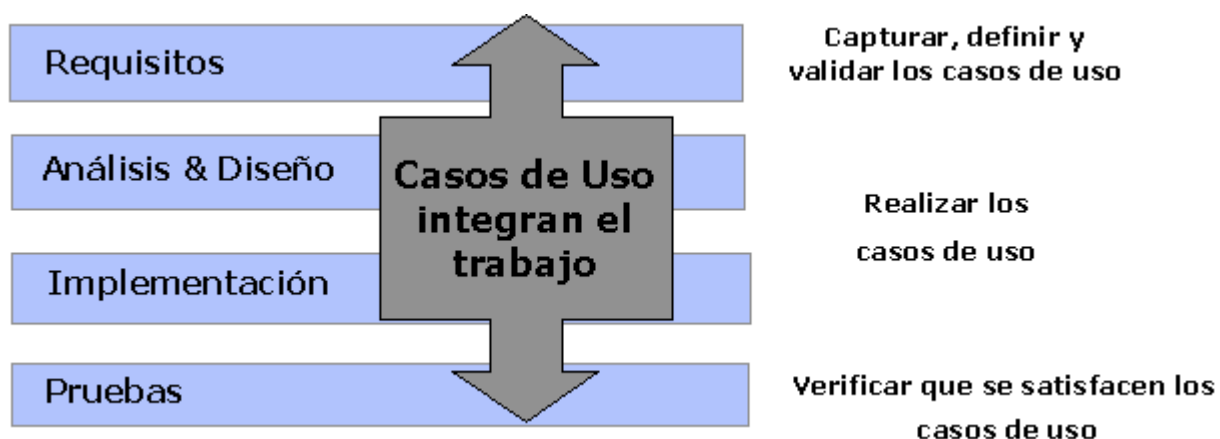


Ilustración 5.4: Los Casos de Uso integran el trabajo.

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Como se muestra en la ilustración 5.5, basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso.

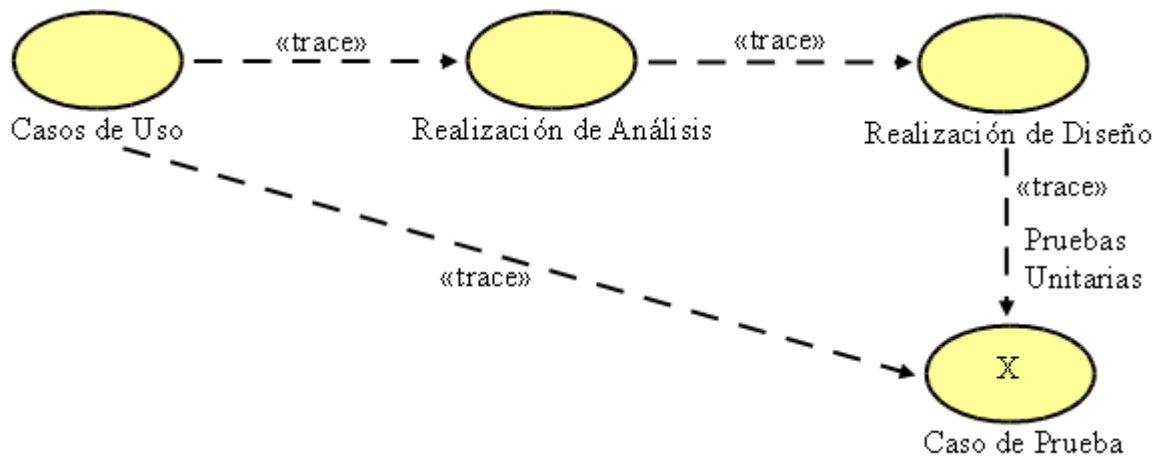


Ilustración 5.5: Trazabilidad a partir de los Casos de Uso.

5.3.2 Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca

que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

En la ilustración 5.6 se ilustra la evolución de la arquitectura durante las fases de RUP. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de *baselines* y se va modificando dependiendo de las necesidades del proyecto.

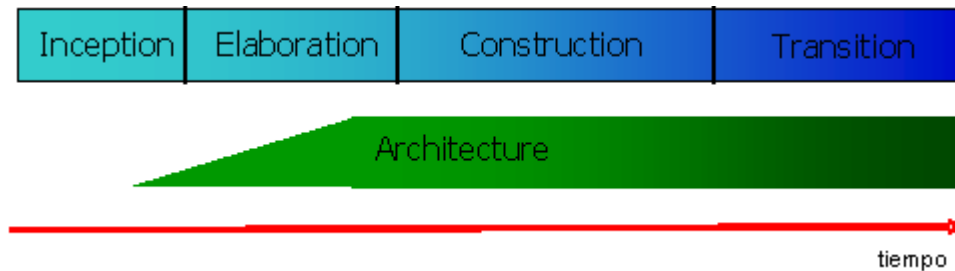


Ilustración 5.6: Evolución de la arquitectura del sistema.

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas.

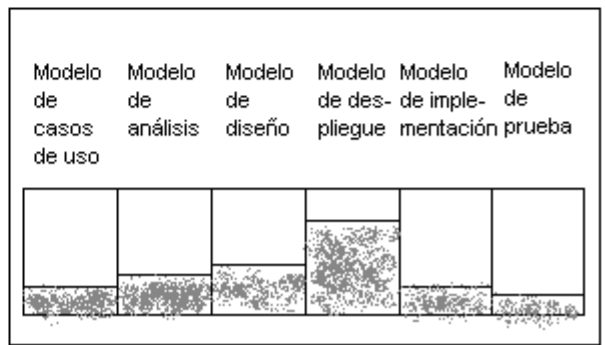
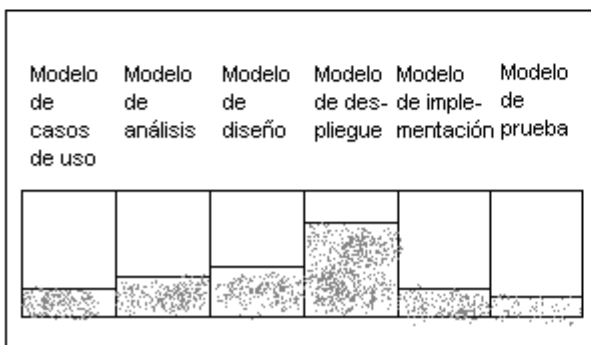
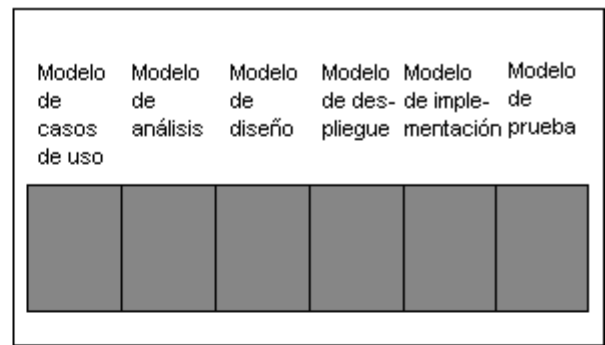
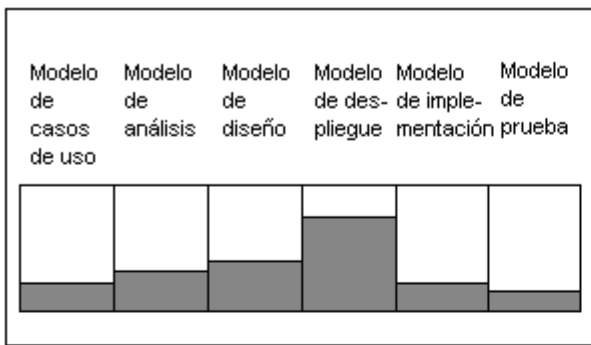


Ilustración 5.7: Los modelos se completan, la arquitectura no cambia drásticamente.

Al final de la fase de elaboración se obtiene una *baseline* de la arquitectura donde fueron seleccionados una serie de Casos de Uso arquitectónicamente relevantes (aquellos que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para el usuario y aquellos que cubran las funcionalidades significativas)

Como se observa en la ilustración 5.7, durante la construcción los diversos modelos van desarrollándose hasta completarse (según se muestra con las formas rellenas en la esquina superior derecha). La descripción de la arquitectura sin embargo, no debería cambiar significativamente (abajo a la derecha) debido a que la mayor parte de la arquitectura se decidió durante la elaboración. Se incorporan pocos cambios a la arquitectura (indicados con mayor densidad de puntos en la figura inferior derecha)

5.3.2.1 Proceso iterativo e incremental

El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se

puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en la ilustración 5.8. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

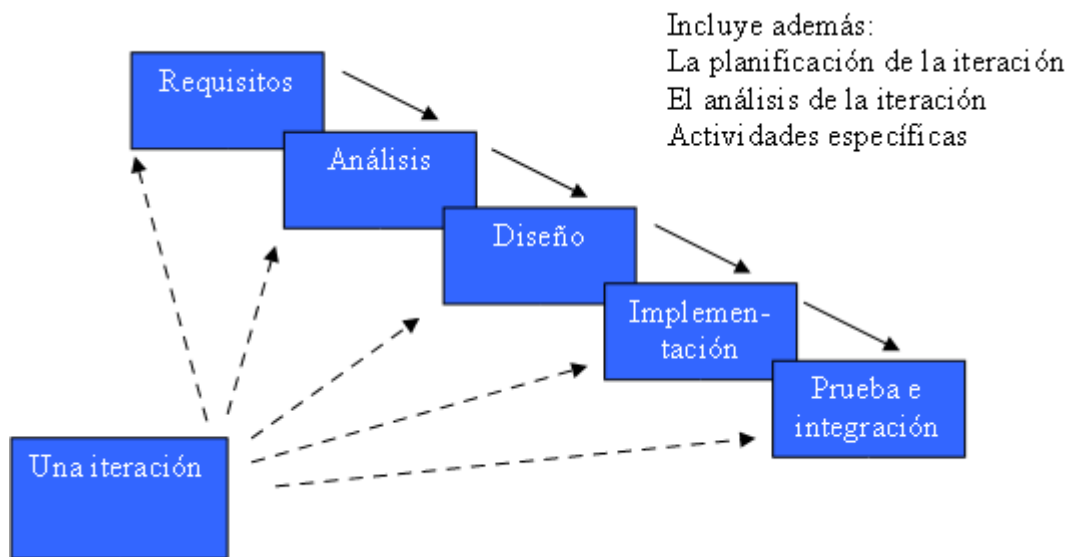


Ilustración 5.8: Los modelos se completan, la arquitectura no cambia drásticamente.

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. En la ilustración 5.9 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Organización a lo largo del tiempo

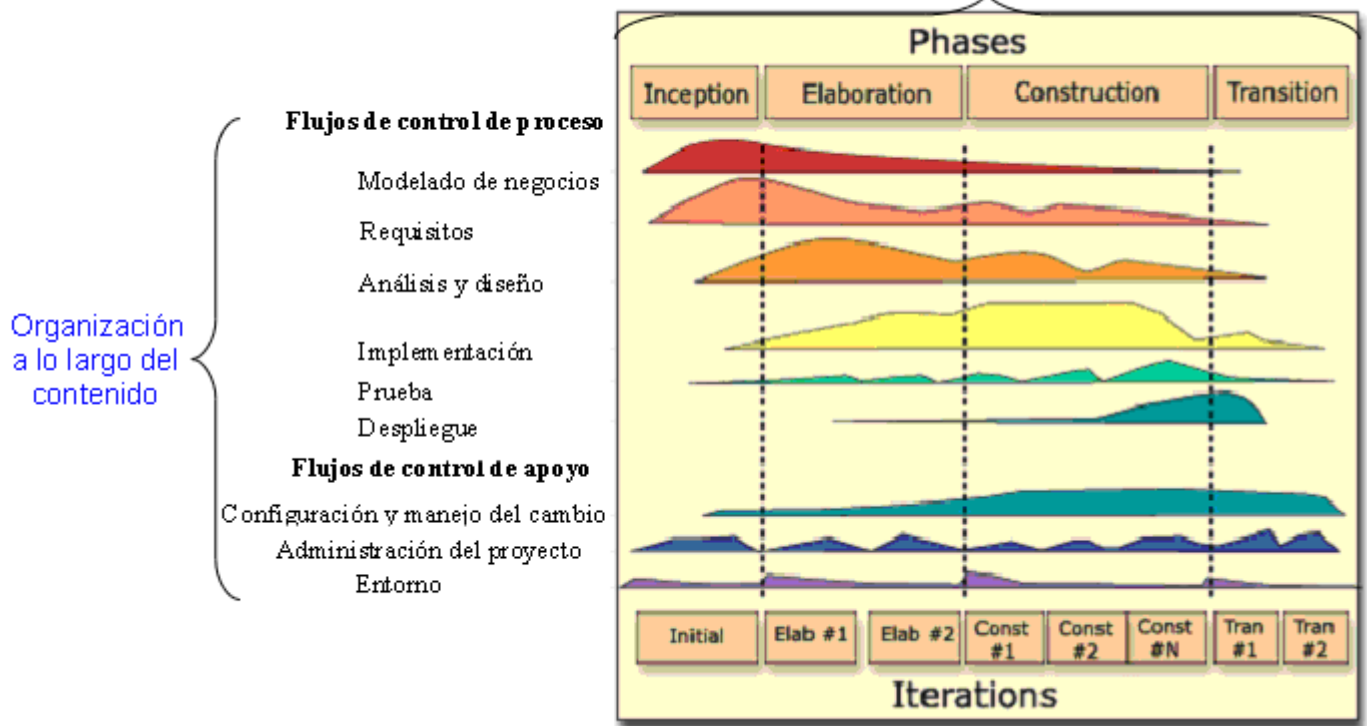


Ilustración 5.9: Estructura global de RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una *baseline* de la arquitectura. Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la *baseline* de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la *baseline* de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

5.3.2.2 Estructura del proceso

Tal y como se explica, el proceso puede ser descrito en dos dimensiones o ejes diferenciados:

- **Eje horizontal:** Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la ilustración 5.10 que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.
- **Eje vertical:** Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

5.3.2.3 Estructura Dinámica del proceso. Fases e iteraciones.

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto.

Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable. Podemos apreciar todo esto en la ilustración 5.10.

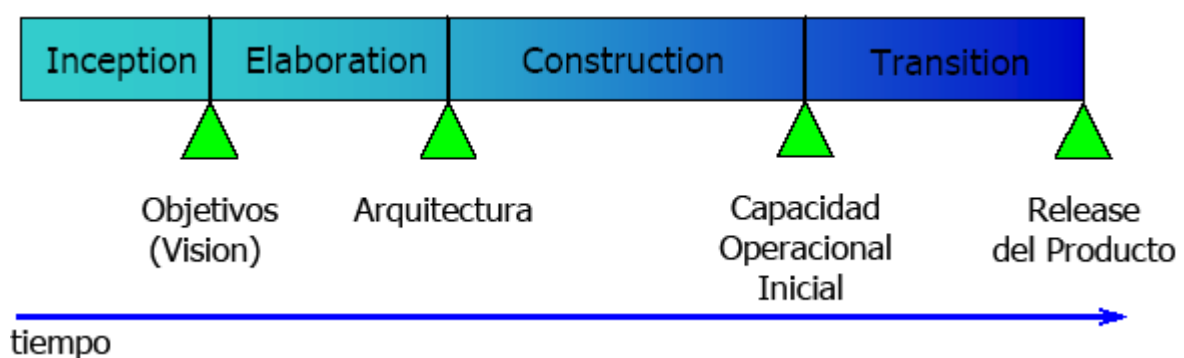


Ilustración 5.10: Fases e hitos en RUP.

5.3.2.4 Estructura Estática del proceso. Roles, actividades, artefactos y flujos de trabajo.

Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos los roles, que responden a la pregunta ¿Quién?, las actividades que

responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? (ver ilustración 5.11).

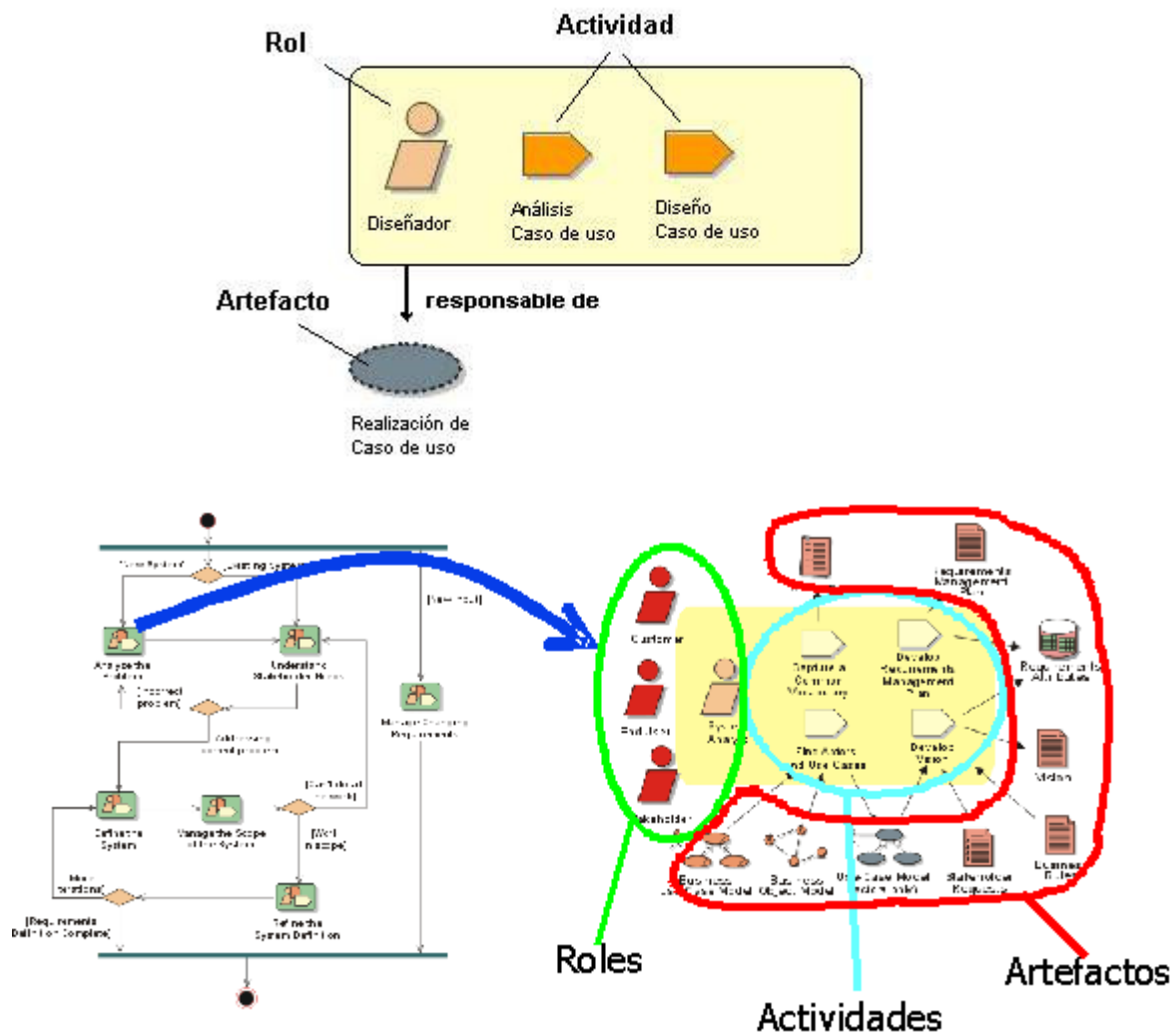


Ilustración 5.11: Estructura estática del proceso.

5.3.2.4.1 Roles

Un rol define el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas.

Las responsabilidades de un rol son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos.

RUP define grupos de roles, agrupados por participación en actividades relacionadas.

Podemos diferenciar los siguientes grupos:

- Grupo Analista
- Grupo Desarrollador

- Grupo Especialista en pruebas
- Grupo Gestor
- Grupo Otros roles

5.3.2.4.2 Actividades

Una actividad en concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.

5.3.2.4.3 Artefactos

Un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final.

Un artefacto puede ser cualquiera de los siguientes:

- Un documento, como el documento de la arquitectura del software.
- Un modelo, como el modelo de Casos de Uso o el modelo de diseño.
- Un elemento del modelo, un elemento que pertenece a un modelo como una clase, un Caso de Uso o un subsistema.

5.3.2.4.4 Flujos de trabajo

Con la enumeración de roles, actividades y artefactos no se define un proceso, necesitamos contar con una secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables.

A continuación se nombrarán los diferentes tipos de flujos de trabajo posibles. Para ilustrar esto, observemos la ilustración 5.12, en la que se detalle el ciclo en el que interviene cada uno de los flujos citados:

1. Modelado de negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Test
6. Distribución
7. Gestión de configuración y cambios
8. Gestión del proyecto
9. Gestión del entorno

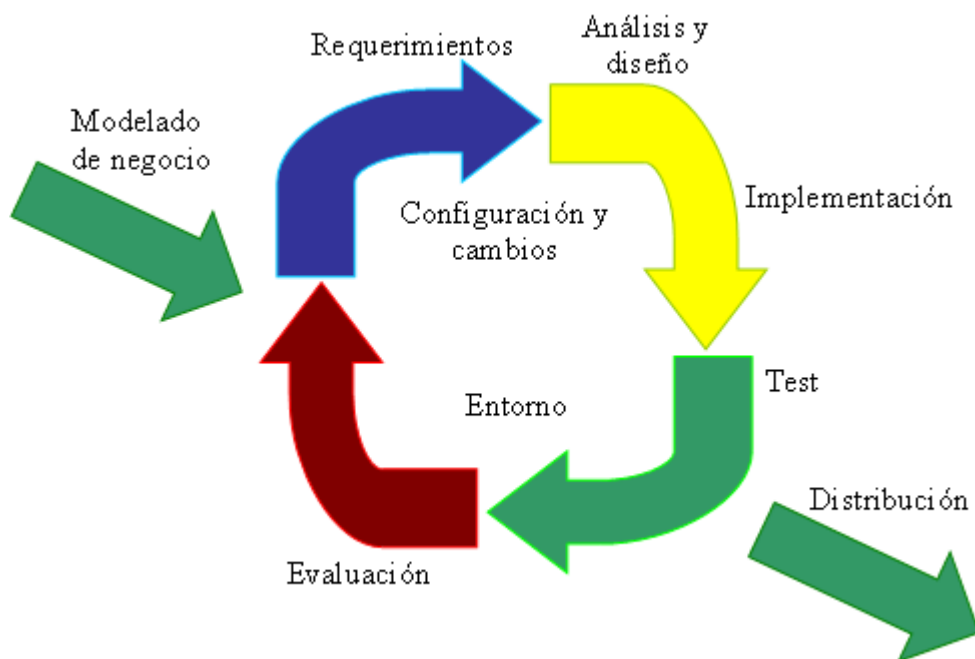


Ilustración 5.12: Flujos de trabajo en RUP.

5.4 eXtreme Programming (XP)

La programación extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP en [11] sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

Las Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas

o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, más quizás una estimación de esfuerzo en días. Beck en su libro presenta un ejemplo de ficha (customer story and task card) en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado cosas por terminar y comentarios.

Uno de las interrogantes (que también se presenta cuando se utilizan casos de uso) es ¿cuál es el nivel de granularidad adecuado para una historia de usuario? La respuesta no es tajante. Jeffries en [10] dice que depende de la complejidad del sistema, debe haber al menos una historia por cada característica importante, y propone realizar una o dos historias por programador por mes. Si se tienen menos, probablemente sea conveniente dividir las historias, si se tienen más lo mejor es disminuir el detalle y agruparlas. Para efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración).

No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Roles XP

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado describiremos los roles de acuerdo con la propuesta original de Beck.

- **Programador**

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

- **Cliente**
El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.
- **Encargado de pruebas (Tester)**
El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker)**
El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.
- **Entrenador (Coach)**
Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor**
Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- **Gestor (Big boss)**
Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

5.4.1 Proceso XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (*Release*), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

5.4.1.1 Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

5.4.1.2 Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de

una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

5.4.1.3 Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Wake en [9] proporciona algunas guías útiles para realizar la planificación de la entrega y de cada iteración.

5.4.1.4 Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

5.4.1.5 Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para

realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

5.4.1.6 Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

5.4.2 Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que describiremos a continuación.

- **El juego de la planificación**

Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos tipos de jugadores: Cliente y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario reconducir el proyecto.

- **Entregas pequeñas**

La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero si que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

- **Metáfora**

En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. Martin Fowler en [12] explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.

- **Diseño simple**

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

- **Pruebas**

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

- **Refactorización (*Refactoring*)**

La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo. Robert Martin [13] señala que el diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto. De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

- **Programación en parejas**

Toda la producción de código debe realizarse con trabajo en parejas de programadores. Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios de la programación en parejas [14], las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo. Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas. En los estudios realizados por Cockburn y Williams este lapso de tiempo varía de 3 a 4 meses.

- **Propiedad colectiva del código**

Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

- **Integración continua**

Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido. Martin Fowler afirma que el desarrollo de un proceso disciplinado y automatizado es esencial para un proyecto controlado, el equipo de desarrollo está más preparado para modificar el código cuando sea necesario, debido a la confianza en la identificación y corrección de los errores de integración.

- **40 horas por semana**

Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

- **Cliente in-situ**

El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada. Jeffries indica que se debe pagar un precio por perder la oportunidad de un cliente con alta disponibilidad. Algunas recomendaciones propuestas para dicha situación son las siguientes: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

- **Estándares de programación**

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

- **Comentarios respecto de las prácticas**

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. Esto se ilustra en la ilustración 5.13, donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí.

La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

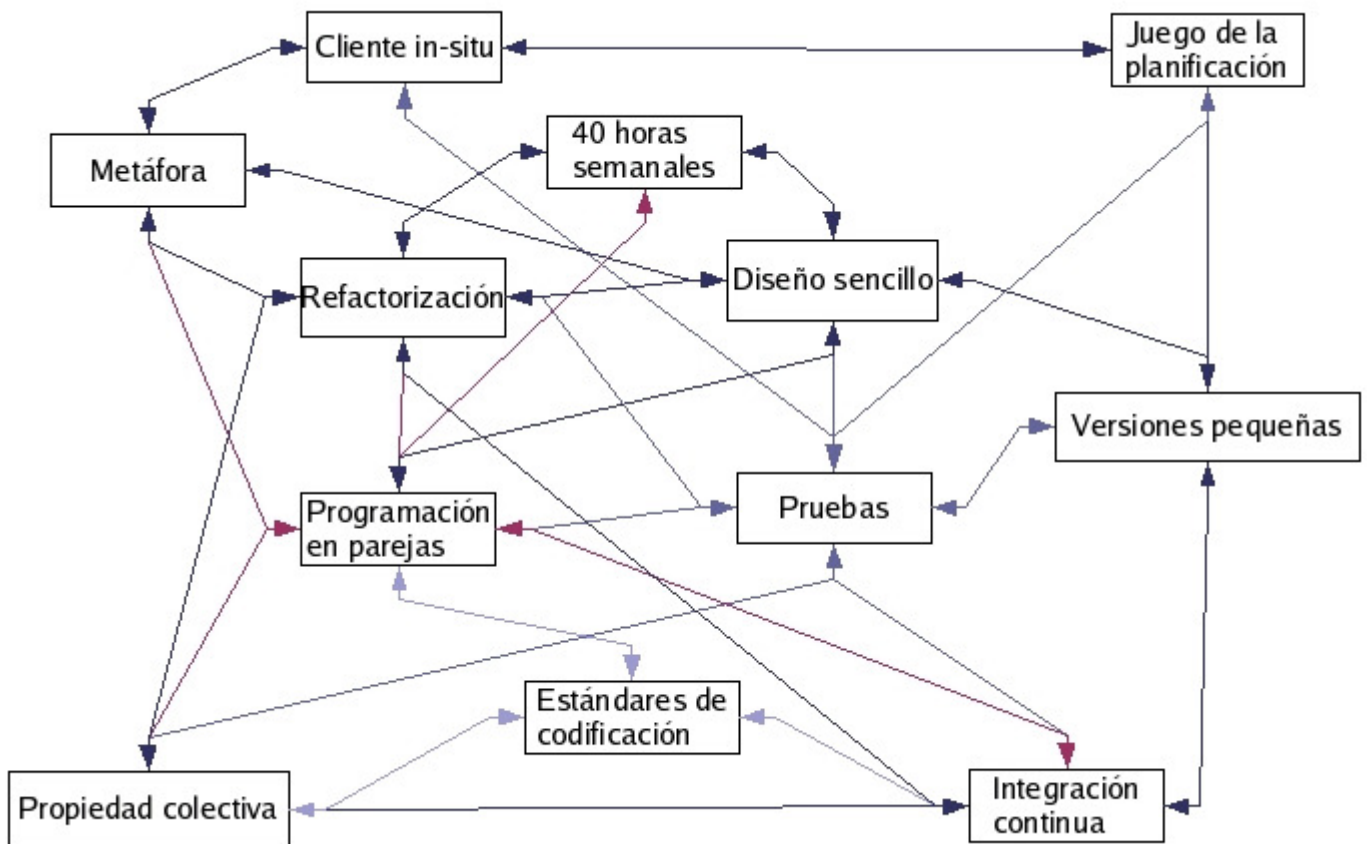


Ilustración 5.13: Flujos de trabajo de RUP

5.5 Conclusiones

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Esto ha llevado hacia un interés creciente en las metodologías ágiles. Sin embargo, hay que tener presente una serie de inconvenientes y restricciones para su aplicación, tales como: están dirigidas a equipos pequeños o medianos (Beck sugiere que el tamaño de los equipos se limite de 3 a 20 como máximo, otros dicen no más de 10 participantes), el entorno físico debe ser un ambiente que permita la comunicación y colaboración entre todos los miembros del equipo durante todo el tiempo, cualquier resistencia del cliente o del equipo de desarrollo hacia las prácticas y principios puede llevar al proceso al fracaso (el clima de trabajo, la colaboración y la relación contractual son claves), el uso de tecnologías que no tengan un ciclo rápido de realimentación o que no soporten fácilmente el cambio, etc.

Se han comentado a lo largo de las anteriores páginas, distintas posibilidades prácticas de desarrollo de software, cada una con sus diferencias, puntos de vista y sugerencias. Todas ellas dejan, sin embargo, la puerta abierta a posibles mejoras que el programador crea pueden beneficiar la mejora del software concreto de la empresa en cuestión. Se trata al fin y al cabo, de intentar imponer ciertas normas y reglas de desarrollo que nos permitan alcanzar la producción de un software de calidad, sin eliminar la libertad de investigación y desarrollo que todo diseñador debe disponer.

6 Borland Delphi

Borland Delphi 7 puede resumirse rápidamente como una herramienta de séptima generación, basada en la metodología orientada a objetos (OMT), el desarrollo rápido de aplicaciones (RAD) y la programación visual. Pero cuando se habla de Delphi 7, no solo hay que referirse al lenguaje, que en realidad se llama Object Pascal 7, sino también al conjunto de herramientas que componen el paquete de desarrollo.

El paquete Profesional consta de los siguientes componentes que se han clasificado independientemente dentro del mundo CASE:

- **Herramienta para el Desarrollo y Diseño de Interfaces**
 - Integrated Development Environment (IDE)
- **Herramientas de Soporte**
 - Herramientas de Documentación
 - Herramientas de Bases de Datos
 - Herramienta de Diseño Gráfico
- **Herramienta de Análisis Dinámico, Intrusiva**
 - Winsight
 - Integrated Debugger
- **Documentación para los Desarrolladores**

A continuación pasamos a comentar brevemente algunas de las funciones más características de este entorno, anteriormente comentadas.

6.1 Herramienta para el Desarrollo y Diseño de Interfaces

6.1.1 Integrated Development Environment (IDE)

El IDE de Delphi es el centro de desarrollo de los proyectos que se van a desarrollar con él. Actúa como enlace y gestor de la comunicación de todos los elementos que componen el paquete.

En su pantalla principal podemos acceder a todas las opciones y herramientas, actuando como un gestor de proyectos cuando se trabaja en grupo desde red. Podemos observar un ejemplo de dicha pantalla, en la Ilustración 6.1 a continuación mostrada.

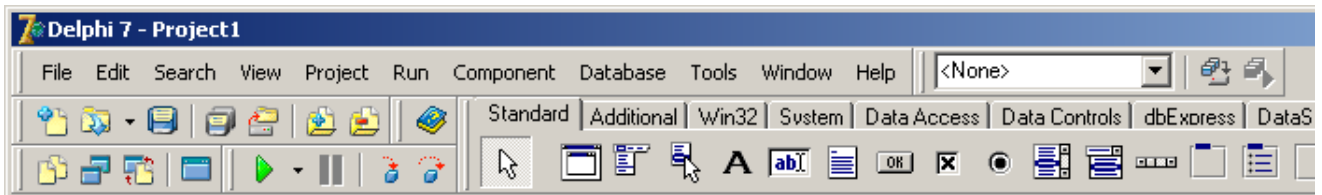


Ilustración 6.1: Sección de la pantalla principal.

Desde el punto de vista de las herramientas CASE Orientadas a Objetos, Borland Delphi se basa en un entorno para la Programación Visual, con generación automática de código, soporte de trabajo en grupo, repositorio de objetos y de expertos y centralización de código para acceso remoto.

No se puede decir que Delphi 7 sea una herramienta CASE, ni mucho menos decir que es una herramienta CASE Orientada a Objetos, pero mantiene un conjunto de características comunes con ésta última, ya que, como entorno de desarrollo, posee un conjunto de herramientas que se centran en las fases de Desarrollo, Codificación y Documentación de un proyecto software.

Las características comunes con las herramientas CASE son las que se detallan a continuación.

6.1.1.1 Generación Automática de Código

La generación automática no se basa en la definición de diagramas de metodología OMT ni de otras similares, sino en la interacción de los desarrolladores con los objetos disponibles en las paletas de componentes visuales (VCL).

En la Ilustración 6.2, en la sección inferior se pueden apreciar una serie de iconos organizados en *paletas*, los cuales serán seleccionados y *arrastrados* (mediante Drag&Drop) sobre el formulario o ficha que se desee. Inmediatamente, el código necesario para incluir ese componente en el proyecto será generado automáticamente, incluyendo desde las declaraciones de librerías necesarias hasta la definición de variables con nombres automáticos para poder referenciarlo.

Otra herramienta es Code Insight, que se encarga de la terminación del código que se está tecleando, si así se elige. De esta forma, a medida que se está escribiendo el nombre de una función, además de aparecer un mensaje (*hint*) donde se indica el nombre más parecido a lo que se lleva escrito, nos indica, una vez determinado si es una función (por ejemplo), el tipo de datos que devuelve, el tipo de los parámetros necesarios para su llamada... agilizando y automatizando en parte la creación del código fuente. En este punto cabe destacar un hecho, y es el siguiente: uno de los focos principales de pérdida

de tiempo a la hora de la implementación de código (dependiendo del conocimiento del lenguaje) es la sintaxis de llamada de funciones, de sentencias, de declaración de clases..., y si se utiliza Code Insight el tiempo perdido se reducirá dramáticamente al no tener que consultar ficheros de ayuda o documentación de cualquier otro tipo. Esto, junto a las características propias del proyecto, pueden ser especificadas individualmente para cada uno. E incluso es posible personalizar la configuración del IDE mediante la pantalla mostrada a continuación:

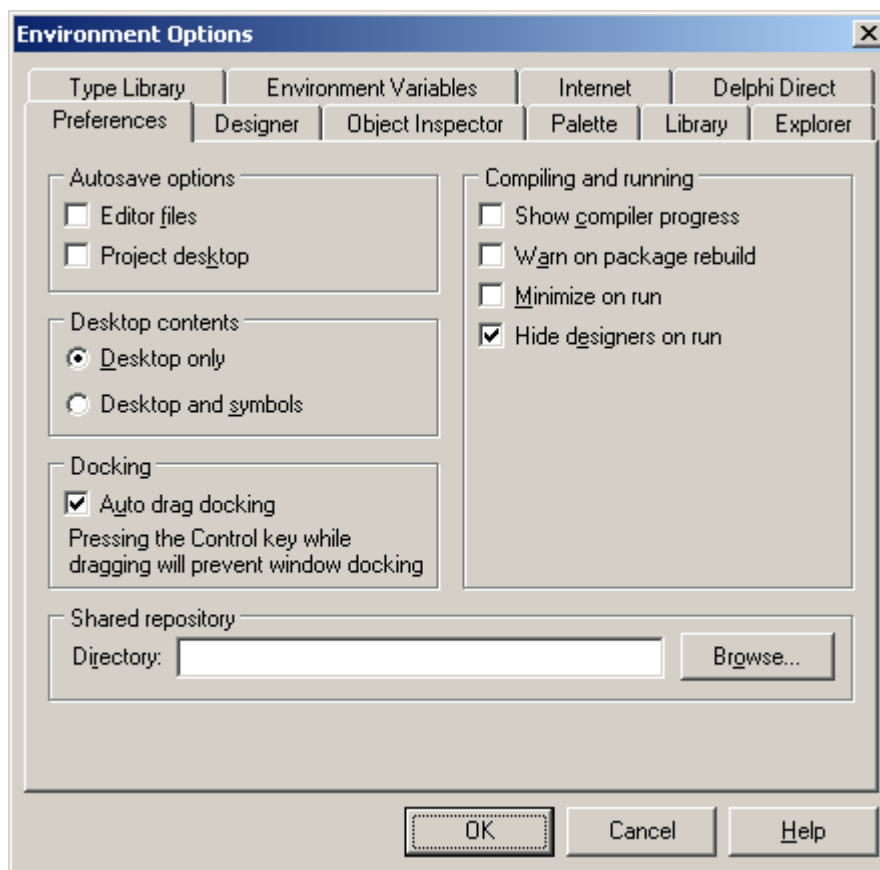


Ilustración 6.2: Configuración del IDE.

6.1.1.2 Desarrollo Visual

Una vez seleccionados los componentes, que en realidad son objetos con un formato especial, interviene otra parte de la programación visual del IDE. Se trata del Inspector de Objetos, mediante el cual se podrán cambiar las características de los objetos visuales (componentes) en tiempo de edición, simplificando sobremanera la codificación manual. Cuando los componentes son configurados, es decir, las propiedades han recibido los valores pertinentes y los eventos y procedimientos han sido implementados, en el momento de la compilación, se *linka* en el ejecutable el código necesario para su inicialización y creación automática, así como el código necesario para asignarle los

valores establecidos mediante el Inspector de Objetos en tiempo de edición, sin que sea necesaria la codificación manual en ningún momento.

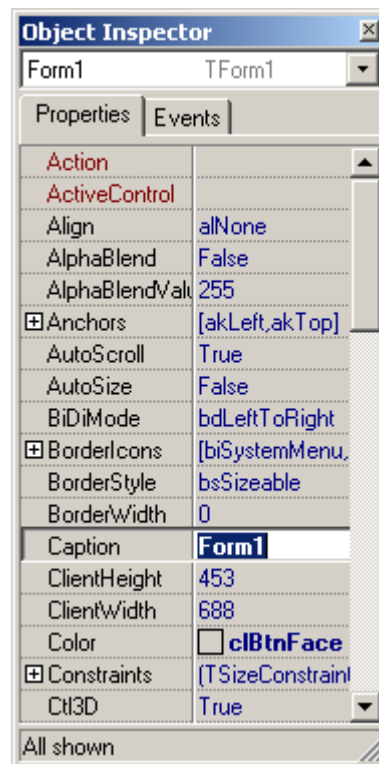


Ilustración 6.3: Inspector de Objetos.

6.1.1.3 Desarrollo Centralizado y Distribuido

Una de las opciones que las herramientas CASE deben poseer es la posibilidad de trabajo en grupo.

Delphi 7 dispone de la posibilidad de trabajar varios puestos en red local sobre el mismo proyecto, encargándose automáticamente de las actualizaciones de versiones del código y del protocolo de red, siendo prácticamente transparente para los desarrolladores.

Para el desarrollo centralizado se dispone de los denominados *paquetes* de componentes, que son librerías compiladas de objetos (VCL, OCX, ActiveX...) que pueden ser accedidas remotamente de forma que cada uno de los IDE remotos que trabajen sobre el mismo proyecto, las utilicen sin necesidad de tener que utilizar copias locales.

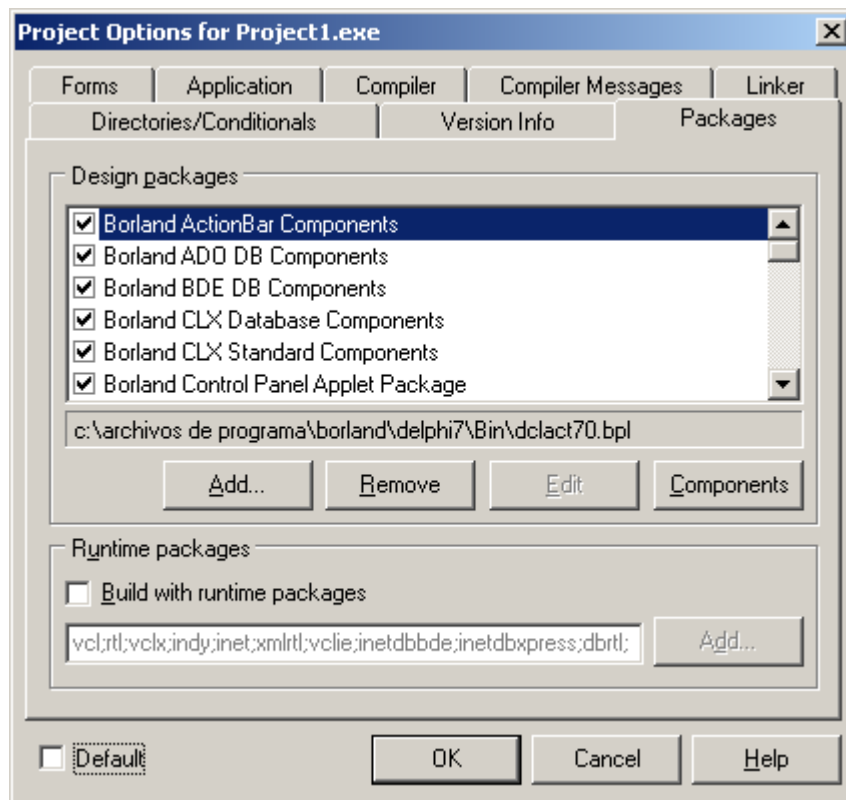


Ilustración 6.4: Configuración del IDE.

De esta forma, la utilización de códigos compilados generados anteriormente, queda excluida de la problemática de actualización continua que sufre un proyecto cuando se está trabajando sobre él desde diversos puestos, empaquetando el trabajo anterior y siguiendo adelante en el desarrollo.

6.1.1.4 Reutilización

Pero esta utilización de componentes u objetos no termina aquí, sino que es posible crear aplicaciones enteras que podrán ser reutilizadas por todos los puestos como si se tratasen de simples objetos, independientemente de su complejidad.

Esto significa un paso más allá de la reutilización de código, ya que se trata en realidad de la reutilización de todo lo que se desarrolle con Delphi.

Para ello se encuentra la *Librería de Expertos*, donde se podrán almacenar desde formularios, asistentes para generar informes, unidades precompiladas en formato DLL o ActiveX, entre otras muchas posibilidades, tal y como se aprecia en la ilustración 6.5.

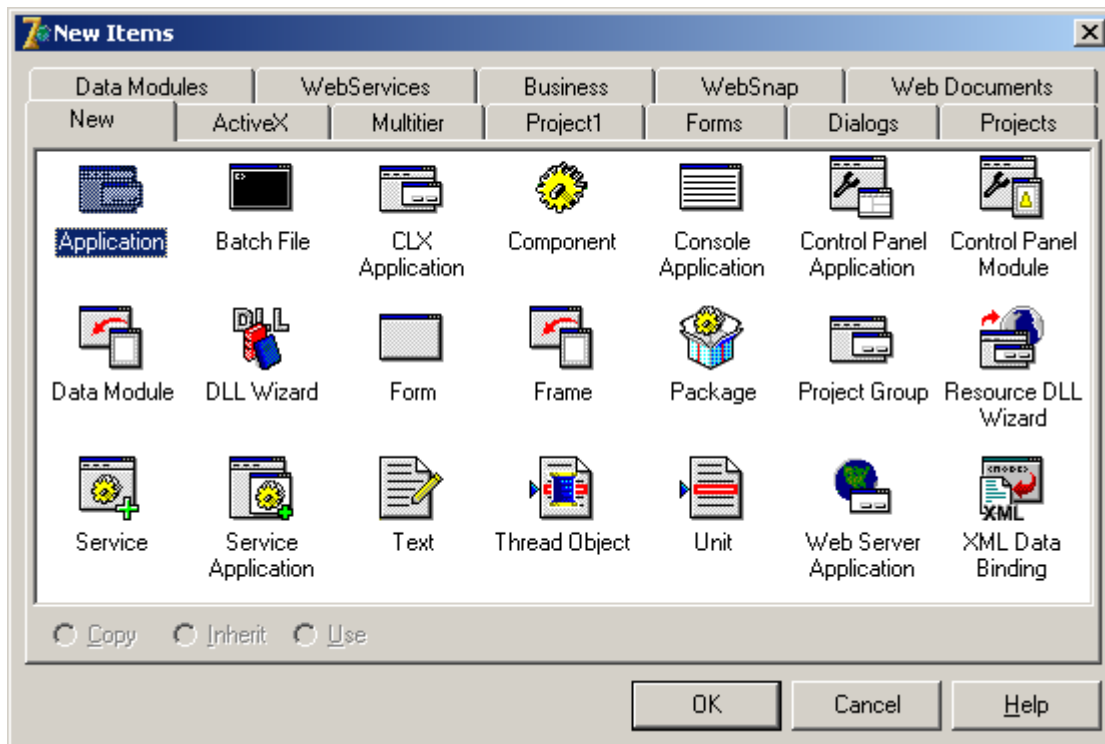


Ilustración 6.5: Librería de Expertos.

Otro paso más allá de los estándares es el uso de Paquetes de Distribución, que son núcleos de código compilados que serán comunes a todos los ejecutables independientemente de su funcionalidad.

Estos paquetes tienen la finalidad de reducir drásticamente el tamaño de los ejecutables, eliminando las partes comunes a todos ellos y centralizando parte de su funcionamiento en librerías externas. Así, varios ejecutables desarrollados con Delphi 7, tan solo ejecutarán unas pocas decenas de KB, compartiendo los paquetes precompilados.

6.1.1.5 Explorador de Código

El Explorador de Código es una de las utilidades del entorno que más va evolucionando en cada versión del lenguaje. Versión tras versión, se han ido incorporando pequeños detalles que hagan más sencilla la navegación por el código fuente, permitiendo acceder rápidamente al código de funciones y procedimientos, variables y constantes, así como un listado de los *uses* incluidos en el proyecto.

Este explorador está estructurado jerárquicamente, siendo el nodo inicial del árbol el formulario (objeto padre) que contiene al resto de componentes y código. La información se presenta siguiendo los patrones de la orientación a objetos, es decir, cada nodo será un objeto cuyos nodos hijos serán o bien otros objetos, o declaraciones de funciones (métodos) y variables (propiedades).

6.1.1.6 Control de Versiones

Otra característica fundamental de las herramientas CASE orientadas a objetos es el control de versiones de los proyectos.

El control que brinda Delphi 7 se encarga de la generación interna (tanto para los desarrolladores como incluida en el ejecutable), de las versiones incrementales en el tiempo que se van creando durante el desarrollo.

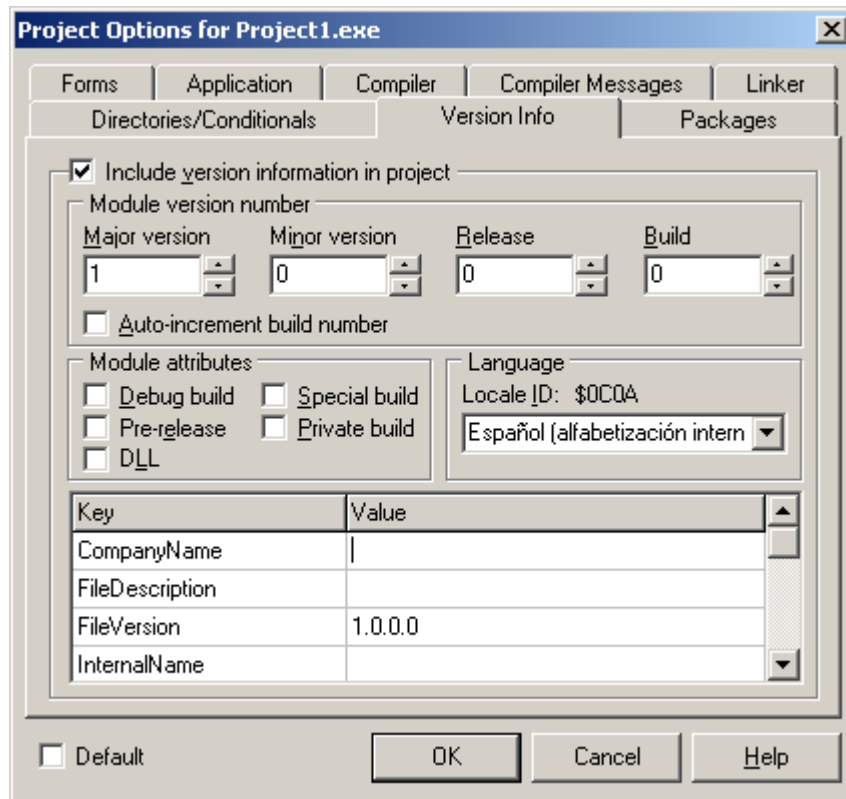


Ilustración 6.6: Control de versiones.

Este control permite en todo momento identificar una copia de seguridad de un determinado proyecto, ya que, como si de una base de datos se tratase, almacena toda la información necesaria para ello.

6.2 Herramientas de Soporte

Junto al IDE, que se encarga de centralizar todo el desarrollo y de ofrecer acceso a otras opciones, existen herramientas también accesibles desde él. Estas herramientas se engloban en las siguientes categorías CASE:

6.2.1 Herramientas de Documentación

Estos dos programas sirven para la creación de la documentación del programa. Principalmente se suelen usar para la distribución final de los ficheros de ayuda que

acompañarán al producto software final que llegará a manos de los usuarios, pero también pueden ser utilizados para la documentación interna del proyecto.

La mejor de ellas es OpenHelp, que es un nuevo formato de ayuda basado en HTML dinámico, permitiendo una mejor estructuración de la información y por consiguiente, de búsqueda y localización de un tema determinado. Por otro lado, WinHelp es el clásico compilador de ayuda de Windows 9x.

6.2.2 Herramientas de Bases de Datos

Delphi es uno de los más potentes entornos de desarrollo de bases de datos, tanto para Desktop y redes locales, como Cliente-Servidor, implementando los protocolos más usados mundialmente.

Las herramientas que se incluyen son aplicaciones que sirven de soporte al desarrollo de las bases de datos de un programa de gestión.

Mediante DataBase Desktop se podrán crear tablas de tipo: Paradox, dBASE, Informix, Oracle, Access, SyBase, Mysql... vía ODBC, o vía los drivers nativos incluidos en Delphi. Esta herramienta se comentará en profundidad en el punto 4.2 de esta memoria.

El BDE Administrator permite la navegación por las bases de datos de los proyectos que se encuentren en una determinada máquina (local o remota), permitiendo una rápida navegación entre las distintas tablas de cada uno. Podemos ver una captura de pantalla de dicha utilidad en la Ilustración 6-7.

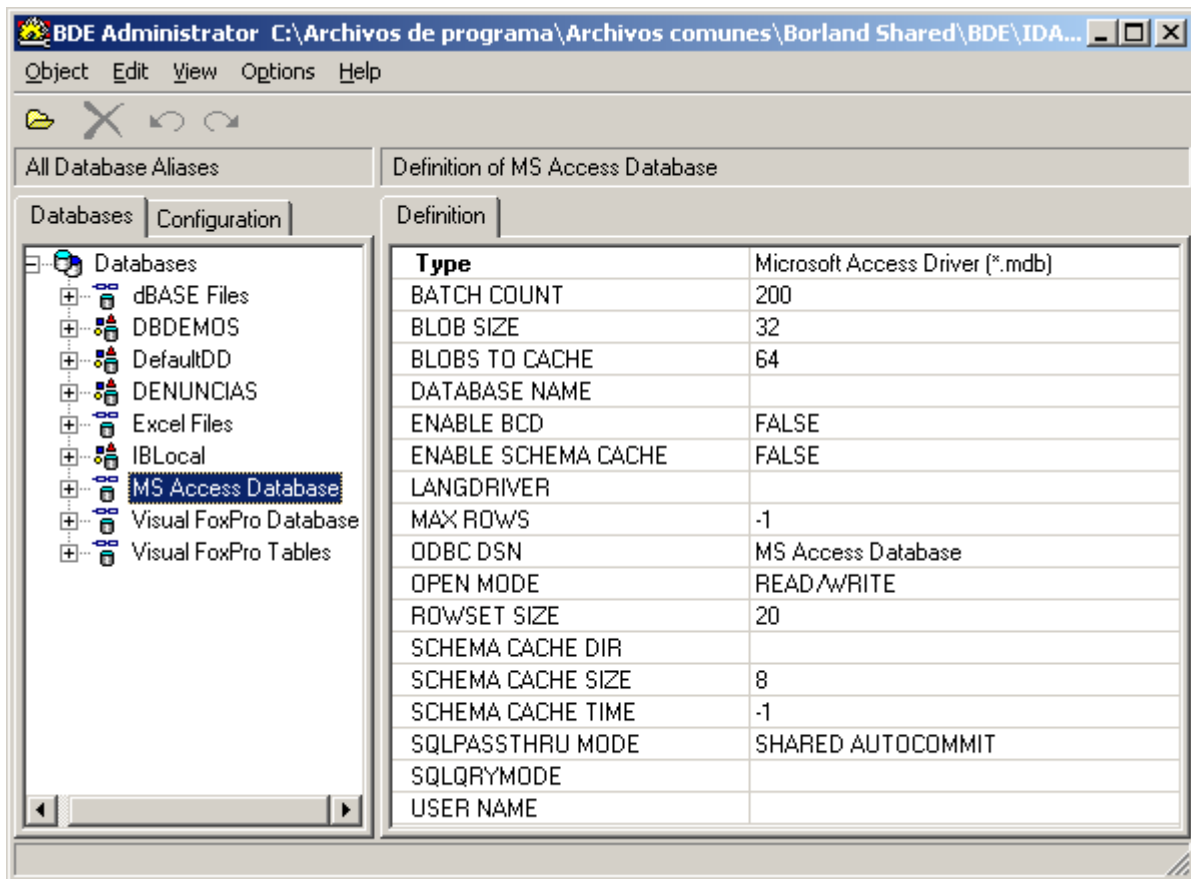


Ilustración 6.7: BDE Administrator

El BDE Administrator puede configurar también los drivers de los distintos tipos de bases de datos que se encuentren instalados en el sistema.

Todas estas herramientas ayudan considerablemente al desarrollo de aplicaciones de bases de datos, evitando la necesidad de utilizar aplicaciones externas para ello y centralizando además de garantizando la compatibilidad del trabajo.

6.2.3 Herramienta de Diseño Gráfico

Delphi incluye, desde la primera de sus versiones, ImageEdit, una pequeña aplicación para la creación de iconos, cursores y pequeños bitmaps que cumple su función específica a la perfección y si bien ha evolucionado poco a lo largo de las distintas versiones del entorno RAD, sigue siendo de gran utilidad para realizar distintos ajustes a los *glyphs* empleados en nuestro proyecto.

6.3 Herramienta de Análisis Dinámico, Intrusiva

6.3.1 Winsight

Winsight es la solución que Delphi aporta para el análisis de programas ya concluidos o en desarrollo. Es capaz de monitorizar toda la actividad que genera un programa en

Windows, capturando y mostrando en tiempo real el tráfico de mensajes entre todos y cada uno de los objetos que conforman el programa.

Se pueden definir filtros para localizar el foco de atención en un determinado tipo de eventos o de mensajes entre objetos, constituyéndose en una herramienta indispensable cuando se utiliza Delphi para aplicaciones que tienen que "bajar" al más profundo nivel de la arquitectura de Windows.

6.3.2 Integrated Debugger

El Debugger que se encuentra integrado dentro del IDE, es una excelente herramienta de depuración y traceado de código, que permite monitorizar el programa durante su ejecución en modo IDE, es decir, con la supervisión del IDE durante su ejecución, a diferencia con Winsight, que se trata de una aplicación aparte que monitoriza el ejecutable final, independientemente del IDE.

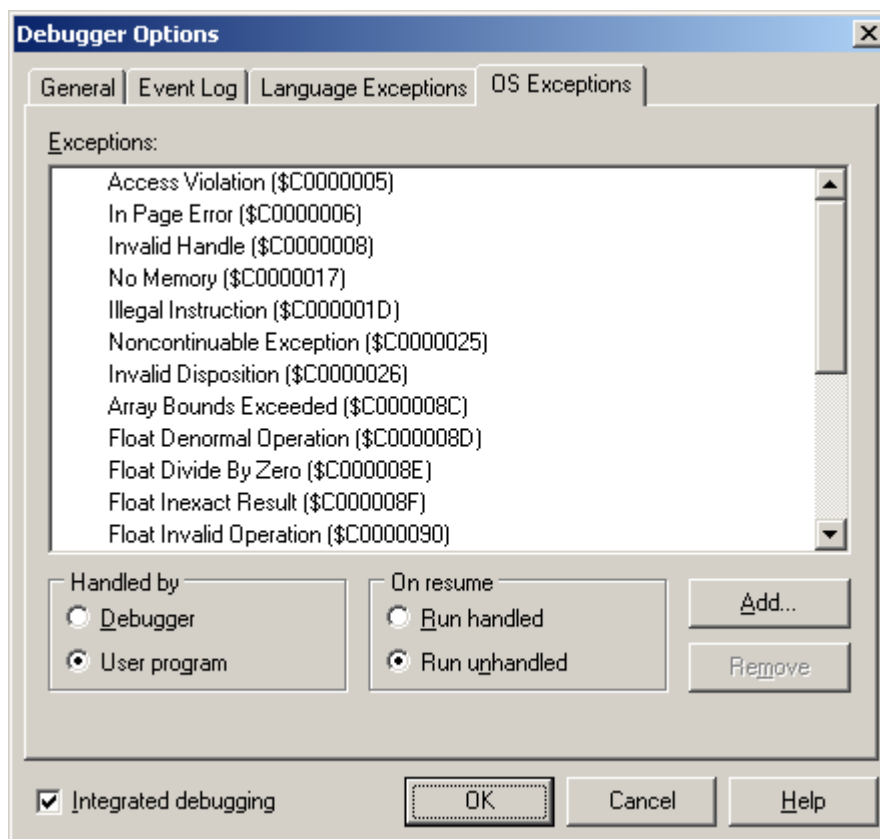


Ilustración 6.8: Integrated Debugger

Además, existe la posibilidad de visualizar todos los valores de las variables locales del punto en donde se encuentre la ejecución del programa en un determinado momento.

Esto, junto al Evaluador de Expresiones, que consiste en que al situar el cursor sobre una expresión dentro de la ventana de código y aparece automáticamente su valor en un *hint*, reduce de una forma importantísima el tiempo de depuración.

6.4 Documentación para los Desarrolladores



Ilustración 6.9: Documentación de desarrolladores.

Una de las cosas indispensables para el desarrollo de programas es la documentación para los programadores y analistas programadores, ya que se minimiza sobremanera el tiempo invertido en la localización de la sintaxis de determinadas funciones, o la búsqueda de cualquier otro tipo de información que pueda ayudar al desarrollo.

Delphi 7 incluye un amplio abanico de documentación en formato WinHelp y OpenHelp que a continuación se muestran:

En resumen, Delphi no puede ser catalogado como una herramienta CASE, debido a que no es su propósito el abarcar todo el ciclo de vida del software, pero posee características que sí pueden ser catalogadas bajo el punto de vista CASE, desde el momento en que se encarga de varias fases: Desarrollo, Codificación, Documentación...

Es ideal para el ciclo de vida de prototipos, para el de orientación a objetos y el de desarrollo visual, no en vano es catalogado como una herramienta RAD (Rapid Application Development), siendo considerado como una herramienta muy potente para realizar programas de Bases de Datos bajo plataformas WIntel.

En cuanto a herramienta de propósito general, tan solo es aventajado por Visual C++ en el momento en que hay que programar al más bajo nivel de máquina. Pero la dificultad de aprendizaje de este último no llega a compensar la poca diferencia de potencia en este campo.

Delphi lleva el desarrollo orientado a objetos al siguiente nivel, permitiendo al desarrollador separar la actividad de la interfaz de usuario de la lógica de la aplicación, reduciendo así las líneas de código, la complejidad del código, la reutilización y la robustez. Delphi 7 incluye muchas acciones estándares predefinidas, reduciendo así más el tiempo necesario para crear aplicaciones Windows ricas en funciones totalmente operativas.

6.5 InstallShield

“InstallShield Express Borland Edition”, diseñado específicamente para cubrir las necesidades básicas de instalación de las aplicaciones desarrolladas con Delphi 2005 tanto para Win32 como para .NET. Merece la pena resaltar, asimismo, que en la creación de instalaciones .NET nos permite distribuir .NET Framework con nuestras aplicaciones.

7 Manual De Usuario

Este capítulo pretende dar una visión global de como operar entre las funciones que los servicios que este programa ofrece. A lo largo de los siguientes apartados se intentara mostrar al usuario la filosofía de trabajo la cual rige el diseño del programa y sus posibilidades.

Comenzaremos revisando el método de instalación del software en una red de área local, para dar paso a la ejecución del mismo desde una máquina cliente.

7.1 Instalación.

7.2 Manual de uso

7.2.1 Identificación y menú principal.

Tras ejecutar la aplicación lo primero que el usuario vera será una ventana, dividida en dos secciones, un panel izquierdo, sobre el cual se despliegue en forma de menús los distintos servicios agrupadas por secciones, y un panel derecho mucho más grande que ocupara el 80% de la pantalla, que es donde el usuario podrá interactuar con los servicios a los que se tenga acceso.

Como el usuario todavía no ha ingresado sus credenciales, el menú de servicios solo mostrara, un menú básico, compuesto por las opciones de Asistencia Remota y Cerrar. Y en el panel derecho se visualiza un formulario de identificación, solicitando las credenciales del usuario (usuario y contraseña).

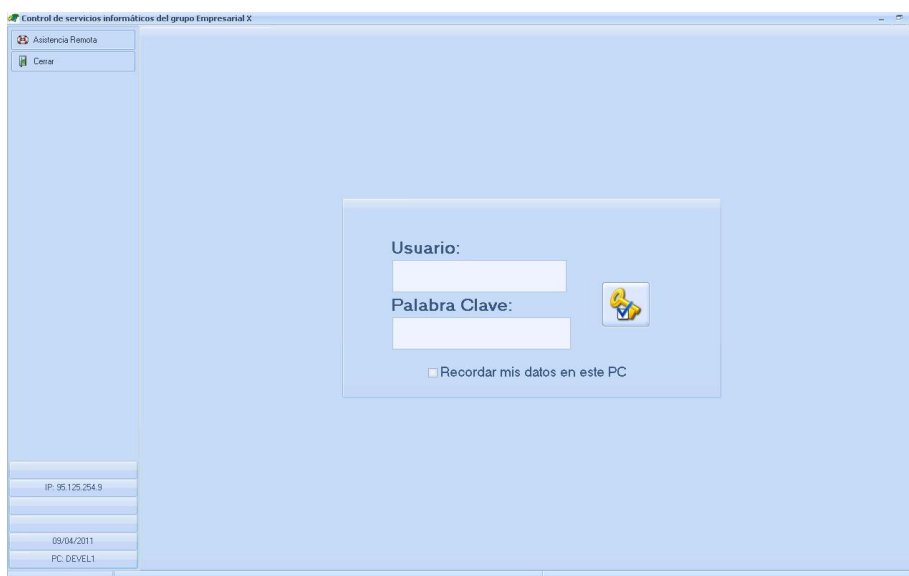


Ilustración 7.1: Pantalla de inicio y login.

Una vez el usuario se haya identificado, la aplicación verificara sus credenciales, y desplegara en un menú lateral izquierdo, los servicios a los que el usuario puede acceder. Así bien, podemos clasificar los siguiente servicios a los que un usuario puede acceder, según se le hayan asignado o no privilegios de acceso.

Todos los usuarios registrados pueden acceder, a la ficha personal, y al servicio de Incidencias.

Pero solo aquellos que tengan permisos asignados podrán acceder a la gestión de incidencias, gestión de cierres, gestión de proveedores, gestión de activos, gestión facturación telefónica y gestión personal.

Un usuario puede tener acceso a más de un servicio, no excluyentes entre si.

Es decir que un usuario puede interactuar en varios servicios si tiene permisos para ello. En caso de que las credenciales introducidas no sean correctas se mostrara un mensaje de error y se volverá a solicitar el usuario y la contraseña.

En caso de que sea un usuario que existe en el sistema pero es la primera vez que accede, la contraseña será en blanco, y se le solicitara que confirme dos veces la nueva contraseña, para así finalizar su registro.

Para este manual utilizaremos un usuario que tiene privilegios para interactuar en todos los servicios y secciones que ofrece la aplicación. En caso de que su usuario no tenga permisos para interactuar en uno o más servicios este u estos no se mostraran en el menú izquierdo (menú de servicios) y por lo tanto no se podrá desplegar en el panel central.

Una vez identificados correctamente, veremos que el menú de servicios se despliega una serie de secciones con los servicios, y en panel derecho una gran cantidad de información asociada al usuario.

Una primera sección con los activos asociados al usuario, teléfonos, tarjetas sims, monitores, portátiles, impresoras, y otros activos que puedan estar asignados al usuario. La segunda sección se trata de un listado de las incidencias que el usuario ha creado, y aun no han sido cerradas, en la pantalla que aparece en la siguiente figura se muestran todas las incidencias, ya que el usuario que ejecuta esta instancia de la aplicación tiene el rol de gestor de incidencias asignado.

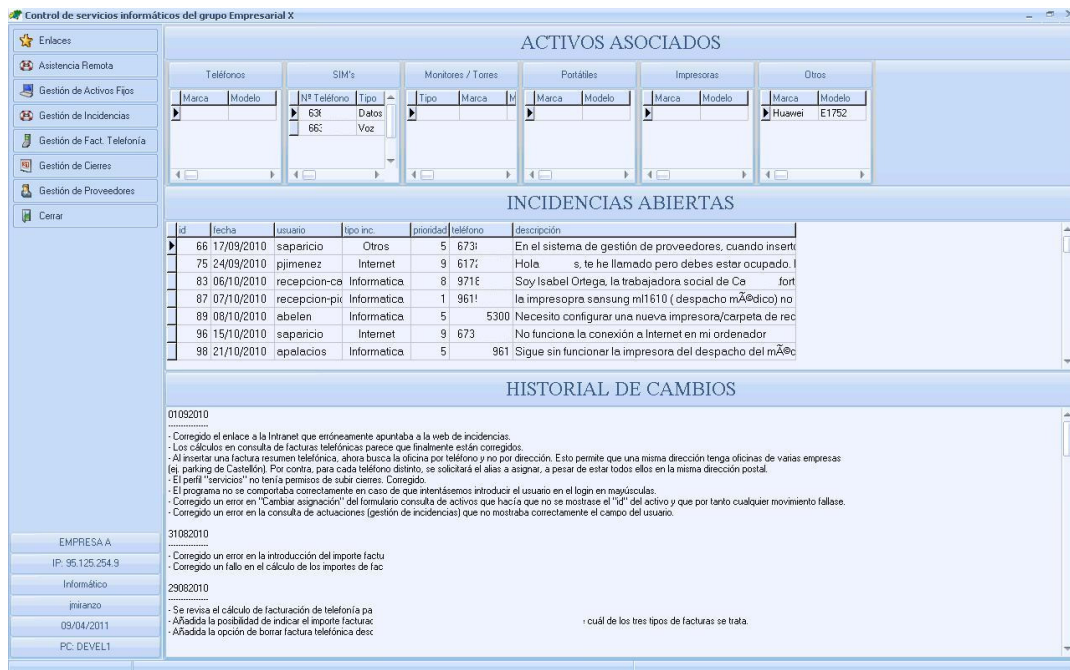


Ilustración 7.2: Pantalla principal.

7.2.2 Enlaces.

Como se puede ver el menú de enlaces nos ofrece los servicios de Colgar archivo, Correo electrónico, e Intranet.

Si clicamos sobre las opciones de Correo electrónico como de Intranet, se ejecutara una instancia de un navegador de internet abriendo las páginas web que hospedan el correo electrónico de la empresa o la intranet de la empresa respectivamente.



Ilustración 7.3: Panel de servicios.

7.2.2.1 Colgar archivos.

Este servicio está pensado para poder ofrecer a un tercero una forma de descargarse ficheros de forma sencilla, al estilo de otros servicios comerciales.

Mediante un formulario nos permitirá colgar en un servidor web un documento que no sobrepase el tamaño de 50Megas que previamente habrá tenido que ser comprimido.

Solo habrá que adjuntar dicho fichero al formulario y este nos devolverá en pantalla una URL con la cual podremos descargar el fichero mediante un navegador web desde donde queramos.

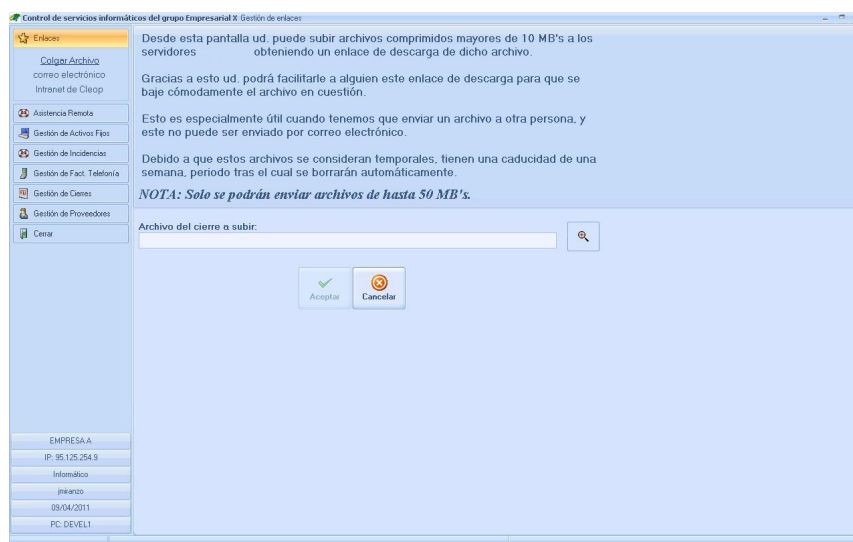


Ilustración 7.4: Subir fichero.

7.2.2.2 Asistencia remota.

Bajo la opción de asistencia remota aparecen los nombres de dos miembros del departamento de informática a los cuales se les puede solicitar mediante esta opción una asistencia remota, para que dicho informático pueda conectarse al ordenador y solucionarle algún problema o duda, ejecutándose una instancia del cliente de VNC.

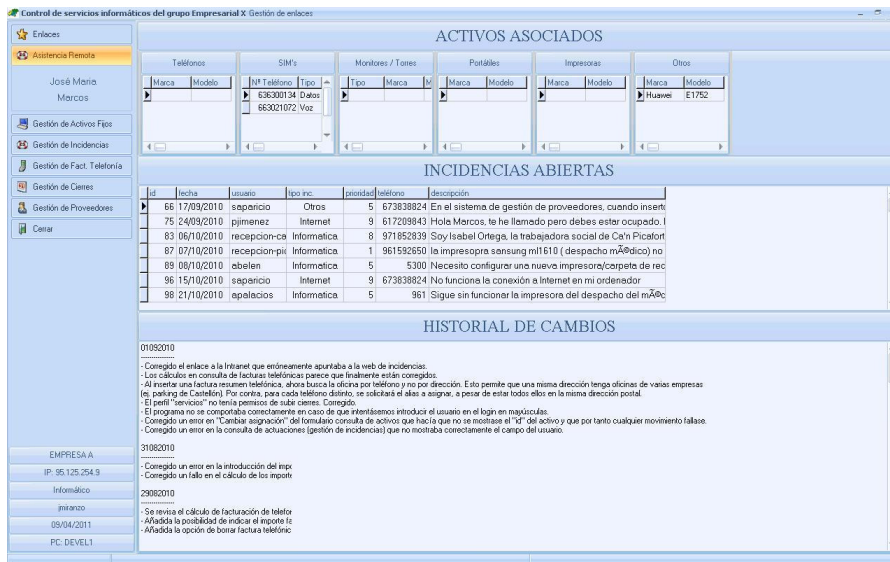


Ilustración 7.5

7.2.3 Gestión de activos Fijos.



Ilustración 7.6: Menú servicios de gestión de activos fijos

Esta sección está dedicada a la gestión de activos informáticos y de telecomunicaciones de la empresa, por parte de los usuarios que se les asigne dicho rol.

7.2.3.1 Añadir activo.

Para añadir un activo que se haya comprado y asignarlo a una persona si así procede tendremos que hacer uso del servicio de Añadir activo, donde rellenaremos un formulario con los datos de dicho activo y se asignara un código de inventario, en cuyo caso de no saber cual hay que asignar se auto rellenará pulsando sobre la tecla interrogante junto a dicho campo. También habrá que asignarle un responsable a dicho activo, pero si no se tiene que asignar a ninguno, porque es para almacén, esta la opción de reserva central. Y finalmente una vez rellenado los datos se pulsara el botón añadir.

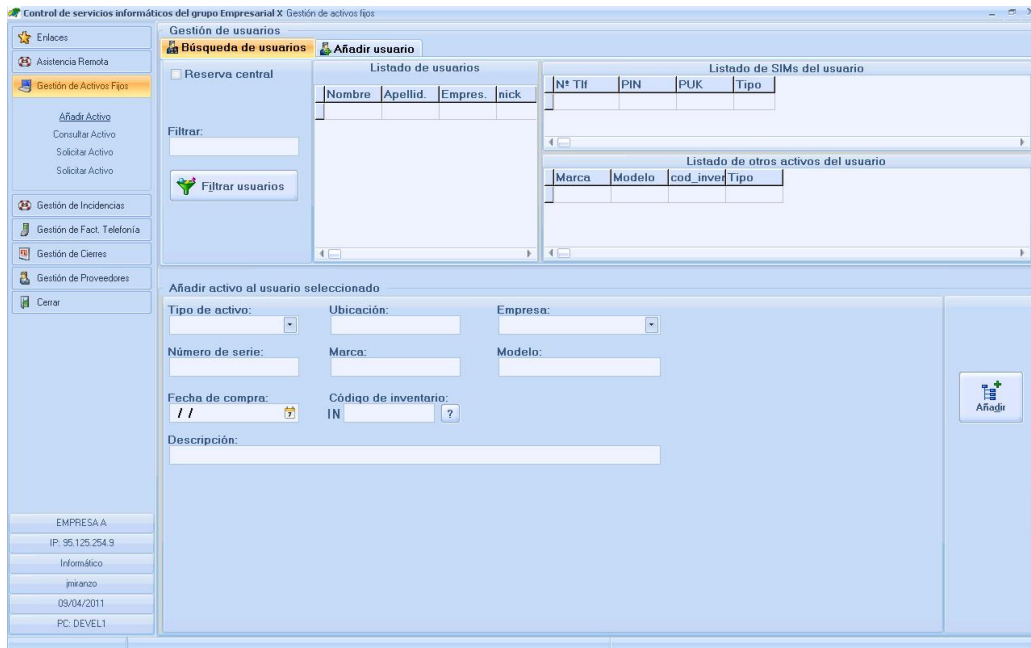


Ilustración 7.7: Añadir activo.

Cuando un activo de tipo SIM, es decir una tarjeta de telefonía/3g es seleccionado para dar de alta los campos a rellena son distintos, mostrándose en pantalla, los campos a rellenar: Numero de icc, pin, puk, y tipo de sim(datos o voz).

7.2.3.2 Consulta de activo y cambiar propietario de activo.

A parte de explicar cómo se realiza la consulta de activos se aprovechara este punto para explicar cómo manejar los resultados que se muestran en la mayoría de tablas que se emplean en el programa, ya que casi todas tienen las mismas funcionalidades.

A causa de la gran cantidad de activos que almacena la base de datos, se pensó un servicio para facilitar la consulta de activos y reasignación.

Cuando se accede a este servicio se mostrara en el panel derecho un listado de todos los activos.

id	Tipo	Responsable	Empresa	Cod. Inv.	marca	modelo	Descripción	id_usuario	num_serie	num_serie2	id_i
1	USB 3G	dllona	S	Y LE >	Huawei	E1752		6	045TAA1051302192		31
2	USB 3G	abas	C	O	Huawei	E1752		213	359853039252563		262
3	USB 3G	abauza	N	/A JA F	Huawei	E1752		137	359853038765896		33
4	USB 3G	Reserva Vale	G	RC LE >	Huawei	E1752		241	045TAA1051308227		34
5	USB 3G	Reserva Vale	G	RI)	Huawei	E1752		241	359853038764428		5
6	USB 3G	Reserva Vale	S	JF 2(I	Huawei	E1752		241	359853039247548		312
7	mifi 3G	mgonzalez	C	O	Novatel	Mifi 2352		1	358204020538076		12
8	USB 3G	jgonzalez	C	O	Huawei	E1752		69			11
10	USB 3G	rruiz	C	O	Huawei	E1752		40	045TAA1051307690		8
11	USB 3G	mcastellano	It	O E(Huawei	E1752		71	045TAA1051308473		9
12	USB 3G	oesquivel	C	O	Huawei	E1752		72	045TAA1051308436		1
13	USB 3G	vesteve	C	O	Huawei	E1752		73	045TAA1051308487		7
14	USB 3G	egrau	S	YV LE >	Huawei	E1752		91	045TAA1050804857		46
15	USB 3G	jmiranzo	C	O	Huawei	E1752		227	045TAA1050801873		45
16	USB 3G	vdominguez	N	V/	Huawei	E1752		76	045TAA1051308422		16
17	USB 3G	abelen	G	RC LE >	Huawei	E1752		117	045TAA1051308426		15
18	USB 3G	gmarban	S	YV LE >	Huawei	E1752		89	359853039247902		14
19	USB 3G	fpatino	C	O	Huawei	E1752		63	045TAA1051308428		47
20	USB 3G	japozo	C	O	Huawei	E1752		80	045TAA1051308471		5

Ilustración 7.8: Listar activos.

7.2.3.2.1 Manejo de los listados.

Este tipo de listado se utilizan en muchas partes del programa y en todos se opera de la misma forma.

Como se puede ver en la siguiente tabla en los títulos de cada una de las columnas si pinchamos sobre la flecha, se mostrara un desplegable con los distintos valores que puede tomar y así filtrar según el valor que seleccionemos.

id	Tipo	Responsable	Empresa	Cod. Inv.	marca	modelo	Descripción	id_usuario	num_serie	num_i
(Todos)		d.llona	S	R C E()	Huawei	E1752		6	() T Y	192
(Personalizado...)		abas	C	E()	Huawei	E1752		213	() 8 0	3
USB 3G		abauza	N	V E A >	Huawei	E1752		137	() 8 0	6
cámara fotos		Reserva Vale	C	R) E(Huawei	E1752		241	() T Y	227
d.duro ext.		Reserva Vale	C	R 1 K	Huawei	E1752		241	() 8 0	8
impresora		Reserva Vale	S	U / OI	Huawei	E1752		241	() 8 0	8
mifi 3G		mgonzalez	C	E()	Novatel	Mifi 2352		1	() 2 0	6
monitor		jgonzalez	C	E()	Huawei	E1752		69		
móvil		rruiz	C	E()	Huawei	E1752		40	() T Y	690
portátil		mcastellano	It	() C I O	Huawei	E1752		71	() T Y	473
router 3G		oesquivel	C	E()	Huawei	E1752		72	() T Y	436
torre		vesteve	C	E()	Huawei	E1752		73	() T Y	487
11 USB 3G		egrau	S	R C E()	Huawei	E1752		91	() T Y	857
12 USB 3G		jmiranzo	C	E()	Huawei	E1752		227	() T Y	873
13 USB 3G		vdominquez	N	V /	Huawei	E1752		76	() T Y	422

Ilustración 7.9: Búsquedas personalizadas.

Entre esos valores hay uno llamado “personalizado....” El cual nos permite personalizar la búsqueda a nuestro antojo rellenando un formulario.

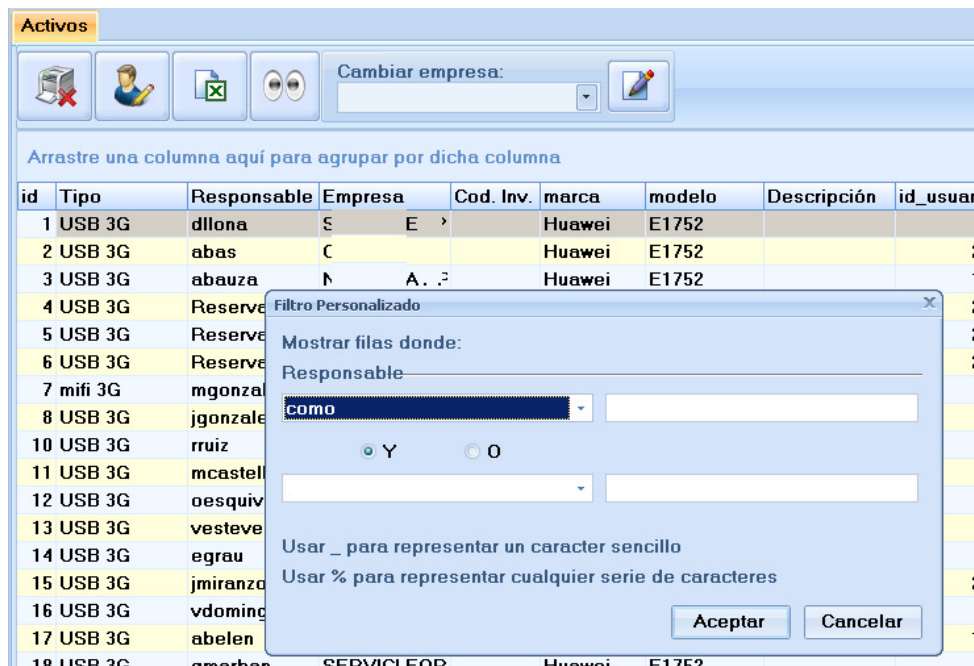


Ilustración 7.10: Búsquedas personalizadas 2.

En caso de solo querer que la lista se nos ordene de mayor a menor o del revés solo hay que pulsar sobre el título de la tabla.

7.2.3.2.2 *Cambiar propietario de activo.*

Como es habitual que ciertos activos se tengan que cambiar de responsable, se facilito mediante un botón la reasignación de un activo a otro usuario. Pulsando sobre el botón con el dibujo de una persona cuando tenemos un activo seleccionado, se mostrara un formulario que permitirá buscar el usuario al que se le reasignara dicho activo.

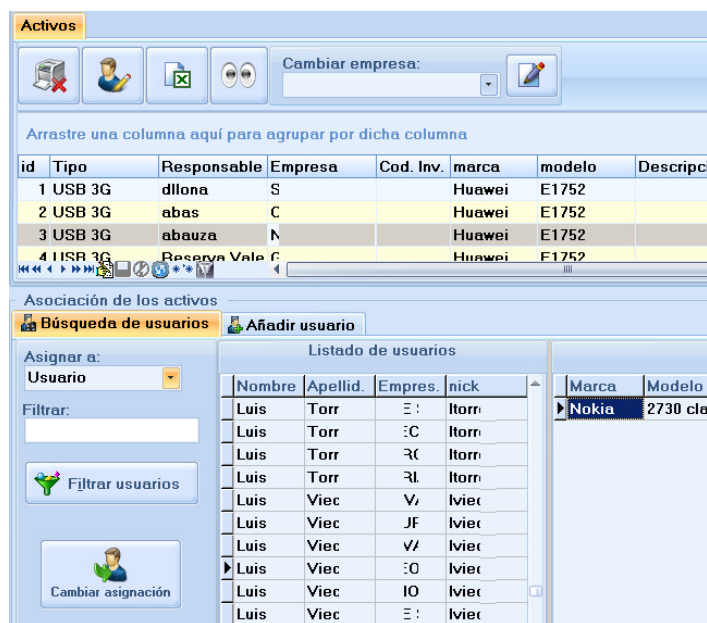


Ilustración 7.11: Cambio de propietario de activo.

7.2.3.3 Solicitud de activo.

Servicio que permite al usuario realizar la solicitud de un activo y que tenga una estimación del valor de estos.

Una vez el usuario seleccione los distintos activos que necesite adquirir y pulse el botón de solicitar, se enviara una notificación (en forma de correo electrónico) al departamento de compras para que evalúe la viabilidad de la compra.

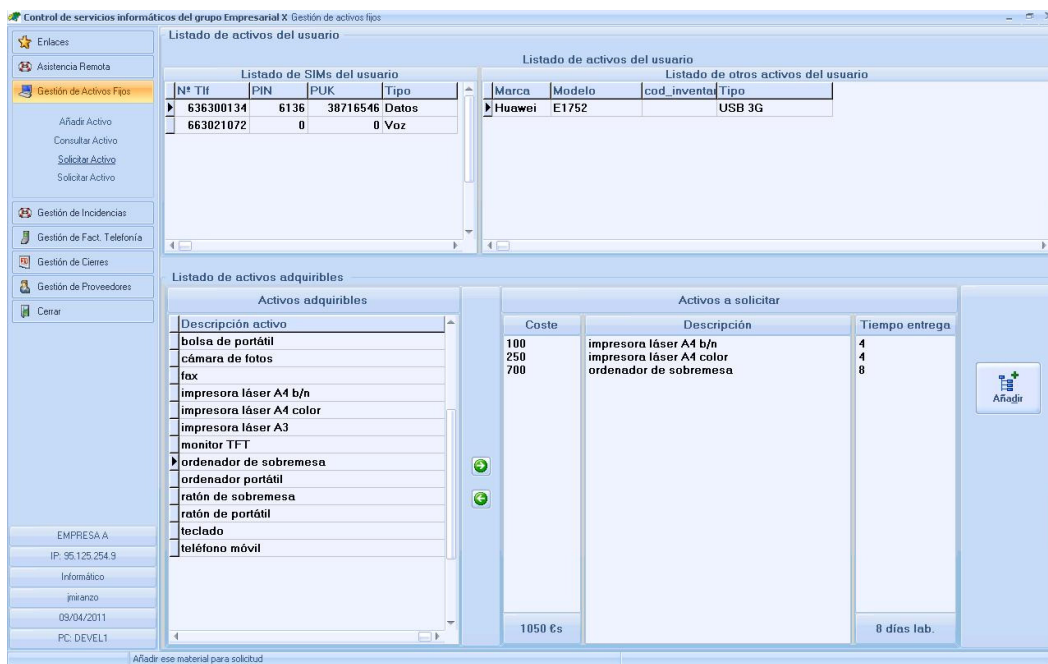


Ilustración 7.12: solicitud de activo.

7.2.4 Gestión de incidencias.

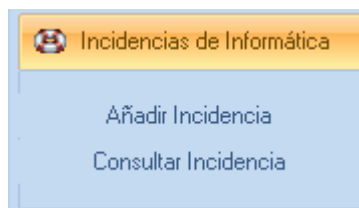


Ilustración 7.13: Menú de servicios de gestión de incidencias.

Sección dirigida para que el usuario pueda crear incidencias y además si el usuario posee el rol de gestión de incidencias, pueda también administrarlas, pudiendo crear actuaciones sobre cada una de las incidencias y cerrándolas.

7.2.4.1 Alta de incidencias.

Todo usuario podrá dar de alta incidencias, mediante un formulario donde se podrá indicar, que tipo de incidencias es (Hardware, software, redes, telefonía), empresa a la que afecta dicha incidencia, número de teléfono de contacto, urgencia que el usuario estima que tiene la incidencia, el código de activo al que afecta, ubicación para así saber a dónde hay que dirigirse, y una breve descripción de los problemas que se tienen. Una vez añadida, se enviara un correo electrónico al usuario indicando que la incidencia ha sido creada correctamente y que el departamento de informática se pondrá en contacto en cuanto pueda.

Introducción de incidencias

Tipo de incidencia: Ordenador

Empresa: [dropdown]

Número de teléfono: [input]

Urgencia: [dropdown]

Cód. Inventario: IN 0

Ubicación puesto de trabajo: [dropdown]

Aceptar

Necesito configurar una nueva impresora/carpeta de red en mi ordenador

Necesito instalar un programa en mi ordenador

No puedo imprimir desde mi ordenador

Mi ordenador no se enciende

Otros (introducir descripción de la incidencia) Restante: 255

Ilustración 7.14: Alta de incidencias.

7.2.4.2 .Gestión de incidencias.

Para la gestión de incidencias se muestra un panel donde el usuario con rol de gestión de incidencias podrá operar, añadiendo actuaciones sobre la incidencia seleccionada o cerrando dicha incidencia con una actuación de cierre.

Resultados								
Listado de incidencias:								
Arrastre una columna aquí para agrupar por dicha columna								
id	tipo	cerrado	Activo	fecha	usuario	priori	teléfono	descripción
359	Informatica	No	270	20/04/2011	arc	1	'50:	4 Por favor, cuando podais me habiltais el servicio de
357	Informatica	No		19/04/2011	rra	1	'57	6 Tengo problemas al acceder a la web de un curso.
327	Informatica	No		13/04/2011	auz	9		1 no se tiene acceso(puede que no este configurado)
325	Informatica	No		13/04/2011	epi	7	'69:	4 La torre del ordenador de administración, parece qu
322	Internet	No		12/04/2011	iaz	8	59:	0 Hola! No me funciona internet, he cmbiado varias v
318	Informatica	No		11/04/2011	ala	9		1 La impresora del médico lleva tiempo sin funcionar,

Descripción de la incidencia:								
cuando queremos entrar en el control center, nos da error. Tambien desde los demas ordenadores no se tiene acceso(puede que no este configurado)								

Listado de actuaciones:								
Arrastre una columna aquí para agrupar por dicha columna								
id	Fecha	Empleado	Descripción					
40	18/04/2011 17:05:12	jo a z	Se le transmite problema a a i dor					
81	19/04/2011 13:14:15	jo a z	Siguiendo instaraión de Ram oraraji uraA.Bøara se pone en contacto con Meet para i e solu					
84	19/04/2011 17:25:53	jo a z	Los de Meet do a (según cor do a A a za) o a no pueden conectarse. Le indico a e auza i					
87	19/04/2011 21:11:26	jo a z	Le facilito a A.lon za el teléfoi ton ér n dímaon eara que se ponga en contacto con el z.					

Ilustración 7.15: Gestión de actuaciones.

En el panel de gestión de incidencia cuando se selecciona una incidencia se mostrarán las actuaciones que dicha incidencia ha ido acumulando en una subsección.

En el momento que una incidencia es cerrada se envía un correo a los usuarios que tienen el rol de gestión de incidencias asignado para así saber que ya se ha cerrado, al igual que al usuario que la creó para así comunicar su resolución.

Solo podrán operar en este servicio los Informáticos.

7.2.5 Facturación telefónica.

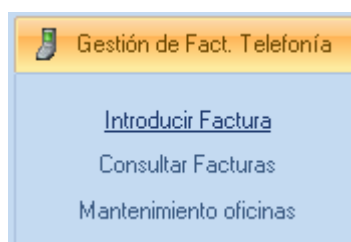


Ilustración 7.16: Menú servicios de gestión de telefonía.

Sección dirigida al administrativo que deberá tener el rol de gestión de facturación telefónica asignado para operar en el.

El conjunto de servicios que componen esta sección ofrece al administrativo una forma de controlar la gran cantidad de líneas contratadas (Saber qué consumo hace cada usuario,

o por empresas para facturación interna), verificar que los descuentos que aplica el proveedor de telefonía y datos es correcto, y en qué medida afectan dichos descuentos,

7.2.5.1 Introducir facturación.

El grupo empresarial X todos los meses recibe de su proveedor de telefonía y datos una gran cantidad de listados con los consumos realizados por todas las líneas contratadas. Estos listados se introducen en la base de datos, mediante el formulario que nos ofrece el servicio de introducir facturación.

En dicho formulario aparte de indicarle el xls hay que darle otros datos como, el mes de facturación, empresa, descripción, cantidad facturada. Una vez rellenado los datos y se haya pulsado el botón de aceptar, si todo es correcto guardara en la base de datos la información necesaria.

Datos de factura (consumo realizado)

Ruta de la factura a procesar:

Mes facturado: // Empresa: Descripción: Aceptar Tiempo transcurrido: Tiempo

Datos de factura real (consumo facturado)

Cantidad facturada: €'s Mes facturado: // Empresa: Tipo de factura: Aceptar

Estado:

General Errores

Ilustración 7.17: Insertar factura telefónica.

7.2.5.2 Consultar facturas.

Ofrece un listado con el desglose del consumos de cada una de las líneas y el usuario que tiene asignado la línea, generados a raíz de un pequeño formulario donde podremos indicar Fecha de facturación, empresa a la que se factura y tipos de líneas (de telefonía fija, o móvil).

Resumen de Factura							
Mes facturado:		Empresa:		Tipo de factura:			
01/02/2011		C C C		Telefonía Móvil			
Responsable /							
id	Tlf origen	Desc Servicio	importe cons	Porcentaje	importe fact	fecha	Nombre empresa
+	Responsable : d	on (SUM=25,05)					
+	Responsable : e	o (SUM=3,61)					
+	Responsable : e	(SUM=12,88)					
+	Responsable : fl	(SUM=41,12)					
+	Responsable : fr	(SUM=0,00)					
+	Responsable : fr	(SUM=10,96)					
+	Responsable : fr	(SUM=24,08)					
+	Responsable : ft	(SUM=16,06)					
+	Responsable : g	eta (SUM=7,68)					
+	Responsable : g	an (SUM=0,00)					
+	Responsable : it	a (SUM=40,44)					
+	Responsable : ja	(SUM=10,60)					
+	Responsable : ja	i (SUM=42,62)					
+	Responsable : jt	o (SUM=0,00)					
+	Responsable : jt	as (SUM=8,10)					
+	Responsable : jc	ria (SUM=44,58)					
+	Responsable : jc	o (SUM=13,62)					
+	Responsable : jc	eo (SUM=16,22)					
+	Responsable : jc	UM=12,67)					
+	Responsable : jc	ez (SUM=112,27)					
+	Responsable : jt	(SUM=32,05)					
+	Responsable : jn	(SUM=36,51)					

Ilustración 7.18: Consulta facturación.

El listado generado puede ser exportado a excel mediante el botón que se puede ver en el esquina superior derecha.

7.2.5.3 Mantenimiento oficinas.

A causa de que los ficheros suministrados por el proveedor que hacen referencia a telefonía fija vienen facturados a una dirección, y ya que cada empresa puede tener múltiples oficinas, hay que dar de alta esa dirección en el sistema para saber a qué empresa hay que facturarlos.

Para ello hay que rellenar un formulario que permite asignar un número de teléfono, con un nombre de oficina, ubicación, y empresa a la que pertenece.

Empresa:

Buscar teléfono:

Buscar oficina:

Listado del listín

Teléfono	Descripción	Nombre Ofi.	Domicilio Ofi.

Listado de oficinas

Nombre Oficina	Domicilio Oficina

Teléfono:

Descripción teléfono:

Oficina:

Añadir **Editar** **Borrar**

Nombre de la oficina:

Domicilio de la oficina:

Añadir **Editar** **Borrar**

Ilustración 7.19: Mantenimiento de oficinas.

7.2.6 Gestión de Obras

Este servicio permite al usuario listar, borrar y dar de alta obras en el sistema. Este servicio es accesible desde las secciones de Gestión de Proveedores y Gestión de cierres, por lo que el usuario tendrá que tener asignado al menos uno de los dos roles mencionados. Pudiendo ser directores, jefes de obra, jefes de grupo, jefe de administración.

Parámetros de búsqueda

Código:

Nombre de obra:

Añadir

Listado de obras existentes

Filtrar: **Filtrar obras**

Resultados:

Listado de obras

Cód. obra	Nombre de la obra

Borrar

Ilustración 7.20: Gestión de obras.

Todas las operaciones se gestionan desde un misma ventana, que o bien relleno el formulario inicial permite el alta de una obra, o bien en el listado que se muestra más abajo, listar las obras aplicando el filtro relleno el formulario que hay sobre el o eliminar la obra seleccionándola previamente y pulsando el botón borrar.

7.2.7 Gestión de Usuarios.

La ventana que muestra el servicio de gestión de usuarios para un usuario que tenga el rol de gestión de obra o gestión de proveedores es mucho más limitado que el que tiene el usuario con rol de gestión de personal, ya que lo único que podrá hacer el usuario será asignar a un usuario una obra, para que este pueda operar en ella ya sea bien o en gestión de cierres o en gestión de proveedores. Pudiendo ser directores, jefes de obra, jefes de grupo, jefe de administración.

Para ello rellena los datos del usuario y busca la obra que se le quiera asignar.

The screenshot shows a web application interface with a light blue background. At the top, there is a section titled "Datos del usuario" (User Data) with input fields for "Nombre:" (Name) and "Apellidos:" (Surnames), a dropdown menu for "Categoría:" (Category), and an input field for "Nombre de usuario:" (Username). To the right of these fields are three buttons: "Añadir" (Add), "Modificar" (Modify), and "Cambiar contr." (Change password). Below this is a section titled "Asignación de obras al usuario" (Work assignment to user) with a search field "Filtrar obras por código o nombre:" (Filter works by code or name) and a "Filtrar obras" (Filter works) button. Underneath, there are two main panels: "Listado de obras" (List of works) on the left, which contains a table with columns "codigo" (code) and "nombre obra" (work name), and "Obras asignadas al usuario" (Works assigned to user) on the right, which is currently empty. A vertical toolbar with two green circular icons is positioned between the two panels.

Ilustración 7.21: Asignar usuarios a obra.

7.2.8 Gestión de cierres.

Al final de cada mes, todos los administrativos de obra (un administrativo al cual se le ha asignado una obra) tienen que mandar el cierre (facturación de la obra) para ser revisada y aprobada por los jefes de obra. A su vez estos cierres tienen que ser aprobados por el director de construcción.

En caso de existir discrepancias, el jefe de obra o si se ha escalado al siguiente nivel el director de construcción, podrá no aprobar dicho cierre, para que se rectifique.9.2.9 Gestión de proveedores.

Listado de obras asignadas:

Filtrar por nombre/código de obra
 Filtrar cierres por mes
 Mostrar solo cierres válidos

21/04/2011

Código Obra	Nombre Obra
A0085	CONSTRUCCION I.E.S. DE SANTA POLA
A0088	36 VIVIENDAS EN FINESTRAT
A0090	21 VIVIENDAS EN FINESTRAT 2
A0091	URB.VIVIENDAS FINESTRAT
A0092	AREA DE RODAJE CIUDAD DE LA LUZ
A0094	I.E.S. NUEVO EN TORREVIEJA (ALICANTE)
A0096	REFORMA INSS MURCIA
A0099	CUENTE VARIOS
A0101	REST. MEDIOAMBIENTAL DEL RIO AMADORIO. FASE IV
A0102	EDIFICIO TESORERIA TORREVIEJA
A0103	39 VIVIENDAS EN FINESTRAT
A0104	148 VIVIENDAS PLAYA SAN JUAN
A0105	REPARACION PLACAS DE CUBIERTA TORRE GENERALITAT
A0106	CONSERVATORIO DE ELDA
A0107	CONSEJO REGULADOR D.O. ALICANTE
A0108	COLEGIO PUBLICO HABANERAS (TORREVIEJA)
A0109	OBRA EMERGENCIA EN CALPE
A0110	DERRIBO EDIF CENTRO ESPEC. DENIA
A0111	CEIP BRACAL EN MURO DE ALCOY

Subir cierre

Cierres de la obra seleccionada:

Últ. sub. fe.	Últ. sub. ho.	Título	fecha	Usuario
30/03/2011	9:42:54	A0101:31-03-2011	31/03/2011	ecalero
01/03/2011	16:28:02	A0101:28-02-2011	28/02/2011	ecalero
04/02/2011	9:30:20	A0101:31-01-2011	31/01/2011	ecalero
04/02/2011	9:55:05	A0101:31-12-2010	31/12/2010	ecalero
04/02/2011	9:54:08	A0101:30-11-2010	30/11/2010	ecalero
04/02/2011	9:53:30	A0101:31-10-2010	31/10/2010	ecalero
04/02/2011	9:43:47	A0101:30-09-2010	30/09/2010	ecalero
04/02/2011	9:43:03	A0101:31-08-2010	31/08/2010	ecalero
02/08/2010	18:11:41	A0101:31-07-2010	31/07/2010	ecalero

Bajar cierre

No válido

Ilustración 7.22: Gestión de cierres.

El servicio de gestión de cierres muestra al usuario las obras que tiene asignadas, seleccionando una obra y pulsando el botón Subir cierre, le aparecerá una ventana indicándole el fichero para así subir y guardar la facturación del mes de la obra.

Cuando se selecciona una obra, en un listado bajo el listado de obras aparece el histórico de los cierres de la obra. Teniendo las opciones de descargar el cierre seleccionado y de mostrar inconformidad en caso de encontrar algún error.

7.2.9 Gestión de proveedores.

Gestión de Proveedores

- [Alta proveedor](#)
- [Consultas](#)
- [Editar proveedor](#)
- [Obras](#)
- [Usuarios](#)

Ilustración 7.23: Menú servicios gestión de proveedores.

El objetivo del servicio de alta de proveedores es la de centralizar las certificaciones que tiene cada uno de los proveedores que suministran a las obras. Se podrá asignar a un empleado la responsabilidad de encargarse de actualizar estos certificados solicitándolos al proveedor cuando se acerque a su vencimiento, y ver para qué obra trabaja cada uno de los proveedores.

7.2.9.1 Alta de proveedores.

Para poder dar de alta un proveedor hay que rellenar el formulario, con el CIF, nombre de proveedor, y adjuntar a este formulario los certificados pertinentes.

Cada formulario tiene una fecha de caducidad que permitirá al encargado de actualizar dicha documentación, saber cuando pierde caduca dicho certificado.

	Fecha caduc:	Documento:
Certificado de Hacienda:	30/12/1899	<input type="text"/>
Certificado Seguridad Soc:	30/12/1899	<input type="text"/>
Seguro Convenio Colectivo	Pago:	<input type="text"/>
	Contrato:	<input type="text"/>
Seguro Responsabilidad Civil	Pago:	<input type="text"/>
	Contrato:	<input type="text"/>
Servicio de Prevención / Mútua	Pago:	<input type="text"/>
	Contrato:	<input type="text"/>
TC1 / TC2:	30/12/1899	<input type="text"/>
R.E.A.:	30/12/1899	<input type="text"/>

Ilustración 7.24: Alta de proveedores.

Desde el mismo formulario se le puede asignar obras, y contratas al proveedor.

codigo	nombre obra

Ilustración 7.25: Asignación de obras a un proveedor.

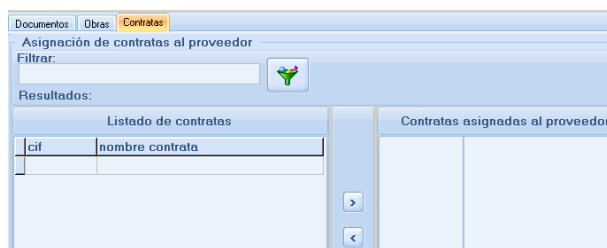


Ilustración 7.26: Asignación de contrata a un proveedor.

7.2.9.2 Consulta.

Permite por medio de un filtro de búsqueda listar los documentos de los proveedores que trabajan en una obra o bien los documentos de un proveedor específico.

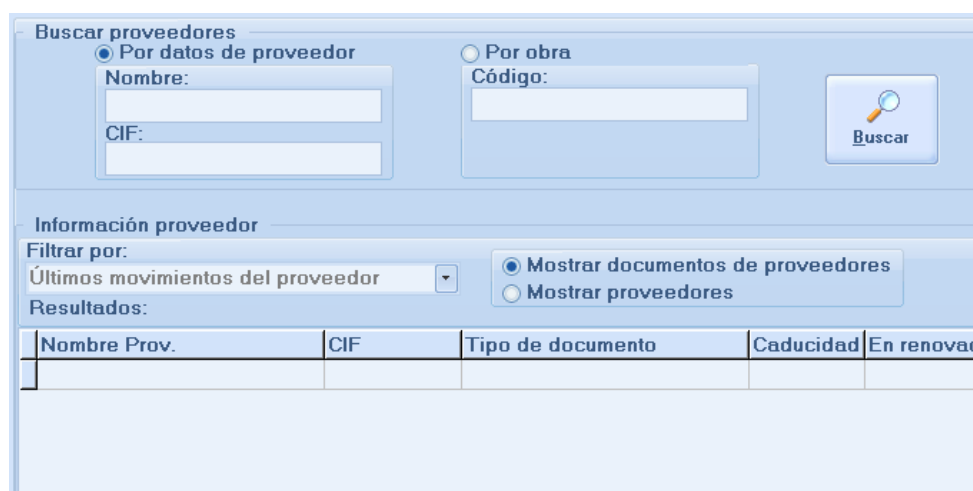


Ilustración 7.27: Consulta de proveedores.

7.2.9.3 Edición de proveedores

Ofrece un formulario muy parecido al lata de proveedores para hacer las modificaciones oportunas, ya se la de actualizar documentos que estén caducados, asignaciones a obras o a una contrata.

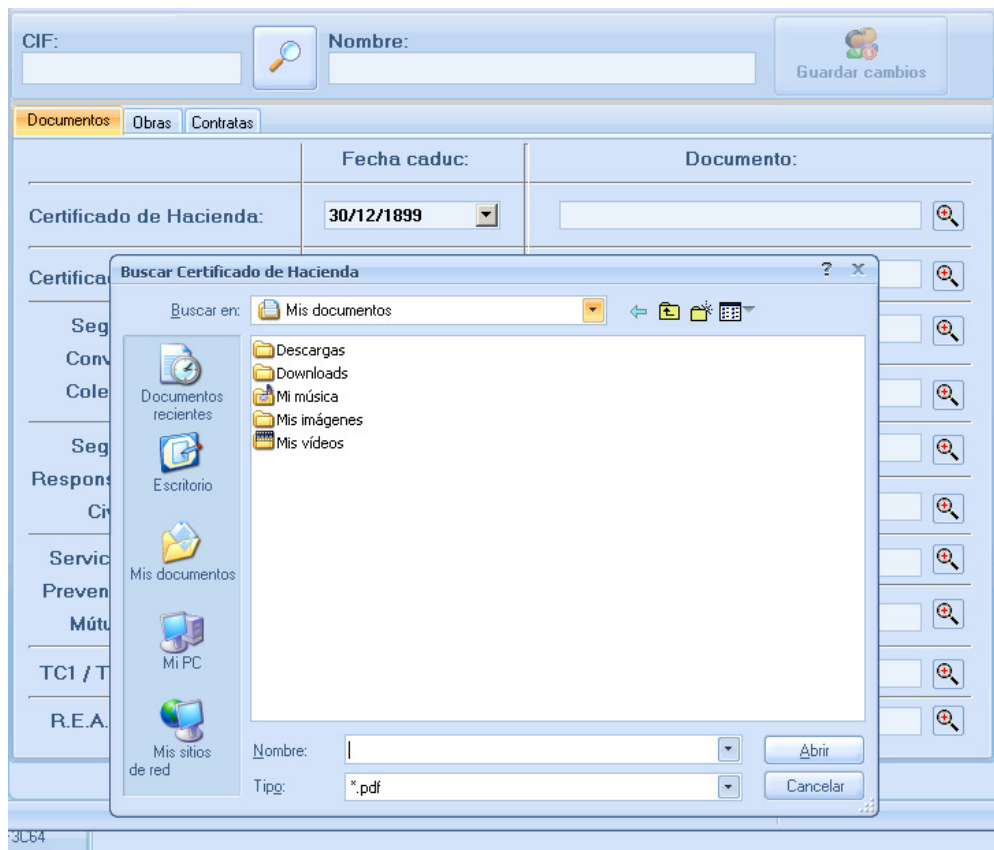


Ilustración 7.28: Subir documentación de proveedor.

7.2.10 Gestión de personal.



Ilustración 7.29: Menú servicio gestión de personal.

Esta sección es la encargada de dar de alta los usuarios en el sistema, asignarles el rol que desempeñarán, asignarles una obra, una empresa y su superior.

Todo se realiza desde el mismo formulario, y tendremos dos pestañas, una para asignarle una obra y otra para asignarle un superior.

Los datos a rellenar son Nombre y Apellidos del empleado.




Empresa, una empresa de las que componen el grupo empresarial X.

La categoría definirá el rol que desempeñe dentro de la aplicación, de los cuales podremos encontrar:

Datos del usuario

Nombre: Apellidos:


Categoría: Nombre de usuario:

Asignación de obras | Asignación de superior del usuario

Asignación de obras al usuario

Filtrar obras por código o nombre:



Resultados:

Listado de obras		Obras asignadas
codigo	nombre obra	




Ilustración 7.30: Alta de usuario

Administrativo, Administrativo telefonía, Administrativo de RRHH, jefe de obra, jefe de administración, jefe de grupo, Director, Informático. A su vez un administrativo, jefe de obra y jefe de grupo pueden tener una o varias obras asignadas. Según estas combinaciones se generan los posibles permisos para operar en los servicios. Una vez el usuario es dado de alta por el departamento de recursos humanos, se emite una notificación al departamento de informática y a sus superiores para que realicen las gestiones pertinentes.

Datos del usuario


Nombre: Apellidos:

Categoría: Nombre de usuario:

Asignación de obras | **Asignación de superior del usuario**

Filtrar usuarios por nick:



Resultados:

nombre	apellidos	nick

Ilustración 7.31: Asignar superior.

7.3 Listín telefónico.

Este servicio ofrece al todo usuario registrado en la aplicación poder acceder al listín telefónico de todo el grupo empresarial, pudiendo consultar cualquier numero de teléfono, email, extensión, etc...

La aplicación ofrece una serie de campos con la cual poder hacer la consulta.

LISTÍN TELEFÓNICO

Búsqueda:

Nombre / Apellidos N° de Teléfono
 Nombre: Número:

Resultados:

Arrastre una columna aquí para agrupar por dicha columna

Empresa	nick	Nombre	Apellidos	Nº Teléfono	Ext.	Tif Mesa	email	Cargo
C	a	A	G	67	0	1310	0 a eno@	Topografa
C	js	z Jr	uel G z	67	12	1259	js zalez@	Director
C	n	ez M	G z Diaz	62	6	1212	n izalez	Informatic
C	a	ez A	el G z Mora	66	14	1301	a zalez@	Administra
C	n	o M	G	64	3	1243	n nero@	Administra
S	ea	e	Ei G	60	8	8703	e i@ser	Gerente Gr
S	ea	je	ez Jc onio G z	66	0	8706	je ierrez@	Encargado
C	a	le A	H ez	64	7	1297		Encargado
C	v	V	Ig	62	5	1231	v il@cle	Jefa de Pro
C	ju	Jc	ia Ir larin	66	2	1293	ju nzo@	Informatic
N	ik	rt a	A Jr	67	12	5202	a i@ger	Subdirecto
C	y	or Y	DA L. DOR MAR	61	13	1371	y ador@	Administra

Ilustración 7.32: Listín telefónico.

8 Conclusiones

El proyecto ha pasado por varias etapas, las cuales se pueden ver en los distintos apartados de esta memoria, las etapas son: etapa de análisis, diseño y por último de implementación. Después la realización del manual que se incluyen en la memoria.

- El análisis se realizó mediante un estudio extenso sobre el producto pedido por un grupo empresarial X, manteniendo varias reuniones a lo largo de toda la etapa. Se realizó la especificación de requisitos, casos de uso, diagramas de secuencia y la estructura más adecuada de la base de datos. En dichas reuniones se acordó los colores, estructura de la aplicación, etc.
- El diseño donde se ha plasmado los resultados del análisis, se realizó un pequeño prototipo, en él se aplicó el diseño y esa parte de la aplicación se realizó sobre este diseño, tras la aceptación por parte de la empresa. Además en esta fase se eligió la tecnología a utilizar para la realización de la aplicación.
- La implementación conllevó un trabajo adicional, pues al no tener mucho conocimiento sobre dicha tecnología tuve que documentarme y posteriormente empezar a implementar la aplicación.

La aplicación permite muchas vías de ampliación por su carácter modular.