

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

I.T. Telecomunicación (Sist. de Telecomunicación)



**UNIVERSIDAD
POLITECNICA
DE VALENCIA**



**ESCUELA POLITECNICA
SUPERIOR DE GANDIA**

“Técnicas y herramientas para implementación de flujos de trabajo en el proceso docente”

TRABAJO FINAL DE CARRERA

Autor/es:

Carmen María Juliá Gil

Director/es:

Dr. Roberto Llorente Sáez

Dña. María Morant Pérez

GANDIA, 2010

Qué triste sería la vida
sin sueños ni ilusiones.
Qué triste sería la vida
sin la esperanza de hacerlos
algún día realidad.

A mis padres y hermanos,
por todo su apoyo en la distancia.

A Miguel Ángel,
por estar siempre ahí.

ÍNDICE

| | |
|--|-------------|
| ÍNDICE DE FIGURAS | V |
| ÍNDICE DE TABLAS | VI |
| ÍNDICE DE ANEXOS | VII |
| ACRÓNIMOS | VIII |
| 1. OBJETIVOS Y METODOLOGÍA DEL PROYECTO | 1 |
| 1.1 Objetivos del proyecto | 2 |
| 1.2 Fases del desarrollo del proyecto | 2 |
| 2. INTRODUCCIÓN | 5 |
| 2.1 Motivación del proyecto | 6 |
| 2.2 Estado del Arte..... | 9 |
| 2.3 Ingeniería del Software | 11 |
| 2.4 ¿Web 2.0 o escritorio? | 13 |
| 3. PLANTEAMIENTO Y SOLUCIÓN AL PROBLEMA | 19 |
| 3.1 Programación Orientada a Objetos | 21 |
| 3.2 Arquitectura Orientada a Servicios..... | 23 |
| 3.3 RIA – Aplicaciones Ricas de Internet..... | 26 |
| 3.4 Solución al Problema | 26 |
| 4. LINEAS DE ACTUACIÓN | 29 |
| 5. BASE DE DATOS | 33 |
| 5.1 Diseño conceptual..... | 35 |
| 5.2 Conceptos del Modelo Entidad-Relación | 35 |
| 5.3 Diagrama Entidad-Relación..... | 38 |
| 5.4 Estructura de la Base de Datos..... | 39 |
| 5.5 Sistemas de Base de Datos..... | 40 |
| 5.5.1 Oracle | 40 |
| 5.5.2 PostgreSQL..... | 41 |
| 5.5.3 Firebird | 42 |
| 5.5.4 MaxDB | 42 |
| 5.5.5 MySQL..... | 43 |
| 6. TECNOLOGÍA DE IMPLEMENTACIÓN | 45 |
| 6.1 Oracle Forms. Developer 2000 | 47 |
| 6.2 Ajax..... | 49 |
| 6.3 OpenLaszlo | 49 |
| 6.4 Microsoft Silverlight..... | 50 |
| 6.5 JavaFX | 50 |
| 6.6 GWT (Google Web Toolkit)..... | 51 |
| 6.7 Adobe Flex..... | 51 |
| 6.8 Adobe AIR..... | 53 |
| 6.8.1 MXML..... | 55 |
| 6.8.2 ActionScript..... | 56 |
| 7. LENGUAJES DEL LADO DEL SERVIDOR | 57 |
| 7.1 Java (JSP)..... | 59 |
| 7.2 ASP | 60 |
| 7.3 PHP | 60 |
| 8. SOFTWARE EMPLEADO | 63 |
| 8.1 XAMPP..... | 64 |
| 8.3 Flex Builder | 65 |
| 9. DESARROLLO DE LA APLICACIÓN | 69 |
| 9.1 Base de Datos..... | 73 |
| 9.2 Interfaz de Usuario..... | 74 |
| 9.3 Acciones por parte del Servidor..... | 82 |

| | |
|---|------------|
| 10. TEST DE PRUEBAS | 89 |
| 11. MANUAL DE USUARIO..... | 93 |
| 11.1 Manual de Administradores | 97 |
| 11.1.1 Mantenimiento de Usuarios..... | 98 |
| 11.1.2 Mantenimiento General | 102 |
| 11.1.3 Definición de Procesos | 104 |
| 11.1.3.1 Gestión de Tipos de Procesos..... | 104 |
| 11.1.3.2 Gestión de los Procesos..... | 108 |
| 11.2 Manual de Usuarios | 113 |
| 12. PRESUPUESTO..... | 119 |
| 13. CONCLUSIONES..... | 121 |
| 14. LÍNEAS FUTURAS | 123 |
| 15. BIBLIOGRAFÍA Y REFERENCIAS | 127 |
| 16. ANEXOS | 131 |

ÍNDICE DE FIGURAS

| | |
|--|-----|
| Figura 1. Distribución por domicilio familiar del alumnado..... | 7 |
| Figura 2. Distribución del alumnado con domicilio en la CAV por provincias y comarcas..... | 8 |
| Figura 3. Estructura de los modelos de Ingeniería de Software..... | 12 |
| Figura 4. Mapa conceptual de la Web 2.0..... | 14 |
| Figura 5. Representación de objeto..... | 22 |
| Figura 6. Estructura de lenguajes orientados a objetos..... | 23 |
| Figura 7. Arquitecturas Orientadas a Servicios..... | 23 |
| Figura 8 Diagrama del flujo de la información al utilizar una base de datos..... | 35 |
| Figura 9. Representación gráfica de los conceptos del modelo entidad-relación extendido..... | 38 |
| Figura 10. Ejemplo diagrama E-R..... | 39 |
| Figura 11. Estudio Forrester 2009 sobre Gestores de Base de Datos..... | 40 |
| Figura 12. Porcentaje de usos de diferentes plataformas..... | 53 |
| Figura 13. Jerarquía de nodos..... | 55 |
| Figura 14. Lenguajes de servidor más representativos..... | 59 |
| Figura 15. Pantalla de control de XAMPP..... | 64 |
| Figura 16. Interfaz aplicación phpMyAdmin..... | 65 |
| Figura 17. Imagen inicial al ejecutar la aplicación..... | 66 |
| Figura 18. Vista de diseño de un componente en Flex Builder..... | 67 |
| Figura 19. Vista de código de un componente en Flex Builder..... | 67 |
| Figura 20. Ejecutable de una aplicación desarrollada en AIR..... | 67 |
| Figura 21. Pantalla de creación de firma digital..... | 68 |
| Figura 22. Flujo de trabajo de aplicaciones Flex..... | 70 |
| Figura 23. Flujo de trabajo extendido de una aplicación Flex..... | 70 |
| Figura 24. Esquema de funcionamiento de la aplicación..... | 71 |
| Figura 25. Proceso carga: tipo de proceso definido por el administrador..... | 76 |
| Figura 26. Ejecutable de la aplicación..... | 94 |
| Figura 27. Pantalla de Instalación..... | 94 |
| Figura 28. Pantalla opciones de instalación de la aplicación..... | 95 |
| Figura 29. Proceso de instalación de la aplicación..... | 95 |
| Figura 30. Pantalla principal de la aplicación..... | 96 |
| Figura 31. Barra de herramientas principal de la aplicación..... | 96 |
| Figura 32. Botón de cierre de la aplicación..... | 96 |
| Figura 33. Pantalla de identificación del usuario..... | 96 |
| Figura 34. Detalle codificación contraseñas..... | 97 |
| Figura 35. Indicación de acceso al servidor no disponible..... | 97 |
| Figura 36. Barra de herramientas personalizada..... | 97 |
| Figura 37 Opción de mantenimiento de permisos..... | 98 |
| Figura 38. Pantalla de mantenimiento de permisos..... | 98 |
| Figura 39. Filtro de búsqueda de usuarios..... | 99 |
| Figura 40. Resultado de la búsqueda..... | 99 |
| Figura 41. Detalle selección de registro..... | 99 |
| Figura 42. Detalle de carga de datos..... | 100 |
| Figura 43. Visualización de los datos obtenidos..... | 100 |
| Figura 44. Pantalla asignación de nuevo permiso..... | 100 |
| Figura 45. Detalle de los posibles tipos de permisos..... | 101 |
| Figura 46. Pantalla modificación de permiso ya otorgado..... | 101 |
| Figura 47. Mensaje de éxito de la operación realizada..... | 101 |
| Figura 48. Detalle botones dinámicos..... | 102 |
| Figura 49. Mensaje de error de la operación realizada..... | 102 |
| Figura 50. Opción de mantenimiento automático de los procesos..... | 102 |
| Figura 51. Lista de los distintos tipos de procesos dados de alta en ese momento..... | 103 |
| Figura 52. Detalle de selección de un tipo de proceso..... | 103 |

| | |
|---|-----|
| Figura 53. Mensaje de proceso de mantenimiento de procesos completado..... | 103 |
| Figura 54. Detalle del menú de gestión de procesos | 104 |
| Figura 55. Detalle del submenú de Tipos de Procesos (arriba) y de Procesos (abajo)..... | 104 |
| Figura 56. Pantalla de alta y modificación de Tipos de Procesos | 105 |
| Figura 57. Alta de un nuevo tipo de proceso..... | 105 |
| Figura 58. Mensaje de confirmación de proceso realizado correctamente..... | 105 |
| Figura 59. Detalle del estado de la pantalla tras guardar un nuevo tipo de proceso..... | 106 |
| Figura 60. Lista de tipos de procesos | 106 |
| Figura 61. Detalle de selección de tipo de proceso | 107 |
| Figura 62. Modificación de un tipo de proceso vigente | 107 |
| Figura 63. Mensaje de confirmación de modificación de tipo de proceso correcta | 107 |
| Figura 64. Mensaje de confirmación de eliminación de tipo de proceso correcta | 108 |
| Figura 65. Pantalla de alta y modificación de un proceso..... | 108 |
| Figura 66. Detalle de generación automática de código identificativo del proceso | 108 |
| Figura 67. Detalle de la fecha de alta del proceso..... | 109 |
| Figura 68. Detalle de selección del tipo de proceso, título y detalle en el proceso de alta de un proceso | 109 |
| Figura 69. Campo Documento Asociado a un proceso | 109 |
| Figura 70. Ventana de selección del documento adjunto asociado al proceso..... | 110 |
| Figura 71. Ayuda del campo del responsable del proceso | 110 |
| Figura 72. Validación del campo Email Responsable..... | 110 |
| Figura 73. Mensajes de ayuda del campo de la fecha de baja | 111 |
| Figura 74. Detalle de validación de la fecha de fin | 111 |
| Figura 75. Detalle de parámetros de la definición de procesos..... | 111 |
| Figura 76. Mensaje de confirmación de proceso realizado correctamente..... | 112 |
| Figura 77. Lista de procesos vigentes a esa fecha..... | 112 |
| Figura 78. Detalle de la barra de herramientas con el nuevo tipo de proceso | 113 |
| Figura 79. Detalle del menú de los tipos de procesos | 113 |
| Figura 80. Mensaje de carga de los procesos de un tipo | 113 |
| Figura 81. Lista de procesos de un tipo vigentes hasta ese momento | 114 |
| Figura 82. Carga de un proceso y su documento asociado | 114 |
| Figura 83. Ventana de respuestas de un proceso..... | 115 |
| Figura 84. Detalle de indicación de pregunta a contestar..... | 115 |
| Figura 85. Cuadro de texto donde se insertará la respuesta | 115 |
| Figura 86. Mensaje de validación de respuesta guardada. | 116 |
| Figura 87. Detalle de activación de los distintos botones | 116 |
| Figura 88. Habilitación de los botones en la última pregunta | 116 |
| Figura 89. Ventana del sistema asociada a adjuntar documentos a las respuestas..... | 117 |
| Figura 90. Lista de documentos asociados a las respuestas | 117 |
| Figura 91. Mensaje de confirmación de respuestas enviadas correctamente | 117 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Datos de procedencia alumnado UPV..... | 6 |
| Tabla 2. Alumnos de la CAV por provincias/comarcas | 7 |
| Tabla 3. Limitaciones postgresQL..... | 41 |
| Tabla 4. Lenguajes de programación - popularidad | 62 |

ÍNDICE DE ANEXOS

| | |
|--|-----|
| ANEXO I. ESTRUCTURA BASE DE DATOS..... | 133 |
| ANEXO II. ESQUEMA RELACIONES BASE DE DATOS | 137 |
| ANEXO III. DIAGRAMA ENTIDAD – RELACION BASE DE DATOS | 139 |
| ANEXO IV. CLASE USUARIO | 141 |
| ANEXO V. CLASE TIPO PROCESO | 143 |
| ANEXO VI. CLASE PROCESO..... | 145 |
| ANEXO VII. EVENTO USUARIOEVENT..... | 147 |
| ANEXO VIII. CARGA DE UN TIPO USUARIO | 149 |
| ANEXO IX. CLASE TITLEBACKGROUND..... | 151 |
| ANEXO X. DBCONN.PHP..... | 153 |
| ANEXO XI. CABECERAXML.PHP | 155 |
| ANEXO XII. BORRADODIRECTORIO.PHP | 157 |

ACRÓNIMOS

- AJAX: Asynchronous Javascript And XML
- API: Interfaz de Programación de Aplicaciones
- ASP: Active Server Pages

- BD: Base de Datos
- BSD: Berkeley Software Distribution

- CAV: Comunidad Autónoma Valenciana
- CGI: Common Gateway Interface
- COM: Component Object Model
- CORBA: Common Object Request Broker Architecture
- CSIC: Consejo Superior de Investigaciones Científicas

- DBMS: DataBase Management System
- DCOM: Distributed COM
- DNI: Documento Nacional de Identidad

- ECMA: European Computer Manufacturers Association
- ER: Entidad-Relación

- GPL: General Public Licence
- GWT: Google Web Toolkit

- HD: High Definition (Alta Definición).

- IAS: Internet Application Server
- IDE: Entorno de Desarrollo Integrado
- IIOP: Internet Inter-ORB Protocol

- JSP: Java Server Pages
- JVM: Máquina Virtual de Java

- MXML: Macromedia eXtensible Markup Language

- OLAP: On-Line Analytical Processing
- OLTP: On-Line Transactional Processing
- OMG: Object Management Group

- PGDG: PostgreSQL Global Development Group
- PHP: Hipertext Preprocesor
- POO: Programación Orientada a Objetos

- RAD: (Rapid Application Development).
- RIA: Aplicación Rica de Internet
- RMI: Remote Method Invocation
- RPC: Remote Procedure Call

- SAP: Systems, Applications, Products
- SDK: Kit de Desarrollo de Software

- SGBD: Sistema Gestor de Base de Datos
- SOAP: Simple Object Access Protocol (Protocolo Simple de Acceso a Objetos)
- SQL: Structured Query Language

- TIC: Tecnología de la Información y Comunicación

- UPV: Universidad Politécnica de Valencia

- WWW: World Wide Web

- XAMPP: X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl

1. OBJETIVOS Y METODOLOGÍA DEL PROYECTO

Los objetivos principales de este proyecto son el desarrollo e implementación de una herramienta software dinámica que permita a la comunidad universitaria, tener acceso a distintos servicios que se puedan ofrecer desde la Universidad Politécnica de Valencia, UPV, en su etapa de formación, al mismo tiempo permitiéndole al profesorado la implementación del flujo de trabajo de dichos servicios y en donde la eficacia y eficiencia no se encuentren en contraposición.

Esta herramienta tendrá que adaptarse en todo momento al perfil del usuario que se encuentre ejecutando la aplicación mostrándole las opciones definidas a su perfil, sin relegar por ello a un segundo plano la calidad de la interfaz de usuario y la posibilidad de dispersión de la herramienta en distintos dispositivos móviles.

Para tal fin se llega a la decisión, tras contemplar diferentes posibilidades, de desarrollar una aplicación RIA (Aplicaciones Ricas de Internet, del inglés *Rich Internet Application*) en el escritorio mediante el motor de ejecución de Adobe® AIR®.

1.1 Objetivos del proyecto

Para el desarrollo de este proyecto se propone un objetivo principal:

- El desarrollo de herramientas que permitan la implementación de flujos de trabajo en entornos gestionados, como el proceso docente.

A éste, se le puede asociar dos objetivos adyacentes como:

- Integración de la información sobre la que aplicar flujos de trabajo.
- Implementación de flujos de trabajo asociados a documentación docente.

1.2 Fases del desarrollo del proyecto

De las fases de las que constará el desarrollo de los objetivos del proyecto serán:

- Análisis de los requisitos para la integración de información e implementación de flujos de trabajo sobre en las actividades típicas de la gestión docente (evaluación continua, por ejemplo)
- Análisis del estado del arte y selección del entorno de desarrollo software según los requisitos de la tarea anterior.
- Desarrollo de una herramienta de integración de información
- Definición del interfaz de la herramienta de integración
- Definición de la estructura de almacenamiento de la información (Base de datos)
- Implementación de la herramienta de integración de información y de la estructura de almacenamiento
- Test funcional de lo desarrollado hasta este punto
- Desarrollo de una herramienta de gestión de los flujos de trabajo

- Definición del interfaz de la herramienta de gestión de los flujos de trabajo
- Definición de los posibles flujos de trabajo
- Implementación de la herramienta de gestión de flujos de trabajo
- Test funcional de la herramienta de gestión
- Redacción de la memoria que contemple el desarrollo del proyecto

1. OBJETIVOS Y METODOLOGÍA DEL PROYECTO

2. INTRODUCCIÓN

2. INTRODUCCIÓN

Es innegable que, hoy en día, la enseñanza y la Tecnología de la Información y Comunicación (TIC's) van unidas de la mano creando una simbiosis perfectamente ligada y que no podemos menospreciar. La irrupción de las nuevas tecnologías en la educación ha facilitado la labor de la enseñanza y ha hecho que ésta sea accesible a una sociedad en continuo movimiento.

Hoy en día se encuentran en el mercado aplicaciones para diversas plataformas, que sirven de apoyo en la docencia pero, ¿consiguen cubrir sus objetivos?, ¿abarcan los deseos del profesorado?, ¿y del alumnado? Aun cumpliendo estos objetivos, ¿son realmente eficientes o para poder llegar a sacarles totalmente partido a estas aplicaciones es necesario emplear parte considerable de nuestro tiempo, hoy en día tan escaso por la vertiginosa velocidad a la que se mueve el mundo, que realmente resultan más engorrosas que beneficiosas?

Por ello, intentamos con este proyecto dar respuesta a estas cuestiones entre otras y encontrar soluciones adecuadas a la aplicación que estamos buscando.

2.1 Motivación del proyecto

Este proyecto nace en el entorno docente universitario de la UPV promovido por la necesidad del colectivo docente de poder disponer de una herramienta que les facilite el trabajo en la gestión, actualización y flujo de documentos y datos, para la ardua tarea de mantener a sus alumnos al corriente de los diferentes recursos que se les facilita durante su enseñanza.

En la actualidad los temarios, prácticas, catálogos y notas informativas, entre otros, son actualizados, sino casi a diario, semanalmente por medio del "PoliformaT" al que los alumnos tienen acceso remotamente vía Web. Al mismo tiempo, en ciertas ocasiones, se pone a disposición de los alumnos la posibilidad de poner a prueba sus conocimientos mediante ejercicios, prácticas, cuestiones, etc. que se van resolviendo en la distancia a través de recursos de la misma índole que pueden obtener documentación.

Bien es verdad, que estos recursos no hacen sino facilitar la enseñanza del alumnado al poder ser considerada como una enseñanza no presencial. Tenemos que tener en cuenta que la UPV es considerada la décima universidad española en el 2.010 según encuestas del CSIC [1] y cuenta con un 14% de estudiantes procedentes de otras comunidades autónomas o de otros países [2] según datos proporcionados por el rectorado de la UPV referentes al curso académico 2001/2005, que aunque sean datos no muy recientes, los números no han variado considerablemente con respecto al curso actual, y que podemos ver en la Tabla 1, donde se contempla el número de alumnos matriculados agrupados por su procedencia y en Figura 1 donde se muestra el porcentaje que representa de cada sector, por lo que con estas opciones se está facilitando que todos tengan por igual acceso a diversos recursos, evitando los desplazamientos a veces imposibles.

| Domicilio Familiar en | Alumnos DNI - Curso 04/05 |
|---------------------------------------|----------------------------------|
| Comunidad Autónoma Valenciana | 31.125 |
| Resto comunidades autónomas españolas | 3.072 |
| Extranjero | 1.880 |
| Total | 36.077 |

Tabla 1. Datos de procedencia alumnado UPV

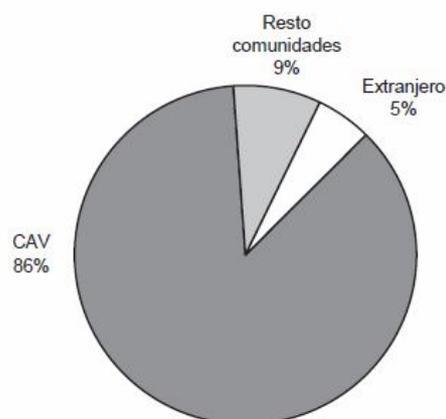


Figura 1. Distribución por domicilio familiar del alumnado

En la Tabla 2 se muestra, con respecto al mismo año de referencia que datos anteriores, la cantidad y porcentaje de alumnos de procedencia valenciana agrupados por comunidades autónomas y comarcas. Si bien es verdad, que la red de carreteras y ferroviarias en la actualidad han mejorado considerablemente y proporcionan una interconexión entre diferentes comarcas cada vez mejores, hay que tener en cuenta según estos datos, que todavía hay un 15,53% de estudiantes provenientes de las comarcas de Alicante y Castellón, que prácticamente todos los días tienen que hacer uso de esas redes de desplazamiento para asistir a las clases, y sumados a lo largo de toda la etapa estudiantil suponen un tiempo considerable que es desaprovechado.

| Comunidad Autónoma Valenciana PROVINCIA / COMARCA | Alumnos DNI Curso 04/05 | % respecto total CAV | Comunidad Autónoma Valenciana PROVINCIA / COMARCA | Alumnos DNI Curso 04/05 | % respecto total CAV |
|--|----------------------------|-------------------------|--|----------------------------|-------------------------|
| VALENCIA | 26.291 | 84,47% | ALICANTE | 3.357 | 10,79% |
| VALÈNCIA | 11.928 | 38,32% | L'ALCOIÀ | 897 | 2,88% |
| L'HORTA OEST | 2.634 | 8,46% | LA MARINA ALTA | 588 | 1,89% |
| L'HORTA NORD | 2.093 | 6,72% | L'ALACANTÍ | 462 | 1,48% |
| LA RIBERA ALTA | 1.841 | 5,91% | EL COMTAT | 296 | 0,95% |
| LA SAFOR | 1.577 | 5,07% | EL BAIX SEGURA | 284 | 0,91% |
| L'HORTA SUD | 1.244 | 4,00% | LA MARINA BAIXA | 254 | 0,82% |
| EL CAMP DEL TÚRIA | 1.102 | 3,54% | EL BAIX VINALOPÓ | 233 | 0,75% |
| EL CAMP DE MORVEDRE | 768 | 2,47% | EL VINALOPÓ MITJÀ | 206 | 0,66% |
| LA RIBERA BAIXA | 720 | 2,31% | L'ALT VINALOPÓ | 137 | 0,44% |
| LA VALL D'ALBAIDA | 697 | 2,24% | CASTELLÓN | 1.477 | 4,75% |
| LA COSTERA | 543 | 1,74% | LA PLANA BAIXA | 526 | 1,69% |
| LA PLANA DE UTIEL-REQUENA | 412 | 1,32% | LA PLANA ALTA | 509 | 1,64% |
| LA HOYA DE BUÑOL | 359 | 1,15% | EL ALTO PALANCIA | 205 | 0,66% |
| LOS SERRANOS | 164 | 0,53% | EL BAIX MAESTRAT | 170 | 0,55% |
| LA CANAL DE NAVARRÉS | 122 | 0,39% | L'ALCATALÉN | 28 | 0,09% |
| EL VALLE DE AYORA | 67 | 0,22% | ELS PORTS | 17 | 0,05% |
| EL RINCÓN DE ADEMUZ | 20 | 0,06% | L'ALT MAESTRAT | 11 | 0,04% |
| | | | EL ALTO MIJARES | 9 | 0,03% |
| | | | No consta | 2 | 0,01% |
| Total CAV | 31.125 | 100% | | | |

Tabla 2. Alumnos de la CAV por provincias/comarcas

Referente a los datos anteriores, podemos tener una mejor visualización en la Figura 2 que nos ayuda a tener una perspectiva general del porcentaje de alumnos de la Comunidad Autónoma de Valencia distribuida por comunidades y comarcas permitiéndonos tener una

2. INTRODUCCIÓN

visualización de la distancia geográfica entre las distintas comarcas y la ciudad de Valencia, ciudad central donde se localiza la Universidad Politécnica de Valencia.

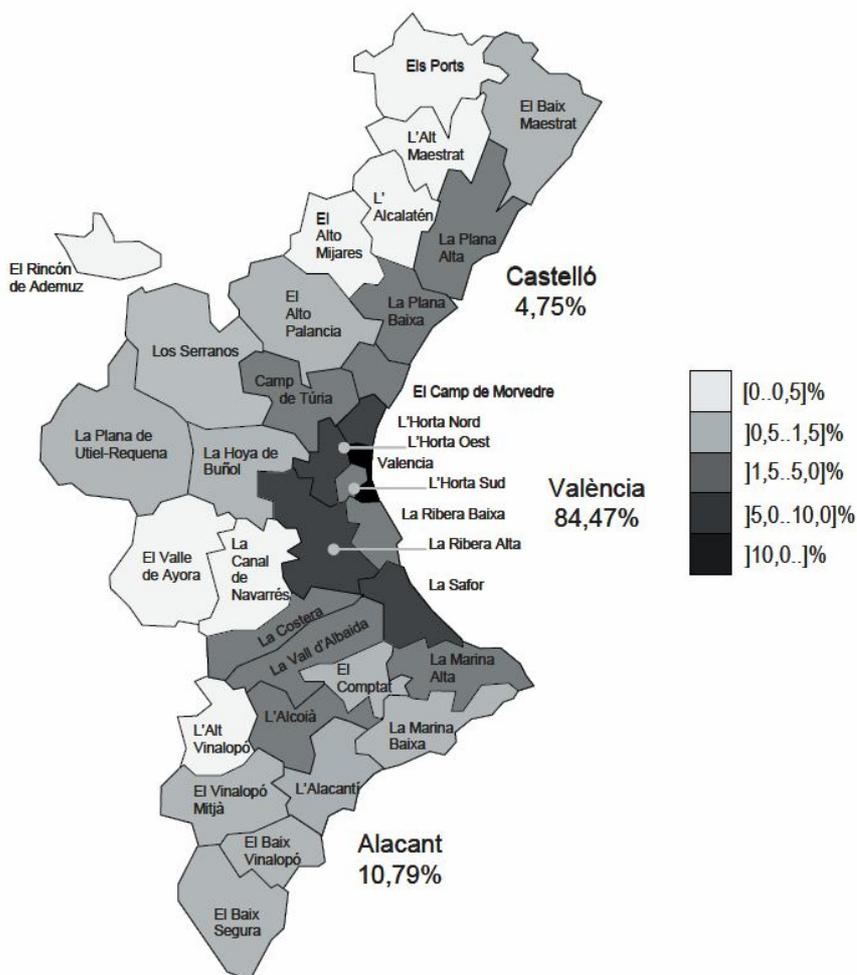


Figura 2. Distribución del alumnado con domicilio en la CAV por provincias y comarcas

Tal y como se ha comentado, este proyecto está desarrollado para favorecer al alumnado. Pero las tareas que hacen que esto sea posible consumen un tiempo considerablemente amplio el cual, en ocasiones, el profesorado no tiene disponible y a la vez son tareas engorrosas y nada fáciles.

Dentro de las obligaciones que tiene el profesorado, además de la docencia universitaria, están el mantener en buenas condiciones los materiales de los laboratorios, bien sea de forma directa o supervisando a los individuos a los que se les ha relegado estas funciones, mantener contacto con proveedores de materiales, supervisar, gestionar y realizar pedidos de materiales, etc. entre otras diversas funciones que si no se llevan con cierto control y orden pueden llegar a ser una cantidad considerable de tareas que consumen mucho tiempo y recursos del profesorado.

Es evidente que el valor del uso de las nuevas tecnologías como recurso educativo de apoyo a la docencia es innegable. No sólo por la riqueza de los contenidos multimedia, que facilitan la asimilación y comprensión de los contenidos por parte del alumno, sino también por su accesibilidad y disponibilidad. Obviamente, las nuevas tecnologías, al igual que los libros de la biblioteca, constituyen un recurso educativo, suponen un complemento de gran valor para la tarea docente, pero además, nos ofrecen la virtualización de otros recursos presentes en la enseñanza como ciertas tareas administrativas del centro (matriculación, consulta de notas y expediente académico, inscripciones a seminarios, cambios de grupo, petición de certificados, etc.) o incluso en algunas áreas educativas y según qué metodología se emplee, la virtualización del profesor. Con esto la pregunta que aparece rápidamente en nuestras mentes sería: ¿Podría

conseguirse una plataforma tecnológica capaz de ayudar de forma eficaz el papel del centro docente (escuela, facultad, academia, etc.) durante un porcentaje adecuado de la carga lectiva, de forma que no constituya un recurso educativo sino el medio en el que se construye el aprendizaje del alumno, al mismo tiempo que no suponga un esfuerzo considerablemente extra para el profesorado? En este proyecto vamos a intentar presentar una solución flexible y versátil capaz de hacer algunas de estas funciones.

2.2 Estado del Arte

En niveles educativos como la Enseñanza Universitaria se usan las nuevas tecnologías como soporte al proceso de aprendizaje del alumno, incorporando herramientas como el correo electrónico para consultas, los chats, los foros de debates y otras utilidades de la Web junto con unos cursos on-line que proporcionan al alumno la posibilidad de construir su aprendizaje de forma virtualmente guiada.

Según Salinas [3], la llegada de las tecnologías de la información y la comunicación al sector educativo viene enmarcada por una situación de cambios que no pueden ser considerados al margen de los cambios que se desarrollan en la sociedad, relacionados con la innovación tecnológica, con los cambios en las relaciones tecnología-sociedad que determinan las relaciones tecnología-educación. La educación no puede ser ajena a estos cambios puesto que los nuevos medios configuran una nueva sociedad [4], a la que el sistema educativo tendrá que seguir formando adecuadamente a los alumnos, garantizando su integración en esta nueva sociedad, socializándolos conforme a estos nuevos medios de comunicación, a estas nuevas relaciones sociales y empleando los medios utilizados en la comunicación social.

En los últimos años han aparecido sistemas informáticos para la enseñanza y, aunque el objetivo de todos ellos es muy similar, los medios mediante los cuales se llega a dicho objetivo varían en gran medida. Muchos de estos sistemas, mal identificados como “sistemas de eLearning”, únicamente se centran en la gestión de documentos y en su provisión a discentes y docentes, y aunque ciertamente faciliten la tarea de búsqueda y organización de información, no realizan un seguimiento del proceso de aprendizaje del alumno. Por eso, entender lo que es y no es el eLearning puede resultar confuso debido a la gran cantidad de términos que han aparecido.

El eLearning permite superar algunas de las barreras existentes en los sistemas de enseñanza asistida por ordenador, entre las que se pueden destacar:

- Eliminar las distancias y favorecer la movilidad de los usuarios alumnos.
- Aumentar el número de destinatarios que pueden seguir un curso simultáneamente.
- Permitir flexibilidad horaria.
- Permitir la alternancia de diversos métodos de enseñanza.
- Favorecer la interacción entre alumnos. Está demostrado que la no presencia física minimiza la timidez y favorece el establecimiento de comunicación entre los alumnos, especialmente en la adolescencia.
- Anonimato¹.

¹ Como indica Gabriel Kaplun en [5] Capítulo 2, pág. 24, algunas personas no realizan cursos de formación por vergüenza a reconocer que necesitan seguir formándose, especialmente si son personajes populares.

2. INTRODUCCIÓN

- Seguimiento y tutoría del progreso del alumno a través de los canales de comunicación establecidos.
- Posibilidad de escoger entre gran variedad de materiales, cursos y especialidades.
- Minimizar los costes de formación continua en la empresa.
- Favorecer la convivencia familiar para alumnos con responsabilidades familiares a su cargo.

Además de todas estas ventajas intervienen otros factores que favorecen la implantación de sistemas eLearning:

- Factores económicos: mejor relación coste-beneficio en la producción y desarrollo aprovechando la reutilización de componentes tecnológicos y materiales de aprendizaje. Es un factor interesante a la hora de aumentar los niveles de formación en países en desarrollo, con un alto ritmo de crecimiento económico y con grandes necesidades de trabajadores cualificados.
- Alta disponibilidad de recursos digitales. Las grandes empresas multinacionales necesitan distribuir materiales de aprendizaje a sitios geográficamente dispersos, para que estén disponibles en cualquier momento y desde cualquier lugar. La existencia de un gran número de recursos digitales libres y gratuitos en Internet (imágenes, clips de audio y vídeo, animaciones, etc.) favorece su reutilización y aprovechamiento por parte de las grandes empresas (o terceros, como puede ser una empresa especializada en la creación de cursos o implantación de sistemas de eLearning) para la creación de cursos a través de sistemas eLearning.
- Penetración social. La alta penetración en la sociedad de las nuevas tecnologías en general, y de Internet en particular, favorece la aceptación de nuevas vías de información y de comunicación.
- Ayudas estatales: Los programas de subvenciones por parte del Estado, las Comunidades Autónomas y el Fondo Social Europeo, han incentivado la creación y desarrollo de un sector empresarial dedicado a la formación on-line. Estas subvenciones han hecho posible la aparición de programas como los de Formación Continua de trabajadores, que contribuyen a la adaptación de los trabajadores a las más novedosas tecnologías.

Pero a pesar de todo lo expuesto hay que aclarar que no todo son ventajas. Algunos inconvenientes en el empleo de sistemas de eLearning son:

- Preparación del estudiante: es necesario un esfuerzo para asegurar que los estudiantes tienen las habilidades y conocimientos técnicos, así como el acceso al hardware y software necesarios para completar satisfactoriamente el curso basado en las TIC's. Tanto la gestión del tiempo como las habilidades meta cognitivas están relacionadas con las actitudes y la motivación del estudiante.
- Personal dedicado: al igual que los estudiantes, los profesores deben tener habilidades técnicas, conocimiento y acceso al hardware y software necesarios en este caso para facilitar el diseño y desarrollo del curso basado en las TIC's. Deben tener un excelente manejo del tiempo y la motivación para proporcionar asistencia y llevar el seguimiento del estudiante. No obstante algunos autores diferencian el rol

del profesor, encargado de la selección de contenidos, seguimiento y asistencia al alumno, del rol del técnico, encargado del diseño y creación del curso eLearning a partir de los contenidos, objetivos y metodologías, estableciendo de esta forma la necesidad de diferentes perfiles.

- Gestión de la información: a pesar de que se posean unas habilidades técnicas y un manejo del tiempo excepcionales, tanto los profesores como los alumnos requieren de interfaces que reduzcan las cuestiones logísticas y técnicas. El uso de boletines y listas de distribución pueden ayudar a manejar la sobrecarga de información.
- Equidad: no todos los usuarios cuentan con las mismas facilidades de acceso a Internet. La tecnología incrementa las diferencias entre los que tienen y los que no tienen tales posibilidades.
- Ancho de banda: uno de los mayores inconvenientes desde hace una década y que está desapareciendo rápidamente con la llegada de nuevas y mejores líneas de transmisión. Actualmente en Europa el ancho de banda es aceptable y permite transmitir con buenos resultados audio y vídeo sincronizados sin los indeseables “saltos” de antaño.

Ventajas e inconvenientes, disyuntiva que tras sopesar unos y otros, no nos cabe duda que ganarían las ventajas sin más que mirar a nuestro alrededor y observar las opciones tecnológicas que nos rodean, opciones en continuo desarrollo y cada vez más integradas en la sociedad.

Una filosofía de vida a la que no podemos cerrar los ojos y en la que parece ser que si no estás integrado en las nuevas tecnologías no eres nadie. Una sociedad cuya estructura alcanza el rango de “ente” dinámico, propiamente dicho, en el que los jóvenes se integran más temprano al mundo laboral o los mayores lo hacen en el mundo estudiantil teniendo ambos que compaginar ambos mundos, labor que a veces resulta tan difícil.

Por conclusión, todo aquello que facilite la enseñanza a distancia, sin ocasionarnos un quebradero de cabeza, nunca está de más.

2.3 Ingeniería del Software

La Ingeniería del Software es la disciplina que abarca los métodos, modelos, procedimientos y herramientas necesarios para la construcción de software eficiente.

Un proyecto Software es el proceso de gestión para la creación de un determinado Software, el cual engloba un conjunto de actividades de gestión (presupuesto, recursos, plazos), técnicos (metodología, técnicas, herramientas) y de comunicación (elaboración de documentación y manuales) y que deben cumplir ciertas necesidades alcanzando determinados niveles de calidad, plazos y costes.

La ingeniería del Software tiene varios modelos o paradigmas de desarrollo en los cuales se puede apoyar para la realización de software, de los que podemos destacar los siguientes por ser los más utilizados y completos:

- Modelo en Cascada
- Modelo en Espiral
- Modelo de prototipos

2. INTRODUCCIÓN

- Desarrollo por etapas
- Desarrollo iterativo y creciente
- RAD (*Rapid Application Development*)

En la Figura 3 se presentan las etapas que conforman cada uno de los cuatro modelos más representativos de Ingeniería del Software.

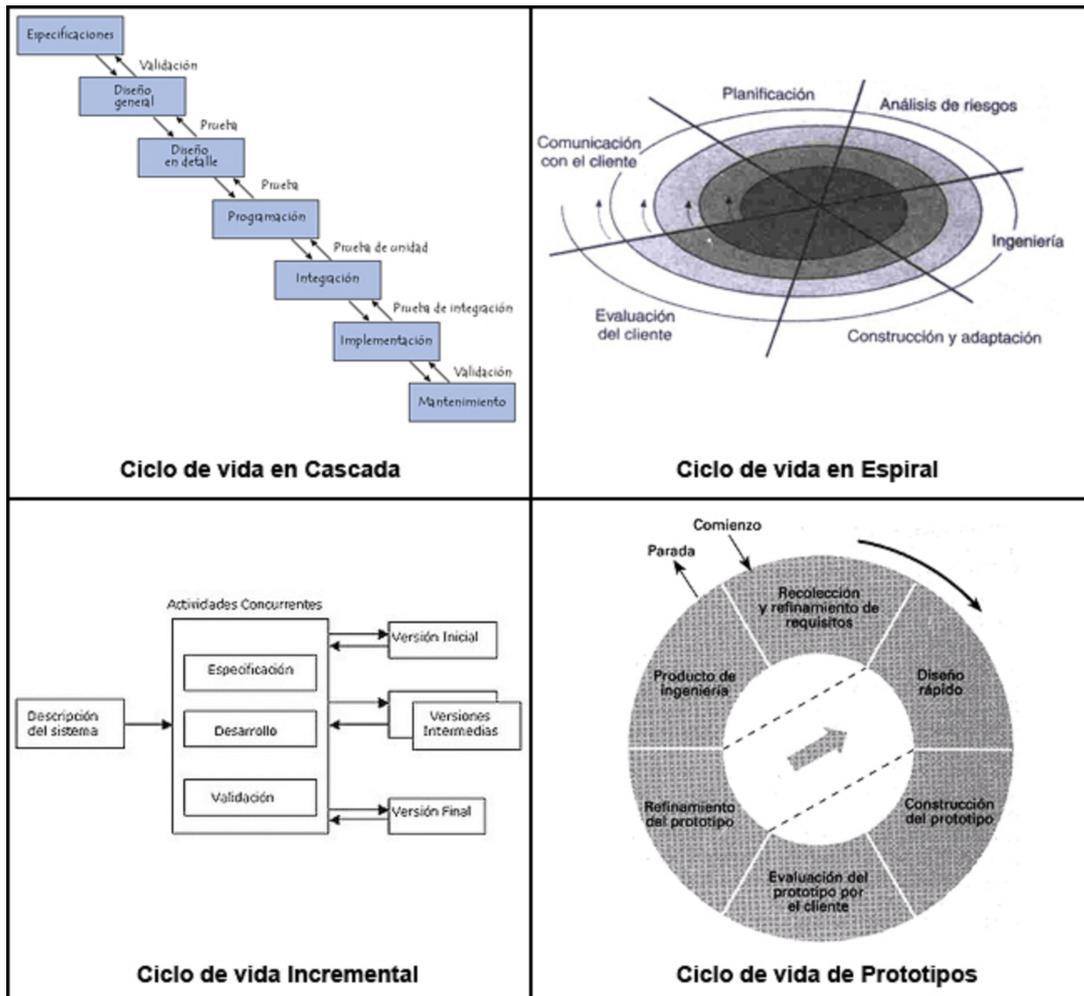


Figura 3. Estructura de los modelos de Ingeniería de Software

Aún sin decantarnos por un modelo u otro, conforme iba avanzando el desarrollo del proyecto nos dimos cuenta que nos estábamos basando en el modelo de prototipos por varios motivos, principalmente porque dependíamos de la evaluación que iba haciendo el cliente final de nuestro desarrollo, pues teníamos como principal consideración la que éste tenía del producto ya que será él quien finalmente tenga que trabajar con la aplicación y uno de nuestros objetivos es facilitarle el desempeño de su labor y por un segundo motivo que tras presentarle un prototipo inicial en una plataforma escogida por diferentes consideraciones como se expondrán en apartados posteriores, como fue Adobe Flex, se decidió volver a la fase de diseño para adaptar lo ya desarrollado a Adobe AIR por causas que también se expondrán posteriormente.

Hay que tener en cuenta, se elija el modelo de ciclo de vida que se elija, que las principales dificultades que se plantean a la hora de desarrollar software (aplicativos de negocios) corresponden a la etapa de capturar los requerimientos de los usuarios, ya que después de aquello se tiene la esperanza por parte de los desarrolladores que estos requerimientos sean

estáticos y no varíen conforme se está procediendo al desarrollo del software, sin hacerse cargo de los continuos cambios que tienen los modelos de negocios, producto de la competencia, otros productos y en general situaciones de mercado en muchos casos no previsibles. Según la empresa Standish Group² cerca del 60% de los requerimientos cambian a lo largo de un proyecto, tal y como ocurrió en nuestro caso que implicó un cambio de plataforma de desarrollo como se ha comentado en el párrafo anterior.

2.4 ¿Web 2.0 o escritorio?

La Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones Web enfocadas al usuario final. La Web 2.0 es la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través de la Web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio.

Y es que cuando la Web³ inició, nos encontrábamos en un entorno estático, con páginas en HTML que sufrían pocas actualizaciones y no tenían interacción con el usuario.

Pero para entender de donde viene el término de Web 2.0 tenemos que remontarnos al momento en que Dale Dougherty de O'Reilly Media⁴ utilizó este término en una conferencia en Octubre del 2004 en la que compartió una lluvia de ideas junto a Craig Cline de MediaLive. En dicho evento se hablaba del renacimiento y evolución de la Web.

Constantemente estaban surgiendo nuevas aplicaciones y sitios con sorprendentes funcionalidades. Y así se dio la pauta para la *Web 2.0 Conference* que arrancó en el 2004 y se realiza anualmente en San Francisco.

En la charla inicial del *Web Conference* se habló de los principios que tenían las aplicaciones Web 2.0:

- La Web es la plataforma
- La información es lo que mueve al Internet
- Efectos de la red movidos por una arquitectura de participación.
- La innovación surge de características distribuidas por desarrolladores independientes.
- El fin del círculo de adopción de software pues tenemos servicios en beta perpetuo

Web 2.0 no significa precisamente que existe una receta para que todas las aplicaciones Web entren en este esquema. Sin embargo, existen varias tecnologías que están utilizándose actualmente y que deberíamos de examinar con más cuidado en busca de seguir evolucionando junto a la Web.

El uso del término de Web 2.0 está de moda, dándole mucho peso a una tendencia que ha estado presente desde hace algún tiempo. En Internet las especulaciones han sido causantes de grandes burbujas tecnológicas y han hecho fracasar a muchos proyectos.

² <http://www.standishgroup.com/>

³ Referencia a WWW (World Wide Web). fue creada alrededor de 1989 por el inglés Tim Berners-Lee y el belga Robert Cailliau y publicado en 1992

⁴ antes llamada O'Reilly & Associates, es una empresa editorial estadounidense principalmente enfocada a libros de tecnología e informática

2. INTRODUCCIÓN

Además, nuestros proyectos tienen que renovarse y evolucionar. La Web 2.0 no es precisamente una tecnología, sino es la actitud con la que debemos trabajar para desarrollar en Internet. Tal vez allí está la reflexión más importante del Web 2.0.

La Figura 4 muestra un 'mapa meme' de la Web 2.0 que fue desarrollado en una sesión de *brainstorming* durante el FOO Camp, una conferencia en O'Reilly Media. Es fundamentalmente trabajo en curso, pero manifiesta las muchas ideas que irradian desde el núcleo de la Web 2.0.

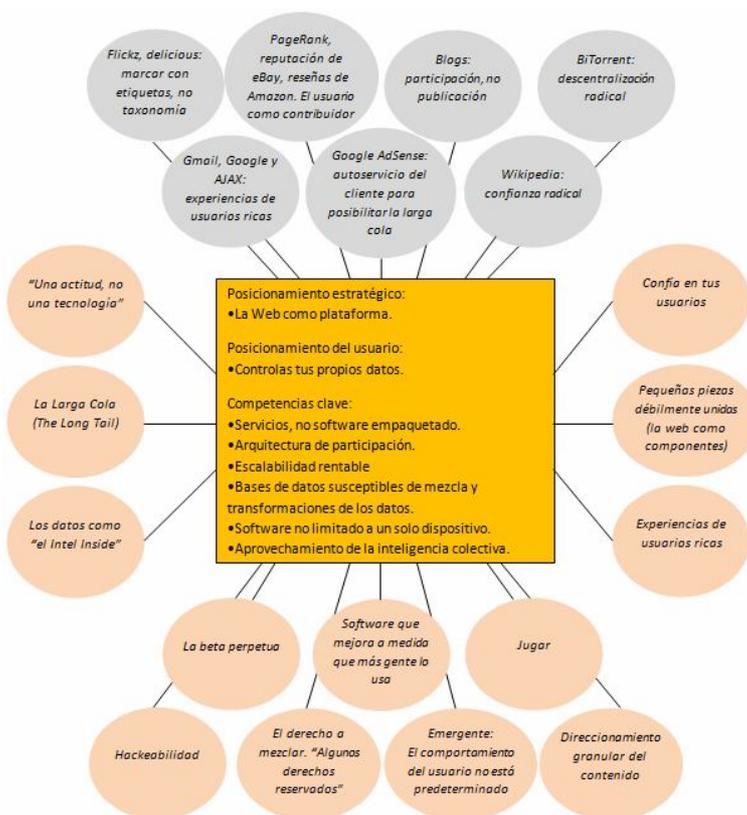


Figura 4. Mapa conceptual de la Web 2.0

Las principales características que presenta la Web 2.0 es que las páginas son dinámicas, integran recursos multimedia como vídeos, sonidos, que se pueden compartir. Los formatos utilizados para diseñarlas son Javascript, PHP, u otras similares, que permiten más funcionalidad. Emplean interfaces de fácil entendimiento para la interacción del usuario. La información se puede presentar en varias formas (escrita, audiovisual), y que esta se comparta entre los usuarios o entre estos y los dueños de las páginas. Permite que el usuario cree su propio contenido. La información se puede transmitir unidireccional o bidireccionalmente.

Pero no todo son ventajas, falta implementar estrategias de seguridad informática, el constante intercambio de información y la carencia de un sistema adecuado de seguridad ha provocado el robo de datos e identidad generando pérdidas económicas y propagación de virus.

La seguridad es fundamental en la tecnología, las empresas invierten en la seguridad de sus datos y quizás el hecho de que la Web aún no sea tan segura, crea un leve rechazo a la transición de algunas personas con respecto a la automatización de sus sistemas.

La Web 2.0 ha modificado definitivamente el concepto de programa. Ya no es sólo un software que uno debe instalar en su ordenador y ejecutarlo cada vez que lo necesita, sino que

también puede tratarse de una herramienta accesible en la Red para todo el mundo, con la ventaja de que nos permite compartir nuestros trabajos.

Se ha desplazado el ámbito que monopolizaba el PC de escritorio hacia Internet, que ya no es exclusivamente una plataforma para el correo y la navegación. Desde que se han popularizado los procesadores y hojas de cálculo on-line, se ha abierto un debate interminable sobre ventajas y desventajas de uno y otro sistema. El clásico contra el moderno, la historia frente al presente futuro, lo conocido versus lo experimental. En fin, ganas de polemizar.

Durante las últimas dos décadas, la mayoría de empresas han adoptado la World Wide Web como una plataforma estándar para la entrega de aplicaciones a los consumidores, socios, clientes, e incluso usuarios finales internos. La Web ofrece a las empresas más fácil despliegue, una interfaz de navegador estándar que requiere una formación mínima, y el acceso en cualquier lugar del mundo en cualquier momento y en cualquier dispositivo.

Algo que distingue a Internet es su capacidad para adaptarse, cambiar y renovarse ante las diversas necesidades y tendencias que mandan en el mercado. Esta constante evolución, hizo que la información que viajaba en una sola dirección, se convirtiera en una autopista de dos vías para lograr mayor participación de los usuarios. Fue entonces, cuando los usuarios comenzaron a darse cuenta de su verdadero poder y, de esta forma, pasaron de ser un mero espectador a transformarse en actores mucho más participativos de la gran película sin fin que ofrece Internet. Con esta creciente interrelación y con el usuario convirtiéndose en generador de contenidos, se establece el paradigma de la Web 2.0 y una nueva etapa en la gran red de redes.

De la mano de este fenómeno, también aparecen lo que se conoce como aplicaciones Web, es decir, herramientas on-line que permiten realizar tareas que antes sólo estaban disponibles en aplicaciones de escritorio. Desde soluciones de correo, pasando por agendas, editores de texto, editores de imágenes, unidades de almacenamiento virtual, planillas de cálculo y software más complejos, capaces de operar directamente desde Internet, sin que el usuario necesite más que tener un navegador actualizado en su computadora.

Con la evolución de las tecnologías para el desarrollo Web, las aplicaciones Web fueron creciendo, siendo cada día más robustas y ofreciendo cada vez más opciones para el usuario. Ya queda poco de aquellas primeras aplicaciones con opciones básicas e interfaz poco atractiva. Hoy en día las aplicaciones Web deben dar soluciones reales a los usuarios y mostrar una interfaz que esté a la altura de las aplicaciones de escritorio, sin perder eficiencia.

Las ventajas de las aplicaciones Web, tal y como se han citado previamente, son bien conocidas, pero vale la pena repasarlas. La primera y fundamental, como ya mencionamos: el usuario no debe instalar software adicional, sólo debe tener un navegador actualizado y una buena conexión a Internet. En relación a esto, también surge que ya no se necesitan varios GB en el disco duro para almacenar una aplicación que no hay que instalar. Por otro lado, como la aplicación corre en el navegador, no hay problema de compatibilidad de sistemas operativos, por lo cual es una solución multiplataforma. Tampoco debemos pensar en instalarla, actualizarla o qué pasará si se rompe nuestro disco duro. Al estar la aplicación en Internet, ella vive en un servidor del cual se ocupará el equipo de mantenimiento, el usuario sólo deberá recordar la dirección para acceder y la clave de acceso. Esto permite resolver de una manera mucho más sencilla y eficiente problemas críticos y cualquier necesidad de actualización. Esto también beneficia el trabajo en equipo que se puede realizar con personas que se encuentren en diversos lugares del mundo sin ningún tipo de problema.

Evidentemente hay aplicaciones que tienen una complejidad que aún hoy es difícil llevarla a la práctica en una solución completa como herramienta on-line. Sin embargo, la evolución de las técnicas existentes, nos demuestran día a día que el mundo está cada vez más permeable a utilizar herramientas on-line y a probar cosas nuevas. También vale la pena mencionar que las

2. INTRODUCCIÓN

“webapp”⁵ son una solución muy interesante y cada vez más difundidas para móviles, lo cual es otro aspecto que todo desarrollador debe tener en cuenta como un interesante camino por explorar.

Personalmente, no veo ninguna necesidad de convertir la posibilidad de elegir en un debate: no hay por qué absolutizar las preferencias y convertirlas en incompatibles, como si se tratase de una guerra fundamentalista, pero desarrolladores acaban abocados a la disyuntiva sobre qué plataformas desean desplegar sus aplicaciones, disyuntiva que deben solventar antes de comenzar con sus desarrollos.

Esto no significa que vamos a dejar de usar las clásicas aplicaciones de Escritorio. Sinceramente, ¿quién puede prescindir de la férrea llamada del Escritorio a la hora de editar una imagen, reproducir música o convertir un vídeo? Nadie, simplemente porque no se puede hacer todo vía Web. Además, el Escritorio sigue siendo la opción más potente y cómoda de usar, al menos por ahora.

Pero si queremos anticiparnos un poco al futuro, lo más probable es que de ahora en adelante las aplicaciones on-line copen cada vez más nuestro interés. No podemos quedarnos impasibles ante sus evidentes ventajas, el componente social y la movilidad ofrecida. Ante todo, son aplicaciones sociales y son accesibles desde cualquier dispositivo, basta con que tengas un navegador Web abierto. Tampoco es tan descabellada la idea de que nuestros equipos se vayan convirtiendo poco a poco en meros clientes de acceso a distintos servicios, redes sociales y aplicaciones Web... almacenamiento on-line, álbumes de fotos y convertidores de vídeo on-line; la evolución es imparable.

Volviendo al presente, más que una selección natural lo más probable es que ahora nazcan nuevas aleaciones tecnológicas para que ambas especies convivan. Más que probable, esto es lo que está pasando. Así es. Si bien asistimos a una transición bastante suave, las viejas costumbres no se destierran tan fácilmente. En la mente del usuario permanece la insistente llamada del Escritorio que, por otro lado, tiene sus incuestionables ventajas.

Ahora, nuevas plataformas, tecnologías y entornos de desarrollo nacen también con esta idea en mente: establecer una pasarela entre las aplicaciones Web y el Escritorio.

Algunas propuestas tratan de mejorar nuestra experiencia como usuario 2.0 recién estrenado, ofreciendo acceso off-line a las aplicaciones Web, aportando más capacidad o aprovechando simplemente las funcionalidades que ya ofrece el Escritorio. Es el caso de Google Gears. Por otro lado, otras pretenden llevarse directamente al Escritorio las aplicaciones Web. Es el caso de Adobe AIR o Microsoft Silverlight.

En el caso de Silverlight quiero aportar un pequeño matiz. Está pensado para impulsar el desarrollo de aplicaciones Web por medio de tecnologías conocidas de los desarrolladores de aplicaciones de Escritorio, como por ejemplo .NET. En cambio, Adobe quiere aprovechar los conocimientos de los desarrolladores de aplicaciones Web para exportarlas al Escritorio con ayuda de AIR.

En este apartado, hemos presentado la disyuntiva de la clasificación de las aplicaciones sobre qué plataformas son desplegadas: aplicaciones de escritorio y aplicaciones Web. Pero para nuestro caso en particular, ¿sobre qué plataformas deseamos implementar nuestra aplicación? ¿Cuál es mejor? Ambas tienen sus ventajas e inconvenientes como hemos presentado y que pueden resumirse en:

⁵ Webapp. Terminología que hace referencia a aplicaciones web.

Aplicaciones Web:

- Desarrollo barato, sencillo y rápido
- Acceso ubicuo, sin necesidad de distribución e, idealmente, con pocos requerimientos técnicos.
- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Permiten el desarrollo de comunidades que dan valor a las aplicaciones (software social).

Aplicaciones de escritorio:

- No necesita a Internet por lo que hacen que el acceso a estas aplicaciones esté al alcance de todos.
- La interactividad se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.
- Mejor interacción con el usuario. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.
- Nos evitamos el problema de encontrarnos diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones Web.

En conclusión, unas no son mejores que otras, todo depende de qué es lo que deseemos conseguir y para quienes esté destinada la aplicación. No es otro hecho fuera de lo común sino un punto más que debe ser analizado y estudiado en la fase de análisis de los requisitos en la fase previa al desarrollo de una aplicación.

2. INTRODUCCIÓN

3. PLANTEAMIENTO Y SOLUCIÓN AL PROBLEMA

Esta diversidad de medios de distribución y los dispositivos usados como receptores (televisores digitales, teléfonos móviles, PDAs, etc.) hace necesaria la conversión de formato de un mismo contenido para que pueda ser visualizado de forma eficiente y adecuada a las características de cada dispositivo. En estos tiempos en los que la globalización y la dependencia de la Red se acentúan cada vez más, un acceso flexible a la información, independientemente del lugar, del tiempo y de los dispositivos de comunicación usados, marcará la diferencia a la hora de aumentar la competitividad y productividad así como reducir los costes dedicados a la formación [6].

La implantación obligatoria de la TV digital y el consecuente “apagón analógico” (en 2010 en España y en Europa a partir de 2012), la telefonía móvil, los ordenadores de bolsillo, la automoción, etc., introducen un nuevo grado de complejidad al problema de interoperabilidad visto para diferentes sistemas de eLearning. No sólo han de buscarse soluciones para asegurar la interoperabilidad entre dispositivos, sino también soluciones que permitan el despliegue y descubrimiento de aplicaciones en diferentes dispositivos de forma transparente para el usuario, posibilitando la instalación y configuración de dichas aplicaciones en función de las necesidades del usuario y del contexto en que se encuentra. Nos encontramos frente a problemas de adaptación de contenidos y conciencia del contexto (*context awareness*).

La expresión “*context awareness*” o conciencia del contexto fue introducida por Schilit [7] en 1994 para referirse, expresamente en el campo informático de los sistemas distribuidos, al hecho de que los dispositivos puedan simular que perciben lo que ocurre a su alrededor, sienten y son capaces reaccionar en base al entorno que les rodea, simulando así esa conciencia acerca del contexto en el que se encuentran. Este contexto engloba algo más que simplemente la ubicación del usuario, pues puede haber factores variables como luces (encendidas o apagadas), el coste de las comunicaciones en una determinada franja horaria, (para facturar con la empresa más barata en cada momento), el nivel de ruido, las conexiones de red inalámbrica existentes y sus características, etc. Chen y Kotz [8] definen el contexto como “el conjunto de estados del entorno que determinan el comportamiento de una aplicación o en qué eventos se producen cambios interesantes para el usuario”.

Otro problema a resolver en la plataforma será el relativo a la interfaz gráfica. Será necesario abstraer las características comunes en la mayoría de interfaces gráficas para facilitar que pueda implementarse cualquier tipo de interfaz según requiera el dispositivo en que se encuentre, sin que ello afecte a los datos o la funcionalidad de los servicios registrados. Igualmente habrá que diferenciar qué aplicaciones cuentan con interfaz gráfica de usuario y cuáles no.

Resumiendo lo expuesto hasta ahora, presentamos a continuación algunos puntos que nos ayudarán a tener una mejor idea de la relevancia del proyecto:

- Desde el punto de vista económico, al utilizar sólo software libre, no sujeto a patentes, se facilitará su uso en centros de enseñanza.
- Desde el punto de vista social, si tenemos en cuenta la influencia que ha tenido la telefonía móvil en la sociedad en los últimos años, la integración de ésta en un proyecto destinado a la educación no parece muy lejos.
- Desde el punto de vista tecnológico, se trabajarán con un gran número de tecnologías, atacando problemas que aún no han sido resueltos, resultando en un proyecto de alto valor tecnológico e innovador.
- Desde el punto de vista de la enseñanza, las aplicaciones educativas serán de mayor calidad, acortando a su vez las distancias alumno-centro, alumno-profesor, aportando un nuevo campo de experimentación en cuanto a metodologías de

aprendizaje, pues se pretende una participación interactiva del alumno ante un medio en el que tradicionalmente el receptor de la información ha sido un actor pasivo. La creación de canales educativos personalizables constituye un objetivo interesante.

Como consecuencia, en este trabajo se aportan ideas para el desarrollo de una aplicación dinámica y adaptativa, de arquitectura orientada a servicios cuya funcionalidad principal es la distribución de recursos educativos y el seguimiento de la evolución en el aprendizaje del alumnado facilitando el rol del profesorado, lo que supone un enfoque novedoso, puesto que el enfoque más extendido actualmente sigue siendo las tradicionales aplicaciones Web que deben ser ejecutadas desde el navegador del usuario.

Se persigue definir una arquitectura orientada a servicios por medio de lenguajes orientados a objetos, abierta, que posibilite diferentes modos de uso para favorecer tanto el aprendizaje on-line como el aprendizaje off-line (usando dispositivos móviles como PDA's o teléfonos móviles) y facilitando la incorporación de plug-ins que permitan añadir la funcionalidad necesaria para lograr una plataforma capaz de adaptarse a la necesidades del usuario en pro de un aprendizaje personalizado.

Las arquitecturas orientadas a servicios promueven el diseño de aplicaciones como un conjunto de elementos poco acoplados que interaccionan entre sí, llamados servicios. Como definición de servicio podríamos decir que “es una entidad que puede ser usada por una persona, programa u otro servicio, que realiza funciones de cómputo, de almacenamiento o de comunicación con otros usuarios, dispositivos físicos o incluso con otros servicios” [10].

A continuación veremos pequeñas nociones de diversos conceptos que nos ayudará a entender y comprender lo expuesto en el desarrollo del proyecto.

3.1 Programación Orientada a Objetos

La programación orientada a objetos o POO (OOP según sus siglas en inglés, *Object Oriented Programming*) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Nació en los años 70 y su uso se popularizó a principios de la década de los años 90. En la actualidad existe variedad de lenguajes de programación que soportan la orientación a objetos.

Como definición de objeto podríamos decir que “es una entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa).”, definición que podemos ver representado en la Figura 5. Todas las propiedades y métodos comunes a los objetos se encapsulan o se agrupan en clases. Una clase es una plantilla o un prototipo para crear objetos, por eso se dice que los objetos son instancias de clases.

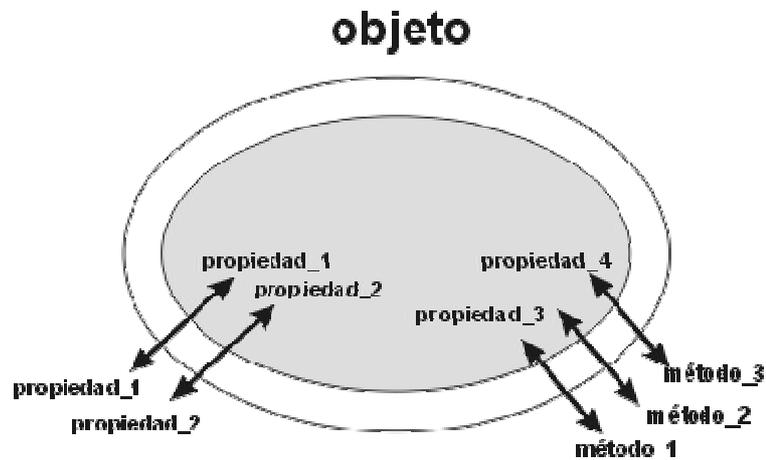


Figura 5. Representación de objeto

La orientación a objetos ha tomado por asalto y en forma legítima al mundo del software. Como medio para la generación de programas, tiene varias ventajas, entre las que podemos resaltar que fomenta una metodología basada en componentes para el desarrollo de software, de manera que primero se genera un sistema mediante un conjunto de objetos, luego podrá ampliar el sistema agregándole funcionalidad a los componentes que ya había generado o agregándole nuevos componentes y finalmente podrá volver a utilizar los objetos que generó para el sistema cuando cree uno nuevo, con lo cual reducirá sustancialmente el tiempo de desarrollo de un sistema.

La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas [9] [10].

En cuanto a la funcionalidad y la metodología interna, la programación orientada a objetos se fundamenta en la abstracción, modularidad, la encapsulación, la herencia y el polimorfismo, arquitectura representada gráficamente en la Figura 6. La encapsulación significa que en una clase se declaran los tipos de datos y el medio de manipularlos (sus métodos). La herencia supone crear clases derivadas de otras existentes, que heredan sus tipos y métodos y pueden contener otros nuevos. Si una nueva clase hereda propiedades de más de una antecesora, se denomina herencia múltiple; no todos los lenguajes orientados a objetos lo permiten. El polimorfismo facilita la programación de funciones o procedimientos que ejecutarán acciones que dependerán de los objetos sobre los que se apliquen; por ejemplo, aumentar el tamaño de un objeto, independientemente de su forma.

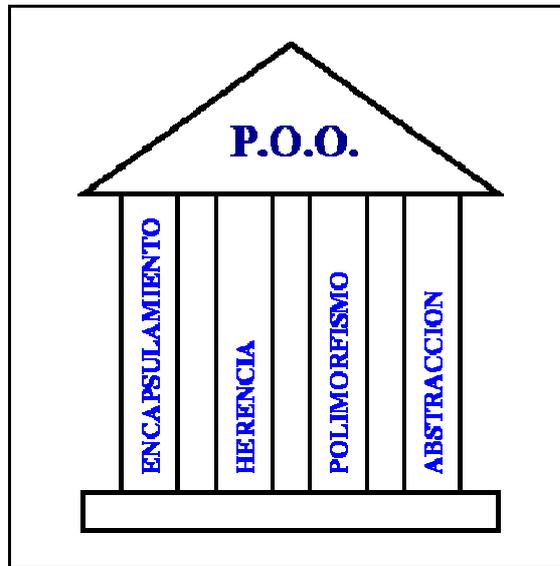


Figura 6. Estructura de lenguajes orientados a objetos

3.2 Arquitectura Orientada a Servicios

La falta de interoperabilidad, la dependencia a la arquitectura de trabajo, así como el lenguaje de programación ha llevado a la industria a considerar un nuevo modelo de computación distribuida de objetos, sin tener la dependencia de plataformas, modelos de desarrollo y lenguajes de programación usados y como una medida de solución nace SOAP (*Simple Object Access Protocol*) que es una estrategia de desarrollo de aplicaciones distribuidas usando tecnologías diversas adoptada por las diferentes organizaciones del mundo para resolver los problemas de falta de interoperabilidad entre las tecnologías anteriormente mencionadas, tomando como base protocolos ya establecidos y con gran aceptación en Internet, como HTML y XML.

SOAP no es más que un protocolo estándar que permite la comunicación y la interoperabilidad entre diversas aplicaciones Web desarrolladas bajo tecnologías diferentes, quizás sea más comprensible si observamos la Figura 7 que tenemos a continuación, donde el cliente de servicio sería la propia aplicación ejecutada por los usuarios, el registro de servicios sería la plataforma intermedia que pone en contacto las aplicaciones que hacen uso de esos servicios y los servicios, y el proveedor de servicios sería donde están alojados los servicios en sí como puede ser acciones en una base de datos que es nuestro caso en particular.



Figura 7. Arquitecturas Orientadas a Servicios

3. PLANTEAMIENTO Y SOLUCIÓN AL PROBLEMA

La evolución tecnológica y búsqueda de soluciones a la computación distribuida no es un problema reciente, es por ello que desde el año 1980 se dieron los inicios en este tema aunque los protocolos de comunicación no era objeto de interés de los desarrolladores en ese momento; realizar aplicaciones que dentro de una misma máquina se comunicaran entre sí, era suficiente.

Posteriormente en el año de 1990 alcanzaron popularidad objetos como COM (*Component Object Model*) introducido por Microsoft y CORBA (*Common Object Request Broker Architecture*) introducido por OMG (*Object Management Group*).

En general, COM y CORBA son modelos para escribir y encapsular código binario. Estos son componentes que pueden ser fácilmente llamados desde cualquier aplicación que soporte COM o CORBA. Sin embargo estos modelos no son fácilmente interoperables, de tal manera que COM puede solamente llamar a COM, y lo mismo ocurre con CORBA.

Conectar una maquina a otra se transformó en una prioridad cuando las redes locales se generalizaron, fue entonces que OMG estableció IIOP (*Internet Inter-ORB Protocol*) como el protocolo de comunicación para CORBA. Microsoft creo DCOM (*Distributed COM*), más tarde Sun Microsystems lanzo al mercado RMI (*Remote Method Invocation*).

Con estos protocolos se pueden llamar componentes que se encuentren en otras computadoras a través de la red. Estas llamadas se realizan bajo la forma de RPC (*Remote Procedure Call*). Es necesario aclarar que estos protocolos no son interoperables.

Para resolver estas dificultades de interoperabilidad se desarrolló SOAP, el cual se dio a conocer en 1999 y fue un resultado de desarrolladores de Microsoft Corp., DevelopMentor Inc. y Userland Software Inc. SOAP 1.1 fue liberada el 8 de Mayo del 2000, por W3C. Esto fue un buen signo de la industria para aceptar e implementar estándares basados en protocolos interoperables. Actualmente este protocolo está siendo desarrollado por el XML Protocol Working Group de la W3C

SOAP (Simple Object Access Protocol, Protocolo Simple de Acceso a Objetos) es un protocolo de mensajes entre computadores. SOAP especifica el formato de mensaje que accede e invoca a los objetos, más que un objeto en particular.

La idea detrás de SOAP es la misma que RPC. También define un protocolo para llamadas a métodos remotos, sin embargo SOAP contiene:

- Información adicional incluida en el documento XML (lenguaje de marcado extensible), que describe el contenido y como podría ser procesada.
- Definición de la especificación de algunas estructuras en XML, tales como arrays.
- El modelo descentralizado, esto significa que puede ser procesado por varios intermediarios.
- Características específicas para operaciones clásicas de RPC con parámetros in/out, etc.

Entre las ventajas de SOAP se tiene que:

- Es sencillo de implementar, probar y usar
- Atraviesa "firewalls" y routers, pues estos "piensan" que es una comunicación HTTP.

- Tanto los datos como las funciones se describen en XML, lo que permite que el protocolo no sólo sea más fácil de utilizar sino que también sea muy sólido.
- Es independiente del sistema operativo y procesador.
- Se puede utilizar tanto de forma anónima como con autenticación (nombre/clave)
- Facilidad para utilizar cualquier lenguaje: los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación y la plataforma.
- No se encuentra fuertemente asociado a ningún protocolo de transporte: la especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- No está atado a ninguna infraestructura de objeto distribuido: la mayoría de los sistemas de objetos distribuidos se pueden extender, y alguno de ellos admiten SOAP.
- Aprovecha los estándares existentes en la industria: los principales contribuyentes a la especificación SOAP evitaron, intencionalmente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes, los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.
- Permite la interoperabilidad entre múltiples entornos: SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en un PC puede comunicarse con una aplicación del back-end ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

Entre las desventajas de SOAP se tiene que:

- Las desventajas de la utilización de SOAP recaen en la dificultad para entender las especificaciones del protocolo, puesto que es un complejo esquema de codificación en el cual es necesario precisar que todos los mensajes se incluyan en un sobre, con el contenido del mensaje dentro de un elemento de cuerpo para que puedan ser entendidos por cada una de las aplicaciones Web que procesan el mensaje.
- SOAP convierte en opcionales elementos como encabezados y ofrece un amplio margen con respecto a lo que se puede incluir en el elemento de cuerpo y además cambia los nombres de métodos en etiquetas secundarias del cuerpo y los argumentos en etiquetas secundarias del nombre del método, lo que puede generar ciertos problemas de interoperabilidad.

3.3 RIA – Aplicaciones Ricas de Internet

Bien es verdad que las aplicaciones Web actuales ya no son lo que eran. Nos encontramos en un marco conocido y definido previamente como Web 2.0 donde la inmensa mayoría de estas aplicaciones pueden englobarse dentro de las conocidas aplicaciones ricas de Internet o RIA

Estas aplicaciones son independientes del sistema operativo y se ejecutan sin necesidad de una instalación previa. Una aplicación RIA es ejecutada en el interior de un navegador y se caracteriza y diferencia de las aplicaciones Web corrientes por presentar un comportamiento similar al de las de escritorio. El aspecto fundamental es que no se precisa una actualización de la página completa, que actúa como interfaz de usuario cada vez que se interactúa con la funcionalidad que ofrece el programa.

A pesar de la mejora que implica en el comportamiento de las aplicaciones Web, la disponibilidad de una interfaz de usuario más rica y ágil, no fundamentada exclusivamente en la transición entre páginas de un sitio, siguen existiendo limitaciones inherentes a su propia naturaleza.

Una aplicación Web es, por definición, un software alojado en la Red y que precisa de ésta para su correcto funcionamiento. En el momento en que carezcamos de una conexión a Internet, no resulta posible su uso, ya que no se puede acceder al servidor que aloja la interfaz de usuario y los datos empleados por la aplicación.

En la rápida mejora de escenarios tecnológicos para la creación de aplicaciones dinámicas de Internet (RIAs) se ha vuelto difícil para seleccionar la tecnología adecuada que se cumplan todos los requisitos de la industria, tales como la interacción del usuario, apariencia, rendimiento, desarrollo, mantenimiento, seguridad, compatibilidad con navegadores cruzados, plataformas cruzadas de compatibilidad e integración con una empresa existente y las aplicaciones heredadas.

El desafío que se plantea con vista al futuro en el proceso de desarrollo de software es comprimir el tiempo, esto es, reducir el tiempo entre el surgimiento de la necesidad y contar con la aplicación que la soporta. A partir de esto es que han surgido metodologías cuyo propósito es precisamente reducir ese tiempo y hacerse cargo de que el mundo sigue girando con su consecuente impacto en los sistemas de soporte al negocio.

3.4 Solución al Problema

Es por todos estos puntos, que se han ido resumido parcialmente hasta ahora, e intentando seguir en la línea del estándar que sigue la Universidad Politécnica de Valencia de presentar a los alumnos una plataforma vía Web donde pueden disponer de documentación, aplicaciones, datos personales... , por lo que inicialmente, a pesar de que en ocasiones no se pueda disponer de conexión, que se opta por desarrollar una aplicación RIA mediante tecnología Adobe Flex, apoyando esta decisión el atractivo gráfico que puede llegar a lograrse con esta plataforma al mismo tiempo que se reducen costes en el desarrollo de la aplicación, siendo Adobe Flex una plataforma Open Source⁶; pero tal y como se ha comentado anteriormente, por deseo del cliente y exigencias que se desea cubrir con la aplicación, se migrará a una aplicación RIA fuera de línea (off-line) adaptada al escritorio adaptando lo ya desarrollado al motor de ejecución de Adobe AIR.

⁶ Open Source (Código Abierto): aquellos que ponen a disposición de la gente el código fuente del programa para que pueda ser estudiado, modificado y mejorado.

Tal y como se ha comentado anteriormente, una aplicación Web cuenta con inconvenientes en temas como la seguridad, el acceso a documentos del sistema donde se esté ejecutando pero principalmente en cuestiones de conexiones a la red. Se plantean cuestiones tales como, ¿Qué pasaría si en momentos determinados los usuarios disponen de conexiones limitadas?, ¿y si se plantean procesos que deben ir resolviéndose en diferentes etapas distanciadas en el tiempo? ¿Dónde almacenar lo ya resuelto y que pueda estar accesible en momentos posteriores sin que el sistema nos pueda resultar un conflicto ya que programas que son ejecutados desde la red puedan considerarse como amenazas? Estas vicisitudes son resueltas si se implementa una aplicación de escritorio pero, ¿y si no se desea dejar a un lado la cuestión de accesibilidad remota que pueda tener la aplicación? Por eso se opta por desarrollar una aplicación RIA en el escritorio y tanteando el mercado de ofertas, contemplamos que para tal fin la mejor opción es apoyarnos en nuestro desarrollo en Adobe AIR.

Adobe AIR cada día gana más adeptos entre desarrolladores de software, además, no podemos menospreciar que estamos hablando de código gratuito eximiéndonos de pagar licencia. Pertenece a la compañía Adobe Systems Incorporated, empresa de software con sede en San José (California, USA) fundada en diciembre de 1982 por John Warnock y Charles Geschke. Destaca en el mundo del software por sus programas de edición de páginas Web, vídeo e imagen digital, y como quien dice, veintiocho años en este mundo, aunque Adobe AIR sea una “tecnología” reciente, nos transmite confianza y seguridad.

Tras haber decidido el tipo de aplicación que deseamos llevar a cabo, queda una cuestión principal por resolver ¿Cómo gestionar todo esto? ¿Cómo dejar constancia de todo proceso que se incorpore en la aplicación y cuáles ha de ser sus fases? ¿Cómo identificar al sistema el grado de responsabilidad de cada usuario que ejecute la aplicación?, ¿o si un usuario puede o no trabajar con la aplicación? ... la respuesta es obvia, no nos queda más opción que ayudarnos de una base de datos.

Es por esto que nos encontramos con dos líneas de actuación, en las cuales la una sin la otra no tienen sentido: una base de datos donde se almacenan todos los datos, los cuales analizaremos a continuación, y una aplicación cuyas funciones estarán recogidas de dicha base.

3. PLANTEAMIENTO Y SOLUCIÓN AL PROBLEMA

4. LINEAS DE ACTUACIÓN

4. LINEAS DE ACTUACIÓN

Puestos anteriormente en antecedentes hagamos una síntesis de aquello que se nos pide y a tener en cuenta:

- Se desea tener una aplicación que se empleará para las resoluciones de distintos tipos de procesos los cuales serán definidos por los gestores responsables de estos procesos.
- Cada usuario que acceda a dicha aplicación tendrá un perfil propio según se hayan otorgado unos permisos u otros. Se deberán contemplar dos perfiles bien definidos y distinguidos: unos gestores/administradores que definirán, modificarán y gestionarán los tipos y procesos a resolver y unos usuarios finales que serán quienes tendrán la posibilidad de poder responder a los procesos anteriores. El primer perfil definido tendrán la opción, a su vez, de otorgar permisos a los segundos individualmente para acceder a unos procesos u otros.
- La aplicación debe ser dinámica, es decir, podrá cambiar permisos y procesos a elección de los gestores cambiando la interfaz gráfica de los usuarios finales cuando accedan a la aplicación según permisos y procesos otorgados.
- Tiene que moverse en un marco actual. Es decir, nos encontramos en un punto en el que la tecnología avanza vertiginosamente en el mundo informático tanto a nivel visual como de ejecución por lo que no podrá descuidarse ninguno de estos dos temas.
- Debe ser posible el acceso remoto para que así los usuarios tengan la libertad de poder trabajar desde donde deseen y cuando deseen.
- A ser posible, se emplearán software de desarrollo Open Source, programa de código abierto los cuales implican el no pago de licencias para su disfrute con lo que se reducirá considerablemente el coste económico.

Tras estudiar detenidamente las distintas opciones que se desea que cumpla la herramienta a desarrollar, hay una característica que queda perfectamente definida: el dinamismo. Y es que lo que se está deseando no es la implementación de una aplicación que tan sólo cargue procesos sin más que luego, un usuario final, deberá resolver pues uno de los puntos claves a tener en cuenta es que no se tiene claro qué tipos de procesos se van a abarcar finalmente, ni que cantidad. Además no todos los usuarios deben poder visualizar todos los tipos de procesos por lo que debe quedar una aplicación abierta que vaya construyéndose y remodelando a medida en función de nuestras exigencias.

Visto el punto anterior y reflexionando sobre ello, queda definida una de las líneas de desarrollo y es que quizás no hay otra posibilidad de acción que guardar los datos referentes a los procesos que los gestores quieran definir en una base de datos a la que todos los usuarios, tanto gestores como usuarios finales, tengan acceso a consultar esa información.

Es por este punto anterior que ya inicialmente podemos marcar dos líneas a desarrollar:

1. Una base de datos donde recoger datos de los tipos, procesos y permisos de los usuarios para ir confeccionando la aplicación personalizada en función de la información recogida tras consultar estos datos.
2. Implementar una interfaz gráfica que sea capaz de ir confeccionándose a sí misma tras consultar una base de datos que le indicará con la información recogida de ésta mediante diversas consultas las opciones que se visualizarán.

Además de estas dos líneas obvias a contemplar, tenemos que considerar un tercer y último punto, que no por ello menos importante, y es que hay que tener en cuenta que necesitaremos un lenguaje del lado del servidor que nos haga de puente entre nuestra base de datos y la aplicación. Lenguaje a través del cual consultaremos, modificaremos, borraremos e introduciremos datos en la base de datos y le indicará a la aplicación los resultados de estas operaciones.

4. LINEAS DE ACTUACIÓN

5. BASE DE DATOS

Una vez seleccionada la plataforma, hay que tener en cuenta el marco contextual para el que se va a desarrollar esta aplicación. Se trata del entorno docente de la UPV, el cual ya cuenta con una base de datos, a partir de ahora haremos referencia con las siglas BD, donde ya se encuentran almacenados los datos tanto del personal docente como del personal estudiantil.

Para que la herramienta pueda integrarse con el entorno actual, hay que aprovechar esa BD y esos datos que ya tienen almacenados sobre los usuarios, por lo que en cuestión de la entidad en sí de “usuarios” no nos centraremos, tan sólo contemplaremos la manera de reflejar los diferentes permisos que a estos se les podrá otorgar, tal y como se ha comentado con anterioridad, para que puedan acceder a las diferentes opciones de las que contará la aplicación.

Aunque la base de datos que deberemos implementar para el desarrollo de la aplicación ocupa un papel importante para el buen funcionamiento de esta, tanto para su estructura como su eficiencia, el estudio del arte de éstas es un terreno diferente del tema que abarca este proyecto, pues no se trata de un proyecto de BD sino del desarrollo de una aplicación que facilite el trabajo al personal docente en la gestión de diferentes procesos que deben realizar a lo largo de su docencia. Pero es por su importancia dentro de éste, que en el transcurso temporal ha ocupado una cierta parte, si no mayoritaria sí importante, por lo que hagamos una pequeña mención sobre la ardua decisión de emplear una tecnología u otra y de la estructura implementada.

Hasta ahora hemos estado hablando de base de datos, pero ¿qué es una base de datos? Según Henry F. Korth autor del libro “Fundamentos de Bases de Datos”[11] se define una base de datos como una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permitan a los usuarios acceder y modificar esos datos.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo.

Uno de los propósitos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

En general, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Una de las técnicas para lograrlo consiste en diseñar esquemas que tengan una forma normal adecuada. La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes. Para determinar si un esquema de relaciones tiene una de las formas normales se requiere mayor información sobre la empresa del "mundo real" que se intenta modelar con la base de datos. La información adicional la proporciona una serie de limitantes que se denominan dependencias de los datos.

Con esta visión general del concepto de base de datos, podemos apreciar la importancia de porqué utilizar una base de datos para el desarrollo de este proyecto. Aunque existirán múltiples perfiles de usuarios en función de los múltiples tipos de procesos que se puedan definir, todos ellos tendrán una base común con unos pasos genéricos iguales y lo que no se pretende es reprogramar la aplicación cada vez que cambie un tipo de proceso, por ello, ¿porqué repetir información? ¿Cómo almacenar dicha información de manera no redundante?, ¿y para recuperar esta información de forma fácil y rápidamente? ¿Solución? Almacenar toda esta información en una base de datos.

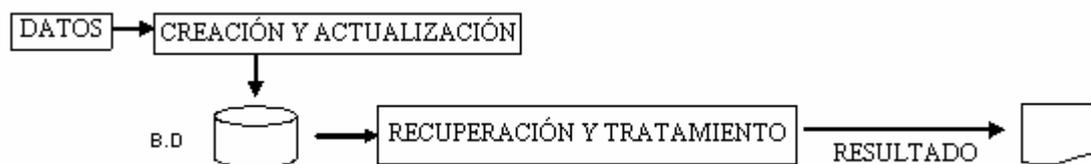


Figura 8 Diagrama del flujo de la información al utilizar una base de datos

Según este enfoque se logramos los siguientes cambios:

- **Independencia** de los datos respecto a los tratamientos y viceversa, lo que evita el importante esfuerzo que origina la reprogramación de las aplicaciones cuando se producen cambios en los datos.
- **Coherencia** de los resultados, con lo que se elimina el inconveniente de las divergencias en los resultados debidas a actualizaciones no simultaneas en todos los archivos.
- **Mejor disponibilidad** de los datos para el conjunto de los usuarios junto con una mayor transparencia respecto a la información existente.
- **Globalidad** ya que todos los usuarios tienen acceso a todos los datos.
- **Mayor valor informativo**, debido a que los distintos elementos están interrelacionados.
- **Documentación** de la información mejor y más normalizada, la cual está integrada con los datos.
- **Mayor eficiencia** en la recuperación, validación y entrada de los datos al sistema.

5.1 Diseño conceptual

Una vez recogidos todos los requerimientos, el siguiente paso para el diseño de la base de datos es crear un esquema conceptual de esta mediante un modelo de datos conceptual de alto nivel.

El esquema conceptual contiene una descripción detallada de los requerimientos de información de los usuarios, y contiene descripciones de los tipos de datos, relaciones entre ellos y restricciones.

Nosotros utilizaremos para el diseño de esquemas conceptuales el modelo E-R (entidad-relación), que describe los datos como entidades, vínculos (relaciones) y atributos.

5.2 Conceptos del Modelo Entidad-Relación

El modelo entidad-relación, a partir de ahora E-R, fue propuesto por Peter P. Chen entre los años 1976-1977. Posteriormente otros muchos autores han investigado y escrito sobre el modelo, proporcionando importantes aportaciones, por lo que realmente no se puede considerar que exista un único modelo E-R.

El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo E-R sólo incluía los conceptos de entidad (datos), relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido.

El objeto básico que se representa en el modelo E-R es la *entidad* que es "cualquier objeto del mundo real con existencia propia, sobre el cual queremos tener información en una base de datos". Una entidad puede ser un objeto con existencia física (una cierta persona, una casa, un empleado, un coche,...) o un objeto con existencia conceptual (una empresa, un puesto de trabajo, un curso universitario,...).

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil, es decir, que depende de sí misma.

El conjunto de entidades "*es la totalidad de las entidades del mismo tipo que comparten las mismas propiedades o atributos*". En los diagramas E-R se representan mediante un *rectángulo* y dentro del mismo se pone el nombre. Por ejemplo: CLIENTE, PROVEEDOR, ARTÍCULO, COCHE, etc. Debemos elegir nombres que comuniquen, hasta donde sea posible, el significado de cada entidad. Normalmente se utilizan nombres en singular.

La relación (interrelación) es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante *rombos* y su nombre aparece en el interior.

"Las entidades que están involucradas en una determinada relación" se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria; etc.

Una relación recursiva "*es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles*". El nombre de estos papeles es importante para determinar la función de cada participación.

La cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es obligatoria (total) si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es opcional (parcial). Las reglas que definen la cardinalidad de las relaciones son las reglas de negocio.

A veces surgen problemas cuando se está diseñado un esquema conceptual. Estos problemas, denominados "trampas", suelen producirse a causa de una mala interpretación en el significado de alguna relación, por lo que es importante comprobar que el esquema conceptual carece de dichas trampas. En general, para encontrar las trampas, hay que asegurarse de que se entiende completamente el significado de cada relación. Si no se entienden las relaciones se puede crear un esquema que no represente fielmente la realidad.

Una de las trampas que pueden encontrarse ocurre cuando el esquema representa una relación entre entidades, pero el camino entre algunas de sus ocurrencias es ambiguo. El modo de resolverla es reestructurando el esquema para representar la asociación entre las entidades correctamente.

Otra de las trampas sucede cuando un esquema sugiere la existencia de una relación entre entidades, pero el camino entre una y otra no existe para algunas de sus ocurrencias. En este caso, se produce una pérdida de información que se puede subsanar introduciendo la relación que sugería el esquema y que no estaba representada.

El atributo es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado “dominio”. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un atributo simple es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un atributo compuesto es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso. Un atributo compuesto se representa gráficamente mediante un óvalo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un atributo monovalente es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un atributo polivalente es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina multievaluados, y pueden tener un número máximo y un número mínimo de valores. La cardinalidad de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es (1,1).

Por último, los atributos pueden ser derivados. Un atributo derivado es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

Para poder hacer una definición de la jerarquía de generalización previamente debemos entender el concepto de generalización.

Una entidad E es una generalización de un grupo de entidades E\$_1\$, E\$_2\$,... E\$_n\$, si cada ocurrencia de cada una de esas entidades es también una ocurrencia de E. Todas las propiedades de la entidad genérica E son heredadas por las subentidades.

Cada jerarquía es total o parcial, y exclusiva o superpuesta.

- Una jerarquía es total si cada ocurrencia de la entidad genérica corresponde al menos con una ocurrencia de alguna subentidad.
- Es parcial si existe alguna ocurrencia de la entidad genérica que no corresponde con ninguna ocurrencia de ninguna subentidad.
- Una jerarquía es exclusiva si cada ocurrencia de la entidad genérica corresponde, como mucho, con una ocurrencia de una sola de las subentidades.
- Es superpuesta si existe alguna ocurrencia de la entidad genérica que corresponde a ocurrencias de dos o más subentidades diferentes.

Un subconjunto es un caso particular de generalización con una sola entidad como subentidad. Un subconjunto siempre es una jerarquía parcial y exclusiva.

En la Figura 9, podemos observar la representación gráfica de los conceptos que se han ido definiendo hasta ahora.

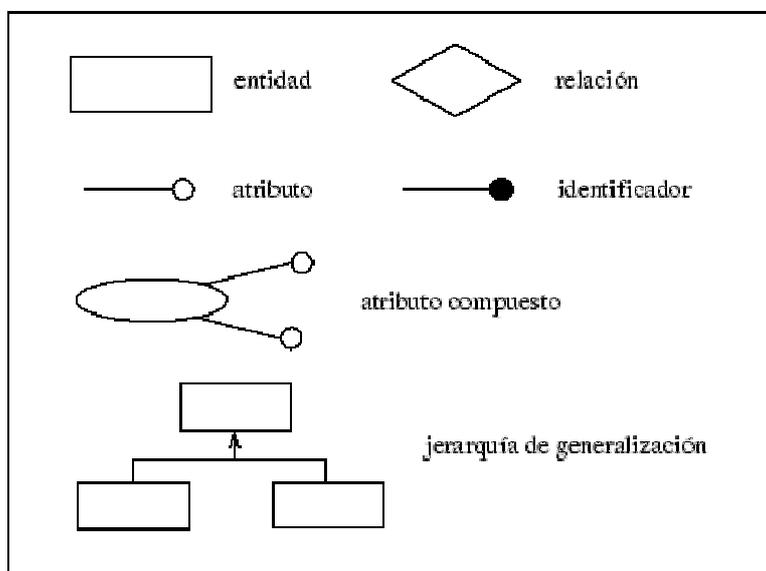


Figura 9. Representación gráfica de los conceptos del modelo entidad-relación extendido.

5.3 Diagrama Entidad-Relación

Los diagramas o modelos entidad-relación (denominado por su siglas, E-R “*Diagram Entity Relationship*”) es una herramienta para el modelado de datos de un sistema de información. Se caracteriza por utilizar una serie de símbolos y reglas para representar los datos y sus relaciones. Con este modelo conseguimos representar de manera grafica la estructura lógica de una base de datos. En la Figura 10 podemos ver un ejemplo de este y en el anexo III queda representado el modelo E-R de cómo sería para nuestro caso particular de nuestra BD.

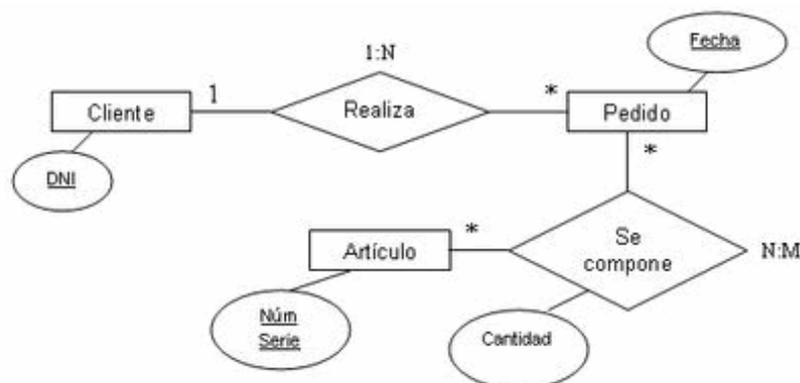


Figura 10. Ejemplo diagrama E-R

Se trata de un modelo bastante potente y a la vez bastante simple. El elemento principal de este modelo es la relación. Por lo que podemos decir que una base de datos relacional está compuesta por un conjunto de relaciones. La relación se representa mediante una tabla, esta tabla representa a lo que en el modelo entidad-relación llamábamos entidad. Esta tabla contiene los atributos (columnas) y las tuplas (filas).

- Atributo: se trata de cada una de las columnas de la tabla. Vienen definidas por un nombre y pueden contener un conjunto de valores.
- Tupla: se trata de cada una de las filas de la tabla. Es importante señalar que no se pueden tener tuplas duplicadas en una tabla.

Cada tupla de una tabla tiene que estar asociada a una clave única que permita identificarla. Una clave puede estar compuesta por uno o más atributos, tiene que ser única dentro de su tabla y no se puede descartar ningún atributo de la misma para identificar una fila.

Existen dos tipos de claves:

- Clave primaria (*Primary Key*): es el valor o conjunto de valores que identifican una fila dentro de una tabla. Nunca puede ser NULL. Un ejemplo claro de clave primaria sería el DNI, que es único para cada persona y no puede ser NULL.
- Clave ajena (*Foreign Key*): es el valor o valores de una tabla que corresponde con el valor de una clave primaria en otra tabla. Esta clave es la que representa las relaciones entre las tablas.

5.4 Estructura de la Base de Datos

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos.

Por consiguiente, una base de datos posee el siguiente orden jerárquico:

- Tablas
- Campos
- Registros
- Lenguaje SQL

El lenguaje SQL es el más universal en los sistemas de base de datos. Este lenguaje nos permite realizar consultas a nuestras bases de datos para mostrar, insertar, actualizar y borrar datos.

5.5 Sistemas de Base de Datos

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se componen de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

Entre los principales sistemas de base de datos que nos encontramos en el mercado podemos mencionar a Oracle, Microsoft, IBM y MySQL. Veamos a continuación una pequeña reseña de cada uno de ellos.

5.5.1 Oracle

El gigante Oracle, sigue siendo el líder en sistemas de bases de datos (los resultados de diferentes estudios de mercado pueden verse sintetizados en la Figura 11). Y hay que tener en cuenta que, aunque haya otros sistemas que le hagan sombra, las principales empresas utilizan Oracle. Sin duda alguna el buque insignia de la compañía Oracle Corporation es su Sistema Gestor de Base de Datos (SGBD).

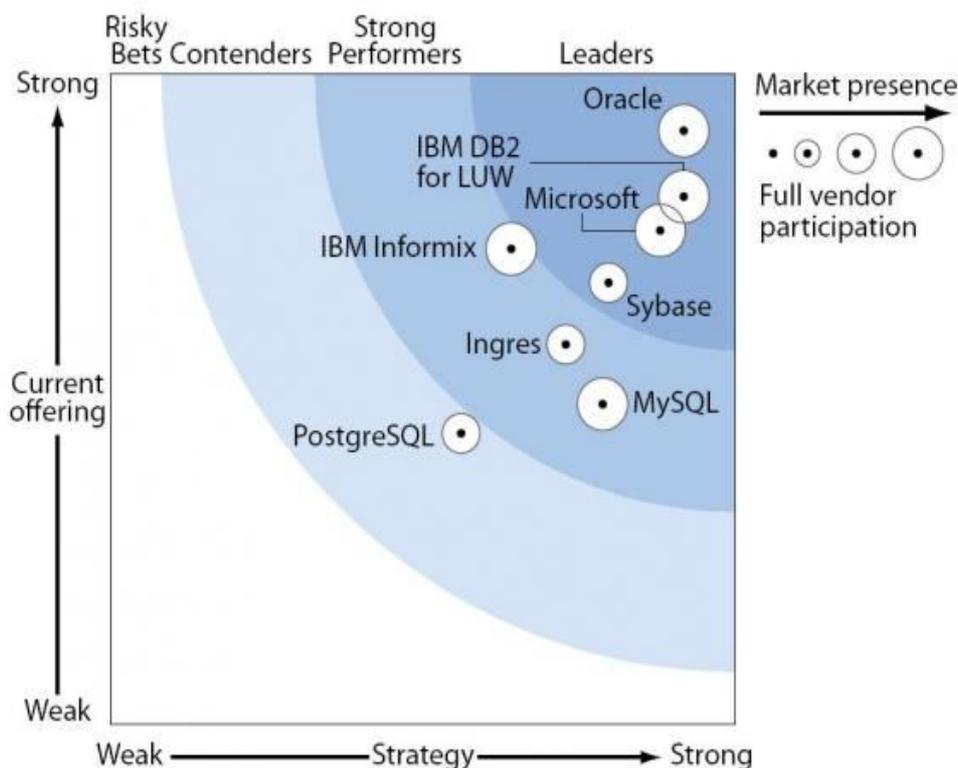


Figura 11. Estudio Forrester 2009 sobre Gestores de Base de Datos

Desde hace casi treinta años, Oracle viene desarrollando los productos más competitivos en el mercado de la base de datos, desde su posición dirigente, ha ido obligando a todos los

desarrolladores de bases de datos a cumplir una serie de criterios. Su funcionamiento esta optimizado para bases de datos relacionales Oracle pero también puede utilizarse con otros gestores como Access y SQL Server de Microsoft o DB2 de IBM y otros mediante el uso de controladores ODBC, su funcionamiento en estos otros gestores es más lento e inestable.

A pesar de las innumerables ventajas que nos ofrece esta opción, y que Oracle nos ofrece una base de datos robusta y una de las más confiables que existen en el mercado, queda desde un principio descartada pues implica el pago a Oracle por sus servicios y existen otras posibilidades interesantes en el mundo de la Open Source que, en ocasiones, pueden salir más rentables.

Los productos OpenSource más conocidos son: MySQL, PostgreSQL, MaxDB, Firebird, Ingres, MonetDB, LuciDb.

5.5.2 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD⁷.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Algunos de los límites de PostgreSQL se presentan en la Tabla 3:

| Limite | Valor |
|-------------------------------------|---|
| Máximo tamaño base de datos | Ilimitado (Dependiendo del sistema de almacenamiento) |
| Máximo tamaño de tabla | 32 TB |
| Máximo tamaño de fila | 1.6 TB |
| Máximo tamaño de campo | 1 GB |
| Máximo numero de filas por tabla | Ilimitado |
| Máximo numero de columnas por tabla | 250 - 1600 (dependiendo del tipo) |
| Máximo numero de índices por tabla | Ilimitado |

Tabla 3. Limitaciones postgresQL

⁷ licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Permite el uso del código fuente en software no libre.

5.5.3 Firebird

El sistema de administración de bases de datos relacional de código abierto Firebird cuenta con un rendimiento excelente y se escala de manera impresionante, desde un modelo integrado y monousuario, hasta desarrollos empresariales con múltiples bases de datos de más de 500 GB, con cientos de clientes simultáneos. Firebird soporta un número grande de plataformas de software y hardware: Windows, Linux, MacOS, HP-UX, AIX, Solaris y más. Funciona en x386, x64 y PowerPC, Sparc y otras plataformas de hardware, y cuenta con un mecanismo de fácil migración entre tales plataformas.

Una de las características claves de Firebird es su arquitectura multigeneracional, que permite el desarrollo y soporte de aplicaciones híbridas OLTP⁸ y OLAP⁹. Esto hace a Firebird capaz de servir simultáneamente como un almacén de datos analítico y operacional, porque las lecturas no bloquean a las escrituras cuando acceden a los mismos datos en la mayoría de las situaciones. Firebird soporta procedimientos almacenados, disparadores, eventos y funciones definidas por el usuario; tiene un gran soporte a SQL92. Está soportado por muchas opciones de conectividad a bases de datos. La alta compatibilidad con los estándares de la industria en muchos frentes hace a Firebird la opción obvia para desarrollar aplicaciones interoperables para ambientes homogéneos e híbridos. La mezcla de características: alto rendimiento, tamaño de distribución pequeño, escalabilidad suprema, instalación sencilla y silenciosa y 100% libre de regalías hacen a Firebird una opción muy atractiva para todos los tipos de desarrolladores de software y distribuidores. Es utilizada por aproximadamente 1 millón de desarrolladores de software en todo el mundo.

5.5.4 MaxDB

MaxDB es un sistema de administración de bases de datos adquirido por la compañía SAP¹⁰ para usarse como un repositorio de datos para las aplicaciones de SAP. Desde su adquisición fue nombrado SapDB y, posteriormente, renombrado a MaxDB por MySQL AB. Antes de SapDB, la base de datos pasó por otros nombres. Originalmente, se llamaba Adabas D, una base de datos pre-relacional.

MaxDB fue liberado desde la versión SapDB 7.2 y es de uso gratuito en ambientes no SAP. Como consecuencia de su adquisición por parte de MySQL AB, MaxDB está bajo licencia GPL¹¹ y las interfaces de programación están bajo licencia LGPL.

Desde la versión 7.5 en adelante SapDB fue adquirido por MySQL AB, la misma compañía que desarrolla la base de datos de código abierto MySQL, aunque el desarrollo todavía está a cargo de SAP AG.

MaxDB opera como un producto cliente/servidor. Fue desarrollado para cubrir las demandas de las instalaciones que requieren de un gran volumen de procesamiento de transacciones. Tanto el back up en línea como expansión de la base de datos están soportados.

⁸ On-Line Transactional Processing. Bases de datos orientadas al procesamiento de transacciones. Una transacción genera un proceso atómico (que debe ser validado con un commit, o invalidado con un rollback), y que puede involucrar operaciones de inserción, modificación y borrado de datos. El proceso transaccional es típico de las bases de datos operacionales.

⁹ On-Line Analytical Processing. Bases de datos orientadas al procesamiento analítico. Este análisis suele implicar, generalmente, la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil.

¹⁰ Siglas de Systems, Applications, Products in Data Processing. Es un sistema informático basado en módulos integrados.

¹¹ 60% del software licenciado como software libre emplea una licencia GPL. Posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia.

MaxDB corre en la modalidad de un sistema cliente/servidor. MySQL puede funcionar tanto cliente/servidor como también en un sistema embebido. Pero presenta el inconveniente de no funcionar en todas las plataformas soportadas por MySQL. Por ejemplo, MaxDB no corre sobre el sistema operativo de IBM OS/2. MaxDB emplea redes con protocolos de comunicación cliente/servidor mientras que MySQL puede emplearse con ambos, TCP/IP (con o sin encriptación SSL), sockets (bajo Unix - como sistema), o las llamadas pipes (bajo sistemas de la familia Windows NT).

5.5.5 MySQL

MySQL, el sistema de gestión de bases de datos relacionales SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB se trata de una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio. Cualquiera puede bajar el software MySQL desde Internet y usarlo sin pagar nada.

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje de consulta estructurado estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL.

Antes MySQL se consideraba como la opción ideal de sitios Web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL es una base de datos robusta que se la puede comparar con una base de datos comercial, es incluso más veloz en el procesamiento de las transacciones y dispone de un sistema de permisos elegante y potente, y ahora, además, incluye un motor de almacenamiento InnoDB compatible con ACID, además dispone de *store procedures*, *triggers*, vistas.

Son muchas las razones para escoger a MySQL como una solución de misión crítica para la administración de datos:

- **Coste:** es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- **Asistencia:** MySQL AB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- **Velocidad:** es mucho más rápido que la mayoría de sus rivales.
- **Funcionalidad:** dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados on-line, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- **Portabilidad:** se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Facilidad de uso:** resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

MySQL es muy popular por su velocidad de procesamiento, además el tipo de licencia que maneja es GPL (*General Public Licence*).

Es difícil encontrar documentación imparcial como para profundizar lo debido en todas (PostgreSQL, MySQL, Firebird, MaxDB, etc.) pero con esta leve visión que se ha presentado, hay varios aspectos que nos hace decantarnos por un servidor de base de datos en particular, nuestra elección: MySQL.

Esos puntos a favor pueden resumirse en que ya que no prevemos grandes transiciones de datos, MySQL nos ofrece grandes posibilidades de fiabilidad, velocidad, fácil manejo además de que debido a su gran distribución y popularidad en el mundo comercial nos facilita el trabajo pudiendo encontrar bastante información en cuestiones técnicas y uno de los puntos clave es que es de libre licencia, por lo que supone una reducción importante en el coste del proyecto, ventaja indiscutible que nos hace discriminar desde un principio a Oracle.

6. TECNOLOGÍA DE IMPLEMENTACIÓN

Tras la decisión del servidor de base de datos, queda otra importante, sino la más, de qué tecnología o framework emplear para llevar a cabo tal aplicación. Nuevamente, debido a la amplia oferta que encontramos en el mercado, esta decisión no resulta nada fácil.

Quizás la primera decisión que se debe llevar a cabo es sobre qué plataformas se va a ejecutar la aplicación, es decir, ¿se desea una aplicación que esté desplegada en la Web o se desea una aplicación de escritorio?, ¿se ejecutará sobre pc's de sobremesa o sobre otros dispositivos móviles?

Si esta misma pregunta nos la hubiéramos hecho hace unos cuantos años, quizás no hubiera surgido duda alguna, las ventajas de movilidad, atractivo y opciones que presentaban por entonces las aplicaciones de Internet sobrepasaban a las que podían desarrollarse sobre las aplicaciones de escritorio, las ya tan conocidas RIA Pero tras documentarnos, nos encontramos con plataformas de desarrollo (los llamados entornos de desarrollo integrado IDE) que pueden llegar a construir aplicaciones de escritorio con las mismas, o incluso mejores, que las ya nombradas RIA ofreciéndonos una mayor robustez y seguridad.

Tal y como se nos indicó desde un principio, el deseo de tener una aplicación móvil, es decir, que podamos ejecutar desde cualquier lugar en el que se encuentre el usuario y desde cualquier dispositivo móvil, es quizás por ello, que nos replanteamos en un principio el desarrollar una aplicación que sea ejecutada desde Internet, una aplicación Web.

Una de las principales ventajas que presentan las aplicaciones Web ante las aplicaciones de escritorio o consola es el hecho de que no dependen de ningún sistema operativo ni configuración de hardware específica; para su ejecución simplemente basta con teclear su dirección URL en cualquier navegador Web. De igual manera sus actualizaciones se hacen de forma muy sencilla, sin necesidad de hacer descargas, instalaciones o comprar físicamente el producto, bastará con ser actualizadas en el servidor y todo usuario que acceda a ella estará constantemente ejecutando la aplicación actualizada.

Esta flexibilidad ha sido uno de los principales motivos por los que cada vez son más utilizadas las RIA para una creciente diversidad de tareas, aunque al día de hoy todavía no cubren todos los aspectos, puesto que hay aplicaciones de consola más aptas para ciertas tareas al igual que hay aplicaciones de escritorio para otras, principalmente para tareas de bajo nivel que hacen interacción directa con algún hardware.

Por esta decisión inicial, comenzamos a ver las diferentes posibilidades que nos encontramos en el mercado que nos permita llevar a cabo nuestro objetivo sin todavía descartar el realizar una aplicación de escritorio o una aplicación Web. Si bien es verdad que el abanico es muy amplio, nos limitaremos a comentar aquellas que en su principio nos replanteamos.

Entre estas opciones podemos mencionar al grande en gestión de base de datos, Oracle pudiendo utilizar para su implementación por ejemplo la plataforma gratuita que ofrece la empresa como es el caso de JDeveloped, el archiconocido en aplicaciones RIA desarrollados en AJAX que tras ella cuenta con el poderoso JAVA, apoyándonos en el desarrollo de diferentes plug-ins¹² gratuitos de Eclipse, y el que va subiendo puestos cada día más en la implementación de RIA, Adobe Flex que gracias a su entorno de desarrollo que ofrece la compañía Adobe mediante el IDE(Entorno de Desarrollo Integrado) Flex Builder y su SDK (kit de desarrollo de software , del inglés software development kit) facilitan bastante el trabajo.

¹² Son programas que permiten visualizar e interpretar ficheros de texto, de video o sonido, de distinto formato. La característica principal de estos programas es que, una vez instalados, son transparentes para el usuario que los utiliza y están totalmente integrados en los sitios Web.

Podemos encontrarnos además, otras muchas herramientas para la creación de entornos RIA. Entre estas se puede mencionar las plataformas OpenLaszlo, Silverlight de Microsoft, JavaFX Script de Sun Microsystems, Bindows y Javascript.

6.1 Oracle Forms. Developer 2000

Oracle Forms, un componente de Oracle Fusion Middleware, familia de productos, que aportan una mayor agilidad, mejor toma de decisiones, y el coste reducido y el riesgo a diversos entornos de las tecnologías de la información hoy en día, es la tecnología para diseñar y construir aplicaciones empresariales de forma rápida y eficiente, aplicaciones basada en formularios, incluidas las propias de Oracle E-Business Suite de Oracle a apoyar y mejorar la tecnología de las formas por muchos años por venir. Oracle mantiene su compromiso con el desarrollo de esta tecnología y la liberación permanente como un componente de la plataforma Oracle. Este continuo compromiso con la tecnología de formularios le permite aprovechar su inversión existente al permitir el mejoramiento y la integración de Oracle Forms para aprovechar las tecnologías web y arquitecturas orientadas a servicios (SOA).

Es una herramienta integrada para el desarrollo de aplicaciones visuales que precisen acceder a bases de datos desde plataformas Windows (aunque se están desarrollando también nuevas versiones para otros entornos como Linux).

Su funcionamiento esta optimizado para bases de datos relacionales Oracle pero también puede utilizarse con otros gestores como Acces y Sql Server de Microsoft o DB2 de IBM y otros mediante el uso de controladores ODBC, su funcionamiento en estos otros gestores es más lento e inestable.

Developer 2000 está compuesto de cuatro aplicaciones básicas, a saber:

- FORMS. Genera pantallas y trata dichas pantallas.
- REPORTS. Creación de informes sobre una tabla.
- GRAPHICS. Genera gráficos estadísticos partiendo de los valores contenidos en una tabla.
- BOOK. Utilidad para generar documentación sobre las aplicaciones.

Se trata de una herramienta cliente-servidor que permite modificar, crear y borrar datos en la BD mediante la interacción del lenguaje PL/SQL y la programación visual orientada a eventos.

Forms se compone de tres programas principales:

- Forms Designer (a partir de Forms 5.0 se denomina Forms Builder): crea las pantallas y da funcionalidad a las mismas.
- Forms Generate: convierte el fichero designer en un fichero pseudo ejecutable
- Forms Runform: ejecuta el pseudo ejecutable

Es importante saber que Forms no genera ficheros *.exe.

A pesar de ser ficheros independientes y de poder generarse por separado, nos es más cómodo realizar las tres fases desde Forms Designer, dado que este nos lo permite.

Designer genera tres tipos de ficheros principales sin los que una aplicación nunca podrá existir en Forms:

- *.fmb: incluye todas las partes que intervienen en el desarrollo, pantallas, procedimientos, funcionalidad, etc.
- *.fmt: contiene el fmb traducido a lenguaje C.
- *.fmx: pseudo ejecutable

Es considerada de 4ª Generación y permite construir aplicaciones transaccionales muy robustas. La versión 6 de este producto tiene una máquina virtual de Java para su presentación en Internet.

Oracle ofrece ventajas con respecto a sus competidores como puede ser: Un estándar de programación plenamente adaptado a las normas de SQL Oracle; un entorno de programación mejorado para facilitar el desarrollo y la posibilidad de desarrollar junto al servidor IAS (*Internet Application Server*) una solución para Internet; todo ello, siguiendo un estándar, y una normalización de la seguridad de acceso a datos.

Pese a estas ventajas, cabe mencionar, que peca en cuestiones de interfaz gráfica si hacemos comparaciones con otras tecnologías que se encuentran actualmente en el mercado haciéndolas más atractivas visualmente de cara al usuario.

Es por la necesidad de emplear una base de datos para el desarrollo de la aplicación por la que nos replanteamos esta opción, pero ¿de verdad necesitamos una herramienta tan potente en cuestiones de base de datos cuando nuestra aplicación no se basa en sí sobre la gestión de esta? ¿Estamos dispuestos a descuidar la interfaz gráfica por motivos de la robustez y seguridad en cuestiones trascendentales? Quizás no. Es verdad que no pretendemos introducir grandes novedades gráficas, pues la aplicación debe regirse dentro del marco de diseño que actualmente se encuentran las aplicaciones, software, páginas WEB y demás del mundo de la Universidad Politécnica de Valencia, pero no es por ello que tengamos que prescindir totalmente de estilos que sean atractivos para el usuario; así pues, esta opción la dejamos inicialmente a un lado.

Tras descartar esta opción de una aplicación de escritorio, nos centramos en las aplicaciones Web. Existen diversos estudios de mercado de los diferentes avances que podemos encontrarnos en este entorno y por ello no podemos menospreciar el insistente inclusión de las RIA por ello nos proponemos en desarrollar una aplicación dentro de éstas, pero, ¿Cuál es la mejor plataforma para desarrollar una aplicación RIA?

En la actualidad se está debatiendo una guerra por la supremacía de las RIAs, aún en pañales. Tenemos en el escenario varios actores: Flex, Ajax, OpenLaszlo, Silverlight y JavaX. Es posible que pudiera aparecer algún nuevo actor, pero dada la consolidación de estas plataformas, se hace difícil competir en este terreno, a no ser que incorpore una revolución en cuanto a innovación, tecnología o prestaciones.

RIA, como ya se ha comentado anteriormente, es una tecnología para crear interfaces de aplicaciones ricas para Internet. La gran cualidad de esta tecnología es que permite interactuar con datos con un servidor y actualizar las interfaces (pantallas, formularios, etc.) sin tener que re-dibujar todo nuevamente. Asimismo, incorporan nuevos elementos de interfaz que permiten una experiencia de usuario muy agradable y usable, añadiendo propiedades multimedia (imágenes, vídeo, sonido...).

Para evaluar qué plataforma es mejor, hay que considerar varios factores, principalmente las necesidades que uno tiene para la aplicación a desarrollar. En el presente análisis se van exponer qué cosas fuertes y flojas tiene cada una de estas plataformas sin entrar muy a fondo en ellas pues no es el objetivo del proyecto, simplemente ver una base de cada una de ellas que nos permita decidirnos por una u otra para nuestro objetivo.

6.2 Ajax

Ajax (*Asynchronous Javascript And XML*) no es una tecnología. Es realmente muchas tecnologías... uniéndose en poderosas nuevas formas, aprovecha la penetración y uso de Javascript¹³ en los navegadores Web. El casi 100% de los navegadores tiene esta capacidad, sin importar en qué sistema operativo se esté ejecutando. La pega es que los navegadores podrían tener alguna pequeña incompatibilidad en alguna característica de Javascript (supone un porcentaje ínfimo), especialmente entre Internet Explorer y Mozilla Firefox.

El código Ajax se integra perfectamente con el código HTML y CSS de las páginas Web, pudiendo, incluso incrustar otros elementos tales como películas Flash. El código no se compila, si no que el navegador lo analiza, lo interpreta y lo ejecuta. Esta característica lo puede hacer más lento que otras opciones (que veremos más adelante). Las últimas versiones de navegadores están optimizando el motor de javascript, alcanzando niveles de ejecución realmente rápidos (actualmente ganan esta carrera Opera, Chrome y Safari). Ajax usa el navegador como su tiempo de ejecución, y los navegadores intercambian una pequeña cantidad de datos con el servidor Web (por detrás de las escenas) y, por tanto elimina la necesidad de actualizar la página entera.

Ajax, en su esencia, puede resultar un poco artesano y arduo de programar, sobre todo si se desea cosas avanzadas en interfaces. Existen frameworks disponibles que facilitan esta tarea, tales como MooTools, Dojo, ExtJS, JQuery o Prototype. Asimismo, hay extensiones para Eclipse (uno de los IDE por excelencia para el desarrollo y depuración de aplicaciones Java) para desarrollar en Ajax que facilitan su implantación.

A favor tiene la adopción unánime de la comunidad de Internet, lo que le erige como un estándar abierto y universal.

6.3 OpenLaszlo

OpenLaszlo lleva ya diez años en este escenario de las RIA's. A favor tiene que es software libre 100%, lo que lo convierte en un buen candidato para los puritanos. Esta plataforma tiene una dependencia importante del servidor de aplicaciones Apache Tomcat y de la máquina virtual de Java, pues se trata, principalmente, de un servlet (programa que se ejecuta en un servidor) que interpreta el código y lo compila en tiempo de ejecución para devolver al navegador un Ajax o DHTML, o un Flash.

El lenguaje está muy estudiado para que sea interpretado igual en DHTML o en Flash (sinergias entre ambas plataformas), y también, en el caso de Ajax, corrige las posibles incompatibilidades entre navegadores.

¹³ Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos...

Está basado en un lenguaje de marcas XML muy sencillo de aprender, e incluso permite agregar scripts de JavaScript.

Las capacidades multimedia son muy limitadas, si bien puede reproducir algunos medios audiovisuales. También son limitadas las capacidades de interfaz (comparándolas con Flex), pero son sobradamente suficientes para cualquier aplicación de gestión.

La comunidad OpenLaszlo no es muy numerosa, y el número de recursos en la red no es tan numeroso como en las plataformas anteriores. En el sitio oficial de OpenLaszlo, se puede encontrar una buena documentación, así como otros recursos. Fuera de este sitio, es más difícil encontrar soluciones específicas.

6.4 Microsoft Silverlight

Silverlight es la apuesta de Microsoft a este nuevo mercado de oportunidades tecnológicas basadas en RIA[25]. Llegó tarde, cuando las plataformas anteriormente citadas ya estaban consagradas.

En este periplo, Microsoft trae una interfaz rica y con capacidades multimedia, para intentar cubrir lagunas en Ajax, precisamente, con Silverlight extiende sus capacidades de forma notable, haciendo más sencilla la programación gracias también a su entorno de desarrollo visual. La base de su programación es XAML y el acceso a los objetos esta dado por C# y Visual Basic (aunque la versión 1.0 trabajaba a partir de JavaScript). El XAML puede ser usado para marcar los gráficos vectoriales y las animaciones.

Si bien la idea es buena, y la productividad aumenta con esta tecnología, tiene por contra que los clientes han de instalarse un plug-in en su ordenador (como el de Flash), y que además de ser una tecnología que no es open source, tiene gran dependencia con otras tecnologías de la fábrica del gigante de Redmond, como el framework .NET. Asimismo, el plug-in de Silverlight sólo funciona para entornos Windows y MacOS, y que el entorno de desarrollo Visual Studio está únicamente para entornos Windows.

Si el entorno de la aplicación está basado en sistemas Microsoft, con servicios de servidor basados en .NET, Silverlight es el complemento idóneo, ya que se integra a la perfección sin problemas, además de añadir una dosis de productividad que se aprecia en las planificaciones y en la rentabilidad de los proyectos.

6.5 JavaFX

JavaFX llega bastante tarde a esta contienda, y viene con una propuesta interesante, aunque algo escasa en comparación con sus competidores, especialmente a lo que se refiere a capacidades multimedia (de momento no tiene capacidad para 3D y en cuestión de vídeos o audio está muy limitado).

JavaFX requiere de la instalación de la máquina virtual de Java, así como del módulo específico de JavaFX, en el equipo cliente (en donde está el navegador Web).

El IDE de JavaFX es interesante, pues es una extensión que se ha añadido a NetBeans, el IDE de Sun Microsystems.

Las interfaces de usuario están heredadas de las vetustas y anticuadas Swing, aunque han mejorado ciertas cosas. Parece que también han retomado los anticuados Applets (programas

Java para Internet y páginas HTML). En este aspecto, queda un poco humilde en relación a sus competidores.

La carga de cualquier aplicación JavaFX puede resultar desesperante. Del mismo modo que Flash, requiere que se cargue toda la aplicación al principio. En el caso de Flash es bastante rápido, y la espera no suele suponer un problema. En el caso de JavaFX recuerda mucho a la de los antiguos Applets, y es muy lenta.

Otra de las apuestas de JavaFX es el mercado de los dispositivos móviles, donde ya tiene bastante parte del pastel ganado, aunque no es lo mismo un framework ligero como Java ME, que un framework tan pesado como éste.

En cuanto a soporte, JavaFX está a la altura que nos tiene acostumbrados Sun Microsystems (ahora Oracle). El IDE NetBeans es bueno, hay mucha documentación y recursos documentales, así como comunidades en torno a esta tecnología.

6.6 GWT (Google Web Toolkit)

GWT es un framework de desarrollo de Google en Java de código abierto, que permite compilar esa aplicación y traducirla a Javascript y HTML. De esta forma nos permitirá escribir complejas aplicaciones que hagan uso de AJAX, pero sin tener la menor idea de AJAX.

Con GWT, se puede desarrollar y depurar aplicaciones AJAX usando el lenguaje Java en el entorno de desarrollo que se preferencia (me refiero al sistema operativo y a los IDEs). Cuando se ha finalizado el desarrollo de una aplicación (que se ha escrito en Java), GWT compila y traduce dicho programa a JavaScript y HTML compatible con cualquier navegador Web. Es muy popular por estos tiempos, ya que permite crear Webs dinámicas, y asíncronas, esto quiere decir que podemos realizar cambios en la página sin necesidad de que sea recargada.

Una aplicación GWT puede ser ejecutada en dos modos:

- Modo host (Hosted mode): La aplicación se ejecuta como código bytecode de Java dentro de la Máquina Virtual de Java (JVM). Este modo es el más usado para desarrollo, soportando el cambio de código en caliente y el depurado.
- Modo Web (Web mode): La aplicación se ejecuta como código Javascript y HTML puro, compilado a partir del código Java. Este modo se suele usar para el despliegue de la aplicación.

6.7 Adobe Flex

Adobe Flex aprovecha el paquete de implantación de Flash, con un concepto muy interesante y con un gran apoyo técnico por detrás. Los usuarios para poder disfrutar de aplicaciones desarrolladas en Flex deben disponer en sus dispositivos de Flash Player 7 o superior. El plug-in de Flash está instalado en cerca de un 95% de los navegadores, y al ser un lenguaje compilado (en lugar de interpretado, como Ajax) la ejecución es mucho más rápida, optimizando más los tiempos de la CPU y del navegador.

Utiliza un lenguaje basado en tags XML, denominado MXML. Asimismo, puede incrustar código en ActionScript, el lenguaje nativo de Flash. Este lenguaje no es tan universal como Ajax, y por ello, el número de desarrolladores es mucho menor, más especializado. Detrás de

Flex hay una comunidad extensa, con soporte masivo de tutoriales, blogs, foros, libros, librerías, utilidades, etc.

Adobe Flex está muy pensado al diseño, proporcionando herramientas profesionales para crear interfaces de usuario espectaculares y cuidadas, con una velocidad sorprendente. Incluye también una plataforma de comunicación con servidor muy buena, para acceso a servicios y datos remotos, donde la complejidad se reduce al mínimo y la estabilidad y velocidad se aprovechan al máximo. Esta plataforma está especialmente indicada para desarrollos multimedia, debido a sus capacidades de reproducción de vídeo, de música o de animaciones 2D y 3D, así como de la multitud de efectos visuales y de animación que posee.

Esta plataforma tiene sus detractores, entre los que se encuentran aquellos que abogan por el software libre 100%. El plug-in de Flash no está liberado todavía (está liberado el lenguaje Flex), y éste es un factor importante a la hora de confiar la ejecución de un programa en un entorno que no sabes qué hace por detrás ni que puedes modificar. El plug-in de Flash, por contra, tiene a favor que no hay incompatibilidades entre los navegadores ni entre sistemas operativos.

Otro factor a tener muy en cuenta es que casi todos los recursos de un proyecto en Flex se compilan y se integran en un único archivo *.swf. Por un lado es bueno, porque la instalación es sencillísima, y el navegador sólo ha de cargar dicho fichero. Pero, por contra, si alguno de los recursos (como una imagen, un icono, una fuente o un archivo de audio o vídeo) se ha de actualizar, hay que volver a recompilar nuevamente el *.swf, y volverlo a instalar en el servidor.

Uno de los problemas que ven los adeptos a Ajax con respecto a Flex, es que toda la aplicación se carga desde el principio (fichero SWF), cargando innecesariamente la memoria del navegador y ejecutando procesos que no se requieren en ese momento. La carga inicial puede ser tediosa si la aplicación es de tamaño considerable.

La tecnología Flash también permite su ejecución en dispositivos móviles, con el inmenso mercado que hay detrás.

Otro problema añadido a la tecnología Flash es que al estar todo encapsulado en un archivo SWF, y compilado, no hay elementos visibles en la página para que los buscadores puedan indexar. De cara al éxito de búsquedas, no es la mejor opción.

Si hacemos una valoración general llegamos a la conclusión de que, si bien es verdad que son bastante conocidos interfaces de desarrollo de aplicaciones como Oracle Forms, el cual es una fuerte y robusta herramienta para trabajar con aplicaciones de base de datos, o diferentes framework con el apoyo de JAVA, tan extendido hoy en día en aplicaciones Web, no podemos hacer caso omiso a la información que poco a poco va dominando por el mundo de Internet en el avance de la aplicaciones, las llamadas aplicaciones RIA (Aplicaciones Ricas en Internet) y las herramientas que son más conocidas para facilitar su desarrollo aportadas por la compañía Adobe®.

Si bien no hay reglas fijas acerca de cuándo una tecnología es un mejor ajuste, es importante tener en cuenta algunas consideraciones clave como tamaño de la descarga y el apoyo navegador del cliente. Sin embargo, también es importante resaltar que una tecnología no es necesariamente mejor que el otro. A menudo, los requisitos de un proyecto específico de determinar la mejor solución.

En un intento de crear una aplicación robusta, dinámica pero que a la vez no se quede obsoleta en un periodo de tiempo relativamente corto, bien es sabido por todos la vertiginosa velocidad en la que se mueve el mundo de la informática, y debido a que no se trata de una aplicación que haga grandes operaciones en base de datos, se llega a la decisión de apostar por

esta nueva tecnología de aplicaciones RIA como es el caso de Adobe Flex. Esta decisión además, se ve apoyada y reforzada cuando comprobamos que la compañía Adobe ofrece el kit de desarrollo de Flex, denominado Flex Builder, de forma gratuita reduciendo por consiguiente todavía más si cabe el coste del proyecto.

Adobe Flex es una tecnología de Open Source, por lo que también nos exime de pagar ningún tipo de licencia. Esta tecnología comenzó en el 2004 por lo que nos encontramos en un momento en el que nos es fácil acceder a API's en el que poder apoyarnos facilitándonos la tarea de aprendizaje y uso.

Otro punto a favor de decidimos por Flex, es que está construido sobre un lenguaje de programación fuertemente orientada a objetos. Haciendo uso de Adobe Flex en el tiempo de ejecución (Flash Player), estás actuando frente a un 98% de penetración en el mercado de trabajo constantemente a través de diferentes navegadores, que finalmente pueden centrarse en la construcción de un gran RIA en lugar de preocuparse por su compatibilidad y el comportamiento que tendrá la aplicación en los diferentes navegadores como ocurriría en el caso de haber elegido JavaScript de AJAX.

A pesar de que en la Figura 12 podemos observar que el empleo de AJAX está bastante más extendido que Flex, recordamos que Flex es una tecnología más reciente por lo que todavía no está completamente difundida.

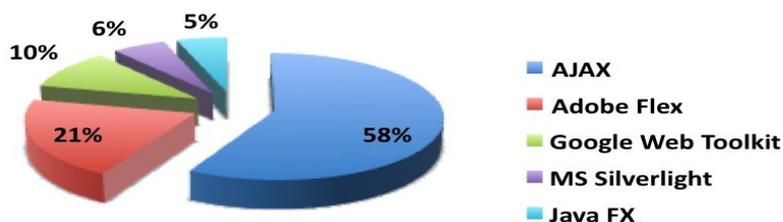


Figura 12. Porcentaje de usos de diferentes plataformas.

Nos decidimos por emplear Adobe Flex para la implementación de nuestra aplicación valorando que nos permite cubrir las especificaciones de los dos últimos puntos exigidos; aplicaciones que corren sobre navegadores WEB dentro de la tecnología WEB 2.0 y pertenecer al grupo de tecnologías Open Source; pero conforme vamos desarrollando el proyecto se nos plantean una serie de dudas como: ¿qué pasaría en el caso que el usuario acceda a la WEB con una conexión limitada?, ¿hasta qué punto se entorpecería su trabajo y hasta qué grado sería un inconveniente? ¿Cómo afecta al diseño la ejecución de la aplicación en los distintos navegadores existentes hasta el momento y los futuros?

Es por estas dudas, entre otras, por lo que llega un momento en el que se decide aprovechar la oferta que nos ofrece la compañía Adobe de poder diseñar una aplicación de escritorio pero con todas las ventajas de una aplicación WEB por medio de la implementación de aplicaciones con su tecnología AIR.

6.8 Adobe AIR

El tiempo de ejecución de Adobe® AIR® permite a los desarrolladores utilizar HTML, JavaScript, Adobe Flash® y ActionScript® para crear aplicaciones de Internet sofisticadas (RIA) que se implanten en el escritorio y para dispositivos móviles que se ejecutan como aplicaciones

clientes independientes sin las restricciones de un navegador. Adobe AIR y la plataforma de Adobe Flash ofrecen un entorno de desarrollo flexible y coherente para la entrega de aplicaciones en dispositivos como teléfonos inteligentes, libros inteligentes, tabletas, netbooks y PC y plataformas con una futura compatibilidad para plataformas adicionales, incluida Android para la cual acaban de lanzar al mercado Adobe AIR 2.

Gracias a Adobe AIR, los desarrolladores pueden utilizar tecnologías Web de eficacia demostrada para crear aplicaciones de Internet sofisticadas que se ejecutan fuera del explorador a través de distintos sistemas operativos, por lo que se elimina las diferentes maneras de actuar JavaScript en distintos navegadores. Se ofrece la ventaja de poder reutilizar del código y con un entorno de desarrollo visual intuitivo para desarrollar aplicaciones multiplataforma permitiendo crear aplicaciones envolventes y coherentes que ofrezcan experiencias del usuario interactivas y expresivas para diversas pantallas.

Las aplicaciones de AIR se ejecutan en diferentes sistemas operativos y se accede a ellas fácilmente utilizando un único archivo de instalación. Bien es sabido que las aplicaciones Web presentan el inconveniente, que por motivos de seguridad, en muchos casos no se tiene acceso a archivos locales o en el caso de poder tener acceso nos tenemos que arriesgar y poner en peligro nuestras plataformas saltándonos dicha seguridad, con AIR combina el poder de los recursos y los datos locales con el alcance de la Web sin llegar a poner en peligro nada, pues trabajamos desde nuestro escritorio.

Estas aplicaciones ofrecen el mismo comportamiento para el escritorio que para el navegador. Las aplicaciones de AIR creadas con Flex utilizan la máquina virtual de alto rendimiento y que se ejecuta en múltiples sistemas operativos del software Adobe Flash® Player 10.1. Las aplicaciones se instalan a la perfección desde el explorador o se descargan y se instalan como una aplicación de escritorio tradicional.

Está surgiendo una nueva clase de aplicaciones cuya tendencia está basada en la nube informática, las redes sociales y el surgimiento de una amplia gama de dispositivos conectado a Internet. Estas nuevas aplicaciones de trabajo a través de múltiples tipos de dispositivos pretenden proporcionar la más atractivas experiencias de usuario en cualquier pantalla. Por ejemplo, una aplicación puede ser ejecutada como una aplicación Web en la que los usuarios ejecuten en un navegador de vez en cuando, o en otro momento, como una aplicación de escritorio que se ejecuta de forma continua, como un teléfono móvil que los usuarios pueden ver mientras se espera el autobús.

Fundamentalmente, el desafío consiste en crear y mantener una o más versiones de los contenidos o aplicación que se ejecute correctamente en un rango de entornos de destino. Gestión de las variaciones en los objetivos que varían mucho, de tamaños de pantalla a la entrada de los distintos dispositivos, mecanismos para las características de rendimiento, etc. puede ser compleja y costosa para desarrollar y mantener. Este desafío para el desarrollo causado por la fragmentación sólo aumenta con el tiempo como el número de puntos de acceso y dispositivos continúa creciendo.

AIR aprovecha las funciones específicas para los dispositivos móviles que proporciona Flash® Player 10.1, y es más, está optimizado para proporcionar un alto rendimiento en las pantallas de los dispositivos móviles y diseñado para aprovechar las capacidades nativas de los dispositivos. Según un análisis de Strategy Analytics de enero de 2010, se espera que más de 250 millones de teléfonos inteligentes sean compatibles con Flash Player finales de 2012. Las plataformas de dispositivos móviles que serán compatibles con Flash Player incluyen Android, la plataforma BlackBerry®, Symbian® OS, Palm® webOS y Windows Mobile®.

En conclusión, viendo los intentos de Adobe AIR por conseguir la compatibilidad de las aplicaciones desarrolladas en diversos entornos, nos trasmite la confianza de poder desarrollar

una aplicación que sea adaptable al mayor número de dispositivos móviles que encontramos hoy en día en el mercado. Ahora que ya hemos decidido sobre qué plataforma trabajaremos, veamos un poco más a fondo de qué se trata y como es MXML y ActionScript.

6.8.1 MXML

Se trata de un lenguaje de programación jerárquico. Esto quiere decir que existe un nodo principal y de este nodo van a ir colgando nuevos nodos. A su vez, cada uno de los nodos descendientes del primero, pueden tener otros nodos descendientes. Un nodo es la unidad que se utiliza cuando se trabaja con jerarquías de elementos. Tal vez convenga verlo con un pequeño ejemplo gráfico como se muestra en la Figura 13.

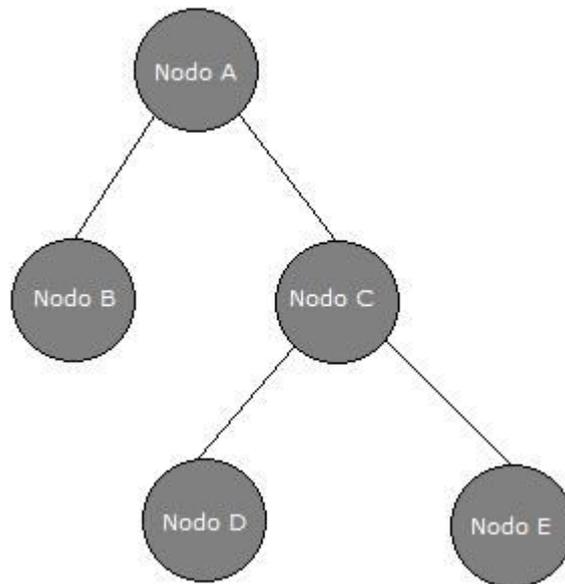


Figura 13. Jerarquía de nodos

En la Figura 13 se puede ver un árbol o grafo. Como se puede observar, los nodos (las unidades dentro del árbol) tienen cierta jerarquía. Veremos dentro de esta figura lo siguiente:

- El Nodo A tiene dos hijos: El Nodo B y el Nodo C.
- El Nodo A no tiene un nodo padre.
- El Nodo B no tiene nodos hijos.
- El Nodo B tiene como padre al Nodo A.
- El Nodo C tiene 2 hijos: El Nodo D y el Nodo E.
- El Nodo C tiene como padre al Nodo A.
- El Nodo D no tiene nodos hijos.
- El Nodo D tiene como padre al Nodo C.
- El Nodo E no tiene nodos hijos.
- El Nodo E tiene como padre al Nodo C.

Definiciones:

- Del Nodo A cuelgan todos los nodos. Esto quiere decir que el Nodo A es la Raíz del árbol. Un árbol jerárquico no puede tener 2 raíces.
- Los Nodos B, D y E no tienen hijos, por ello es que son llamados Hojas.

Volvamos a leer la definición de MXML a ver si nos queda más clara ahora.

Los nodos dentro de MXML, sintácticamente, se llaman tags. Un tag comienza con el símbolo “<” y finaliza con el signo “>”.

MXML se basa en XML y su acrónimo "Macromedia eXtensible Markup Language", que a primera vista se parece a HTML en la manera de describir los elementos, pero con una sintaxis menos ambigua, proporciona una gran variedad e inclusive permite extender etiquetas y crear sus propios componentes. Fue introducido por Macromedia en Marzo de 2004 cuando apareció Flex. Con MXML, al igual que con HTML, se describe el diseño de la aplicación y muy levemente las conexiones con las funciones ActionScript que le dan vida.

6.8.2 ActionScript

ActionScript es un lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones Web animadas realizadas en el entorno Adobe Flash, la tecnología de Adobe para añadir dinamismo al panorama Web. Fue lanzado con la versión 4 de Flash, y desde entonces hasta ahora, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamismo y versatilidad muy altos en la versión 11 (Adobe Flash CS5) de Flash.

ActionScript es un lenguaje de script, esto es, no requiere la creación de un programa completo para que la aplicación alcance los objetivos. El lenguaje está basado en especificaciones de estándar de industria ECMA-262, un estándar para Javascript, de ahí que ActionScript se parezca tanto a Javascript.

ECMAScript es una especificación de lenguaje de programación publicada por ECMA International (Ecma International es una organización internacional basada en membresías de estándares para la comunicación y la información. Adquirió el nombre Ecma International en 1994, cuando la European Computer Manufacturers Association (ECMA), cambió su nombre para expresar su alcance internacional. La organización fue fundada en 1961 para estandarizar los sistemas computarizados en Europa. La membresía está abierta a las empresas que producen, comercializan o desarrollan sistemas computacionales o de comunicación en Europa.).

El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO 16262.

ECMAScript define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C. Soporta algunas características de la programación orientada a objetos mediante objetos basados en prototipos y pseudo clases.

La versión más extendida actualmente es ActionScript 3.0, que significó una mejora en el manejo de programación orientada a objetos al ajustarse mejor al estándar ECMA-262 y es utilizada en las últimas versiones de Adobe Flash y Flex y en anteriores versiones de Flex. Desde la versión 2 de Flex viene incluido ActionScript 3, el cual mejora su rendimiento en comparación de sus antecesores, además de incluir nuevas características como el uso de expresiones regulares y nuevas formas de empaquetar las clases.

Flash está compuesto por objetos, con su respectiva ruta dentro del swf. Cada uno de estos en ActionScript pertenece a una clase (MovieClip, Botones, Vectores (Arrays), etc.), que contiene Propiedades y Métodos o Funciones.

Dentro de Flex o AIR, ActionScript es quien le da el dinamismo a mxml.

7. LENGUAJES DEL LADO DEL SERVIDOR

Como comentamos anteriormente, nos queda por analizar un último punto, no por ello menos importante y es la elección de un lenguaje de servidor que actúe de intermediario entre la aplicación y la base de datos.

Se necesita un script del lado del servidor para poder utilizar los formularios HTML de forma fiable. Por lo general, hay secuencias de comandos simples del lado del servidor para las presentaciones de procesamiento de formularios, como el envío de los datos en formato de texto por correo electrónico a una dirección especificada. Para procesamientos más avanzados, tales como la recogida de datos en un archivo o base de datos o recuperar información y enviarla de nuevo, o hacer algunos cálculos con los datos presentados, probablemente se tendrá que escribir un script de este tipo.

Un lenguaje de servidor es aquél que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página del lenguaje en particular. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

CGI (*Common Gateway Interface*) es una manera de interactuar los servidores Web y los programas del lado del servidor. CGI es simplemente una interfaz entre las formas de HTML y scripts de lado del servidor. Es completamente independiente del lenguaje de programación, sistema operativo y servidor Web. Actualmente es la técnica común de programación del lado del servidor y la mayoría también es apoyada por casi todos los servidores Web. Por otra parte, presenta la facilidad para que pueda hacer un script CGI para un servidor y luego distribuirla a ser ejecutado en cualquier otro servidor Web.

CGI sólo son programas ejecutables ordinarios (o programas interpretados por escrito, por ejemplo en Perl, Python o C++, siempre y cuando el servidor sepa cómo iniciar el programa), que puede utilizar casi cualquier lenguaje de programación que desee. Antes de que el programa CGI se inicie el servidor Web, establece una serie de variables de entorno que contiene la información que el servidor Web recibió en la solicitud. Ejemplos de esto son la dirección IP del cliente, los encabezados de la solicitud, etc.

La forma en que CGI devuelve su salida (encabezados HTTP y el documento HTML) para el servidor es extremadamente simple: se escribe en la salida estándar. En otras palabras, en un script en Perl o Python que acaba de utilizar la instrucción de impresión. En C se utiliza `printf` o equivalente algunos (C++ utiliza `cout <<`), mientras que Java se utiliza `System.out.println`.

Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Actualmente se encuentra un poco desfasado por diversas razones entre las que destaca la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta.

Pero CGI no es la única manera de hacer programas del lado del servidor, además éstos han sido muy criticados por su ineficiencia. Esta ineficiencia tiene algo de verdad si tenemos en cuenta que el programa CGI tiene que ser cargado en la memoria y ejecutado desde cero cada vez que se solicita.

Una alternativa mucho más rápida es la programación en el servidor de la propia API¹⁴. Es decir: hacer un programa que se convierte esencialmente en una parte del proceso del servidor y

¹⁴ (*Application Programming Interface* - Interfaz de Programación de Aplicaciones). Grupo de rutinas (conformando una interfaz) que provee un sistema operativo, una aplicación o una biblioteca, que definen cómo invocar desde un programa un servicio que éstos prestan. En otras palabras, una API representa un interfaz de comunicación entre

utilizar una API expuesta por el servidor. El problema con esta técnica es que la API es dependiente del servidor y que si se utiliza C/C++ (que es común) los errores de programación pueden bloquear todo el servidor.

La principal ventaja de programar del lado del servidor por medio de API es que es mucho más rápido ya que cuando la petición llega el programa ya está cargado en la memoria junto con todos los datos que necesita.

Hay módulos disponibles para los servidores como Apache, que le permiten hacer su servidor API de programación en Perl o Python, que elimina de forma efectiva el riesgo de errores de programación que puedan bloquear el servidor.

También existen diferentes lenguajes de programación (y no propietarios) y técnicas de los servidores Web diferentes. Algunas de las más conocidas son ASP, MetaHTML y PHP.

Vistas ambas opciones, optamos por realizar esta parte empleando API en vez de implementar un CGI. Bien es verdad que debido a su extensión histórica y robustez podemos encontrar una mayor cantidad de servidores que empleen CGI desarrollados en C++ tal y como podemos observar en la Figura 14, tomamos la decisión del empleo de API por el menor coste temporal que pueda llevarnos además de evitarnos por errores nuestros de programación poder llegar en algún momento a bloquear el servidor.

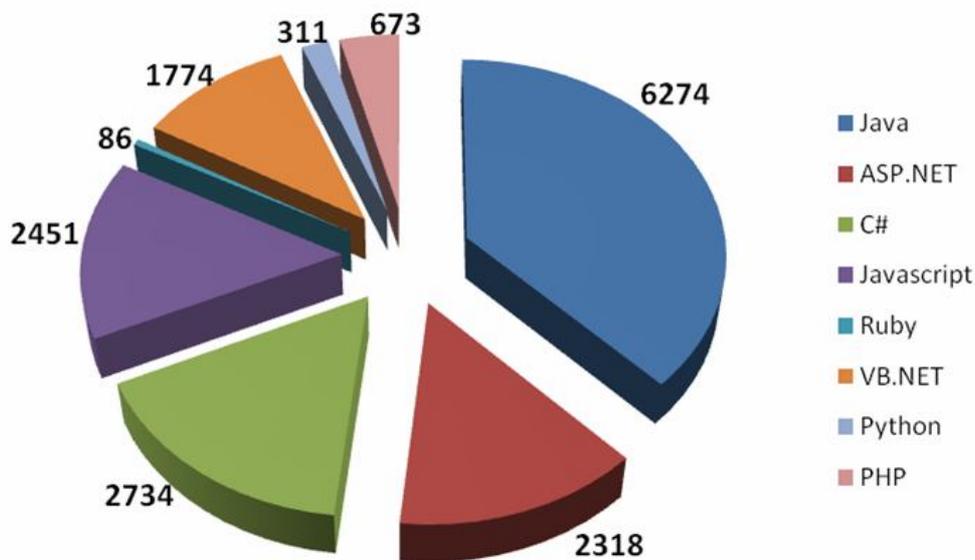


Figura 14. Lenguajes de servidor más representativos

Como más representativos y referenciados, según vamos recabando información en el empleo de lenguajes del lado del servidor por medio de API, podríamos nombrar a PHP, ASP y JAVA. Veamos a continuación algunas características de éstos.

7.1 Java (JSP)

JSP (*Java Server Pages*), como hemos comentado, es el lenguaje que habitualmente utiliza Java para la generación de páginas Web dinámicas. Podemos decir que JSP es un lenguaje

componentes software.

El software que provee la funcionalidad descrita por una API se dice que es una implementación del API. El API en sí mismo es abstracto, en donde especifica una interfaz y no da detalles de implementación.

de programación de páginas “del lado del servidor” que aporta Java. Las páginas JSP tienen el aspecto de las páginas HTML.

Las páginas de servidor Java (JSP) representan una tecnología cada vez más popular para construir aplicaciones Web dinámicas que pueden acceder a bases de datos y proporcionar una experiencia interactiva a los usuarios de un servicio Web.

Como características básicas del lenguaje de programación JSP a fin de introducción a dicho lenguaje podemos destacar el que presenta un aspecto de páginas HTML. Utilizan los llamados “scriptlets”, que no son más que trozos de código, normalmente en Java incrustados en las etiquetas HTML de una página Web. El contenido dinámico se obtiene, en esencia, gracias a la posibilidad de incrustar dentro de la página código Java de diferentes formas, aunque no es aconsejable porque produce páginas complicadas de mantener. Su objetivo final es separar la interfaz (presentación visual) de la implementación (lógica de ejecución). No necesita ser compilado.

JSP se puede ejecutar en los sistemas operativos y servidores Web más populares, como por ejemplo Apache, Netscape o Microsoft IIS. El API JSP se beneficia de la extendida comunidad JAVA existente aprovechando el desarrollo de códigos open source. Los componentes de JSP son reutilizables en distintas plataformas como UNIX y Windows. Las páginas JSP son compilados en Servlets por lo que actúan como una puerta a todos los servicios Java de Servidor y librerías Java para aplicaciones HTTP.

7.2 ASP

ASP (*Active Server Pages*) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).

El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT, aunque también se puede utilizar en un PC con windows 98 si instalamos un servidor denominado Personal Web Server. Incluso en sistemas Linux podemos utilizar las ASP si instalamos un componente denominado Chilisoft, aunque parece claro que será mejor trabajar sobre el servidor Web para el que está pensado: Internet Information Server.

Con las ASP podemos realizar muchos tipos de aplicaciones distintas. Nos permite acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor. También tenemos la posibilidad de comprar componentes ActiveX fabricados por distintas empresas de desarrollo de software que sirven para realizar múltiples usos, como el envío de correo, generar gráficas dinámicamente, y un largo listado de diversos recursos.

Actualmente se ha presentado la segunda versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro.

7.3 PHP

PHP es el acrónimo de *Hipertext Preprocesor*. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Cuenta con algunas ventajas como su gratuidad, independencia de

plataforma, rapidez y seguridad, etc. Cualquiera puede descargar a través de la página principal de PHP www.php.net y de manera gratuita, un módulo que hace que nuestro servidor Web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

PHP, en el caso de estar montado sobre un servidor Linux u Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP.

En cuestiones de seguridad, en este punto también es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

En conclusión, por compatibilidad con distintos sistemas y distintas bases de datos como MySQL, que nosotros hemos escogido, seguridad, rapidez, globalidad, por seguir en la línea de gratuidad, etc. no nos cabe duda en decantarnos por emplear PHP además de que en el momento que comenzamos a recopilar información encontramos ya diversos proyectos realizados con AIR y empleando PHP como lenguaje de servidor por lo que nos resuelve la duda de que no sería una idea descabellada y a la hora de encontrarnos con algún problema en el desarrollo de la aplicación podríamos encontrar ayuda en el mundo de los desarrolladores.

En cuanto a los costes y el apoyo, a partir de PHP es de código abierto y normalmente se ejecuta en una plataforma de código abierto, LAMP (Linux, Apache, MySQL, PHP), en este punto PHP gana sobre ASP.net y JAVA por proporcionar la solución más rentable y proporciona una gran cantidad de recursos y apoyo.

En cuanto a tiempo de implementación (una preocupación adicional por el coste), en promedio, para lograr algo con ASP.net se necesita un número de líneas de código mucho más elevado que para lograr lo mismo en PHP, por lo que el tiempo de implementación es más rápido usando PHP. En cuanto a JAVA, PHP presenta una línea de aprendizaje mucho más corta que JAVA reduciendo así el tiempo de desarrollo cuando se parte del desconocimiento de ambas tecnologías.

LAMP (Linux, Apache, MySQL y PHP) es mucho más popular entre las empresas de alojamiento, y los resultados de popularidad en un mes de alojamiento de menor coste para la lámpara de alojamiento en comparación con Windows de alojamiento.

ASP.net e IIS son gratis si se compra el sistema operativo Windows. Hay un coste sustancial para concesión de licencias de Windows Server de Microsoft, Microsoft SQL Server y actualizaciones futuras

LAMP es la plataforma más popular que el de Windows. La plataforma de base de Julio Netcraft 2010 Encuesta de servidor Web de 205.714.253 sitios, 112.945.968 (54,90%) se alojan en Apache y 53.217.620 (25,87%) están alojados en Windows, y el resto están alojados en otras plataformas.

Es sobre todo por el coste temporal, el punto decisivo que nos hace decantarnos por PHP ya que se parte del desconocimiento de ambas tecnologías reduciendo el tiempo empleado en formarnos sobre estas. Además, debido a su versatilidad en distintas plataformas, y ante no saber en un principio sobre qué servidor se va a alojar los recursos de la aplicación por lo que decidimos realizar esta parte del proyecto en PHP por la incertidumbre de si se tratará de un servidor Linux o Windows.

Es cierto, según muestra la Tabla 4, que existen lenguajes que gozan de más popularidad y con un mayor número de usuarios en la comunidad de desarrolladores, pero ¿qué coste temporal llevaría el poder conocer nociones básicas de JAVA para poder emplear este lenguaje en nuestro proyecto? Y ¿qué coste de complejidad supondría? Es por estas dos razones fundamentalmente, entre otras, por la que nos decantamos por PHP.

| Position Aug 2010 | Position Aug 2009 | Delta in Position | Programming Language | Ratings Jul 2010 | Delta Jul 2009 | Status |
|-------------------|-------------------|-------------------|----------------------|------------------|----------------|--------|
| 1 | 1 | = | Java | 17.994% | -1.53% | A |
| 2 | 2 | = | C | 17.866% | +0.65% | A |
| 3 | 3 | = | C++ | 9.658% | -0.84% | A |
| 4 | 4 | = | PHP | 9.180% | -0.21% | A |
| 5 | 5 | = | (Visual) Basic | 5.413% | -3.07% | A |
| 6 | 7 | ↑ | C# | 4.986% | +0.54% | A |
| 7 | 6 | ↓ | Python | 4.223% | -0.27% | A |
| 8 | 8 | = | Perl | 3.427% | -0.60% | A |
| 9 | 19 | ↑↑↑↑↑↑↑↑ | Objective-C | 3.150% | +2.54% | A |
| 10 | 11 | ↑ | Delphi | 2.428% | +0.09% | A |
| 11 | 9 | ↓↓ | JavaScript | 2.401% | -0.41% | A |
| 12 | 10 | ↓↓ | Ruby | 1.979% | -0.51% | A |
| 13 | 12 | ↓ | PL/SQL | 0.757% | -0.23% | A |
| 14 | 13 | ↓ | SAS | 0.715% | -0.10% | A |
| 15 | 20 | ↑↑↑↑ | MATLAB | 0.627% | +0.07% | B |
| 16 | 18 | ↑↑ | Lisp/Scheme/Clojure | 0.626% | 0.00% | B |
| 17 | 16 | ↓ | Pascal | 0.622% | -0.05% | B |
| 18 | 15 | ↓↓↓ | ABAP | 0.616% | -0.12% | B |
| 19 | 14 | ↓↓↓↓ | RPG (OS/400) | 0.606% | -0.15% | B |
| 20 | - | ↑↑↑↑↑↑↑↑ | Go | 0.603% | 0.00% | B |

Tabla 4. Lenguajes de programación - popularidad

8. SOFTWARE EMPLEADO

Tal y como se ha ido comentando en puntos anteriores, el proyecto podemos agruparlo en tres partes: la base de datos, la interfaz de usuario y los script de parte del servidor. Para estas partes se han desarrollado en MySQL, Adobe AIR desarrollado en Adobe Flex y PHP por los motivos que se han ido indicando en los puntos correspondientes a cada uno.

Para cada uno de estos módulos, exceptuando para la parte del servidor con PHP para el cual basta con un simple editor de texto, nos hemos apoyado en distintas herramientas de trabajo que nos han ido facilitando nuestra labor.

Para poder ir haciendo un desarrollo en local con sus correspondientes pruebas, para la implementación de una base de datos, nos hemos apoyado en el programa XAMPP, el cual nos crea un servidor local con los módulos de Apache, MySQL, PHP, Mercury entre otras herramientas. En el caso del desarrollo de la interfaz gráfica, aunque también podríamos haber empleado un editor de texto, hemos empleado la herramienta de desarrollo gratuita que nos ofrece Adobe como es el caso de Adobe Flex Builder y que al mismo tiempo nos transforma nuestros desarrollos en Flex a una aplicación AIR, el cual es nuestro objetivo. Pero veamos estas herramientas un poco más a fondo.

8.1 XAMPP

Para poder llevar a cabo el diseño de nuestra base de datos de una forma local que luego será traspasado a la base de datos de producción existente en la UPV, mantenerla y poder trabajar con ella, nos hemos apoyado en el programa XAMPP.

XAMPP es un software gratuito que te permite instalar de forma conjunta el servidor Web Apache, una base de datos MySQL, PHP y Perl entre otros módulos como OpenSSL y phpMyAdmin. Actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Para instalarlo sólo requiere descargarlo y extraerlo, con lo que nos resulta muy cómodo y rápido para comenzar enseguida a trabajar con él. Es una herramienta de desarrollo, que permite a los diseñadores de sitios Web y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet.

Tras su instalación, la cual resulta muy sencilla pues sólo hay que aceptar los pasos que se indican por defecto en el caso de querer instalar todos los módulos que contiene la aplicación, si ejecutamos el programa nos encontramos con el panel de control que puede observarse en la Figura 15.

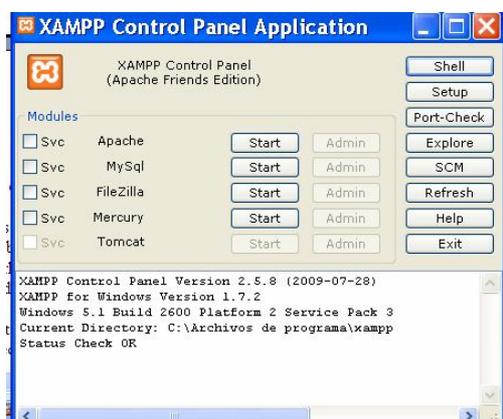


Figura 15. Pantalla de control de XAMPP

Para nuestro desarrollo, es bastante interesante la opción de phpMyAdmin. Esta es una herramienta escrita en PHP de libre distribución en PHP, creado por una comunidad sin ánimo de lucro, se encuentra disponible bajo la licencia GPL, con la intención de manejar la

administración de MySQL a través de una interfaz Web muy intuitiva. Actualmente puede realizar todas las funciones típicas de la base de datos: crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Es a partir de ella con la que crearemos nuestro esquema de base de datos e introduciremos datos de pruebas por medio de su interfaz Web como se aprecia en la Figura 16.

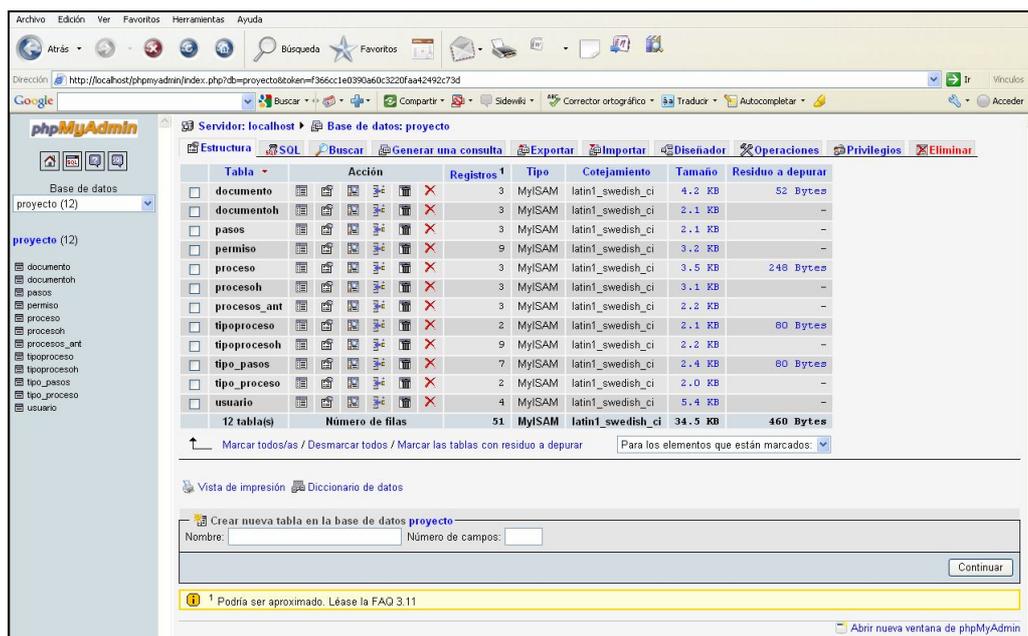


Figura 16. Interfaz aplicación phpMyAdmin

La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor Web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas.

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

8.3 Flex Builder

Flex Builder es el IDE (Entorno de Desarrollo Integrado) de Macromedia para desarrollar aplicaciones Flex basado en el entorno de desarrollo Eclipse tal y como se indica nada más ejecutar la aplicación y puede visualizarse en la Figura 17. En el momento en que nosotros comenzamos el desarrollo del proyecto, nos encontramos con la versión 3, sobre la cual trabajamos, pero actualmente ya se encuentra disponible la versión 4.

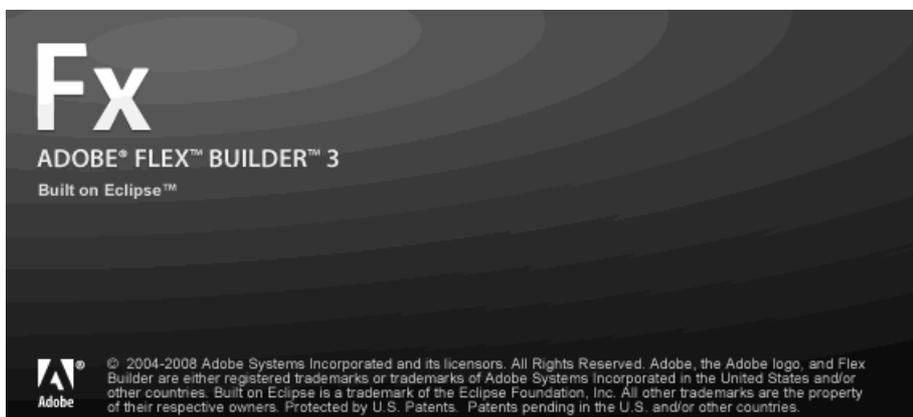


Figura 17. Imagen inicial al ejecutar la aplicación

Aunque para la creación de una aplicación en Flex tan sólo haría falta un editor de texto, Flex Builder permite a los diseñadores y desarrolladores disponer y crear aplicaciones Flex de una forma más productiva. Los diseñadores pueden crear con agilidad prototipos para las interfaces de aplicaciones, disponiendo interfaces de usuarios Flex de una manera rápida y fácil y conectándolas a fuentes de datos del sistema central. Los desarrolladores pueden entonces codificar y depurar aplicaciones Flex de una forma productiva usando el código sugerido y las herramientas de depuración.

Los requisitos del sistema donde vayan a instalarse Flex Builder son de bastante disponibilidad en los PCs de hoy en día, como son:

- Procesador Intel Pentium 3 de 600MHz o su equivalente
- Windows 2000, Windows XP
- 256 MB de RAM (se recomiendan 512 MB)
- resolución de pantalla de 1024 x 768
- 275 MB de espacio disponible en el disco

Flex Builder viene con una interfaz de diseño visual. Lo único que se tiene que hacer es colocar los contenedores y componentes en la aplicación para generar código MXML automáticamente. Vea el código y la disposición de su aplicación con sólo cambiarse entre la vista *Design* (diseño) como se muestra en la Figura 18 y *Code* (código), Figura 19, o si prefiere, puede mostrar ambas a la vez. Las dos vistas de la aplicación se mantienen sincronizadas, de manera que los efectos de los cambios hechos en una, se pueden ver en la otra.

Los componentes personalizados y los componentes de MXML que se creen, son plenamente compatibles con la vista de diseño. Al crear componentes personalizados estos son tomados igual que si fuera con cualquier otro componente de Flex.

Flex Builder se encarga de gestionar los archivos de su aplicación Flex para que el desarrollador se pueda dedicar al desarrollo de aplicaciones estupendas. Siempre que deseemos, los archivos del área de trabajo local pueden sincronizarse con cualquiera de los servidores Flex. Puede ponerse la aplicación que se encuentra desarrollando a prueba sin tener que preocuparse en ningún momento por poner y quitar archivos en el servidor, por qué archivos tiene que pasar y qué archivos pudieran tener dependencias.

Adobe® Flex™ Builder™ 3 proporciona las herramientas para crear aplicaciones de Adobe® AIR®. Flex Builder incluye los componentes de Flex para aplicaciones de Adobe AIR. El flujo de trabajo para el desarrollo de aplicaciones de AIR en Flex Builder es similar al que existe para el desarrollo de la mayoría de las aplicaciones basadas en Flex, proporcionando al final un ejecutable *.air, tal y como se aprecia en la Figura 20, que será el que tras ejecutarlo instale en nuestros dispositivos la aplicación desarrollada.

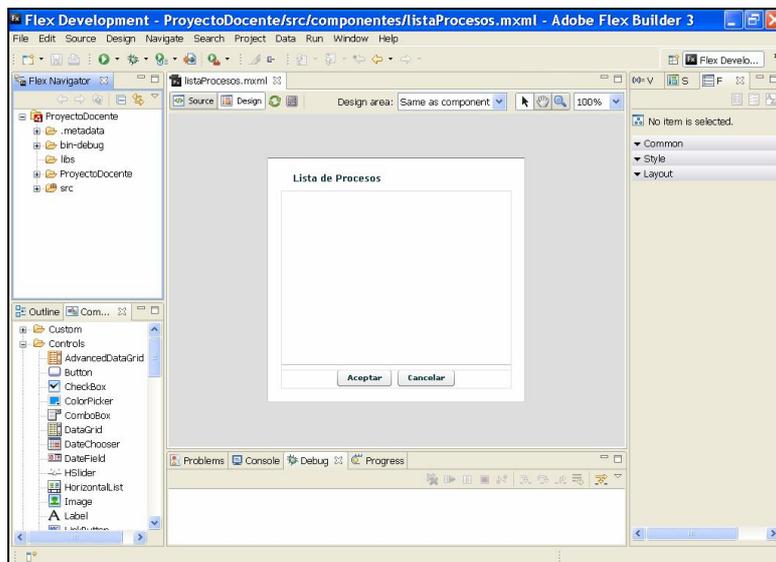


Figura 18. Vista de diseño de un componente en Flex Builder

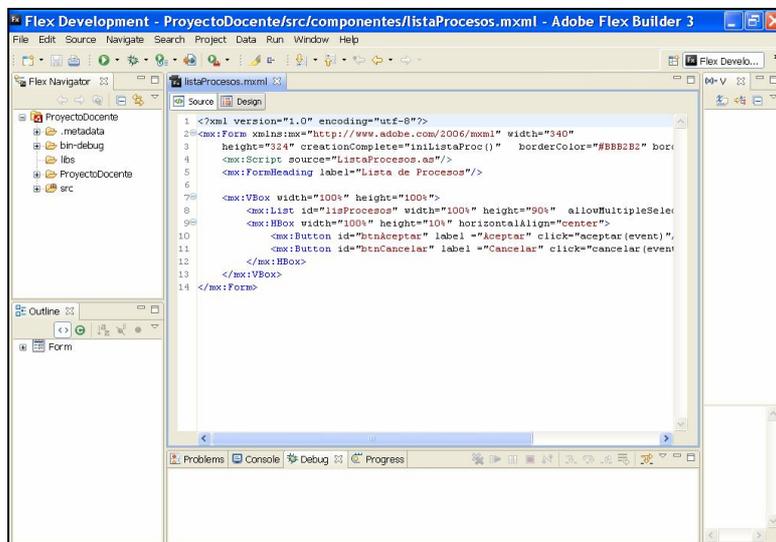


Figura 19. Vista de código de un componente en Flex Builder



Figura 20. Ejecutable de una aplicación desarrollada en AIR

Todas las aplicaciones de Adobe AIR deben estar firmadas para poder instalarse en otro sistema. Sin embargo, Flash permite crear archivos instaladores de Adobe AIR sin firmar para que la aplicación se pueda firmar más adelante. Estos archivos de instalador sin firmar de Adobe AIR reciben el nombre de paquete AIRI. Esta funcionalidad permite manejar situaciones en las que el certificado se encuentra en otro ordenador o en las que la firma se gestiona independientemente del desarrollo de la aplicación. Una firma digital proporciona la garantía de que el código de la aplicación no se ha modificado ni corrompido desde que lo creó el autor del software. Flex Builder te ofrece la posibilidad cuando estás exportando el proyecto a un ejecutable *.air de crear tu propia firma digital tal y como se aprecia en la Figura 21.

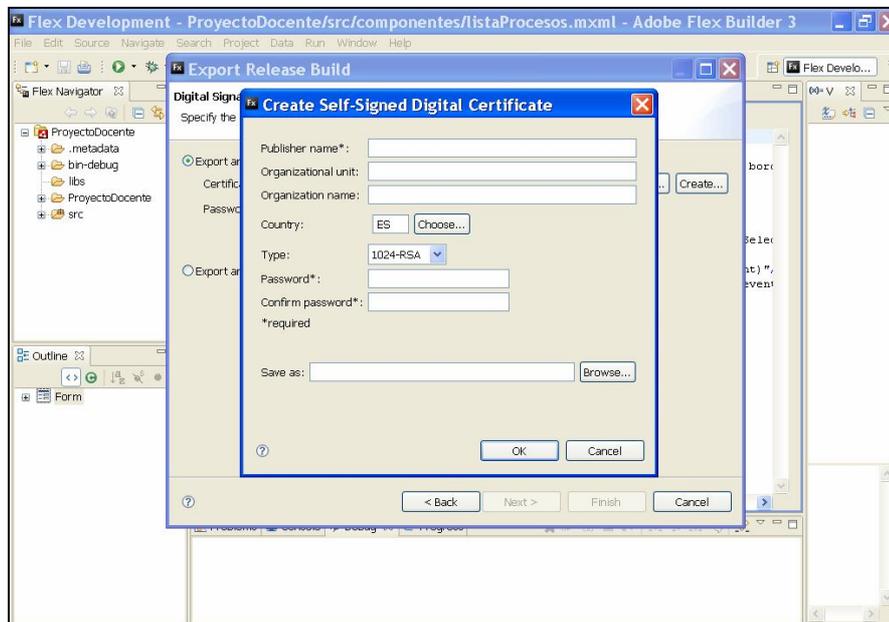


Figura 21. Pantalla de creación de firma digital

A la hora de configurar el directorio donde vamos a guardar los ficheros de nuestra aplicación, tenemos que tener en cuenta que ha de seleccionarse la carpeta “htdocs” creada a partir de la instalación de XAMPP que es el directorio de nuestro servidor local y es donde deben almacenarse nuestros archivos PHP para poder ser interpretados.

9. DESARROLLO DE LA APLICACIÓN

Tal y como hemos comentado en más de una ocasión, Flex es un lenguaje basado en XML que se compila dentro de aplicaciones flash. Podemos ver el proceso descrito gráficamente en la Figura 22 donde se aprecia el flujo básico de trabajo de una aplicación Flex.

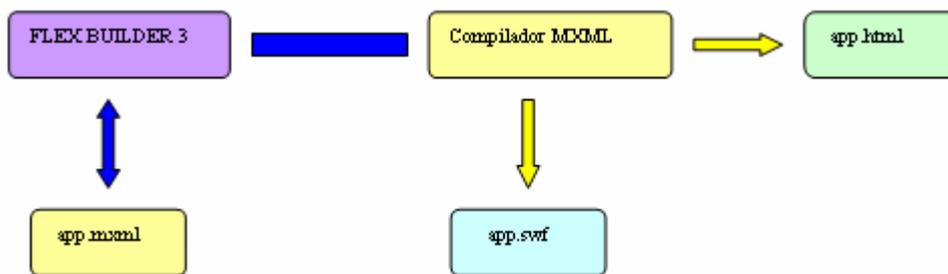


Figura 22. Flujo de trabajo de aplicaciones Flex

En toda aplicación Flex, en todo proyecto Flex mejor dicho, el archivo clave va a ser el archivo *.mxml; este archivo es realmente la aplicación Flex y va a ser a través de Flex Builder como se va a poder ir construyendo estos archivos.

Este archivo se ejecuta a través del compilador mxml que es el encargado de generar un archivo *.swf que es el que contiene la aplicación y un archivo *.html que es un archivo temporal que genera una página Web de prueba que contiene a su vez el archivo *.swf para que podamos testearlo mientras lo vamos creando.

Estos dos archivos, el *.swf y el *.html, se muestran en color distinto al archivo *.mxml porque son temporales, son reemplazados cada vez que se compile la aplicación.

Tras entender este esquema básico, podemos ampliarlo bastante más ya que Flex no sólo trabaja con archivos de aplicaciones sino que maneja proyectos formados por un grupo de componentes y aplicaciones Flex que también pueden tener otros contenidos. Por eso, el esquema anterior quedaría tal y como se muestra en la Figura 23.

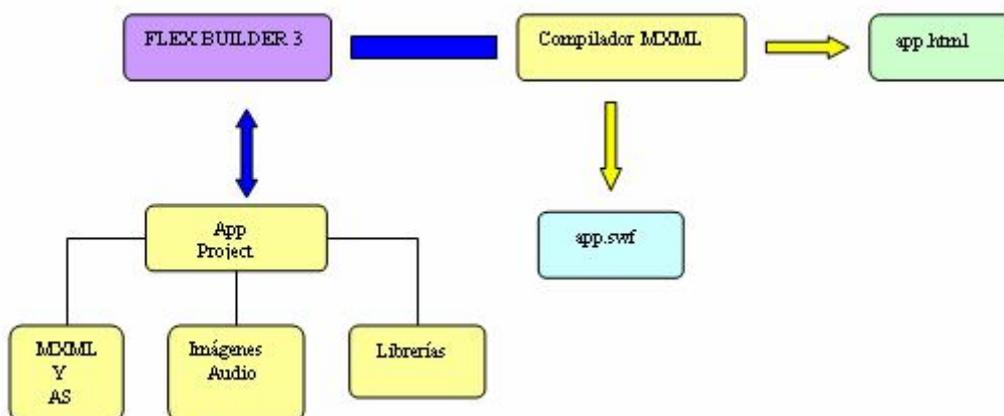


Figura 23. Flujo de trabajo extendido de una aplicación Flex

A continuación explicaremos el funcionamiento de la aplicación o mejor dicho, lo que queremos conseguir para poder llegar a una valoración de lo que necesitamos.

Para poder visualizar mejor este funcionamiento podemos esquematizarlo desde dos puntos de vista: desde el lado del personal docente que tendrán permisos de administrador y el que decidirá qué tipos y procesos desea controlar mediante el programa y desde el lado del alumnado el que tendrá que responder a lo que se pida en los procesos.

Como se ha comentado anteriormente, el personal docente podrá definir los diferentes tipos de procesos entre los que se agruparán los procesos pendientes los cuales también serán definidos por ellos. Además, tendrán la opción de un mantenimiento general que de forma transparente a ellos se realizará, valga la redundancia, un mantenimiento sobre los procesos que estén vigentes; este mantenimiento se explicará más detalladamente en puntos posteriores. Para poder entender esto mejor podemos apreciar en la Figura 24 el diagrama de flujos.

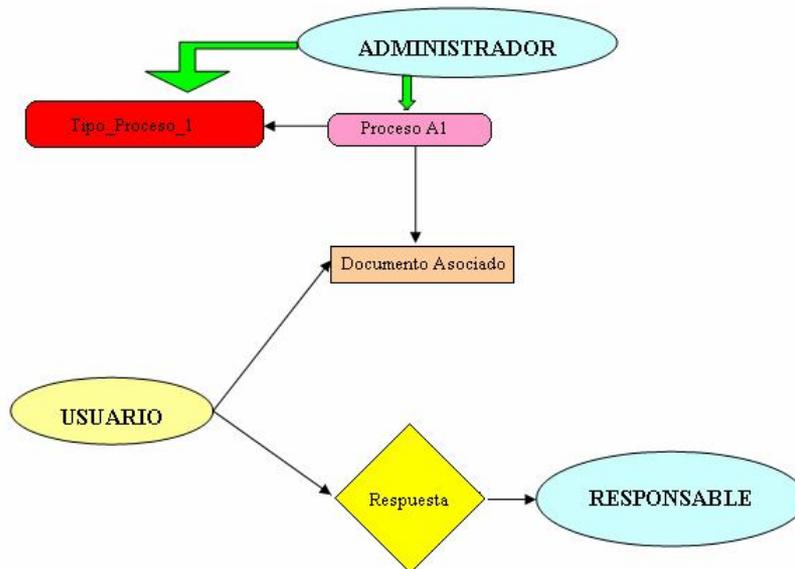


Figura 24. Esquema de funcionamiento de la aplicación

Los tipos de procesos y procesos (véase Figura 24) se dejan a elección de los administradores la envergadura que estos puedan alcanzar. Entiéndase por procesos prácticas de laboratorio, exámenes, trabajos, pedidos, etc. y tipos de procesos según quiera el usuario agruparlos de una forma u otra, bien sea por asignaturas (ej. sistemas digitales, cálculo matemático...), por alcance de procesos (ej. prácticas, exámenes...) u otro sistema que se le ocurra a éste. Debe tenerse en cuenta que cada proceso deberá contar de manera obligatoria de un documento con extensión *.pdf el cual contendrá lo que es en sí el proceso, sean los enunciados de problemas, prácticas, hojas de pedidos... y el número de preguntas del que consta el proceso pues los usuarios que tengan que resolver estos procesos visualizarán el documento *.pdf asociado a ellos y tendrá que responder de forma individualizada a tantas preguntas como se indicó en ellos.

Al mismo tiempo se les eximirá de tener que realizar el tedioso mantenimiento de los procesos uno a uno disponiendo para ello de un proceso que, de forma transparente, pasará al histórico aquellos procesos cuya fecha de baja sea posterior a la fecha del lanzamiento de este proceso, facilitándole así el trabajo al administrador.

Por parte de los usuarios, serán ellos los que visualizarán los documentos asociados a cada proceso vigente y de los tipos de procesos sobre los cuales se les ha otorgado permiso. Serán ellos quienes deben contestar o complimentar lo que se pida en el proceso cargado y una vez finalizado todos los pasos que se han definido en estos procesos, sus respuestas serán enviadas, por medio de la aplicación, vía correo electrónico al responsable del proceso.

Pero antes de ponernos a describir con más detalle cada funcionalidad de la que cuenta la aplicación y de lo que sería un manual de usuario, estudiemos un poco de las partes que cuenta el proyecto.

La aplicación va a estar formada por nueve bloques o funcionalidades que en su conjunto van a permitirnos que esta sea desarrollada.

1. Para empezar necesitamos una BD, MySQL. BD en la cual guardaremos la información referente a los usuarios, tipos de procesos y procesos, de la que se recuperará la información referente a éstos, permitiendo así a través de esta información que se muestren unas opciones u otras cuando el usuario se identifique.
2. En segundo lugar vamos a necesitar unas páginas PHP que van a interactuar con la BD y van a ser también las encargadas de generar el código XML que necesitamos. Estas páginas serán las mediadoras entre la aplicación y el servidor remoto.
3. En tercer lugar necesitamos un archivo de aplicación MXML que es el que va a contener la interfaz y que creamos desde Flex Builder.
4. En cuarto lugar vamos a necesitar también archivos ActionScript que es donde van a contenerse el código que va a trabajar con la interfaz MXML que es en el fondo la clave de funcionamiento de Flex.
5. Vamos también a necesitar definir unas clases ActionScript personalizadas para cargar y procesar el contenido XML inicial.
6. Otras clases ActionScript ocuparían un sexto lugar que nos permitan crear eventos personalizados.
7. Un séptimo lugar también lo ocupan otras clases ActionScript que nos permiten almacenar los detalles del usuario que ejecuta la aplicación y los detalles del proceso que se encuentra respondiendo y así en todo momento tener acceso a estos datos cargados inicialmente sin tener que volver a ejecutar consultas a BD.
8. En octavo lugar necesitaremos componentes personalizados para mostrar los detalles de los procesos y las respuestas de los usuarios, a través del cual se podrán definir nuevos procesos o modificar los ya existentes así como modificar o responder por primera vez las respuestas de los usuarios a estos procesos respectivamente.
9. Y por último necesitaremos una última clase personalizada que nos provea de funciones de conversiones de datos y de otras funciones que nos sean necesarias.

Veamos un poco más a fondo cada uno de los puntos citados anteriormente.

Para comenzar, necesitaremos una base de datos MySQL donde se almacenará todos los datos relacionados con la aplicación. Tal y como se ha comentado anteriormente en varios momentos, se quiere que la aplicación sea una aplicación dinámica, es decir, tenemos una aplicación cuyas posibilidades de trabajo dependerán de los tipos y los procesos que se encuentren disponible en cada momento y del usuario que ejecute la aplicación si tienen permisos o no, los cuales, al igual que los procesos, pueden e irán variando en el tiempo por lo que no podemos definir unas opciones estáticas para unos procesos y usuarios en concreto. Es por ello que necesitamos que esto quede reflejado en algún lugar al que la aplicación pueda acceder a consultar para mostrar unas opciones u otras; por ello es la razón por la que se opta en mantener esta información en una base de datos.

9.1 Base de Datos

Tras estudiar las diferentes posibilidades que se pueden presentar, mantener reuniones con el personal docente responsable y escuchar sus peticiones se ha optado por el siguiente desarrollo de la base de datos para tal fin tal como describimos a continuación:

- Al inicio de todo, debemos partir del supuesto que se debe preservar la aplicación para que no toda persona ajena al entorno docente (entiéndase personal docente al cuerpo profesoral, estudiantil, administrativo...) pueda hacer uso indecente de dicha aplicación. Por lo que se parte de una tabla central denominada “USUARIO” en la que quedarán reflejados los datos de todo el personal perteneciente a los grupos mencionados anteriormente. Para aprovechar la base de datos ya existente en entorno de la UPV, la cual es un Sistema Gestor de Base de Datos Oracle totalmente compatible con MySQL empleado en el proyecto, se intentará emplear los datos ya existentes.
- Además de la restricción de que el usuario que desee ejecutar la aplicación pertenezca al ámbito de la UPV, también se desea restringir el acceso a unas opciones u otras que ofrezca la aplicación. Un claro ejemplo es la distinción del cuerpo profesoral el que podría definir unos procesos existentes u otros y el cuerpo estudiantil el cual su función será el realizar las diferentes fases de los procesos existentes; por lo que se define otra tabla de permisos, “PERMISO”, en el que quedará reflejado los diferentes permisos de cada usuario sobre los diferentes tipos de procesos. Los diferentes permisos que se pueden otorgar son:
 - ‘0’ -> El usuario con este permiso sobre un tipo de proceso no tendrá acceso a los procesos de este tipo, no reflejándose así en la aplicación la línea de menú que hace referencia a estos procesos. Este tipo de permiso se debe asignar a la persona en cuestión cuando se quiera que no tenga constancia de dichos tipos de procesos.
 - ‘1’ -> Este permiso se otorgará a los usuarios que deben realizar los procesos del tipo al que se ha asignado. Es decir, el usuario podrá cargar y realizar los procesos de este tipo pero no podrá modificar nada de la gestión de los tipos y procesos.
 - ‘2’ -> Por último tenemos permisos de administrador. Este permiso lo tendrán asignado aquellos usuarios encargados de definir, modificar y eliminar tipos de procesos y procesos. Nada más que posea este permiso en algún tipo de proceso tendrá acceso al mantenimiento general de todos los tipos de procesos y procesos ya que en las opciones de administrador no existe restricción sobre los tipos de procesos, pudiendo visualizarlos todos, ya que se decidió que este tipo de permiso lo tendrán personal suficientemente responsable como para poder visualizarlo todo y no entorpecer el trabajo de compañeros.
- Existirán diferentes tipos de procesos, que será el propio personal encargado de definirlos junto a sus características en su momento. La agrupación de estos tipos de procesos pueden ser varios, por ejemplo por asignaturas, por envergadura (bien sea prácticas, exámenes...), esta elección corre a cargo del personal citado. Por esto se define una tabla, “TIPOPROCESO”, donde a vista del usuario solamente definirá un título que lo identifique.

- Los procesos en sí quedarán reflejados en una tabla denominada “PROCESO” y serán agrupados por tipos. El tipo de proceso que se le asigne debe existir en la tabla nombrada en el punto anterior. Estos procesos se caracterizarán por un periodo de vigencia contemplado entre una fecha de alta y baja. Se definirá un número de pasos (como por ejemplo pasos, preguntas...) que todo usuario que deba realizar dicho proceso deben completar íntegramente. Se le asignará un responsable al que se le enviarán los resultados de los usuarios que vayan completando los pasos de los procesos vía e-mail y tendrán un documento asociado el cual mostrará la descripción u objetivos de los distintos pasos que debe seguir el proceso.
- Los documentos asociados a cada proceso serán almacenados en el servidor con una jerarquía definida a nuestra conveniencia como:

directorio_aplicacion/documentos/tipo_proceso/documento_asociado.pdf.

Su referencia quedará almacenada en la tabla de documentos, ‘DOCUMENTO’, que tendrá su relación con el proceso mediante el código del proceso al que está asociado.

Toda esta estructura, procesos, documentos y demás podrán darse de alta y baja a elección de los usuarios responsables de tal efecto, por lo que para apoyar esta estructura y mantener un control de todo esto se define una estructura auxiliar para el mantenimiento de los datos que se den de baja formado por tablas históricas referentes a los procesos, documentos y tipos de procesos denominados ‘PROCESOH’, ‘DOCUMENTOH’ y ‘TIPOPROCESOH’ respectivamente.

Estas estructuras y relaciones pueden apreciarse mejor en los anexos I y II.

9.2 Interfaz de Usuario

En esta apartado no pretendemos mostrar ni explicar cada uno de los archivos *.mxml y ActionScript que se han creado para el desarrollo de la aplicación, pues se haría demasiado extenso, por ello nos centraremos en aspectos y archivos que consideramos interesantes para tal fin.

Esta parte es la encargada de la gestión de las acciones relacionadas al usuario, mientras que será del lado del servidor el encargado de la gestión de datos, consiguiendo así separar el rol del usuario del rol de la aplicación.

La aplicación, como cualquier aplicación desarrollada en AIR, cuenta con una pantalla principal que puede ser considerada como la pantalla padre a la que se irá adhiriendo el resto de componentes o ventanas hijos. Esta ventana principal, que en nuestro caso hemos denominado ProyectoDocente.mxml, cuenta con un archivo de respaldo ActionScript en el que se recogerán todas las acciones y eventos asociados a ésta.

La aplicación además cuenta con tres clases principales, cuyo código puede verse en los anexos IV, V y VI, y son: usuarios, tipos de procesos y procesos.

La clase usuarios cuenta con características como las que siguen:

- Nombre: Cadena de caracteres que contendrá el nombre y apellidos del usuario que ejecuta la aplicación.

- Dni: Cadena de caracteres que contendrá el DNI del usuario conectado y que hace referencia a la clave primaria de la tabla 'USUARIO', ya que aprovechamos la característica de único para cada persona.
- Email: Cadena de caracteres que contendrá la dirección electrónica del usuario que ejecuta la aplicación.
- Procesos: Array con el código de los tipos de procesos sobre los que se le han otorgado algún tipo de permiso.
- Permisos: Array con el código del tipo de permiso de cada uno de los procesos contenidos en el array de procesos anterior y relacionados en el mismo orden en el que se encuentran en el array de procesos. Es decir el permiso en la posición uno del array de permisos hace referencia al tipo de proceso indicado en la posición uno del array de procesos.
- Procesos Descripción: Array que contiene la descripción de los tipos de procesos que contiene el array procesos y con la misma relación con este array que se explicó para el array de permisos. Este array, fundamenta su existencia por motivos de definición de tipos de procesos. Es decir, serán los administradores quienes definan la descripción de los tipos de procesos, es por ello que estos datos son variables y se tendrán que recuperar cada vez que un usuario se conecte.
- esAdministrador: Booleano que tomará valor cierto (true) si se le ha otorgado permiso de administrador sobre algún tipo de proceso existente, en caso contrario tomará valor falso (false).

La clase de tipos de procesos cuenta con las características:

- Código Tipo: Es un número entero otorgado de forma aleatoria al insertar en base de datos y que no se permitirá modificar por los administradores y que recoge la clave primaria de la tabla 'TIPOPROCESO'.
- Descripción Tipo: Contiene la descripción que el administrador definió en el alta del tipo de proceso o en su modificación posterior. Esta descripción será la que aparecerá en la barra de herramientas de la aplicación para cargar o limpiar los procesos agrupados en este tipo de procesos tal y como se observa en la Figura 25



Figura 25. Proceso carga: tipo de proceso definido por el administrador

La clase proceso cuenta con las características de:

- Código Proceso: Número entero asignado de manera aleatoria al insertar un nuevo proceso en base de datos y que no podrá ser modificado. Hace referencia a la clave primaria de la tabla 'PROCESO'.
- Tipo: Número entero que hace referencia al código del tipo de proceso en el que será agrupado el proceso y bajo del cual colgará de dicho menú en la barra de herramientas. Es por ello que primero ha de existir el tipo de proceso y después definir un proceso de dicho tipo.
- Título: Título que se desea poner al proceso.
- Descripción: Breve descripción, como máximo 250 caracteres, donde se explique en qué consiste el proceso.
- Fecha de alta: Fecha en la que se da de alta el proceso. Toma el valor del sistema no dejando modificarlo.
- Fecha de baja: Fecha hasta la cual el proceso estará vigente en la aplicación. Hay que tener en cuenta que en el caso de ejecutar el procedimiento de mantenimiento de procesos automático, dará de baja aquellos procesos del tipo seleccionado cuya esta fecha de baja sea anterior al momento en el que se lanza el proceso.
- Número de preguntas: Número de preguntas que el usuario debe contestar para completar el proceso. Hay que tener en cuenta que ha de ser idéntico al número de preguntas del documento asociado.
- Documento: Documento asociado al proceso con extensión *.pdf en donde se muestran las preguntas que los usuarios deben contestar, documentación, información, etc.
- Responsable: Responsable del proceso al que se enviarán las soluciones de los usuarios a ese proceso.
- Adjuntos: Check que indicará si el proceso, justo en el momento antes de enviar las soluciones de los usuarios de este al responsable, permite enviar documentos adicionales a las soluciones que serán adjuntados con la solución de los usuarios en el mismo correo electrónico.

En cuestión del estilo, se ha definido una clase denominada TitleBackground.as, la cual tiene embebida el logotipo de la UPV y esta clase es cargada en la pantalla principal desde el momento en que se inicia la aplicación.

Se ha definido una barra de herramientas sobre la aplicación principal, definida temporalmente como ProyectoDocente, que se adaptará en función del usuario que ejecute la aplicación y de los permisos que este posea sobre los diferentes tipos de procesos definidos hasta ese momento. De esta forma conseguidos uno de los objetivos clave que se pretendía conseguir: dinamismo.

Existe una opción de dicha barra de herramientas que permite que todo aquél que instale la aplicación sobre sus dispositivos tenga acceso. Esta opción, bajo el identificativo de “Archivo”, presenta un submenú con dos opciones:

- Identificarse: como la propia definición indica, tras hacer clic sobre ella se abre una pantalla cargando el componente *identificación.mxml* donde el usuario tendrá que introducir su login y contraseña con los que se dio de alta en base de datos para seguir cargando el resto de opciones sobre los que se le ha dado acceso, en el caso de que el usuario exista en base de datos, o en caso contrario no se seguirá cargando opciones, impidiendo así alguna acción en la aplicación. Todas estas acciones son respaldadas por el archivo ActionScript denominado Identificación.as, y las consultas en base de datos se llevan a cabo del lado del servidor por un archivo PHP denominado getUsuario.php.
- Salir: si se selecciona esta opción se cerrará la aplicación. Es equivalente a pulsar el botón de cierre de cualquier aplicación común.

Cuando el usuario tenga perfil de administrador se cargarán en esta barra de herramientas tres opciones de mantenimiento que no se presentarán en el resto de usuarios: Mantenimiento de Procesos, para el mantenimiento de *tipos de procesos* y *procesos*, Mantenimiento General, para el mantenimiento de *procesos* vigentes en ese momento, y Mantenimiento de Usuarios, para el mantenimiento de los permisos de los usuarios.

El resto de opciones se irán cargando de forma recursiva hasta completar tantas opciones como tipos de procesos sobre los que se le ha otorgado permisos al usuario distinto de ‘0’. Recordemos, como se vio en el apartado de desarrollo de base de datos, que un permiso con valor ‘0’ indica que no se tiene permiso alguno.

Vemos en el siguiente código cómo, a partir del XML generado desde el PHP correspondiente tras identificarse el usuario, se cargará un objeto de clase Usuario con los datos referentes a éste, datos que han sido recuperados de base de datos. En el momento de ser asignados a una clase Usuario, el array comentado anteriormente como Definición de Procesos almacenará las denominaciones de los tipos de procesos que venían en los nodos hijos definidos mediante la etiqueta <procesos> y que harán referencia a los tipos de procesos sobre los que el usuario tiene algún permiso. A partir de esta variable del usuario, la aplicación cargará tantas opciones como valores contenga, quedando reflejadas en el menú de herramientas bajo etiquetas de valores a los valores contenidos en dicho array y que corresponde con el título que se dio al tipo de proceso en el momento de su definición.

El xml recibido tendrá una estructura tal como:

```
<item>
  <guid>2</guid>
  <login>login</login>
  <password>contrasena</password>
  <email>usuario@usuario.es</email>
  <nombre>nombre, apell apel2</nombre>
  <procesos>
```

```

        <proceso>1</proceso>
        <permiso>2</permiso>
        <descripcion>SISTEMAS DE AUDIO</descripcion>
    </procesos>
</procesos>
<procesos>
    <proceso>2</proceso>
    <permiso>2</permiso>
    <descripcion>ELECTRONICA BASICA</descripcion>
</procesos>
</item>

```

Tras ser cargados estos datos en un objeto de tipo *Usuario*, recorreremos el array de *procesosDesc* donde estarán almacenados los valores que venían bajo la etiqueta de `<descripcion>` en el xml:

```

For(var i:int=0; i<usuario.procesosDesc.length; i++){
    if (usuario.permisos[i] > 0){
        var titulo:String =
            UtilString.inicioMayusculas(usuario.procesosDesc[i]);
        var men:NativeMenu = new NativeMenu();
        var btn1:NativeMenuItem = new
            NativeMenuItem("Cargar");
        var btn2:NativeMenuItem = new
            NativeMenuItem("Limpiar");
        men.addItem(btn1);
        btn1.data = usuario.procesos[i];
        btn1.addEventListener(Event.SELECT, cargarFicheros);
        men.addItem(btn2);
        btn2.addEventListener(Event.SELECT, limpiarFicheros);
        this.menuPrincipal.nativeMenu.addSubmenu(men,
            titulo);
    }
}

```

Cada línea del menú de herramientas, al ser pulsada, hará saltar un evento¹⁵ similar a botones tal y como se aprecia con la línea de código `btn.addEventListener`, cuyas acciones serán respaldadas por el archivo *ActionScript* principal de la aplicación cargando, unos componentes hijos u otros y cerrando los que se encontraban abiertos.

En este caso, `Event.SELECT` son eventos que vienen definidos en las librerías propias de *Flex*, pero podemos crear eventos personalizados como hemos hecho para la recepción de datos de la BD, que captan cualquier suceso que llega a la aplicación con estructura xml, ya que los resultados de cualquier consulta sobre la BD se devolverá a la aplicación por parte de los archivos `*.php` del lado del servidor con ese esquema xml, como ya se ha indicado en otros puntos del proyecto y que veremos con más profundidad en el punto siguiente. Un ejemplo de estos eventos personalizados los podemos encontrar en los anexos VII y VIII, en el que se muestra el evento cuando se cargan los datos del usuario.

Para tal gestión de acciones debemos tener presentes tres archivos: la clase *Usuario*, la clase *UsuarioLoader* y la clase del evento *UsuarioEvent*. Expliquemos brevemente como es gestionado todo esto de forma esquemática que nos ayudará a comprenderlo mejor:

1. Se recibe desde la aplicación el login y contraseña del usuario. Con estos valores se debe consultar en BD. Para tal fin, con la clase *UsuarioLoader*, se forma la llamada pertinente al servidor mediante una sentencia *URLRequest* para que sea

¹⁵ Suceso que ocurre en un sistema. Eventos pueden ser un clic, doble clic, minimizar una ventana, etc.

allí donde el archivo *.php oportuno realice la consulta y devuelva el resultado de esta.

2. Mientras la consulta es llevada a cabo por parte del servidor, la aplicación queda a la espera. En el caso de que todo haya ido correctamente, la aplicación recibirá los datos obtenidos de la consulta en formato xml, que hará que se dispare el evento *XMLEvent* propio de librerías de Flex desde el archivo *UsuarioLoader*.
3. Desde *UsuarioLoader* cargará el evento personalizado de tipo *UsuarioEvent* pasándole como variables un objeto de tipo *Usuario* que ha sido cargado previamente con los valores recibidos en el xml e indicándole el tipo de evento, y será desde ese evento cuando se haga accesible desde cualquier punto de la aplicación, desde la que se invocó a este evento, los datos del objeto *Usuario* y *UsuarioLoader*, mediante *dispatchEvent* hará saltar el evento.

Por otro lado, sobre esta pantalla principal se irán cargando y cerrando los diversos componentes de la aplicación, según se elijan unas opciones u otras de la barra de herramientas. En este aspecto deberemos tener en cuenta a la hora de cerrar los componentes que el fondo de la aplicación nombrada anteriormente y definida *Background* nunca debe cerrarse y también habrá veces en las que haya varios hijos cargados y sólo se tendrá que cerrar alguno en particular, pero no todos. Por esto es interesante observar cómo se realiza este proceso mediante una función que hemos definido como *cerrarPendientes* y cuyo código mostramos a continuación:

```
Public function cerrarPendientes(pendiente:String, borrar: olean):void{

    var hijos:Array = Application.application.getChildren();
    var i:int = 0;

    while (i < hijos.length){
        if(hijos[i].id != "background_")
            if (pendiente != ''){
                if(borrar){
                    if(hijos[i].id == pendiente)
                        Application.application.removeChild(hijos[i]);
                }else{
                    if(hijos[i].id != pendiente)
                        Application.application.removeChild(hijos[i]);
                }
            }else
                Application.application.removeChild(hijos[i]);
        i++;
    }
}
```

Esta función recibe dos parámetros: *pendiente*, que recibe el id (identificador) del componente que llama a esta función o el id del componente en particular que se quiere cerrar o no y *borrar*, que indica si es el componente que quiere borrar (tomaría valor true) o no (valor false). En el caso de no indicar ningún valor en *pendiente* se cerrarían todos los componentes hijos.

En el código anterior, donde se mostraba como se cargaba de forma recursiva los componentes, se ha podido apreciar que se ha empleado una clase denominada `UtilString` en la línea `UtilString.inicioMayusculas(usuario.procesosDesc[i])`. Esta clase es una clase personal que hemos definido para llevar a cabo, mediante diversas funciones, los diferentes cambios de formato que se quiera realizar sobre las variables. Por ejemplo, en este caso, la función `inicioMayusculas` transforma la cadena como variable de entrada en una cadena por variable de salida, con la primera letra de cada palabra en mayúsculas y el resto en minúsculas. Así, en este caso en particular que se le está pasando el título de cada proceso, conseguimos un formato estándar para todas las opciones del menú de herramientas independientemente de cómo el usuario dio de alta este proceso en base de datos.

En esta clase hemos implementado otras funciones como:

- `convertDateStringMMDDYYYY (theDate:String):String` → Convierte una fecha pasada con formato mes, día y año ('MMDDYYYY') a formato año, mes y día ('YYYYMMDD')
- `convertDateStringYYYYMMDD(theDate:String):String` → Convierte una fecha pasada con formato año, mes y día ('YYYYMMDD') a formato día, mes y año ('DDMMYYYY')
- `fechaActual():String` → obtiene la fecha actual en formato DD/MM/YYYY.
- `reemplazaCaracteresRaros(cadena:String):String` → Reemplaza los caracteres acentuados por sus respectivos, sin acentuar, para evitar conflictos al grabar en base de datos.

También serán llevadas a cabo por esta parte las acciones de almacenar las respuestas que los usuarios vayan grabando en su local antes de ser enviadas por correo electrónico al usuario. Estas acciones no son llevadas desde archivos del servidor PHP ya que no se trata de modificar datos globales sino locales al dispositivo. Adobe AIR pone a nuestra disposición varias librerías para tratar documentos, tanto de texto o imágenes, ya sea para abrir archivos existentes, modificarlos, crear nuevos o eliminar. También nos proporciona librerías que permiten movernos entre directorios locales.

Aprovechándonos de estas ventajas y librerías, y salvaguardándonos de posibles conflictos de seguridad que puedan ocurrir en los sistemas cuando se intente grabar en diferentes directorios, se irán creando los distintos archivos de extensiones *.doc con las respuestas en el directorio propio del usuario como es:

C:\Documents and Settings\usuario\respuestas\tipo_proceso\proceso\num_pregunta

```
var myFile:File = File.userDirectory.resolvePath("respuestas" + "/" +
    Application.application.proc.tipo + "/" +
    Application.application.proc.titulo +
    "/" + numPregunta);
myFile.createDirectory();
myFile = File.userDirectory.resolvePath("respuestas" + "/" +
    Application.application.proc.tipo + "/" +
    Application.application.proc.titulo +
    "/" + numPregunta +
    "/" + numPregunta+".doc");

var myFileStream:FileStream = new FileStream();
myFileStream.openAsync(myFile, FileMode.WRITE);
```

```
myFileStream.writeUTFBytes(
    UtilString.reemplazaCaracteresRaros(texto_txt.text));
myFileStream.addEventListener(Event.CLOSE, guardadoHandler);
myFileStream.addEventListener(IOErrorEvent.IO_ERROR,
    writeIOErrorHandler);
myFileStream.close();
```

En el instante antes de enviar estas soluciones por correo electrónico, al haber pulsado el botón *Enviar Respuestas* por parte del usuario, será cuando la aplicación, de forma ajena y transparente al usuario, leerá todos y cada uno de los documentos Word de cada una de las respuestas que se han ido generando y grabando conforme se han ido respondiendo a las cuestiones del proceso agrupándolos en un solo documento. A este documento se le denominará *general.doc*, será enviado al responsable del proceso y tendrá una estructural tal como:

número_de_respuesta_contestada "-" respuesta_dada_por_el_usuario

hasta un número igual al número de pasos de los que conste el proceso y se almacenará en el directorio de ruta:

C:\Documents and Settings\usuario\respuestas\tipo_proceso\proceso

Por cuestiones de ahorro en el coste temporal, se ha intentado aprovechar al máximo las distintas opciones y librerías que nos ofrecía por defecto Adobe AIR, como hemos podido comprobar hasta ahora. En otros puntos donde se ha hecho utilidad de esto, ha sido el momento de validar distintos campos cuando se está procediendo a grabar nuevos proceso o modificar los ya existentes, como son las validaciones del campo de la dirección del correo electrónico del responsable del proceso o el formato de la fecha de baja de éste. Tales validaciones son llevadas directamente sobre los archivos *.mxmll pertinentes mediante códigos como el que se presenta seguidamente:

```
<mx:DateValidator id="validarFecha" source="{fecBaja}" property="text"
    inputFormat="yyyy-mm-dd" required="true" />
<mx:EmailValidator id="emV" source="{emailResponsable}" property="text"
    required="false" />
```

Estas validaciones son clases extendidas de la clase de Adobe Flex Validator. Ambas combinan componentes gráficos visuales, como es el mostrar los campos en color rojo en el caso de que no pasen la validación como componentes de código empleando comparaciones del texto introducido en el campo pertinente con el formato que se desea que tenga ese campo, entre otras herramientas.

Otro de los componentes de los que nos hemos valido ha sido el emplear un editor de texto rico (lo que Flex Builder denomina RichTextEditor) para las repuestas de los usuarios a los procesos. Este componente es un editor de texto que combina las diferentes posibilidades gráficas como el tamaño del texto, color de la fuente, diferentes fuentes, etc. para ofrecer al usuario una imagen atractiva del texto que está insertando en ese momento. A pesar de estas posibilidades, cabe destacar que estos estilos sólo son un valor añadido en el preciso momento de la inserción de texto en la pantalla, pues cuando se vaya a grabar estas respuestas en un fichero de texto en el directorio local, las respuestas se contemplarán como texto simple, siendo este formato y estilo despreciado.

Teniendo en cuenta que la aplicación está pensada para que pueda ser ejecutada en dispositivo móviles en los que los tamaños de pantalla son reducidos y poseen bajas resoluciones, en el momento de cargar un proceso, si los documentos asociados a éstos se cargaban directamente en la pantalla de la aplicación producía que los componentes o fueran demasiado pequeños, con lo que no sería nada cómodo para los usuarios, o tendría que haber barras de desplazamiento demasiado extensas, por lo que se optó por abrir estos documentos

asociados en otras ventanas adicionales. Con esto conseguimos un mayor atractivo visual a la vez que una mayor comodidad en el momento de la resolución de los procesos. Para tal fin, seguimos aprovechándonos de las ventajas de las librerías que nos proporciona el SDK de Flex Builder, que mediante el siguiente código se consigue generar estas ventanas:

```
var windowPdf:Rectangle:= new
    Rectangle(20,20,Capabilities.screenResolutionX/1.6,Capabilities.s
        creenResolutionY/1.6);
var pdf:HTMLLoader= HTMLLoader.createRootWindow(true, options, true,
    windowPdf);
```

En estas ventanas tenemos la posibilidad de configurar el tamaño, resolución y otras opciones indicadas en la variable *options*, como es el tipo de ventana. En nuestro caso hemos elegido un tipo estandar y la transparencia de esta, opciones que podemos ver a continuación:

```
var options:NativeWindowInitOptions = new NativeWindowInitOptions();
    options.systemChrome="standard";
    options.transparent=false;
```

La aplicación en general, en el aspecto de la interfaz de usuario y desde el punto de vista de programación tiene muchos más detalles, pero quizás aquí se han remarcado los más interesantes, además no se intenta dar un curso de cómo programar en Adobe Flex y tras pasarlo al motor de ejecución de Adobe AIR, sino el mostrar el resultado de esto, una aplicación que facilite a los usuarios en la labor para la que se desarrolló, por eso no entramos más en detalle en este apartado.

9.3 Acciones por parte del Servidor

Esta parte de la aplicación se encargará de la obtención de datos y modificación de los que procedan de base de datos, ya que esta base de datos, MySQL según se ha optado, estará desarrolla en el servidor, siendo así accesible para todos los usuarios. También será la encargada de realizar los procesos que requerirá el intérprete de PHP, como por ejemplo el envío de los correos electrónicos a los responsables de los procesos con las soluciones de los usuarios, pues aunque esta acción no necesita interaccionar con la base de datos sí es necesario un servidor para realizarla.

Con esta separación entre la parte de presentación y los datos conseguimos, además, se agilizar la ejecución de la aplicación, pues no tendrá que conectarse directamente a base de datos sino que se hará mediante un puente, valiéndonos de estos archivos PHP.

Los archivos PHP deberán ser colocados en el directorio de publicación del servidor para que sean accesibles desde la aplicación. Por motivos de organización, dentro de este directorio se ha optado por crear una ruta tal como ProyectoDocente/php.

Ya que en todos los archivos que se traten de consultas, inserciones u otras funciones en base de datos deberán conectarse a esta, se ha optado por crear un fichero dbConn.php, que conectará con la base de datos y después, mediante la función de PHP de include, incluirla en todos aquellos en los que sea preciso. Este fichero dbConn.php puede verse en el anexo X. De la misma manera, hemos operado para la inserción de la cabecera de los archivos xml, que serán devueltos a la aplicación, ya que ésta será común a todos y que también puede verse en el anexo XI, que hace referencia al fichero cabeceraXML.php.

Es muy útil separar estos trozos de código en un archivo a parte y llamar a este archivo con las funciones PHP *include()* o *require()*. Así, si un día modificamos estos datos, sólo los tendremos que modificar en un archivo o, si cambia la base de datos que utilizamos sólo tendríamos que modificar el archivo que hace la conexión a la base de datos una vez, en lugar de ir cambiándolo en todas las páginas PHP que abrían las bases de datos. Estos archivos no son páginas independientes, sino fragmentos. Seguramente si los ejecutamos por separado no mostrarían ningún resultado válido, incluso podrían dar mensajes de error.

Uno de los procesos que se lleva a cabo desde esta parte es el proceso de Mantenimiento General de los procesos, para el que la aplicación invocará al script `mantenimientoProcesos.php`. A este script se le pasa como único parámetro el tipo de proceso sobre el que se quiere llevar a cabo el mantenimiento y la fecha actual del sistema en el momento en que sea ejecutado.

La funcionalidad de este procedimiento radica en la reducción de la carga de la base de datos ya que lo que hace es traspasar datos de tablas vivas a históricas, consiguiendo así en las distintas consultas que tengan que hacerse sobre las tablas de la base de datos, como puede ser, por ejemplo, en el caso de obtener los distintos procesos, se reducirá el coste temporal que esto supondrá si tenemos en cuenta que si este procedimiento no se llevara a cabo, con el paso del tiempo algunos elementos como la tabla de procesos o documentos alcanzarían un número elevado de registros. Entonces ya no tendrían sentido porque no serían visibles a los usuarios, al tratarse de procesos dados de baja, aumentando el tiempo de la consulta cuando se quisiera cargar los procesos todavía vigentes.

Como hemos visto en el apartado de base de datos, nos encontramos con dos tipos de tablas idénticas relacionadas dos a dos: PROCESO con PROCESOH y DOCUMENTO con DOCUMENTOH. Cuando se procede a realizar el mantenimiento general de un tipo de proceso, se hace una consulta sobre la tabla de PROCESO filtrando por el tipo de proceso que ha sido pasado por parámetro y cuya fecha de baja sea anterior a la fecha actual. También pasado como parámetro, y para todos esos registros recuperados, se hace una copia sobre la tabla PROCESOH y se borran de la tabla PROCESO. Posteriormente, para estos mismos registros, se recuperan los códigos de los documentos asociados y se procede también a hacer una copia de los registros de la tabla DOCUMENTO, cuyos códigos sean los recuperados a la tabla DOCUMENTOSH, borrando igualmente estos registros de la tabla DOCUMENTOS. Una vez que estos dos volcados de datos se han realizado correctamente y con los valores de los códigos de documentos, se accede al directorio del servidor

```
/temp/ProyectoDocente/documentos/tipo_proceso
```

y se borran aquellos documentos cuyo nombre es igual a los códigos de los documentos recuperados de base de datos y extensión *.pdf. Estos documentos no son traspasados a una carpeta histórica, ya que como no se ha contemplado la posibilidad de recuperar un proceso finalizado, no volverá a ser cargado el documento asociado nunca más, por lo que no va a tener sentido tener documentos en la memoria del servidor que nunca van a volver a ser empleados, evitando así con el paso del tiempo una carga excesiva de memoria del servidor con “documentos basura”.

Tanto las acciones de alta, modificar o borrar un nuevo tipo de proceso o los permisos de un usuario, son acciones simples en base de datos de INSERT, UPDATE o DELETE sobre las tablas TIPOPROCESO y USUARIO respectivamente, por lo que no tiene mayor interés; en cambio, cuando se va a dar de alta o modificar un proceso estas acciones, tienen una mayor grado de complejidad como explicaremos a continuación.

Estas acciones de alta o modificación de procesos, al igual que en el proceso de mantenimiento general se pueden distinguir dos partes: una consistente en acciones sobre base de datos y otra en memoria del servidor con la carga del documento asociado.

En el caso de dar de alta un nuevo proceso, la aplicación llamará a un script del lado del servidor llamado nuevoProceso.php y quedará en estado de ejecución hasta recibir confirmación por parte del servidor. A este script se le pasarán todos los campos referentes a un proceso, los cuales se han visto en apartados anteriores, como título, número de preguntas, descripción, etc.

Los pasos que este script realiza en la grabación de un nuevo proceso son:

1. Con el nombre del documento que se ha asociado al proceso, se grabará un registro en la tabla DOCUMENTO, cuya denominación se corresponderá con ese nombre y una vez insertado el registro se recuperará el código que ha sido asignado de forma aleatoria a este registro.
2. Tras insertar en DOCUMENTO y recuperar el código, se procede a insertar en la tabla PROCESO donde, en el campo de documento asociado, se grabará el valor del código anteriormente recuperado.
3. En el caso de grabarse todo de forma correcta, con los datos pasados por la aplicación del documento que se ha seleccionado como asociado al proceso, se subirá este documento al servidor al directorio de ruta:

Temp/ProyectoDocente/documentos/tipo_proceso

y empleando la función de la librería php:

```
$destino = "Temp/ProyectoDocente/documentos/tipo_proceso";  
$moved = move_uploaded_file($_FILES['Filedata']['tmp_name'],  
    $destino.$cod_documento.".pdf");
```

4. Si todo ha ido bien, el script generará un XML que será devuelto a la aplicación y con el que se le indicará que todo ha ido correcto, junto con el código del nuevo proceso insertado como el código del nuevo documento. En caso que haya ocurrido algún error se le devolverá un XML con el mensaje donde se le indica que ha fallado.

Tras devolver el control a la aplicación, si se ha devuelto un mensaje de error se mostrará por pantalla y quedará la pantalla en modo de inserción. En otro caso se mostrará también el mensaje de confirmación pero quedando la pantalla en modo no editable.

Para la modificación de un proceso ya existente los pasos son similares, con la salvedad de que si no ha sido modificado el documento asociado a éste, tan solo se procederá a actualizar los datos de base de datos relacionados con la tabla PROCESO; en caso de que sí se haya modificado el documento asociado se actuará como:

1. Se recuperará filtrando por el código del proceso de la tabla PROCESO el código del documento asociado al proceso que se desea modificar y su tipo de proceso inicial. Se comprobará si se ha modificado el documento asociado, comprobando el título del documento pasado por parámetro desde la aplicación y los datos recuperados de la tabla DOCUMENTO con el código del documento inicial recuperado de PROCESO.
En caso afirmativo:
 - a. Se insertará un nuevo registro en la tabla DOCUMENTO para el nuevo documento, seleccionado recuperando el nuevo código asignado aleatoriamente.
 - b. Se actualizan los datos del proceso de la tabla PROCESO, introduciendo en el campo del documento asociado el nuevo código asignado al nuevo documento.

- c. Se borra el registro de la tabla de DOCUMENTO referente al anterior documento que tenía asociado el proceso.
- d. Se borrará el documento asociado inicialmente de la carpeta:

Temp/ProyectoDocente/documentos/tipo_proceso_inicial

por medio de la función de las librerías de php según:

```
$dir="Temp/ProyectoDocente/documentos/tipo_proceso_inicial/";
$borrado = unlink($dir.$cod_inicial_doc.".pdf");
```

- e. Se subirá el nuevo documento asociado del dispositivo del usuario al servidor en la ruta:

Temp/ProyectoDocente/tipo_proceso_nuevo

con título al nuevo código del documento al ser insertado en la tabla DOCUMENTO.

En caso de que el documento asociado no haya sido modificado:

- a. Se actualizarán los datos de la tabla PROCESO
2. El script generará un XML que será devuelto a la aplicación, donde se indicará un mensaje con el resultado de estas operaciones, tanto si todo ha ido correctamente como si ha fallado algo.

En el caso de haber modificado el documento asociado, se ha procedido en ese orden de actuación, ya que hay en base de datos, tal y como se comentó en el punto pertinente, la restricción de clave ajena entre el código del documento que se grabe en el campo de la tabla PROCESO con la tabla DOCUMENTO, por lo que si previamente no se modifica el documento que tiene asociado el proceso no se podrá borrar el registro al que hace referencia de la tabla DOCUMENTO, pero para modificar este valor en la tabla PROCESO previamente hay que insertar en la tabla DOCUMENTO el nuevo registro, pues sino no tendremos el valor del nuevo código de ese documento, ya que este valor es asignado al grabar el nuevo registro en la tabla DOCUMENTO.

Otra de las operaciones que se realizan desde el lado del servido es la recuperación de las distintas listas de valores que aparecen en la aplicación, como la lista de los distintos tipos de procesos o la de procesos. Estas acciones son consultas simples sobre base de datos de los distintos valores existentes en las correspondientes tablas.

También alcanza cierto grado de complejidad el proceso que envía del correo electrónico a los responsables de los procesos con las soluciones de los usuarios a los procesos y los documentos extra asociados, en el caso de que haya o se permita esta opción. Para tal acción empleamos la función propia de PHP llamada *mail* ().

El script email.php es el archivo que hace tal procedimiento. A éste se le debe pasar todos los datos referentes a la cabecera del correo electrónico, como son el nombre y dirección de correo electrónica del usuario que está enviando su solución, la dirección del responsable del proceso que recibirá el e-mail, el asunto del correo y los datos adjuntos.

Los datos referentes del usuario se tienen en la aplicación desde el instante en que la persona accede a la aplicación y se identifica. La dirección de correo electrónico del responsable se obtuvo en el momento de cargar un proceso, ya que se han cargado todos los datos referentes al proceso cargado y se guardaron en un objeto de clase Proceso. El asunto que se asignará al correo será variable en el que se indicará de forma fija para todos "SOLUCIÓN AL PROCESO:", e irá concatenado con la denominación del proceso que se está resolviendo. Y los documentos a enviar serán, por una parte, un único documento con las soluciones al proceso que la aplicación se ha encargado de generar con todas las respuestas a todos los pasos de los que constaba el proceso que, como se ha comentado en el punto anterior, se denomina general.doc, y, por otro lado, un array con todos y cada uno de los documentos extra que el usuario ha decidido enviar junto a sus soluciones, en el caso de que el proceso presente esa opción. En el caso de que esta opción no esté disponible este array estará vacío.

Antes de enviar el correo electrónico, este procedimiento copiará los documentos que ha de adjuntar a un directorio del servidor cuya ruta será:

directorio_publicacion/ProyectoDocente/dni_usuario

que, tras enviar el e-mail, serán borrados todos estos documentos junto con este directorio creado temporalmente de forma recursiva, según puede verse en el anexo XII.

Recordemos que estamos trabajando sobre una base de datos MySQL, en la que cabe citar que está basada en lenguaje SQL. Mencionar que en este lenguaje, por si no se conoce o no se recuerda, el carácter '%', en las consultas, es un carácter comodín siempre y cuando se emplee el comando LIKE, que se usa para cadenas de caracteres que no se conocen, pero ocupan una posición específica, allí donde se haya colocado este carácter, dentro de una cadena tanto sea de texto como numérica, por eso cuando los usuarios deseen filtrar sus búsquedas en la pantalla de usuarios para recuperar los permisos de algún usuario en particular, deben emplear ese carácter como se ha indicado en apartados previos y como se especifica en el punto posterior a éste, en el Manual de Usuario. Así la consulta quedará tal como:

```
$sql = "SELECT DISTINCT PERMISO.DNI DNI, TIPOPROCESO.TITULO TITULO,
        PERMISO.PROCESO PROCESO,
        PERMISO.PERMISO PERMISO,
        CONCAT (USUARIO.NOMBRE,' ', USUARIO.APELLIDO1,'
        ',USUARIO.APELLIDO2) NOMBRE FROM PERMISO, USUARIO,
        TIPOPROCESO
        WHERE PERMISO.DNI = USUARIO.DNI
        AND USUARIO.DNI LIKE CASE WHEN '$dni' = " THEN
        USUARIO.DNI
        ELSE
        '$dni'
        END
        AND CONCAT (USUARIO.NOMBRE,' ', USUARIO.APELLIDO1,'
        ',USUARIO.APELLIDO2)
        LIKE CASE WHEN '$nombre' = " THEN
        CONCAT (USUARIO.NOMBRE,' ', USUARIO.APELLIDO1,'
        ',USUARIO.APELLIDO2)
        ELSE
        '$nombre'
        END
        AND TIPOPROCESO.COD = PERMISO.PROCESO
        ORDER BY DNI, NOMBRE";
```

La función de PHP empleada para acciones sobre base de datos, bien sea para consultas como inserciones, modificaciones o borrados, es la `mysql_query ()` a la que se le pasará la cadena con la acción a realizar. Por ejemplo, si queremos realizar la consulta anterior para la obtención de los permisos de un usuario se implementará como:

```
$query = mysql_query ($sql);
```

donde el resultado, `$query`, es una variable booleana que tomará valor `TRUE` en caso de que la acción se haya podido realizar, sea una consulta o una modificación de datos, o `FALSE` en el caso de que no se haya podido realizar, como ocurriría en el caso de que no se haya conectado con la base de datos.

Esta variable recuperada no indica que se hayan recuperado datos o no, simplemente si se ha podido llevar a cabo el script en base de datos. Para saber si se han obtenido datos o no empleamos, continuando con el ejemplo anterior, la sentencia:

```
mysql_num_rows ($query)
```

que devuelve el número de registros recuperados de la consulta. Para poder evaluar cada registro recuperado lo haremos de forma recursiva, mientras existan valores, por medio de:

```
while ($row = mysql_fetch_array($query))
```

tal que en la variable `$row` es asignado cada uno de los registros recuperados pudiendo analizarse el valor de cada columna de estos registros como:

```
$vble_a_recuperar = $row ['nombre_columna'];
```


10. TEST DE PRUEBAS

Todo desarrollo e implementación en el mundo de la ingeniería de software debe pasar por ciertos procesos de pruebas y, una vez finalizado su desarrollo, sería óptimo mantener el resultado de este desarrollo durante un tiempo, en periodo de pruebas, donde los usuarios puedan validar que todo funciona conforme a lo especificado y de forma correcta, pues no hay nadie más que conozca mejor el funcionamiento que debe tener la aplicación, ya que fueron ellos quienes definieron sus requisitos.

Según se han ido desarrollando los diferentes módulos de la aplicación, hemos ido realizando diferentes pruebas con la ayuda de la implementación de un servidor local, que hemos podido montar gracias al software XAMPP, como se ha explicado en los puntos anteriores.

Estas pruebas iniciales deberemos trasladarlas fuera del entorno local, accediendo a un servidor remoto y comprobar que todo sigue funcionando de forma correcta y según lo esperado, pues cuando la aplicación sea puesta en el entorno de producción, esta deberá acceder a un servidor remoto donde estará implementada la base de datos necesaria sobre la que está montada la aplicación y donde estarán albergados los script PHP que gestionan todos los procesos de base de datos.

Para estas pruebas nos proponemos una serie de puntos a comprobar y que deberán ser pasados todos y cada uno de ellos tal como especificamos a continuación:

1. Instalación de la aplicación:

El ejecutable de la aplicación debe permitir ser instalada en distintos dispositivos bajo distintos sistemas operativos sin problema alguno, siempre y cuando el usuario tenga instalado el motor de ejecución de AIR propio a cada sistema operativo

2. Entorno gráfico:

Cuando la aplicación sea instalada en distintos dispositivos, bajo distintos sistemas operativos, el diseño gráfico de la aplicación deberá ser coherente a lo diseñado, manteniendo el tamaño de las pantallas acordes al dispositivo.

3. Acceso al entorno de trabajo:

Una vez instalada la aplicación deberá ser capaz de tener conexión a un servidor remoto y ser capaz de conectarse a la base de datos implementada para tal fin, validando correctamente los datos del usuario.

4. Inserción de datos:

Cuando se introduzcan nuevos datos serán validados y grabados de forma correcta en base de datos y al mismo tiempo si se tiene que realizar algún proceso más también será llevado a cabo.

5. Modificación de datos:

Se tendrá que someter a pruebas la modificación de datos, tanto comprobando aquellos que no puedan ser modificados como aquellos sensibles a la modificación de forma que se realice correctamente junto al resto de procesos que cuelguen de estas acciones.

6. Eliminación de datos:

Cuando se eliminen datos se deberán realizar de forma correcta en base de datos, comprobando que no se dejan datos basura ni elementos inaccesibles.

7. Respuestas usuarios:

Debe probarse que las respuestas de los usuarios son grabadas correctamente en el dispositivo local donde esté instalada la aplicación y al enviar el correo sea generado

el documento común con todas las respuestas, al igual que tiene que comprobarse que estas son enviadas y recibidas.

8. Gestión de las distintas pantallas:

Se someterá a pruebas que sean abiertas y cerradas de forma correcta las distintas pantallas que conforman la aplicación.

Una vez estipulados unos puntos de pruebas procedemos a llevarlos a cabo. Para ello nos permiten el acceso a un servidor remoto perteneciente a la UPV al que, en el caso de acceder desde fuera del recinto de la UPV, habrá que acceder mediante VPN¹⁶. Una vez tenemos la dirección IP de este servidor que emplearemos como prueba, configuramos la aplicación para indicarle hacia donde tiene que hacer referencia cuando llame a los distintos archivos PHP, al igual que también configuramos del archivo PHP de la aplicación dbConn.php con esta misma dirección IP, que es la que indicará donde está instalada la base de datos a la que se conectará. Una vez realizadas estas dos modificaciones volvemos a generar el ejecutable de la aplicación y tras pasamos los ficheros *.php al directorio de publicación del servidor bajo la ruta de: ProyectoDocente/php, según se ha comentado anteriormente y es como se ha especificado en la aplicación.

Tras estos pasos, nos ponemos en contacto con el personal de mantenimiento de sistemas para que nos presten su ayuda e instalen el intérprete de PHP en el servidor, ya que no se encontraba instalado y nosotros no teníamos permisos suficientes para tal fin. La base de datos MySQL la procedimos a configurar y crear nosotros mismos, pues tenemos los permisos suficientes, al mismo tiempo que hacemos una carga inicial directamente sobre ésta con datos de pruebas, tanto de usuarios como tipos de procesos sobre los que otorgar permisos.

Realizados todos estos pasos previos correspondientes a la etapa de sistemas comenzamos a evaluar cada uno de los puntos estipulados que debe pasar en el periodo de pruebas la aplicación.

Para estas pruebas se ha instalado la aplicación en tres tipos de dispositivos distintos: un ordenador de sobremesa con el SO de Windows XP SP3 y pantalla de 17 pulgadas, un ordenador portátil con el SO de Windows XP SP3 y pantalla de 10'5 pulgadas y un ordenador portátil con SO Linux y pantalla de 14 pulgadas.

En todos ellos la instalación corrió sin problema alguno. Al ser ejecutada, la interfaz gráfica era coherente en todos los casos, teniendo en cuenta que en cada uno se estaba visualizando sobre pantallas de diferentes tamaños. Cabe señalar que, en el ordenador portátil de pantalla 10'5 pulgadas, las pantallas presentaban barras de desplazamiento tanto vertical como horizontal a pesar de tener las páginas maximizadas pero no presentaba una incomodidad en exceso en el momento de trabajar.

Cuando se procedió a trabajar con ella, inicialmente encontramos problemas para poder acceder a la aplicación ya que no conseguíamos que nos validara como usuarios, pero descubrimos que estos problemas eran debidos a configuraciones de usuarios sobre la base de datos MySQL al acceder remotamente y a configuraciones de PHP, es decir, eran problemas propios al servidor y no a la aplicación. Tras solventar estos problemas, modificando los correspondientes ficheros php.ini en el servidor y otorgar acceso remoto a la base de datos, pudo accederse a la aplicación y trabajar sin problemas.

Para los puntos de prueba de alta, modificación y borrado de datos de base de datos todo fue correcto. El problema surgió en la restricción de permisos que se tenía sobre diferentes

¹⁶ VPN. Red Privada Virtual, es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada

directorios del servidor en los que se había pensado inicialmente albergar los documentos asociados a los procesos, por lo que hubo que modificar estas rutas para hacerlo en directorios a los que sí se tenían permisos, como la carpeta /temp y como se ha podido comprobar hasta este momento que al final así ha quedado. Estos cambios de directorio no supuso ninguna modificación de la aplicación ni volver a generar un nuevo ejecutable, ya que estos datos sólo son empleados en archivos del lado del servidor, *.php, por lo que se modificaron desde un simple editor de texto y se volvieron a subir al servidor.

Para procedimientos en los que la aplicación tiene que trabajar con archivos y directorios locales, han sido ejecutados sin problema e incluso se ha comprobado que los correos electrónicos llegan a su destino correctamente. Cuando se ha tratado de la subida de documentos desde local hacia el servidor, también nos encontramos con la restricción que no podían subirse documentos cuyo tamaño superaran los 2 MB, pero esta limitación también es aportada por el servidor y no por parte de la aplicación, por lo que también se tendrá que tener en cuenta cuando se proceda a poner en producción la aplicación y configurar de manera correcta el servidor sobre el que se va a trabajar.

Sometida la aplicación a las diversas pruebas de forma concienzuda, podemos comentar que por nuestra parte sería este el momento de entregarse una primera versión para que sea sometida a pruebas por los usuarios finales.

11. MANUAL DE USUARIO

ProyectoDocente es una aplicación que le permitirá gestionar todos sus procesos de diversa índole proporcionando una plataforma común para que todos sus alumnos puedan acceder a ellos y puedan proceder a resolverlos. Le permitirá gestionar además los perfiles de sus alumnos permitiéndole otorgar acceso o restringírsele en cualquier momento; todo esto por medio de una interfaz gráfica atractiva y rompiendo las barreras de las distancias geográficas.

Esta aplicación puede ser instalada en dispositivos de sobremesa o portátiles con sistemas operativos WINDOWS XP o posteriores, Linux, MAC o Android.

Previamente, deberá instalar en el dispositivo donde se vaya a proceder a llevar a cabo la instalación de la aplicación, el entorno Adobe AIR, el cual podrá descargar directamente y de forma gratuita desde la página oficial de Adobe, www.adobe.com.

Posteriormente, descargarnos el ejecutable que se muestra en la Figura 26 de la Web donde esté alojado, procedemos a su instalación en nuestro dispositivo.



Figura 26. Ejecutable de la aplicación

Inicialmente se nos muestra una pantalla de seguridad donde nos pide la confirmación de si deseamos instalar la aplicación según vemos en la Figura 27. En esta pantalla se muestran datos relativos a la aplicación que deseamos instalar en función del certificado digital que ésta posea, es decir, si a la aplicación se le ha agregado un certificado digital válido el símbolo en rojo de “?” que se aprecia en la figura cambiará a “√” en verde. En este caso, a la aplicación se le ha asignado un certificado digital proporcionado por el propio programa de desarrollo, por lo que no se puede asegurar la seguridad en el sistema donde se vaya a instalar la aplicación.

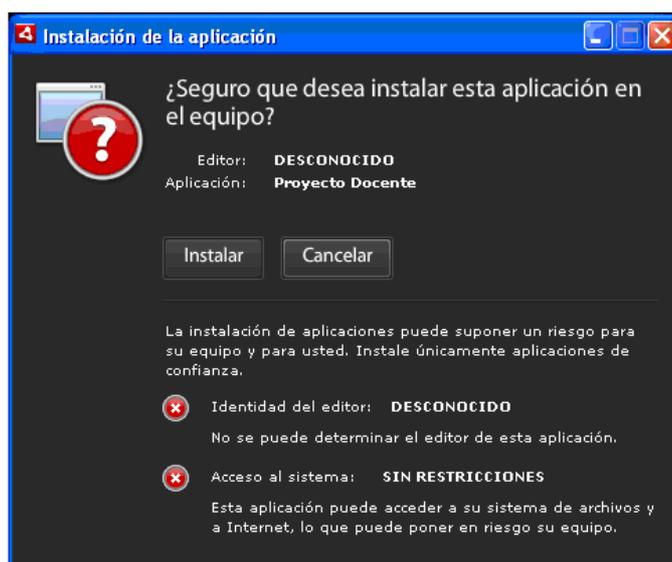


Figura 27. Pantalla de Instalación

En el caso de aceptar la instalación, se nos presentan diferentes opciones de instalación, como se observa en la Figura 28, con los valores por defecto siendo posible cambiarlos, como en el caso del directorio de instalación.

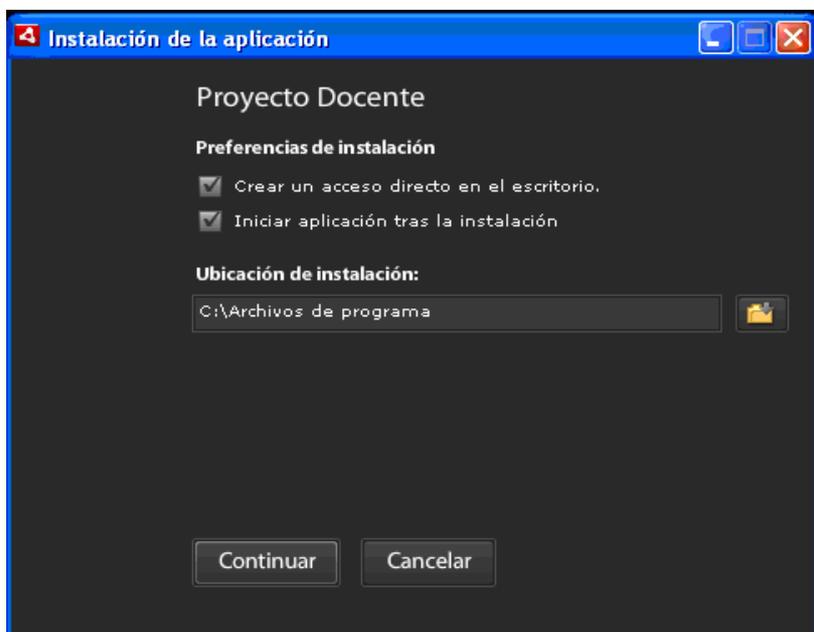


Figura 28. Pantalla opciones de instalación de la aplicación

Una vez modificadas las opciones anteriores a nuestra conveniencia se procede a la instalación. Mientras esto ocurre se nos muestra el estado del proceso como podemos observar en la Figura 29.

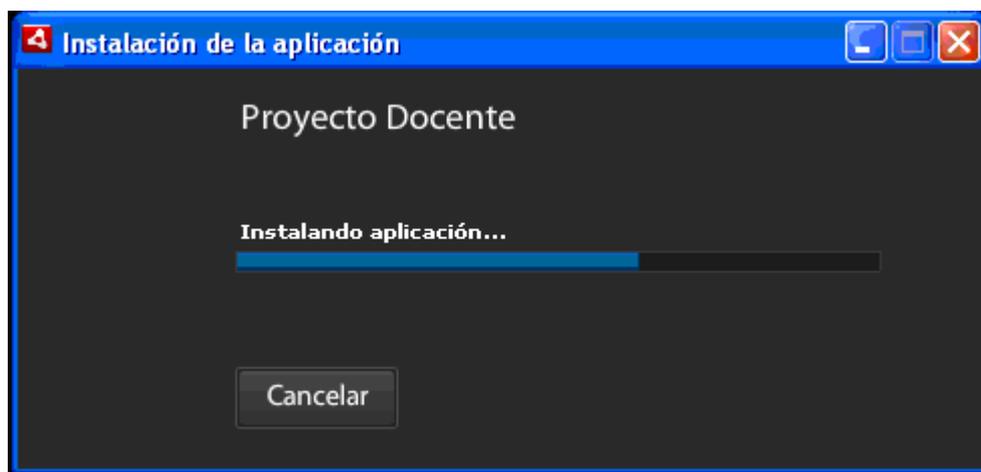


Figura 29. Proceso de instalación de la aplicación

Finalizada la instalación, si se ha mantenido seleccionado el check de iniciar la aplicación tras la instalación, se nos abre la pantalla principal de ésta. Esta pantalla tendrá un aspecto tal y como vemos en la Figura 30. En el caso de no haber dejado seleccionada esta opción, para ejecutar el programa tendremos que hacerlo desde la línea creada en el menú Inicio o en su defecto mediante el icono creado en el escritorio en el caso de haber dejado esta opción seleccionada.



Figura 30. Pantalla principal de la aplicación

Todos los usuarios dispondrán inicialmente en la barra de herramientas el menú “Archivo”. Esta opción contiene un submenú con dos opciones: Identificarse y Salir (Véase Figura 31).



Figura 31. Barra de herramientas principal de la aplicación

En el caso de que después de ejecutar la aplicación se desee salir de ella, podrá hacerlo bien sea mediante el botón de cierre típico a cualquier aplicación como es el mostrado en la Figura 32 o eligiendo la opción de “Salir” del menú “Archivo”.



Figura 32. Botón de cierre de la aplicación

Si se opta por la decisión de querer empezar a trabajar con la aplicación, los usuarios deberán identificarse con el nombre y contraseña que se le haya dado de alta en la base de datos de la UPV, para tal fin tendrán que pulsar sobre la opción de “Identificarse” del menú “Archivo” abriéndose la pantalla que se muestra en la Figura 33.

Figura 33. Pantalla de identificación del usuario

Por motivos de seguridad, los datos introducidos en el campo contraseña serán encriptados como se aprecia en la Figura 34 para evitar posibles intromisiones.

Figura 34. Detalle codificación contraseñas

Si en la pantalla de Identificación pulsamos en el botón Cancelar, cerrará esta pantalla volviendo a la pantalla cargada inicialmente. En el caso de pulsar el botón Aceptar, la aplicación intentará conectarse a la base de datos de un servidor remoto donde validará los datos introducidos en el login y contraseña. Si no se dispone en ese momento de conexión con el servidor remoto y no se ha podido conectar con la base de datos, bien sea porque usted no posea en ese momento de acceso a la red, el servidor esté fuera de servicio o cualquier otro caso que imposibilite tal fin, se mostrará un mensaje tal y como se muestra en la Figura 35 y no se cargará ninguna opción de la aplicación.

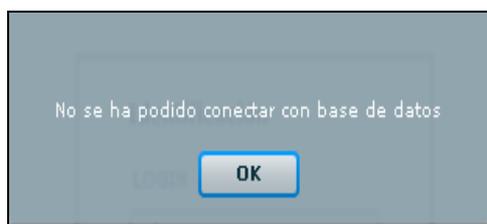


Figura 35. Indicación de acceso al servidor no disponible

En caso contrario, tras validar login y contraseña y comprobar que es correcto, se cargarán en el menú de herramientas de la aplicación, las opciones a las que se le ha otorgado



Figura 36. Barra de herramientas personalizada

11.1 Manual de Administradores

En el caso de que el usuario le haya sido otorgado permiso de administrador en algún tipo de proceso (permiso de código '2' en base de datos según la configuración elegida), independientemente del resto de opciones de la barra de herramientas según los procesos vigentes, se le presentarán tres opciones específicas que le permitirán ejercer su labor de

administrador: ‘Definición de Procesos’, ‘Mantenimiento General’ y ‘Mantenimiento de Usuarios’, que a continuación detallaremos de forma más exhaustiva.

11.1.1 Mantenimiento de Usuarios

Esta pantalla se muestra tras seleccionar la opción de la barra de herramientas que se indica en la Figura 37.



Figura 37 Opción de mantenimiento de permisos

Tras optar por esta opción se nos carga una pantalla tal como se puede ver en la Figura 38.

The image shows a web form titled 'Mantenimiento de Usuarios'. It contains several input fields: 'Nombre' and 'D.N.I.'. Below these fields is a 'Buscar' button. There is also a 'Cargar Datos' button. Below the search fields, there is a section for 'Usuario' with a text input field. At the bottom, there is a table with two columns: 'Tipos Procesos' and 'Permiso'. Below the table are three buttons: 'Nuevo', 'Modificar', and 'Salir'.

Figura 38. Pantalla de mantenimiento de permisos

En ella podemos distinguir tres partes bien definidas:

- En primer lugar tenemos un bloque de búsqueda para restringir la búsqueda por el nombre o DNI del usuario que se desea insertar, modificar o borrar un permiso sobre un tipo de proceso. Es obligatorio insertar algún valor en alguno de los dos campos. En el caso de no conocer los datos exactos de algunos de estos dos valores, se podrá hacer una búsqueda aproximada si se incluye el carácter '%' en aquél lugar donde no se conozca el valor. Es decir, en el caso de desear buscar a un usuario del cual se conoce un solo apellido, por ejemplo *admin*, insertaríamos en el campo del Nombre '%admin%' tal y como se puede apreciar en la Figura 39. Lo mismo ocurriría para el caso del DNI.

Esta condición de obligatoriedad, se ha definido para evitar hacer búsquedas masivas sobre BD y que pueda colapsarla.

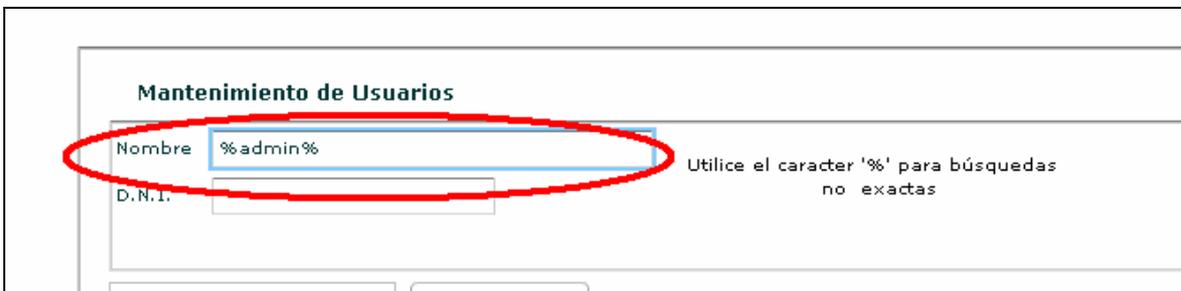


Figura 39. Filtro de búsqueda de usuarios

- Un segundo bloque está formado por una lista de valores donde se mostrarán los resultados de ejecutar la búsqueda indicada en el punto anterior como se señala en la Figura 40, en cuyo ejemplo tan sólo ha recuperado un registro que cumple las condiciones especificadas pero el número de resultados es variable en función del grado de restricción de los datos del bloque de búsqueda.



Figura 40. Resultado de la búsqueda

Los datos que se muestran en esta lista son los mismos por los que se ha restringido la búsqueda. En el caso de desear visualizar los datos relativos a los permisos se deberá seleccionar el registro deseado pudiendo distinguir dicho registro porque queda marcado en azul como vemos en la Figura 41 y posteriormente pulsar el botón de *Cargar Datos*.



Figura 41. Detalle selección de registro

En el caso de no haber seleccionado ningún registro y pulsar el botón de *Cargar Datos*, se mostrará un mensaje informativo indicándole al usuario lo que debe hacer como vemos en la Figura 42.

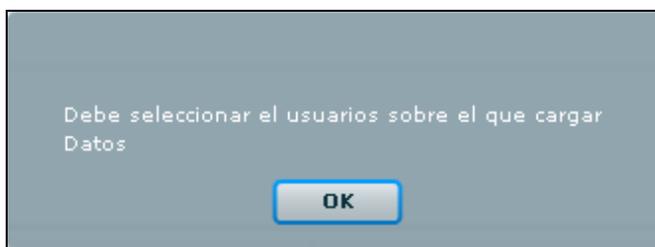


Figura 42. Detalle de carga de datos

- Por último lugar, tenemos el bloque donde son cargados los datos relativos a los permisos del registro seleccionado tal y como podemos ver en la Figura 43. En este bloque a su vez podemos diferenciar dos sub-bloques. El primero de estos, se cargan de forma completa y mejor detallada los mismos datos por los que se puede restringir la búsqueda, es decir el nombre y DNI del usuario buscado. Y en el segundo sub-bloque se presenta una lista de valores con los distintos tipos de procesos y el permiso otorgado a ellos que posee esta persona. En el ejemplo de esta figura, el usuario *ADMINISTRADOR* tan sólo posee permisos sobre el tipo de proceso denominado *PROCESO CARGA* cuyo tipo de permiso es de *Administrador*.

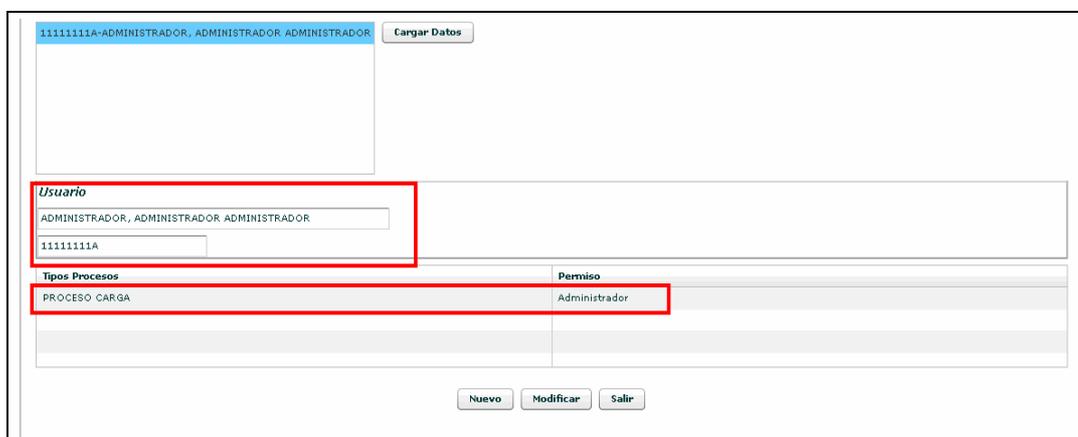


Figura 43. Visualización de los datos obtenidos

Si se desea insertar un nuevo permiso, basta con pulsar el botón de nuevo cuya acción hará cargar la pantalla que se muestra en la Figura 44.



Figura 44. Pantalla asignación de nuevo permiso

En esta pantalla se muestran dos desplegables. En uno se muestran todos los tipos de procesos vigentes en ese momento, en el ejemplo de la figura en ese momento sólo existe el tipo de proceso *Proceso Carga*, y en un segundo desplegable se muestran los tipos de permisos que pueden concederse y pueden verse en la Figura 45 los cuales, como ya hemos referenciado en varias ocasiones, son: *Acceso Denegado*, *Acceso con restricciones* y *Administrador*.

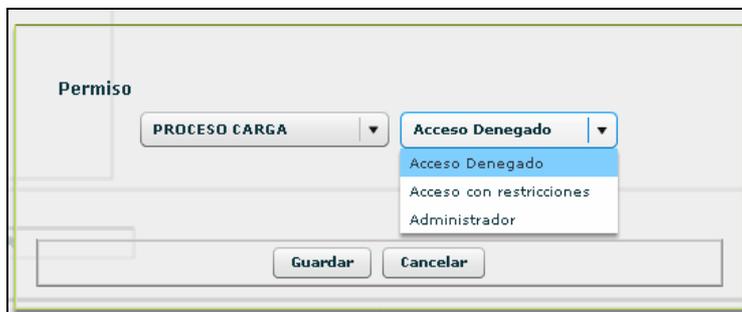


Figura 45. Detalle de los posibles tipos de permisos

Si se desea modificar un permiso ya existente, debe seleccionar previamente el registro deseado de la lista de valores de permisos y pulsar el botón Modificar, tal acción hará que se cargue la misma pantalla que se ha comentado antes pero cargada con los datos referentes al registro seleccionado como se puede ver en la Figura 46.



Figura 46. Pantalla modificación de permiso ya otorgado

Tanto en el caso de insertar un nuevo registro como en el caso de modificar uno ya existente, cuando se pulsa el botón *Guardar* y los valores son actualizados en la BD, se muestra un mensaje de confirmación tal y como se muestra en la Figura 47. Debido a la característica de Adobe AIR, de que no se recarga la pantalla a menos que se fuerce esta recarga o se recarguen datos de la BD, si queremos actualizar la pantalla deberemos volver a ejecutar la búsqueda para ver reflejados los cambios en la pantalla tal y como se indica al usuario en la figura.

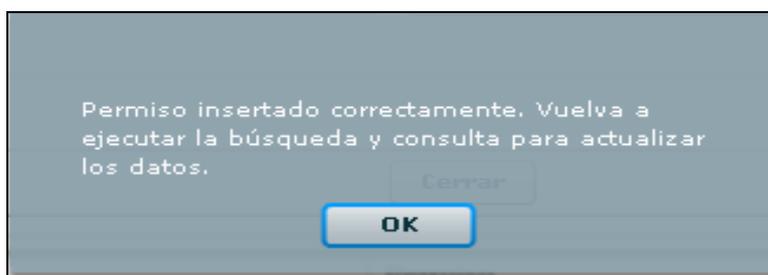


Figura 47. Mensaje de éxito de la operación realizada

Una vez guardados los datos correctamente desaparece el botón de *guardar*, dejando la única opción del botón *Salir* como se aprecia en la Figura 48, el cual tras ser pulsado cerrará la pantalla de modificación de permisos, evitando así posibles conflictos en BD al dejar una tabla o registro bloqueado.

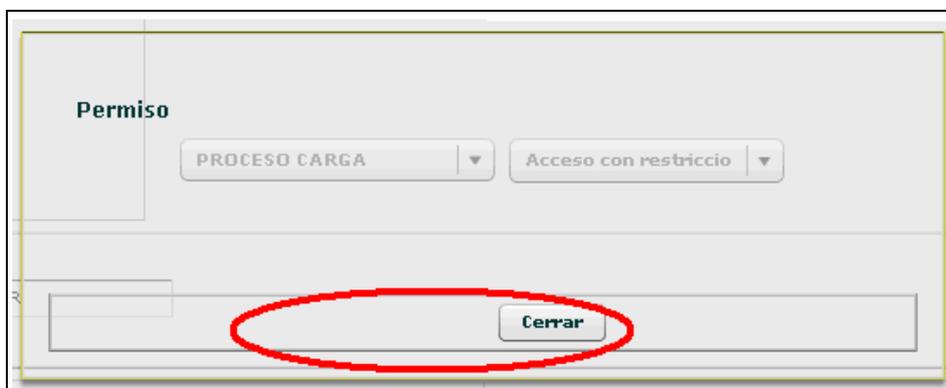


Figura 48. Detalle botones dinámicos

En el caso de que ocurra algún error se mostrará el mensaje de la Figura 49, advirtiéndolo al usuario de tal efecto.

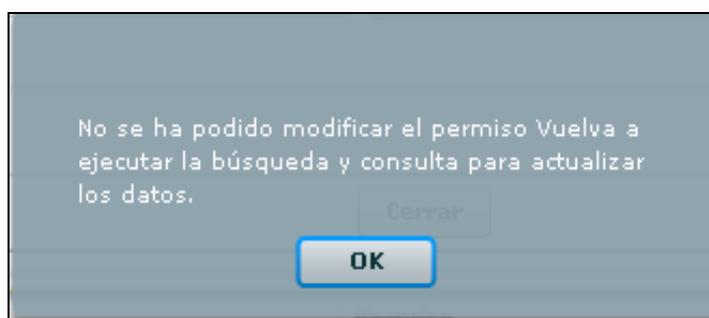


Figura 49. Mensaje de error de la operación realizada

11.1.2 Mantenimiento General

El proceso que cuelga de la línea de menú de la barra de herramientas de *Mantenimiento General*, como podemos ver en la Figura 50, se trata de un procedimiento que se realiza de forma transparente al usuario.



Figura 50. Opción de mantenimiento automático de los procesos

Este procedimiento consiste en traspasar los datos de los procesos que todavía se encuentran activos en la aplicación, pero cuya fecha de baja es anterior a la actual, a tablas históricas para mantener constancia de ellos, aunque ya no serán visibles para los usuarios desde la aplicación.

Este proceso se lanza para los procesos agrupados por su el tipo, es decir, se lanza el proceso para un tipo de procesos y se hará el mantenimiento explicado anteriormente para todos los procesos que sean del tipo seleccionado. Para ello al seleccionar la línea de menú asociada

para tal fin se abrirá una lista de valores de todos los tipos de procesos existentes en ese momento tal y como se muestra en la Figura 51.

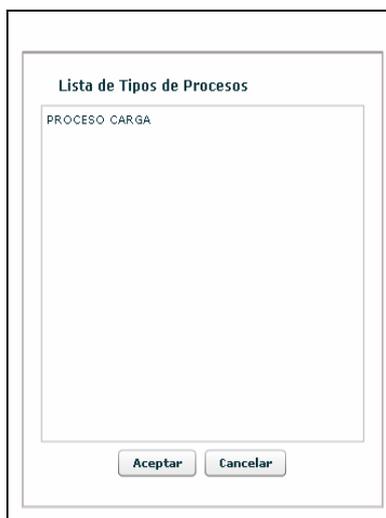


Figura 51. Lista de los distintos tipos de procesos dados de alta en ese momento

Seleccionando el tipo de proceso deseado sobre el que se quiere realizar el mantenimiento y pulsar el botón de *Aceptar*, tal y como vemos en la Figura 52, la aplicación quedará en espera mientras se completa el proceso.

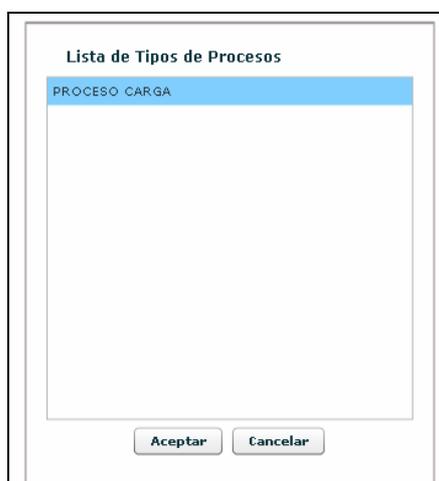


Figura 52. Detalle de selección de un tipo de proceso

En el caso de no existir ningún proceso activo en ese momento del tipo de proceso seleccionado, se mostrará un mensaje tal y como en la Figura 53 donde se indica que no existen procesos pendientes de dar de baja, en otro caso el mensaje será de confirmación satisfactoria.

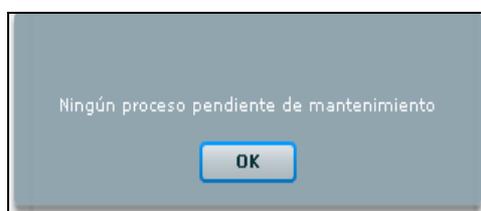


Figura 53. Mensaje de proceso de mantenimiento de procesos completado

11.1.3 Definición de Procesos

Bajo esta línea de menú encontramos dos submenús que podemos visualizar en la Figura 54 : *Tipos de Procesos* y *Procesos*.



Figura 54. Detalle del menú de gestión de procesos

A su vez, estas líneas contienen otros tres submenús idénticos pero cuyas acciones son diferentes como vemos en la Figura 55 .

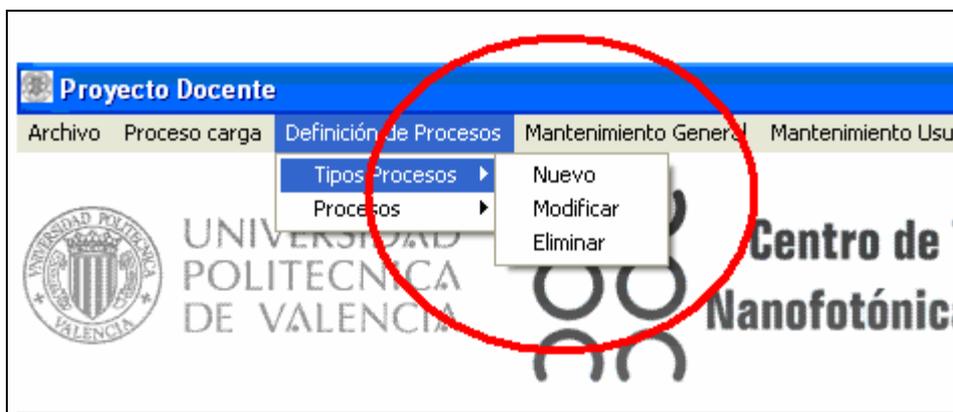


Figura 55. Detalle del submenú de Tipos de Procesos (arriba) y de Procesos (abajo)

11.1.3.1 Gestión de Tipos de Procesos

En el caso de los mantenimientos de tipos de procesos tenemos las opciones de dar de alta un nuevo tipo, modificar uno ya existente o borrarlo. Para los dos primeros casos, al seleccionar

estas opciones de las líneas del submenú correspondiente nos encontramos con una pantalla tal y como podemos apreciar en la Figura 56.

The screenshot shows a web form titled "Tipos de Procesos". It contains two input fields: "Codigo" with a numeric value of 0, and "Titulo del Tipo de Proceso" which is empty. Below the fields are two buttons: "Guardar" and "Cancelar".

Figura 56. Pantalla de alta y modificación de Tipos de Procesos

- **ALTA DE UN TIPO DE PROCESO**

Si se desea dar de alta un nuevo tipo de proceso se deberá elegir la opción de la barra de herramientas para tal fin, lo que hará que se abra una pantalla tal y como se muestra en la Figura 57.

The screenshot shows the same "Tipos de Procesos" form. The "Codigo" field now contains the value 0 and is highlighted with a red border. The "Titulo del Tipo de Proceso" field contains the text "Prácticas_tipo_1" and is highlighted with a blue border. The "Guardar" and "Cancelar" buttons are still present.

Figura 57. Alta de un nuevo tipo de proceso

En esta figura podemos apreciar que se dispone de un campo con un código numérico no editable y que, por tanto, no puede ser modificado. Este valor tomará valor aleatorio al ser introducido un nuevo registro en la BD y será único para cada registro. El único dato que debemos introducir es el título para el nuevo tipo de proceso teniendo en cuenta que el valor que introduzcamos aquí será el que se mostrará al resto de usuarios cuando se les de permiso de acceso a éste en la barra de herramientas para cargar los procesos que cuelguen de este tipo, por eso se recomienda que sea breve pero distintivo.

Tras haber introducido un título y haber pulsado el botón de guardar, si no se ha producido ningún error, nos mostrará el mensaje que se indica en la Figura 58.

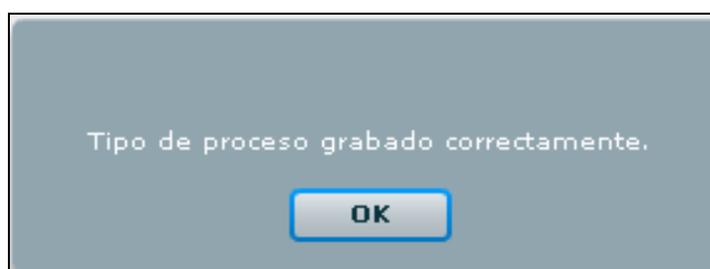


Figura 58. Mensaje de confirmación de proceso realizado correctamente

Tras cerrar el cuadro del mensaje anterior, que nos confirma que todo ha ido correctamente, observamos que en la pantalla del tipo de proceso, se ha actualizado el campo del código del tipo de proceso recuperando de BD el código que le ha asignado a este tipo. Al mismo tiempo, comprobamos que ha desaparecido el botón de *Guardar*, ya que se ha decidido realizar un comportamiento dinámico en los botones de estas pantallas para así obligar a los usuarios a actualizar los datos que se les presenta por pantalla cada vez que realice alguna modificación de estos volviéndolos a recuperar de BD si se desea volver a actualizarlos.

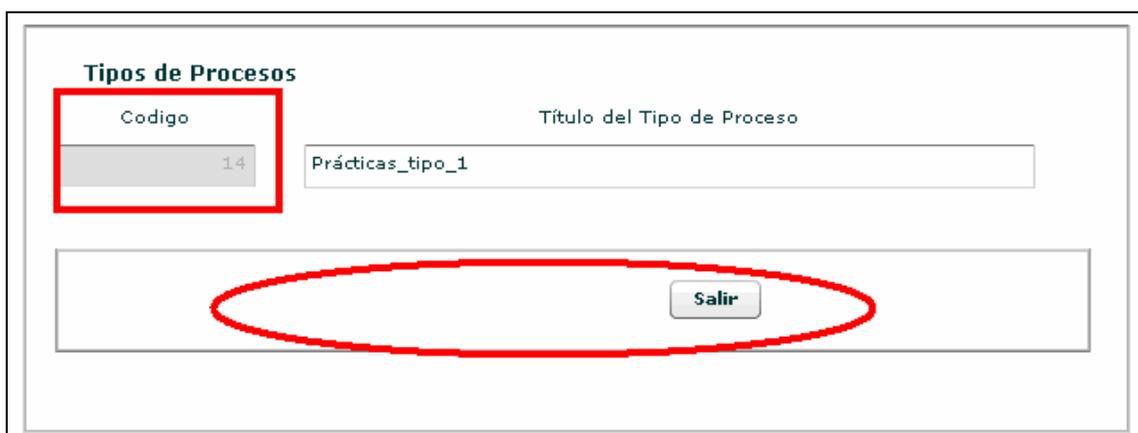


Figura 59. Detalle del estado de la pantalla tras guardar un nuevo tipo de proceso

Se debe tener presente, que una vez finalizada la introducción de un nuevo tipo, se les debe otorgar algún tipo de permiso sobre este tipo de proceso a los usuarios que se quiera que tengan permiso sobre los procesos que cuelguen de este tipo.

- **MODIFICACIÓN DE UN TIPO DE PROCESO**

Para tal fin, seleccionando la línea del submenú correspondiente, se nos abrirá la lista de valores de los diferentes tipos de procesos grabados en BD como se muestra en la Figura 60.

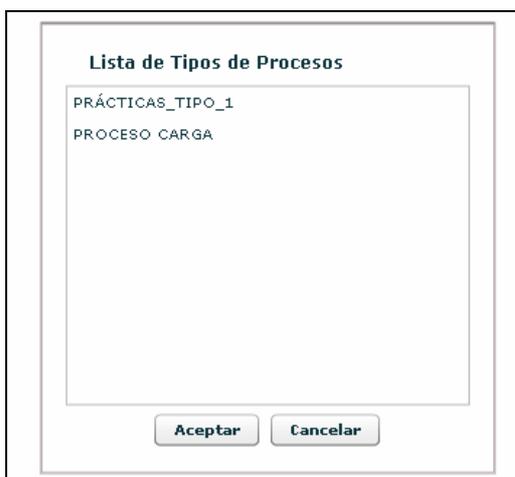


Figura 60. Lista de tipos de procesos

Tras seleccionar el tipo de proceso que deseamos modificar, el cual quedará marcado como se aprecia en la Figura 61, y pulsar el botón Aceptar, se nos cargará una pantalla igual a la anterior pero con los datos referentes al tipo del proceso que deseamos modificar según se muestra en la Figura 62.

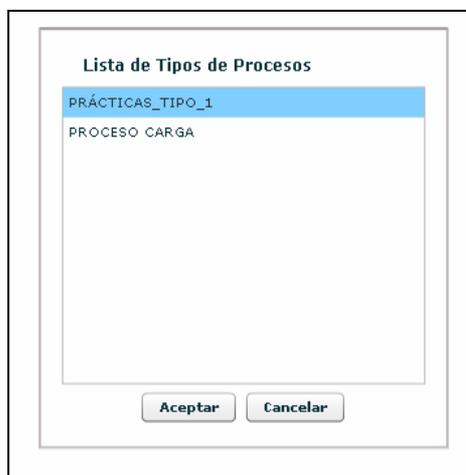


Figura 61. Detalle de selección de tipo de proceso

En esta pantalla apreciamos que no podemos modificar el código recuperado de BD del tipo seleccionado permitiéndonos modificar tan sólo el título de éste.

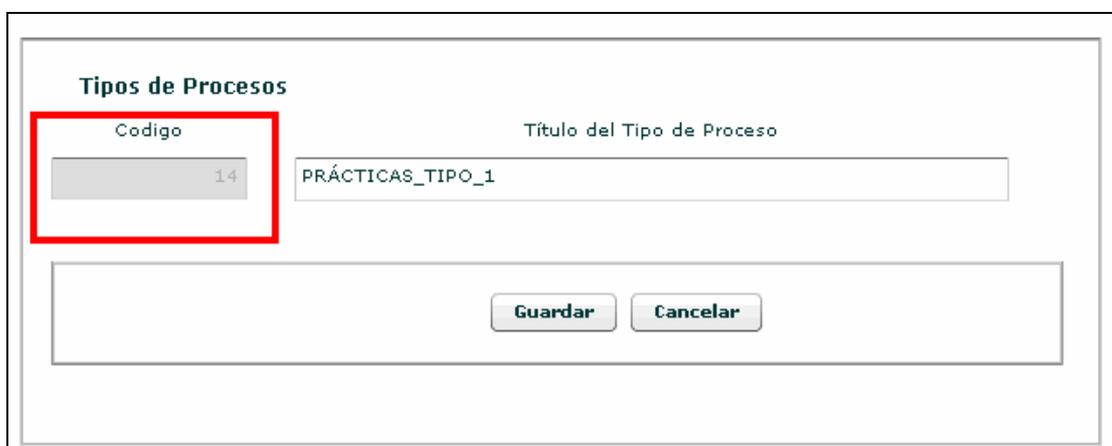


Figura 62. Modificación de un tipo de proceso vigente

Tras realizar los cambios pertinentes y pulsar el botón de Guardar se nos mostrará el mensaje de confirmación tal como presenta la Figura 63.



Figura 63. Mensaje de confirmación de modificación de tipo de proceso correcta

- **ELIMINACIÓN DE UN TIPO DE PROCESO**

En el caso de desear eliminar un tipo de proceso, se nos volverá a mostrar la lista de valores de los tipos existentes tal y como veíamos en la Figura 61. Seleccionando en esta lista el tipo que deseamos eliminar y pulsando sobre el botón Aceptar, la aplicación quedará en proceso de ejecución mientras realiza el proceso correspondiente. Una vez finalizado se mostrará el mensaje de confirmación de la Figura 64.

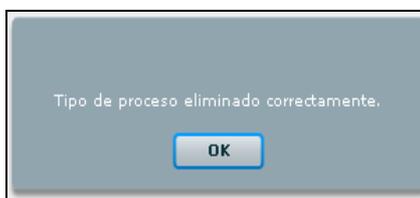


Figura 64. Mensaje de confirmación de eliminación de tipo de proceso correcta

11.1.3.2 Gestión de los Procesos

En el caso de los mantenimientos de los procesos tenemos las opciones de dar de alta uno nuevo, modificar uno ya existente o borrarlo. Para los dos primeros casos, al seleccionar estas opciones de las líneas del submenú correspondiente nos encontramos con una pantalla tal y como podemos apreciar en la Figura 65.

Figura 65. Pantalla de alta y modificación de un proceso

- **ALTA DE UN NUEVO PROCESO**

Si se desea dar de alta un nuevo proceso, deberemos seleccionar la línea del menú de herramientas correspondiente para tal efecto. Esto cargará una pantalla como la que se indicaba en la Figura 65.

En esta pantalla podemos observar, como apreciamos en la Figura 66, que existe un campo bajo la denominación de *Código del Proceso* que no es editable, en éste se indicará un valor numérico aleatorio que, al igual que en los tipos de procesos, será asignado al introducir un nuevo registro en la BD y que será único para cada registro.



Figura 66. Detalle de generación automática de código identificativo del proceso

También apreciamos, como se señala de forma más individualizada en la Figura 67, que existe otro campo el cual tampoco nos permite introducir modificaciones, como es la Fecha de alta del Proceso, que contendrá la fecha actual del sistema que indica la fecha en la que es creado el proceso.


 A screenshot of a web form showing a field labeled "Fecha de alta del Proceso" with the value "2010-10-9". The field is highlighted with a red rectangular border.

Figura 67. Detalle de la fecha de alta del proceso

Valores que debe ser introducido por el usuario son algunos como el Tipo de proceso, el Título del Proceso y la Descripción del Proceso.

El Tipo de proceso debe seleccionarse de la lista de los tipos existentes en ese momento como apreciamos en la Figura 68. El título y descripción serán introducidos por el usuario teniendo en cuenta que el título será el valor que se mostrará al resto de usuarios en las distintas listas de procesos que se muestren en la aplicación por lo que se recomienda que sea cuanto más breve y descriptivo mejor. El campo descripción es un campo que acepta hasta 255 caracteres por lo que si se cree oportuno introducir algún texto descriptivo del proceso se recomienda que sea en este campo donde se haga.


 A screenshot of a web form titled "Procesos". It contains three main sections: "Tipo de Proceso" with a list box showing "PRÁCTICAS_TIPO_1" and "PROCESO CARGA"; "Titulo del Proceso" with a text input field containing "Proceso 1 - Tipo 1"; and "Descripción del Proceso" with a text area containing "Este es el primer proceso ha realizar del primer tipo de prácticas".

Figura 68. Detalle de selección del tipo de proceso, título y detalle en el proceso de alta de un proceso

Otro de los valores obligatorios que se debe introducir es el documento asociado al proceso que se corresponde con el campo que se muestra en la Figura 69.

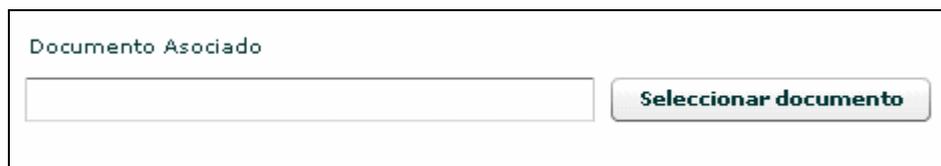

 A screenshot of a web form showing a field labeled "Documento Asociado" with an empty text input box and a button labeled "Seleccionar documento".

Figura 69. Campo Documento Asociado a un proceso

En este campo se deberá rellenar con el documento, como su especificación indica, asociado al proceso. Será este documento el que será en concreto el proceso, es decir, por ejemplo en el caso de tratarse del proceso de una práctica de laboratorio, será el documento que contendrán las preguntas que el usuario tiene que responder. El documento asociado debe tener una extensión *.pdf. Puede introducir este valor desde teclado indicando la ruta completa donde se encuentre dicho documento en su local, por ejemplo C:\Documents and Settings\All Users\Documentos\documentoPrueba.pdf, o emplear el botón asociado para tal efecto de Seleccionar documento el cual al ser pulsado abrirá la ventana del sistema de búsqueda de un archivo como vemos en la Figura 70. Se recomienda que al documento asociado se le asigne como nombre el título del proceso al que es asociado o de manera similar, ya que cuando los usuarios tengan que seleccionar un proceso a responder lo harán eligiendo dentro de una lista de los documentos asociados a los respectivos procesos y no haciendo referencia al título del proceso.

Se ha diseñado de esta manera debido a que podría, en un momento dado, darse de alta varios procesos con el mismo título y del mismo tipo siempre y cuando tengan documentos asociados distintos. Además podrá también así reemplazarse un proceso durante todo un curso lectivo modificando tan solo el documento asociado siempre y cuando se elija adecuadamente la fecha de baja del proceso y el usuario tendrá constancia que ha sido actualizado el proceso.

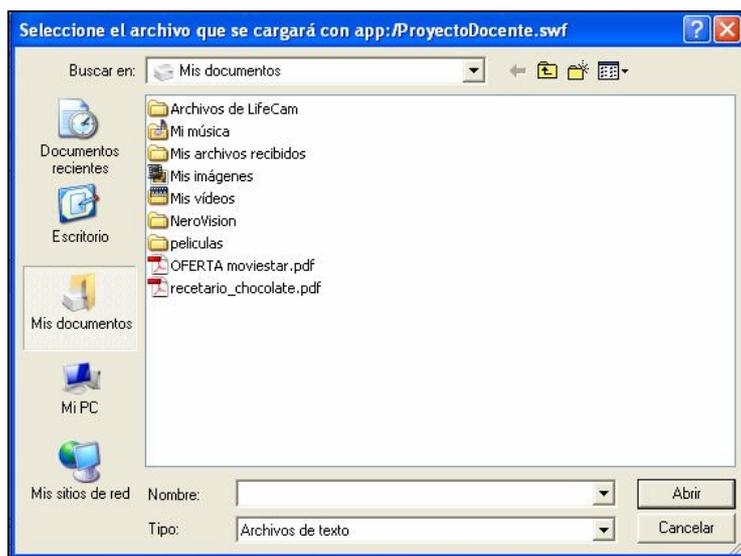


Figura 70. Ventana de selección del documento adjunto asociado al proceso

Deberá también completarse la dirección de correo electrónico de la persona responsable del proceso al cual serán enviadas por este método las soluciones de los usuarios que vayan resolviendo este proceso tal y como se indica en la ayuda de campo asociada a este como podemos apreciar en la Figura 71.

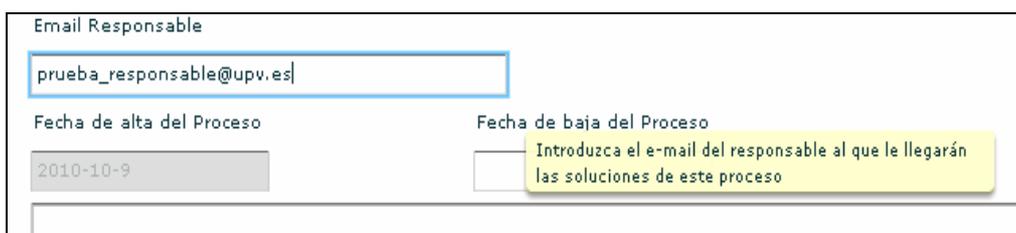


Figura 71. Ayuda del campo del responsable del proceso

Este campo presenta la validación de si se introduce un valor correcto o en caso contrario el valor introducido no presenta la estructura correcta de una dirección de correo, como podemos apreciar en la Figura 72.

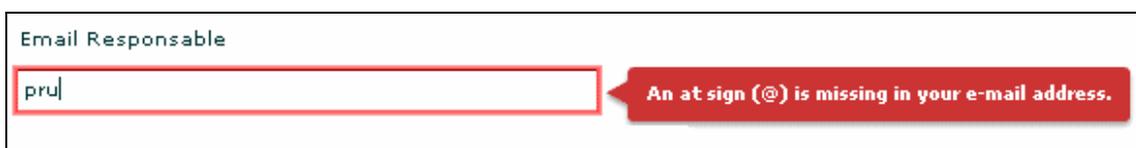


Figura 72. Validación del campo Email Responsable

También será obligatorio introducir una fecha de baja del proceso en formato 'YYYY-MM-DD' como también se especifica en la ayuda del campo asociado y puede observarse en

la Figura 73. Hay que tener en cuenta que el valor de esta fecha indica hasta qué día estará vigente este proceso teniendo en cuenta que si se lanza el proceso del Mantenimiento General un día posterior al indicado en este campo, llevará este proceso al histórico dejando de ser visible para los usuarios.

The screenshot shows a form field titled "Fecha de baja del Proceso". Below the field is a yellow tooltip with the text "Introduzca la fecha de baja de este proceso en formato yyyy-mm-dd". At the bottom of the tooltip are two buttons: "Guardar" and "Cancelar".

Figura 73. Mensajes de ayuda del campo de la fecha de baja

Este campo, al igual que el anterior también presenta validación para el formato de la fecha introducida como comprobamos en la Figura 74.

The screenshot shows a form field titled "Fecha de baja del Proceso". The field is empty and has a red border. To the right of the field is a red error message box that says "This field is required." with an arrow pointing to the field.

Figura 74. Detalle de validación de la fecha de fin

Otros dos campos a rellenar, como apreciamos en la Figura 75, son el número de preguntas de las que consta el proceso y, por tanto, el documento asociado a éste y la opción que posibilita al usuario, una vez finalizado la resolución del proceso, adjuntar documentos adicionales que serán adjuntados junto a la solución en el e-mail que será enviado al responsable.

The screenshot shows a form with two main sections. The first section is titled "Nº de Preguntas" and contains a text input field with the number "5" inside. The second section is titled "¿Posibilidad de adjuntar Ficheros?" and contains the text "Marque el check en el caso de que sí se permita adjuntar ficheros" and a checked checkbox.

Figura 75. Detalle de parámetros de la definición de procesos

Hay que tener en cuenta que tan importante es asociar el documento correcto al proceso como indicar el número correcto de preguntas de las que consta, ya que este número marcará el número de pasos/preguntas por los que los usuarios tendrán que pasar y completar antes de que las soluciones sean enviadas al responsable.

Finalmente, una vez completada la introducción de todos los datos y pulsando el botón de Guardar, si todo ha ido correcto al grabar los datos se mostrará el mensaje de confirmación que aparece en la Figura 76.



Figura 76. Mensaje de confirmación de proceso realizado correctamente

- **MODIFICACIÓN DE UN PROCESO**

Si nuestro objetivo es modificar un proceso ya existente, para tal fin deberá seleccionarse la línea del menú de herramientas para tal fin cuya acción hará que se abra una lista de valores vigentes, es decir, cuya fecha de baja sea posterior al momento actual del sistema, como vemos en la Figura 77.

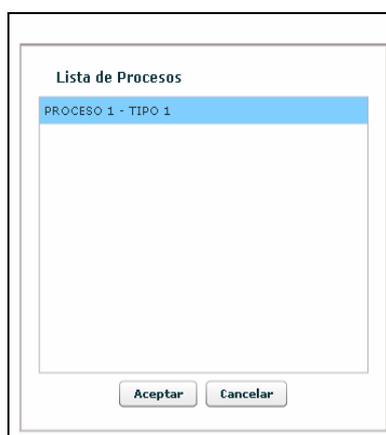


Figura 77. Lista de procesos vigentes a esa fecha

Tras seleccionar el proceso que se desea modificar y pulsar el botón Aceptar, se cargará una pantalla como la indicada anteriormente en la Figura 65, con los datos del proceso seleccionado permitiéndose modificar todos los campos, a excepción de la fecha de inicio del proceso y de su código, como se ha indicado en el apartado de ALTA DE UN NUEVO PROCESO.

Una vez insertados todos los cambios se deberá pulsar el botón de Guardar para actualizar estos cambios en BD y si todo ha ido correctamente volverá a mostrarse un mensaje de confirmación al igual que en el alta de un nuevo proceso y que se muestra en la Figura 76.

- **ELIMINACIÓN DE UN PROCESO**

Nuevamente, seleccionando la línea del menú de herramientas del submenú correspondiente, será cargada la lista de los procesos vigentes en ese momento y que se muestra en el punto anterior en la Figura 77.

Seleccionando el proceso deseado y nuevamente pulsando el botón Aceptar, la aplicación quedará en proceso de ejecución realizando el proceso. Una vez haya finalizado, si todo ha ido correcto, nuevamente volverá a presentar un mensaje de confirmación.

Se debe tener en cuenta que el borrado de un proceso implica el borrado de toda constancia de éste en BD, siendo diferente a darlo de baja con el procedimiento de Mantenimiento General, el cual lo que hace es traspasar los datos a históricos.

11.2 Manual de Usuarios

Tras realizar los previos pasos descritos inicialmente de instalación de la aplicación y acceso a ella tras identificarse, si tiene permiso de acceso a ella, se le cargará en la barra de herramientas diversas opciones personalizadas cuya etiqueta harán referencia a los distintos tipos de procesos sobre los que se le ha otorgado permiso, por ejemplo en la Figura 78, al usuario que ha accedido a la aplicación se le ha otorgado permisos sobre el tipo de proceso *Prácticas_tipo_1* entre otros.

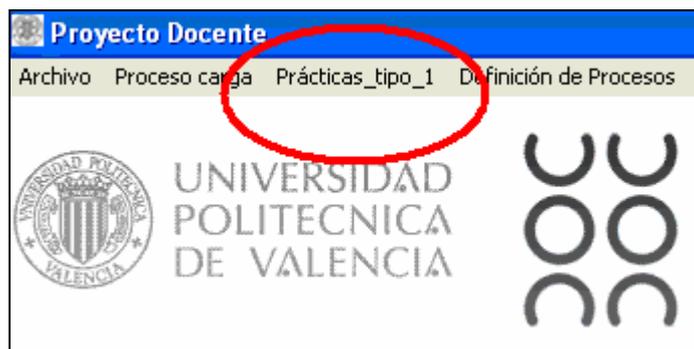


Figura 78. Detalle de la barra de herramientas con el nuevo tipo de proceso

Si se hace clic sobre esta opción, se muestra un menú con dos opciones: *Cargar* y *Limpiar*, tal como podemos ver en la Figura 79.



Figura 79. Detalle del menú de los tipos de procesos

La opción de *Cargar* permite cargar una lista con los distintos procesos de este tipo y la opción de *Limpiar* cierra en el caso de existir algún proceso abierto en la aplicación la pantalla de este. En el caso de elegir la opción de cargar los procesos de este tipo y que no exista ningún proceso vigente, se mostrará el mensaje que se muestra en la Figura 80



Figura 80. Mensaje de carga de los procesos de un tipo

En caso contrario se abrirá una lista de valores donde se muestran los distintos documentos asociados a los distintos procesos vigentes en ese momento como se muestra en la Figura 81, que siguiendo con el ejemplo anterior, "introducción_css.pdf" es el documento asociado a un proceso del tipo *Prácticas_tipo_1*.

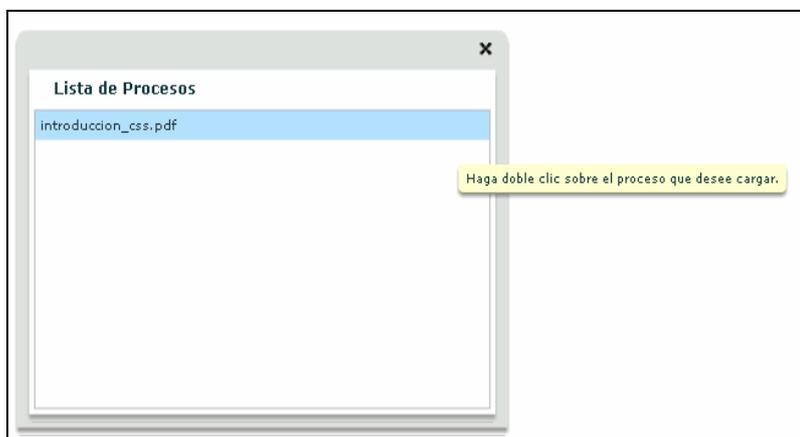


Figura 81. Lista de procesos de un tipo vigentes hasta ese momento

Haciendo doble clic sobre el documento del proceso que se desea cargar se abrirá una ventana adicional con el documento *.pdf seleccionado como se muestra en la Figura 82.

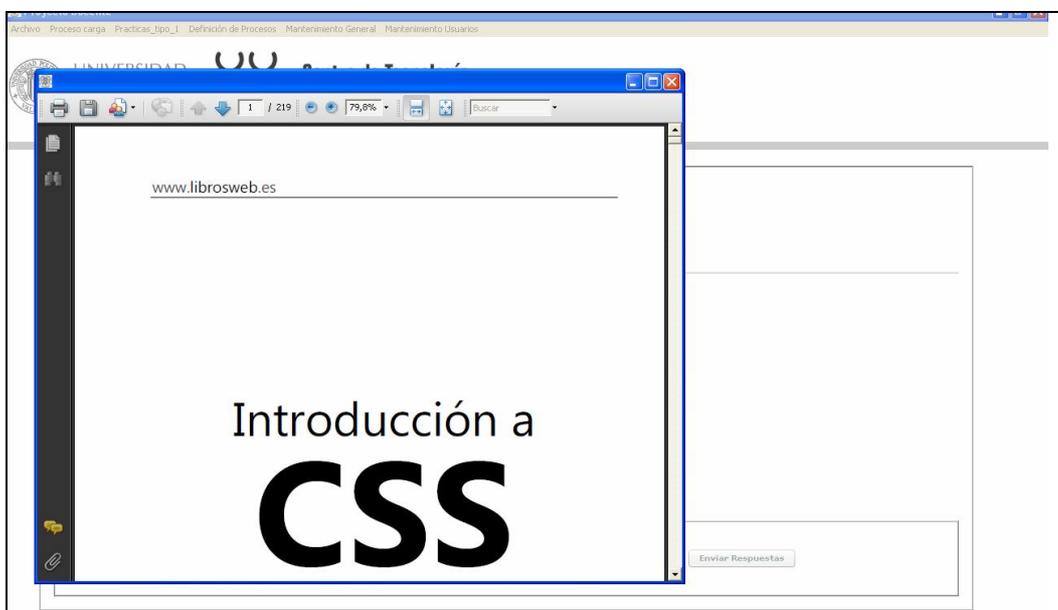


Figura 82. Carga de un proceso y su documento asociado

En la aplicación se abrirá una pantalla donde se contestarán a las preguntas que se muestran en el documento *.pdf de la ventana adicional y que podemos apreciar en la Figura 83

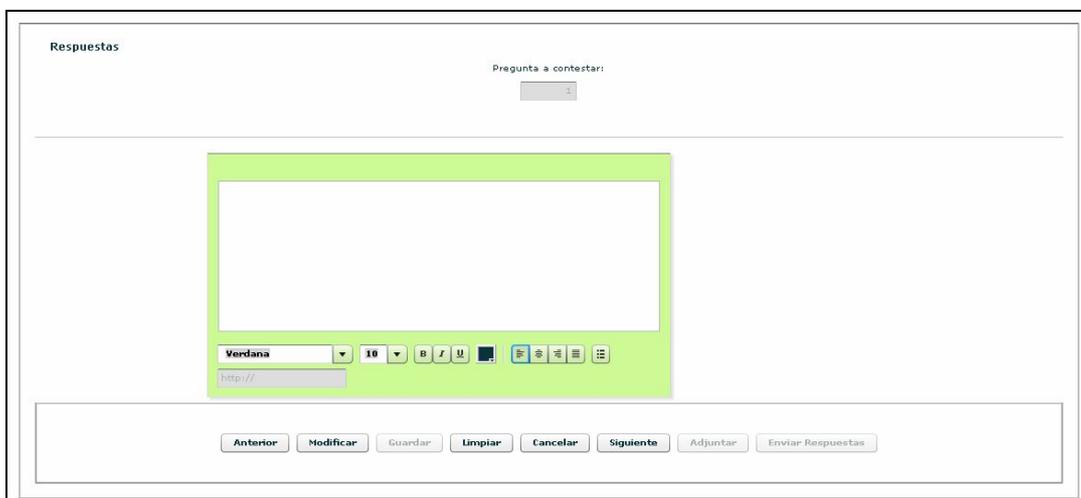


Figura 83. Ventana de respuestas de un proceso

En esta pantalla podemos observar, en la parte superior, un cuadro como apreciamos en la Figura 84, que indica el número de pregunta que en ese momento estamos contestando.

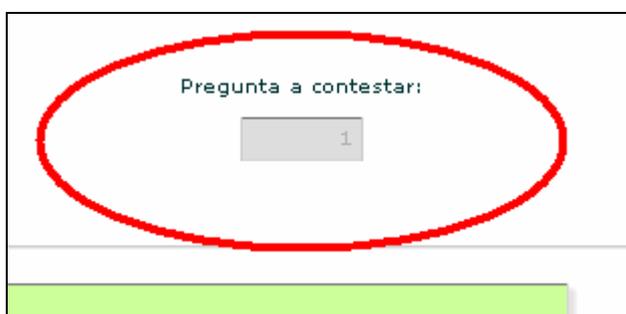


Figura 84. Detalle de indicación de pregunta a contestar

En el centro de esta pantalla se muestra un cuadro de inserción de texto como vemos en la Figura 85 que será ahí donde pondremos nuestra respuesta a la pregunta del documento *.pdf de número al que se indica en el cuadro citado en el párrafo anterior.

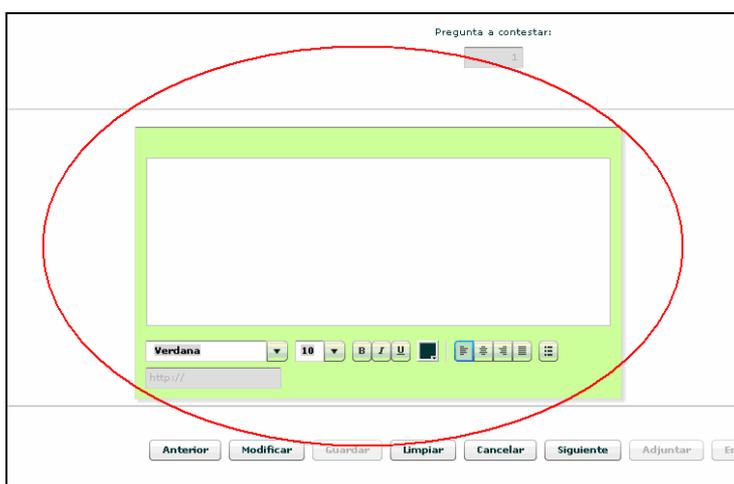


Figura 85. Cuadro de texto donde se insertará la respuesta

Tras insertar nuestra respuesta y pulsar el botón de guardar, si todo ha ido correctamente, se mostrará el mensaje de confirmación como se muestra en la Figura 86.



Figura 86. Mensaje de validación de respuesta guardada.

Tras aceptar el mensaje de confirmación que indica que se ha grabado correctamente, observamos el cuadro de texto que ha quedado de forma no editable y ha quedado deshabilitado el botón de *Guardar* y habilitado el de *Modificar* para que podamos cambiar nuestra respuesta si deseamos en cualquier momento. También estarán deshabilitados los botones de *Adjuntar* y *Enviar Respuesta* en el caso de que no nos encontremos en la última pregunta del proceso. Todo esto podemos contemplarlo en la Figura 87.



Figura 87. Detalle de activación de los distintos botones

En el caso de encontrarnos en la última pregunta y haber grabado ya la respuesta, se habilitará el botón de *Enviar Respuesta* y en el caso de que el proceso permita enviar documentos extra adjuntos a su respuesta también se habilitará el botón de *Adjuntar*, como se muestra en la Figura 88



Figura 88. Habilidad de los botones en la última pregunta

También se puede navegar entre las distintas respuestas ya contestadas con los botones de *Anterior* y *Siguiente*. El botón de *Limpiar* borra lo que haya escrito en el cuadro de texto, siempre y cuando esté habilitado para escritura, y el botón de *Cancelar* cierra la pantalla de respuestas.

Si se desea adjuntar un documento extra a la respuesta, si pulsamos el botón citado anteriormente, se abrirá la ventana de navegación del sistema que se muestra en la Figura 89.

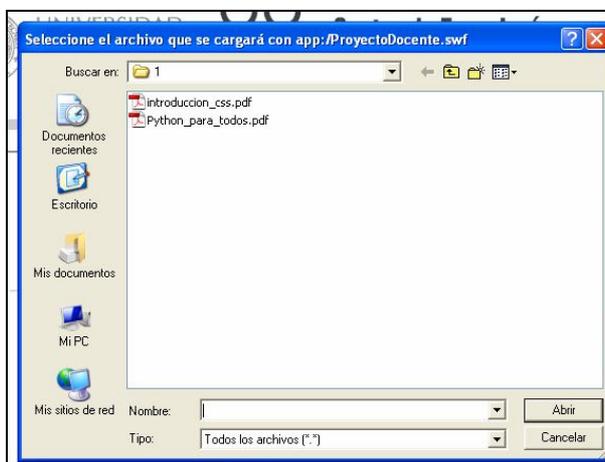


Figura 89. Ventana del sistema asociada a adjuntar documentos a las respuestas

Tras seleccionar el/los documento/s extra que se desee adjuntar, junto al bloque de la respuesta, se mostrará una lista con estos documentos seleccionados como vemos en la Figura 90. En esta lista se presenta también un botón de *Eliminar* que, tras seleccionar un documento de esta lista, se eliminará de ella, por lo que no se adjuntará.

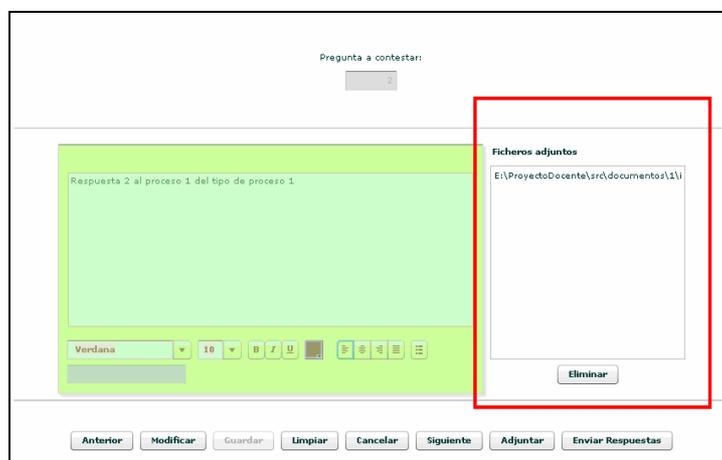


Figura 90. Lista de documentos asociados a las respuestas

Una vez se haya llegado a este punto y habiendo considerado que no se desea realizar ninguna modificación, si se pulsa el botón de *Enviar Respuestas* todas sus respuestas junto a el/los documento/s adjuntos, en el caso de existir o presentar esta opción, se envían vía e-mail al responsable del proceso.

La aplicación quedará en proceso de ejecución hasta que el correo electrónico sea enviado mostrando entonces el mensaje de confirmación que se muestra en la Figura 91.

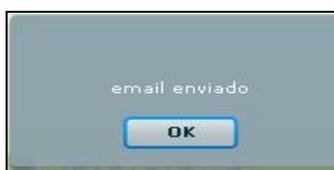


Figura 91. Mensaje de confirmación de respuestas enviadas correctamente

Tenga en cuenta que no es obligatorio contestar a todas las respuestas de las que consta el proceso en ese mismo preciso momento sino que puede salir de la aplicación y en cualquier otro momento que acceda a ella y cargue el proceso, se cargarán las distintas respuestas ya contestadas hasta ese momento, pudiendo volver a modificarlas.

Todas estas respuestas serán almacenadas en local en el directorio:

C: / Documents and Settings/usuario/respuestas/tipo_proceso/proceso/num_pregunta/num_pregunta.doc

y cuando sean enviadas serán agrupadas en el documento:

C:/Documents and Settings/usuario/respuestas/tipo_proceso/proceso/general.doc

que será este el que sea enviado al responsable del proceso.

12. PRESUPUESTO

12. PRESUPUESTO

Gracias a que todo el proyecto ha sido desarrollado e implementado con software libre, hemos conseguido reducir considerablemente el presupuesto empleado para tal fin, ya que hemos evitado pagar las correspondientes licencias de uso y desarrollo.

Se ha realizado una estimación de horas dedicadas a cada tarea de desarrollo en el proyecto para poder realizar un cálculo aproximado de un presupuesto. Las horas asignadas a cada componente se desglosan de la siguiente forma:

- Tarea de jefe de proyecto: 100 horas
- Tarea de analista: 60 horas
- Tarea de diseñador: 90 horas
- Tarea de programador: 250 horas
- Tarea de administrador BD: 10 horas

El precio por hora de cada recurso es algo que se determina por la parte que realiza el proyecto, en función de los honorarios del recurso y el beneficio que se obtendrá de éste. Así pues, no es un baremo estándar. Nosotros realizaremos una estimación sobre estos valores y basándonos en presupuestos consultados a empresas dedicadas al desarrollo de software empresarial, como puede ser el caso de la empresa Sinergia Tecnológica[28]:

- Rol de jefe de proyecto (precio unitario / hora): 30 €
- Rol de analista (precio unitario / hora): 25€
- Rol de diseñador (precio unitario / hora): 19 €
- Rol de programador (precio unitario / hora): 15 €
- Rol de administrador BD (precio unitario / hora) : 24 €

Con lo que el coste final del proyecto quedaría en 10.200 €

13. CONCLUSIONES

Las nuevas tecnologías de la información se encuentran en continua evolución con los propósitos de conseguir acortar las distancias geográficas entre usuarios, ofrecer herramientas que cada día sean más fáciles de usar, más intuitivas y con acceso para todos. Herramientas que están consiguiendo romper con las barreras de adaptación entre distintos dispositivos y distintos sistemas operativos. Un amplio abanico de posibilidades y recursos que rozan el límite de lo ilimitado. Herramientas desarrolladas para una sociedad cada vez más arraigada a las tecnologías, de la que ya es imposible romper estos lazos de unión, en la que se premia el conocimiento y que se mueve a un ritmo de vida vertiginoso.

Con este proyecto hemos conseguido desarrollar e implementar una aplicación, por medio de tecnologías relativamente modernas, pero a la vez desarrollada por una compañía lo suficientemente robusta como es Adobe®, que no tiene otro objetivo principal que facilitar la labor educativa desde dos puntos de vista: la enseñanza por parte del profesorado y el aprendizaje por parte del alumnado.

Una sola aplicación e interfaz de usuario y una infinidad de posibles escenarios adaptables a cada personaje que accede a él desde cualquier lugar, rompiendo las limitaciones de las distancias, de la diversidad de dispositivos y de sistemas operativos.

Una aplicación que puede englobarse dentro del grupo de las tecnologías eLearning pues se trata de una aplicación abierta donde las posibilidades que con ella se pueden contemplar no dependen de otro factor mas que los usos que el usuario quiera darle. Una aplicación dinámica, adaptativa, con datos globales a los que todos los usuarios tienen acceso, flexible y de fácil uso para los usuarios, sin engorrosos procedimientos que entorpezcan más la labor para la que ha sido desarrollada.

Según los objetivos que se propusieron inicialmente para este Trabajo Final de Carrera, se han alcanzado a través de:

- Herramientas que permitan la implementación de flujos de trabajo en entornos gestionados, como el proceso docente: se ha alcanzado con el desarrollo de una aplicación RIA en el escritorio bajo el motor de ejecución de Adobe® AIR®, dinámica, adaptada a las nuevas tecnología y con la posibilidad de su instalación en un rango de diversos dispositivos móviles considerablemente amplio y con la posibilidad de nuevas ampliaciones.
- Integración de la información sobre la que aplicar flujos de trabajo: este objetivo ha sido alcanzado en el momento en el que los procesos, independientemente del usuario que los defina, todos estarán basados en un modelo de estructura idéntica integrada en la aplicación, pudiendo reutilizarse datos previamente definidos por otros usuarios.
- Implementación de flujos de trabajo asociados a documentación docente: se les concede a los usuarios el grado de libertad de ser ellos quienes definan los *tipos de procesos*, los *procesos* y los *pasos* por los que éstos deben pasar, es decir, serán ellos quienes impongan el flujo de trabajo.

14. LÍNEAS FUTURAS

Lejos de quedar desfasada la aplicación desarrollada, pueden contemplarse diversas líneas de actuación futuras. Es verdad que Adobe lanza al mercado Adobe AIR con la intención de que las aplicaciones que sean implementadas en este entorno puedan ser instaladas en el mayor número de variedad de dispositivos móviles como iPhone, Android, ordenadores portátiles con Windows o Linux e incluso se comentaba que estaba en proceso de desarrollo una versión para estas aplicaciones también puedan ser instaladas con Windows Mobile. Según un análisis de Strategy Analytics de enero de 2010, ya que Adobe AIR es ejecutado sobre Flash Player, se espera que más de 250 millones de teléfonos inteligentes sean compatibles con Flash Player a finales de 2012. Las plataformas de dispositivos móviles que serán compatibles con Flash Player incluyen Android, la plataforma BlackBerry®, Symbian® OS, Palm® webOS y Windows Mobile®.

Pero con las experiencias, tanto la de otros usuarios que han trabajado con Adobe AIR como la mía propia, hemos podido observar que la adaptación de estas aplicaciones en dispositivos móviles, que cuentan con pantallas extremadamente reducidas, como por ejemplo iPhone, no es tan idílica y bucólica como se intenta hacer ver. Es por ello que Adobe tiene pensado sacar al mercado un software al que ha denominado 'Hero' el cual, según informan, funcionará al estilo de Flex Builder, pero permitiendo un diseño más optimizado para estos tipos de dispositivos móviles. Por ello, una de las líneas de actuación futuras sería conseguir una mejor adaptación de la aplicación para estos dispositivos por medio del uso de 'Hero' llegando a conseguir una perfecta navegación web de una aplicación expresiva, así como contenidos y vídeos de alta definición (HD) a través de pantallas, incluidas las de los nuevos dispositivos de tableta, teléfonos inteligentes, netbooks, libros inteligentes, ordenadores de escritorio y otros dispositivos electrónicos de consumo.

Además, podrían añadirse implementaciones de muchas más opciones como por ejemplo, llevar un control de los usuarios que han cargado un proceso o han enviado sus soluciones para poder sacar estadísticas; ofrecer la posibilidad de procesos cuyas soluciones sean de tipo test y las propias preguntas sean mostradas y resueltas desde una pantalla sin necesidad de cargar ningún documento de extensión *.pdf o procesos que adjunten videos cargados en la pantalla de la propia aplicación los cuales ayudan tanto de forma visual como auditiva al aprendizaje del alumno.

'ProyectoDocente' se trata de una aplicación sofisticada fuera del navegador del dispositivo móvil, facilitando el trabajo de los usuarios en aquellos momentos en los que no se dispone de una conexión a la red de forma permanente o su velocidad es relativamente lenta pues no se precisa de disponer de ella de forma continua. Bien es verdad que, para poder comenzar a trabajar con ella, es preciso disponer de esta conexión para que pueda validar nuestro acceso a ella, en caso contrario no podremos realizar acción alguna. Por eso otra de las líneas de actuación futuras podría ser el contemplar la posibilidad de que la primera vez que el usuario descargue la aplicación y la instale en su dispositivo, se cree al mismo tiempo una base de datos local SQLite y en el momento en que disponga de acceso a la red descargar en ella los datos que estén relacionados con el usuario como puede ser sus permisos, los tipos y procesos a los que se le ha otorgado permisos, y en el momento de volver a acceder a la aplicación y no disponer de acceso a la red ésta se cargue con los datos de la base de datos local y cuando vuelva a disponer de conexión se actualicen los datos de la base de datos, así este caso como mínimo podría optar a las opciones de las que disfrutó la última vez que ejecutó la aplicación.

Podríamos estar nombrando infinidad de posibles actualizaciones de la aplicación, pues como se ha comentado al comienzo de este apartado, las tecnologías y herramientas software se encuentran en continua expansión y deben ser actualizadas si no queremos pecar de tener aplicaciones que han quedado desfasadas y obsoletas. Aun así, con este desarrollo se ha logrado abarcar los requisitos iniciales que nos pidió el cliente: una aplicación dinámica, cada usuario tendrá una aplicación con herramientas personalizadas, y móvil, puede ser instalada en diversos tipos de dispositivos móviles, que les facilite la labor en la gestión de procesos educativos

permitiéndoles definir el flujo de trabajo de estos a través de la definición del número de pasos por los que deben pasar los procesos, al mismo tiempo que a los alumnos se les permite tener acceso a diferentes herramientas educativas desde la distancia sin tener que disponer de una conexión permanente a Internet para poder disfrutar de ellas.

15. BIBLIOGRAFÍA Y REFERENCIAS

15. BIBLIOGRAFÍA Y REFERENCIAS

- [1] <http://www.empleare.com/ranking-web-de-universidades-de-espana.html>
- [2] <http://www.upv.es/organizacion/documentos/est1-Docencia-es.pdf>
- [3] J. Salinas (2000). El rol del profesorado en el mundo digital. Ed. L. del Carmén, pp. 305-320. Simposio sobre la formación inicial de los profesionales de la educación. Universidad de Girona. 2000.
- [4] Julio Cabero y otros, (2003). Las nuevas tecnologías en la actividad universitaria. Píxel-bit. Revista de Medios y Educación. Nº 20. Enero 2003, pp. 81-100.
- [5] Gabriel Kaplún (2005). Aprender y enseñar en tiempos de internet. Formación profesional a distancia y nuevas tecnologías. Montevideo. CINTERFOR/OIT, 2005.
- [6] J. Dalziel, (2004). Open standards versus open source in e-learning. EDUCAUSE Quarterly, 4. 2004. <http://www.educause.edu/ir/library/pdf/EQM0340.pdf>
- [7] B. Schilit, N. Adams, and R. Want. (1994). Context-aware computing applications. IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), Santa Cruz, CA, US: 89-101.
- [8] Guanling Chen, David Kotz, (2000). A Survey of Context-Aware Mobile Computing. Research, Dartmouth College, Hanover, NH, 2000
- [9] Revista COMPU MAGAZINE, Número 51, Octubre '92
Revista COMPU MAGAZINE, Número 50, Septiembre '92
- [10] J. C. Dueñas, J. L. Ruiz, J. Bermejo, J. A. Alonso, C. Acuña, C. Díaz. (2004) “Plataformas abiertas para la provisión de servicios”, XIV JORNADAS TELECOM I+D. Madrid 23,24,25 Noviembre 2004.
- [11] Sudarshan S.; Korth Henry; Silberschatz Abraham, “Fundamentos de bases de datos, 5ª edc.” McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U., ISBN: 8448146441 ISBN-13: 9788448146443, 5 edición (17/02/2006)
- [12] PHP Práctico Autor: Matt Doyle Editorial: Anaya Multimedia Nº Páginas 848 I.S.B.N. 978-84-415-2689-1 Publicación: 2010
- [13] MySQL Autor: Ian Gilfillan Editorial: Anaya Multimedia bNº Páginas: 880 I.S.B.N.: 84-415-1558-1
- [14] Flash, PHP y MySQL. Contenidos dinámicos Autores: Daniel de la Cruz Heras y Carlos Zumbado Rodríguez Editorial: Anaya Multimedia Nº Páginas: 320 I.S.B.N. : 84-415-1609-X
- [15] <http://www.adobe.com>
- [16] <http://www.phpmyadmin.net/>
- [17] <http://www.flex.org>
- [18] <http://www.postgresql-es.org>
- [19] <http://www.firebird.com.mx>
- [20] <http://www.ajpdssoft.com>
- [21] <http://www.php.net>
- [22] <http://www.programacion.com>
- [23] <http://dev.mysql.com>

[24] <http://ajax.sys-con.com>

[25] <http://silverlight.net>

[26] <http://www.asp.net>

[27] <http://www.todobi.com>

[28] <http://www.sinergiatec.com>

16. ANEXOS

ANEXO I. ESTRUCTURA BASE DE DATOS

Estructura de tabla para la tabla `usuario`

```
CREATE TABLE IF NOT EXISTS `usuario` (
  `ID` int(6) NOT NULL AUTO_INCREMENT,
  `DNI` varchar(10) NOT NULL,
  `NOMBRE` varchar(60) NOT NULL,
  `APELLIDO1` varchar(60) NOT NULL,
  `APELLIDO2` varchar(60) NOT NULL,
  `EMAIL` varchar(150) NOT NULL,
  `LOGIN` varchar(50) NOT NULL,
  `PASSWORD` varchar(20) NOT NULL,

  PRIMARY KEY (`ID`),
  UNIQUE KEY `PASSWORD` (`PASSWORD`),
  UNIQUE KEY `LOGIN` (`LOGIN`),
  UNIQUE KEY `DNI` (`DNI`)
);
```

Estructura de tabla para la tabla `documento`

```
CREATE TABLE IF NOT EXISTS `documento` (
  `COD` int(8) NOT NULL AUTO_INCREMENT,
  `DESCRIPCION` varchar(100) NOT NULL,

  PRIMARY KEY (`COD`,`DESCRIPCION`),
  UNIQUE KEY `DESCRIPCION` (`DESCRIPCION`),
  UNIQUE KEY `COD` (`COD`)
);
```

Estructura de tabla para la tabla `permiso`

```
CREATE TABLE IF NOT EXISTS `permiso` (
  `DNI` varchar(9) NOT NULL,
  `PROCESO` int(1) NOT NULL,
  `PERMISO` int(1) unsigned NOT NULL,

  PRIMARY KEY (`DNI`,`PROCESO`,`PERMISO`),
  KEY `PROCESO` (`PROCESO`)
);
```

Estructura de tabla para la tabla `proceso`

```
CREATE TABLE IF NOT EXISTS `proceso` (
```

```

`COD`                int(8) NOT NULL AUTO_INCREMENT,
`TITULO`             varchar(50) NOT NULL,
`DESCRIPCION`        varchar(500) DEFAULT NULL,
`NUM_PREGUNTAS`      int(2) NOT NULL,
`FEC_ALTA`           date NOT NULL,
`FEC_BAJA`           date NOT NULL,
`DOCUMENTO`          int(8) NOT NULL,
`TIPO`               int(8) NOT NULL,
`RESPONSABLE`        varchar(100) DEFAULT NULL,
`ADJUNTOS`           varchar(1) NOT NULL,

PRIMARY KEY (`COD`),
FULLTEXT KEY `TITULO` (`TITULO`)
);

```

Estructura de tabla para la tabla `tipoproceso`

```

CREATE TABLE IF NOT EXISTS `tipoproceso` (
  `COD`                int(8) NOT NULL AUTO_INCREMENT,
  `TITULO`             varchar(50) NOT NULL,

  PRIMARY KEY (`COD`)
);

```

Estructura de tabla para la tabla `tipoprocesoh`

```

CREATE TABLE IF NOT EXISTS `tipoprocesoh` (
  `COD`                int(11) NOT NULL,
  `TITULO`             varchar(60) NOT NULL,
  `FEC_BAJA`           date NOT NULL,

  PRIMARY KEY (`COD`)
);

```

Estructura de tabla para la tabla `documentoh`

```

CREATE TABLE IF NOT EXISTS `documentoh` (
  `COD`                int(8) NOT NULL,
  `DESCRIPCION`        varchar(100) NOT NULL,
  `FEC_BAJA`           date NOT NULL,

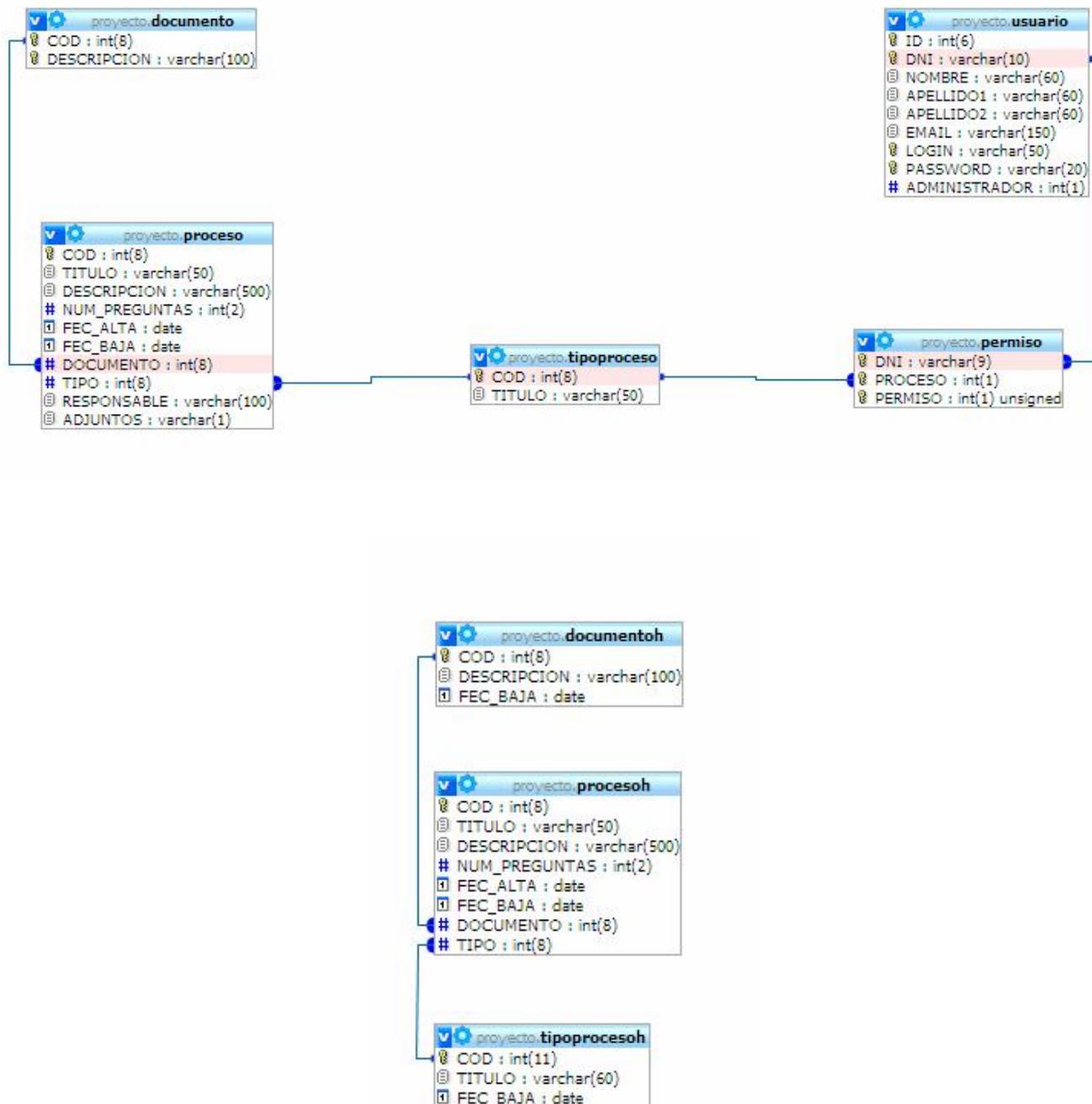
  PRIMARY KEY (`COD`)
);

```

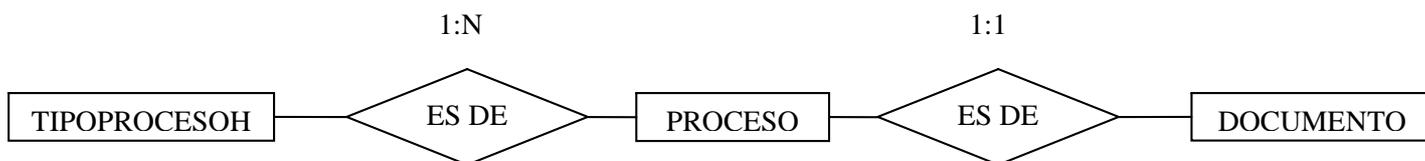
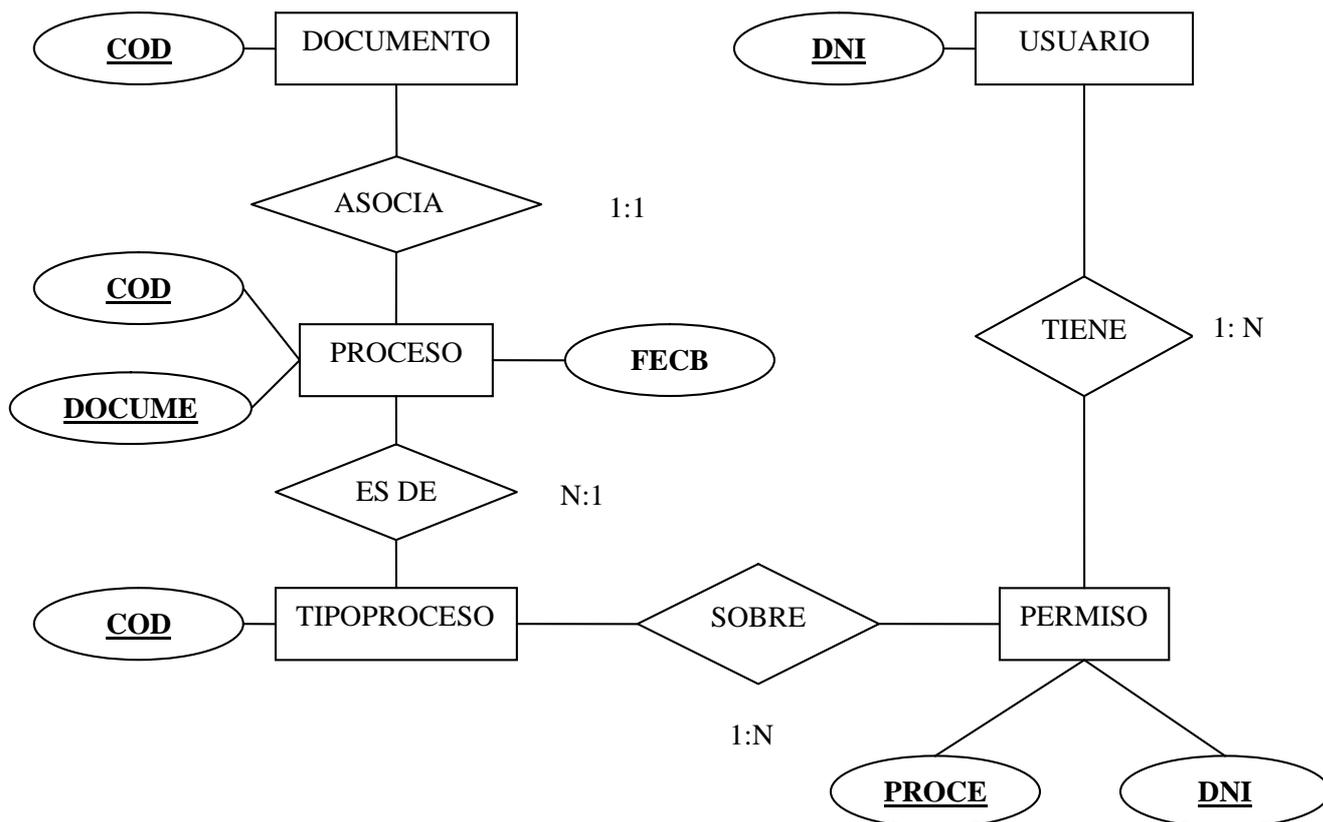
Estructura de tabla para la tabla `procesoh`

```
CREATE TABLE IF NOT EXISTS `procesoh` (  
  `COD` int(8) NOT NULL,  
  `TITULO` varchar(50) NOT NULL,  
  `DESCRIPCION` varchar(500) DEFAULT NULL,  
  `NUM_PREGUNTAS` int(2) DEFAULT NULL,  
  `FEC_ALTA` date NOT NULL,  
  `FEC_BAJA` date NOT NULL,  
  `DOCUMENTO` int(8) NOT NULL,  
  `TIPO` int(8) NOT NULL,  
  
  PRIMARY KEY (`COD`),  
  KEY `TIPO` (`TIPO`)  
);
```


ANEXO II. ESQUEMA RELACIONES BASE DE DATOS



ANEXO III. DIAGRAMA ENTIDAD – RELACION BASE DE DATOS



ANEXO IV. CLASE USUARIO

```

package clases
{
    [Bindable]
    public class Usuario extends Object
    {
        public var procesos:Array=new Array();
        public var permisos:Array = new Array();
        public var procesosDesc:Array = new Array();
        public var esAdministrador:Boolean = false;
        public var email:String;
        public var nombre:String = null;
        public var dni:String = null;

        public function get obtenerProcesos():Array{
            return procesos;
        }
        public function get obtenerProcesosDesc():Array{
            return procesosDesc;
        }
        public function get obtenerPermisos():Array{
            return permisos;
        }

        public function get esAdmin():Boolean{
            return esAdministrador;
        }
        public function get emailUsu():String{
            return email;
        }
        public function get nombreUsu():String{
            return nombre;
        }
        public function get dniUsu():String{
            return dni;
        }

        public function Usuario(e:XMLList):void
        {
            var subXML:XMLList;

            subXML = e.child("procesos");

            for(var j:int=0; j<subXML.length(); j++){
                procesos[j]=subXML[j].child("proceso");
                procesosDesc[j] = subXML[j].child("descripcion");
                permisos[j]=subXML[j].child("permiso");
            }
            email = e.email;
            nombre = e.nombre;
            dni = e.dni;
            esAdministrador = false;

            for (var i:int = 0 ; i< permisos.length; i++){
                if (permisos[i] == 2)
                    esAdministrador = true;
            }
        }
    }
}

```


ANEXO V. CLASE TIPO PROCESO

```
package clases
{
    [Bindable]
    public class TipoProceso extends Object
    {
        public var cod:String=new String();
        public var titulo:String = new String();

        public function get codigo():String{
            return cod;
        }
        public function get tit():String{
            return titulo;
        }

        public function TipoProceso(e:XMLList):void
        {
            var subXML:XMLList;

            subXML = e;

            cod = subXML.child("codigo");
            titulo= subXML.child("descTipo");

        }
    }
}
```


ANEXO VI. CLASE PROCESO

```

package clases
{
    [Bindable]
    public class Proceso extends Object
    {
        public var cod:String=new String();
        public var titulo:String = new String();
        public var descripcion:String = new String();
        public var numPreguntas:String = new String();
        public var fecAlta:String = new String();
        public var fecBaja:String = new String();
        public var documento:String = new String();
        public var tipo:String = new String();
        public var responsable:String = new String();
        public var adjuntos:String = new String();

        public function get codigo():String{
            return cod;
        }
        public function get tit():String{
            return titulo;
        }
        public function get desc():String{
            return descripcion;
        }
        public function get numPreg():String{
            return numPreguntas;
        }
        public function get fecAlt():String{
            return fecAlta;
        }
        public function get fecBaj():String{
            return fecBaja;
        }
        public function get docum():String{
            return documento;
        }
        public function get tip():String{
            return tipo;
        }
        public function get resp():String{
            return responsable;
        }
        public function get adj():String{
            return adjuntos;
        }

        public function Proceso(e:XMLList):void
        {
            var subXML:XMLList;

            subXML = e;

            cod = subXML.child("cod");
            titulo= subXML.child("titulo");
            descripcion= subXML.child("descripcion");
        }
    }
}

```

```
numPreguntas= subXML.child("numPreg");
fecAlta= subXML.child("fecAlta");
fecBaja= subXML.child("fecBaja");
documento= subXML.child("codDocum");
tipo= subXML.child("tipo");
responsable= subXML.child("resp");
adjuntos= subXML.child("adjuntos");
}
}
}
```

ANEXO VII. EVENTO USUARIOEVENT

```
package clases
{
    import flash.events.Event;
    import flash.events.Event;
    import clases.Usuario;

    public class UsuarioEvent extends Event
    {
        public static const CARGAR:String = "cargar";
        public var usuario:Usuario;
        public function UsuarioEvent(type:String,
            theUsuario:Usuario)
        {
            super(type);
            usuario = theUsuario;
        }

        override public function clone():Event{
            return new UsuarioEvent(type, usuario);
        }
    }
}
```


ANEXO VIII. CARGA DE UN TIPO USUARIO

```

package clases
{
    import mensajes.*;

    import adobe.utils.*;

    import flash.events.*;
    import flash.net.URLLoader;
    import flash.net.URLRequest;
    import flash.net.URLRequestMethod;
    import flash.net.URLVariables;

    import mx.events.*;

    [Bindable]
    public class usuarioLoader
    {

        private var xmlContent:XML;
        private var xmlLoader:URLLoader;
        public function usuarioLoader(u:Usuario){
            xmlLoader= new URLLoader();
            xmlLoader.addEventListener(Event.COMPLETE,
                completeHandler);
            xmlLoader.addEventListener(IOErrorEvent.IO_ERROR,
                httpIOErrorHandler);
        }

        public function load(xmlFile:String, vars:URLVariables =
            null):void{
            var request:URLRequest = new URLRequest(xmlFile);
            request.method = URLRequestMethod.POST;

            try{
                if (vars!= null){
                    request.data = vars;
                }
                xmlLoader.load(request);
            } catch (e:Error){
                trace("No se ha podido cargar el documento XML
                    externo");
            }
        }

        private function completeHandler(e:Event):void{
            try{
                xmlContent = XML(e.target.data);
                var message:String =
                    xmlContent.channel.message;
                if (message.length > 0){
                    var messageEventObj:MessageEvent = new
                        MessageEvent(MessageEvent.MESSAGE,
                            message);
                    dispatchEvent(messageEventObj);
                }
            } else{
                var usuario:XMLList =
                    xmlContent.channel.item;
            }
        }
    }
}

```

```

        var b:Usuario = new Usuario(usuario);
        var resultadoEventObj:UsuarioEvent = new
        UsuarioEvent(UsuarioEvent.CARGAR, b);
        dispatchEvent(resultadoEventObj);
    }
} catch(e:Error){
    message = "Error al ejecutar la consulta.
    Inténtelo nuevamente.";
    messageEventObj = new
    MessageEvent(MessageEvent.MESSAGE,
    message);
    dispatchEvent(messageEventObj);
}
}

public function httpIOErrorHandler(event:IOErrorEvent):void
{
    var messageEventObj:MessageEvent = new
    MessageEvent(MessageEvent.MESSAGE,
    messages.CARGAIN0000);
    dispatchEvent(messageEventObj);
}

public function httpErrorHandler(event:HTTPStatusEvent):void
{
    var messageEventObj:MessageEvent = new
    MessageEvent(MessageEvent.MESSAGE,
    messages.CARGAIN0001);
    dispatchEvent(messageEventObj);
}
}
}

```

ANEXO IX. CLASE TITLEBACKGROUND

```

package classes
{

    import flash.display.Bitmap;
    import flash.display.GradientType;
    import flash.display.Shape;
    import flash.display.SpreadMethod;
    import flash.display.Sprite;
    import flash.events.Event;
    import flash.geom.Matrix;
    public class TitleBackground extends Sprite
    {

        public var tileImage:Bitmap;
        private var colors:Array;
        private var alphas:Array;
        private var ratio:Number;

        private var bmp:Bitmap;
        private var degradado:Shape;

        public function TitleBackground(tileImage:Bitmap,
            colors:Array, alphas:Array, ratio:Number = 0.5)
        {
            this.tileImage = tileImage;
            this.colors = colors;
            this.alphas = alphas;
            this.ratio = ratio;
        }

        private function addedToStage(event:Event):void
        {
            removeEventListener(Event.ADDED_TO_STAGE,
                addedToStage);
            stage.addEventListener(Event.RESIZE,
                function():void{update(stage.stageWidth,
                    stage.stageHeight/3)});
        }

        public function update(_width:Number, _height:Number):void
        {
            graphics.clear();
            graphics.beginBitmapFill(tileImage.bitmapData, null,
                false, true);
            graphics.drawRect(-_width, -_height,
                2*_width, 2*_height);
            graphics.endFill();

            var matrix:Matrix = new Matrix();
            matrix.createGradientBox(2*_width, 2*_height,
                0.5*Math.PI);

            graphics.beginGradientFill(GradientType.LINEAR,
                colors, alphas, [0, Math.round(ratio*255),
                255], matrix, SpreadMethod.PAD);
            graphics.drawRect(-_width, -_height, 2*_width,
                2*_height);
            graphics.endFill();
        }
    }
}

```

```
var matrix2:Matrix = new Matrix();
matrix2.createGradientBox(2*_width, _height/3 ,0);

var colorsLinea:Array = [0x00000, 0xFFFFF, 0x00000];

graphics.beginGradientFill(GradientType.LINEAR,
    colorsLinea, alphas, [255, 255,255], matrix2,
    SpreadMethod.PAD);
graphics.drawRect(-_width,  tileImage.height,
    2*_width, 10);
graphics.endFill();
}
}
}
```

ANEXO X. DBCONN.PHP

```
<?php
$db_host='localhost';
$db_user='root';
$db_pass='acceso';
$db_name='proyecto';
mysql_connect($db_host, $db_user, $db_pass) or die("Could not connect
to host.");
mysql_select_db($db_name);
?>
```


ANEXO XI. CABECERAXML.PHP

```
<?php
$xml = new DomDocument('1.0', 'UTF-8');
$root = $xml-> createElement('rss');
$root -> setAttribute('version', '2.0');
$root = $xml -> appendChild($root);
$channelElement = $xml -> createElement('channel');
$channelElement = $root -> appendChild($channelElement);
$titleElement = $xml -> createElement('title', 'Proyecto Docente');
$titleElement = $channelElement -> appendChild($titleElement);
$linkElement = $xml -> createElement('link' ,
    'http://localhost/ProyectoDocente');
$linkElement = $channelElement -> appendChild($linkElement);
$descriptionElement = $xml -> createElement('description', 'Aplicación
    Docente');
$descriptionElement = $channelElement ->
    appendChild($descriptionElement);
?>
```


ANEXO XII. BORRADODIRECTORIO.PHP

```

<?php
/**
 * Borra de forma recursiva un directorio y todos los documentos dentro de este directorio
 *
 * @param string $directorio Nombre del directorio
 * @param boolean $borrarCarpeta Especifica si sólo se tiene que borrar los documentos
 * que se encuentran en el directorio o la carpeta también
 */
function borradoDirectorio($directorio, $borrarCarpeta)
{
    $dirAux = "/ProyectoDocente/".$directorio;
    if(!$dh = @opendir($dirAux))
    {
        return;
    }
    while (false !== ($obj = readdir($dh)))
    {
        if($obj == '.' || $obj == '..')
        {
            continue;
        }

        if (!@unlink($dirAux . '/' . $obj))
        {
            borradoDirectorio ($dirAux.'/'.$obj, true);
        }
    }

    closedir($dh);

    if ($borrarCarpeta)
    {
        @rmdir($dirAux);
    }

    return;
}
?>

```