



APLICACIÓ WEB PER A LA GESTIÓ DE FORMACIÓ INFORMÁTICA A TERRASSA

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Aarón Luque Rubio
i dirigit per
Elisa Heymann Pignolo
Bellaterra, 12 de Juny de 2007.

ÍNDICE

1. INTRODUCCIÓN	
1.1 Introducción del proyecto.....	1
1.2 Motivaciones personales.....	3
2. ESTUDIO DE VIABILIDAD	
2.1 Introducción.....	4
2.2 Descripción de la situación actual.....	5
2.3 Perfil del usuario.....	6
2.4 Objetivos del proyecto.....	7
2.4.1 Diagramas de casos de uso.....	7
2.4.2 Requerimientos funcionales.....	8
2.4.3 Requerimientos no funcionales.....	9
2.5 Fuentes de información.....	10
2.6 Alternativas de solución.....	10
2.6.1 Descripción del sistema a realizar.....	10
2.6.2 Modelo de desarrollo.....	11
2.6.3 Recursos.....	11
2.6.4 Análisis costo – beneficio.....	12
2.6.5 Evaluación de Riesgos.....	12
2.6.6 Planificación.....	13
2.7 Conclusiones.....	15
3. ESTUDIO DE LAS TECNOLOGIAS	
3.1 HTML.....	16
3.1.1 Que es el HTML?.....	16
3.1.2 Historia del HTML.....	17
3.2 JAVASCRIPT.....	18
3.2.1 Que es Javascript?.....	18
3.2.2 Historia del Javascript.....	18
3.2.3 Por qué Javascript?.....	19
3.3 CSS.....	19
3.3.1 Que es el CSS?.....	19
3.3.2 Por qué CSS?.....	20
3.4 PHP.....	20
3.4.1 Que es el PHP?.....	20
3.4.2 Usos del PHP.....	21
3.4.3 Por qué PHP?.....	21
3.4.4 Por qué PHP y no ASP?.....	22

3.5 MySQL.....	23
3.5.1 Que es MySQL?.....	23
3.5.2 Historia del MySQL.....	23
3.5.3 Características del MySQL.....	24
3.5.4 Aplicaciones del MySQL.....	24
3.5.5 Por qué MySQL?.....	25
3.5.6 Por qué MySQL y no SQL Server?.....	25
3.5.7 Por qué MySQL y no PostGreSQL?.....	26
3.6 APACHE.....	27
3.6.1 Que es Apache?.....	27
3.6.2 Por qué Apache?.....	27
3.7 DEBIAN.....	28
3.7.1 Que es Debian?.....	28
3.7.2 Características de Debian.....	28
3.7.3 Por qué Debian?.....	29
4. BASE DE DATOS	
4.1 Estructura de la base de datos.....	30
4.2 Modelo entidad-relación.....	37
5. DISEÑO	
5.1 Restricciones.....	38
5.2 Objetivos.....	39
5.3 Diseño de la aplicación.....	39
5.4 Diseño de los módulos.....	42
5.5 Problemas y soluciones.....	50
6. CODIFICACIÓN	
6.1 Codificación de los módulos.....	52
6.2 Problemas y soluciones.....	71
6.3 Pruebas.....	73
6.4 Jerarquía de la aplicación.....	74
7. CONCLUSIONES	
7.1 Objetivos alcanzados.....	82
7.2 Posibles ampliaciones.....	82
7.3 Valoración personal.....	84
8. BIBLIOGRAFIA.....	85

ANEXO 1 – Manual de usuario

1 INTRODUCCIÓN

1.1 Introducción del proyecto

El proyecto realizado es un proyecto real para el departamento "Sociedad del Conocimiento" del Ayuntamiento de Terrassa. Este departamento se dedica especialmente a servir a la población accesos informáticos con Internet además de ofrecer también cursos de informática básicos para iniciar a la gente dentro del mundo de los ordenadores. El proyecto consiste en la creación de una página web dinámica que junto a una base de datos, permita informatizar diferentes actividades tanto de gestión de usuarios (control de alumnos y profesores), como generación de estadísticas o diferentes herramientas que faciliten algunas tareas a los administradores (control de asistencias, de centros, etc...), para poder llevar una buena gestión de los cursos descritos anteriormente.

El objetivo principal de este proyecto es poner en práctica los conocimientos adquiridos durante la carrera y coger así experiencia en como aplicar todo lo aprendido durante estos años en realizar diferentes aplicaciones. También uno de los objetivos principales es ver como se desenvuelve el alumno delante de un proyecto y como responde a todas sus partes, desde pensar la tecnología a utilizar y el análisis de requerimientos hasta la realización de pruebas o la redacción de la memoria.

La memoria se divide claramente en varios apartados, la primera que coincide con este apartado, que es una breve introducción al proyecto y posteriormente los motivos por el cual ha sido elegido. Después consta de un estudio de viabilidad donde se comprueba que la realización de esta aplicación es totalmente viable.

La tercera sección es un poco la más teórica y consiste en la selección de las tecnologías y herramientas a utilizar para la realización de éste, y la comparación con otras tecnologías similares que finalmente me hicieran decantarme por una o por la otra.

El siguiente apartado de la memoria es el más técnico, ya que se tratan todas las especificaciones del proyecto, la base de datos y como se relacionan entre ellas. También consta de una parte técnica pero no tanto de programación la siguiente sección que trata del diseño de la web. Estos dos apartados anteriores serán detallados también con los problemas surgidos y las soluciones elegidas para la resolución de éstos.

Y para finalizar la memoria se explica un apartado de pruebas realizadas, las conclusiones finales donde se dan a conocer los objetivos alcanzados, posibles ampliaciones y las valoraciones personales y un índice bibliográfico donde se muestran fuentes de información utilizadas.

1.2 Motivaciones personales

Durante el curso pasado estuve combinando los estudios con un trabajo como becario en el ayuntamiento de Terrassa, más concretamente en la sección que se encargaba de dar acceso a Internet y poner a disposición de la población varias salas donde poder conocer la informática y practicarla.

Mi trabajo básicamente consistía en programar diferentes cursos para la alfabetización digital de los habitantes que podían apuntarse totalmente gratis a éste servicio que se ofrecía. Una vez realizada la programación y culminado los cursillos me llegaba la información de los alumnos que lo habían cursado y mi faena era almacenarlo en páginas excel y generar a partir de ahí unas estadísticas requeridas por el ayuntamiento.

Este año al continuar trabajando en el mismo lugar se me ocurrió que podía realizar un proyecto para informatizar y automatizar todo el proceso desde la programación hasta la generación de estadísticas, ya que resulta mucho más sencillo y más viable almacenar los datos en una base de datos que cada cierto tiempo se van haciendo backups que no tenerlos en hojas de excel. Me decanté en realizar una aplicación web ya que creo que para el tipo de aplicación que es, es la mejor opción ya que interactúan administradores y profesores desde diferentes lugares y la forma más accesible es conectarte a una página web.

Además la realización de una página web dinámica es muy interesante ya que juntamente con una base de datos permite dar un potencial de flexibilidad muy amplio a la página.

Finalmente el último aspecto que me hizo decantarme fue la posibilidad de desarrollar un proyecto real de este tipo desde el estudio inicial hasta la implementación del mismo, y que después de ser probado correctamente fuera ser utilizado diariamente en la vida real.

2 ESTUDIO DE VIABILIDAD

2.1 Introducción

El proyecto se basa principalmente, en la realización de una aplicación web para gestionar toda la información acerca de los cursillos realizados a través del servicio Puntxarxa, con el propósito dar más facilidades tanto en la recogida de estadísticas como una ayuda para que los administradores y profesores para que dispongan de mejores métodos de trabajo, facilitando de alguna manera su trabajo.

Ventajas:

1. Mantener actualizada y con un fácil acceso, diferentes estadísticas sobre los cursos del Puntxarxa.

- *Nos permite tener actualizado en tiempo real, el estado de los diferentes cursos y las estadísticas relacionadas con éste, además de obtener un amplio abanico de estadísticas que siempre se puede ampliar de forma sencilla.*

2. Automatización de tareas.

- *Mediante esta aplicación, algunas de las tareas que antes se tenían que hacer manualmente, ahora se realizan de forma automática.*

3. Facilitar el contacto profesor-administrador.

- *Permite un rápido contacto entre el profesor y el administrador a través de un sistema de mensajería sobre esta aplicación.*

4. Permitir un control rápido de las incidencias en las aulas para una posterior solución.

- *Se facilita la tarea de controlar las incidencias y el estado de éstas como también poder tener un historial que permita hacer balance cada cierto tiempo.*

5. Acceso a la aplicación desde cualquier lugar con conexión a Internet.

- *Al ser una página web se puede acceder a ella desde varios lugares a la vez y sin tener que ir instalando la aplicación en cada centro.*

6. Acceso rápido para llevar un buen control de los alumnos.

- *A través de la aplicación se podrá tener un rápido acceso de los alumnos que realizan los diferentes cursos, ver sus asistencias como también visualizar cuantos cursos han realizado y dónde.*

Desventajas:

- 1.** Sesión de aprendizaje para los usuarios de la aplicación.
 - *Se deberá hacer una pequeña formación a los diferentes usuarios de la aplicación para que tengan un correcto y fácil uso de ésta.*
- 2.** Periodo de desarrollo de la aplicación y posterior implantación.
 - *Otra desventaja es que habrá que esperar un tiempo para el desarrollo de la aplicación y la posterior implantación en algún servidor.*
- 3.** Necesidad de un especialista para el mantenimiento y actualización de la herramienta.
 - *Se tendrá que contratar a un técnico que arregle los problemas, si es que los hay, que actualice la herramienta cuando sea necesario y lleve al día la base de datos.*
- 4.** Necesidad de contratar un servidor y un servicio de hosting.
 - *Para poder implantar la página se deberá contratar estos servicios y esto conlleva a un gasto anual aunque no sea muy elevado.*

2.2 Descripción de la situación actual

Hasta ahora, el control de los diferentes cursos del servicio Puntxarxa se realiza de forma “manual”. Todos los pasos desde la programación hasta la generación de estadísticas se hacen a través de hojas de cálculo, siempre insertando manualmente toda la información.

Primero de todo se genera una hoja de cálculo para la programación de los cursos y a partir de ahí se sacan unas hojas de inscripción que se envían a los centros para que los alumnos se puedan apuntar. Esto no cambia demasiado en la aplicación ya que la programación se deberá insertar manualmente también pero ya tendremos una forma más cómoda de generar éstas hojas de inscripción.

El tema de control de asistencia es notablemente mejorado ya que los profesores llevaban el control de las asistencias como también de los datos de los alumnos a través de las hojas de inscripción que después enviaban al coordinador y éste introducía la información manualmente en una hoja de cálculo. Con AD Management, los profesores llevarán el control a través de la web pudiendo así obtener una mayor información de los alumnos y no tener que enviar ésta información al coordinador para que la introduzca en el ordenador y genere las estadísticas.

Estas últimas son generadas de forma notablemente mejor que como se hace actualmente, ya que ahora como ya se ha comentado el coordinador inserta la información a través de la hoja de cálculo y genera unas estadísticas con éstos datos. Ahora las estadísticas se generan automáticamente recogiendo la información de la base de datos por lo que también amplia bastante las posibilidades de realizar estadísticas de varios tipos ya que al tener más facilidad en la generación de éstas, podemos introducir datos de los alumnos que anteriormente no recogíamos y obtener así mucha más información que puede ser muy útil.

Además el contacto entre el profesor y los coordinadores solo se podía dar en persona o mediante teléfono o correo electrónico. Actualmente aún no todos utilizan el e-mail como medio de comunicación, por lo que al utilizar esta aplicación podrán observar de forma clara si tienen algún mensaje nuevo a través del sistema de mensajería de ésta.

De una forma similar cualquier usuario de la aplicación podrá informar de alguna incidencia ocurrida en un centro y así permitir a los coordinadores pensar una rápida solución para ésta.

2.3 Perfil del usuario

Existen dos tipos de usuarios que serán los que utilizarán ésta aplicación, los profesores y los administradores. Los profesores serán los encargados de realizar el curso y llevar el control de éstos, por lo que sus tareas utilizando la aplicación serán introducir a los alumnos en los determinados cursos, marcar sus asistencias y finalizar el curso cuando éste sea terminado.

Por otra parte los administradores serán los encargados de llevar el control de los tipos de cursos, añadir programaciones en los diferentes centros e ir controlando los cursos finalizados para la posterior extracción de las estadísticas. Además si algún profesor ha dado de alta una incidencia, los administradores serán los responsables de pensar la solución a ésta.

La aplicación web consta de una parte privada con acceso restringido en el que tan solo podrán acceder los profesores y administradores que estén dados de alta. Estos últimos tendrán acceso a toda la aplicación mientras que los profesores podrán acceder parcialmente a ésta.

En cuanto al correcto uso de esta web, será necesaria una sesión de aprendizaje para sus usuarios, que no será nada más que unos conocimientos básicos de informática y de Internet, para su perfecto funcionamiento, como también una breve explicación del funcionamiento de la aplicación.

2.4 Objetivos del proyecto

Realizar una herramienta personalizada que tiene como objetivo informatizar la gestión de los cursos Puntxarxa para facilitar y mejorar el control de éstos. Aunque el principal objetivo de este proyecto es poner en práctica algunos de los conocimientos adquiridos durante la carrera.

2.4.1 Diagrama de casos de uso

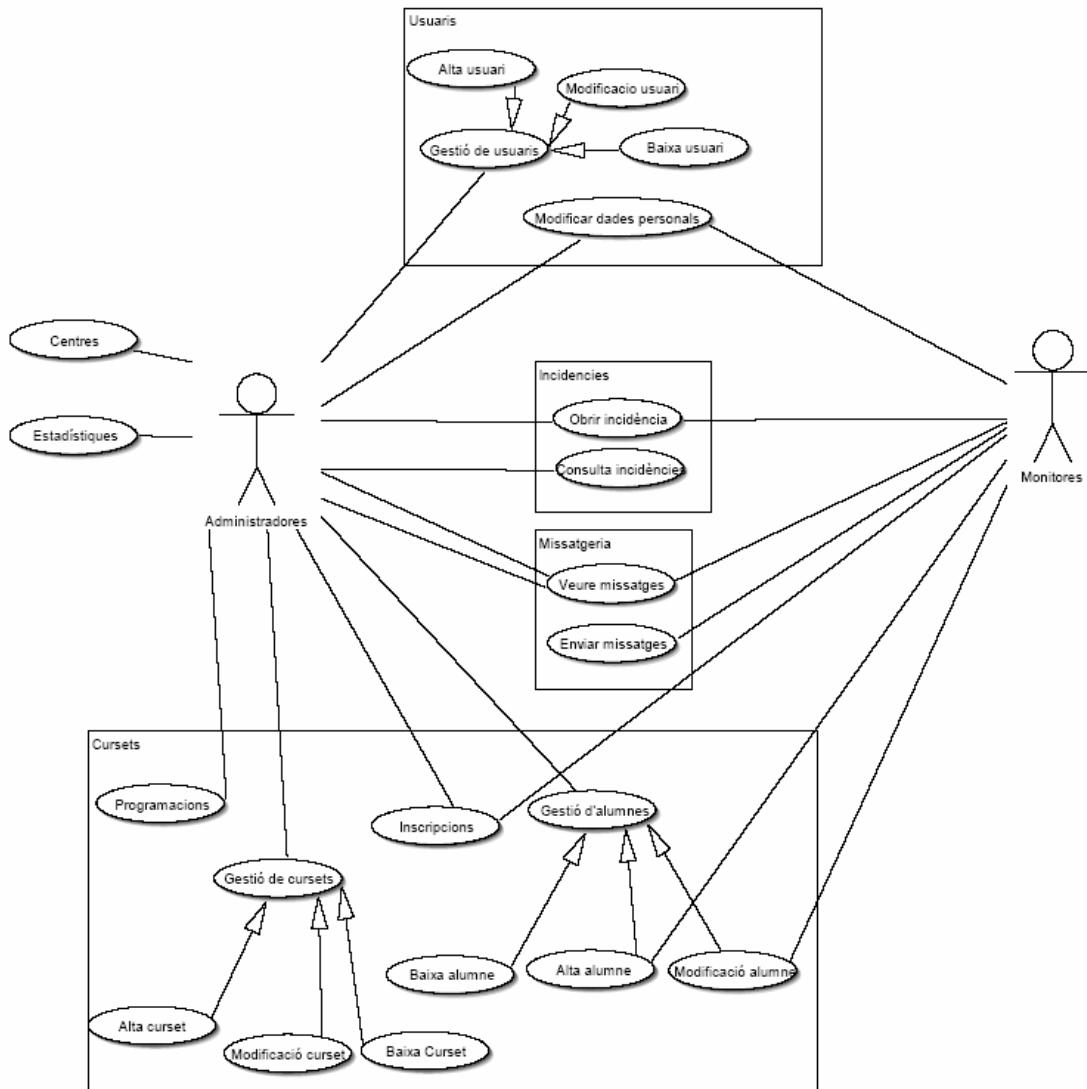


Figura 0 – Realización del diagrama de casos de uso de la aplicación

2.4.2 Requerimientos funcionales

La aplicación debe tener diferentes requerimientos funcionales para satisfacer los objetivos del proyecto. Primeramente, la página debe soportar diferentes tipos de usuarios, concretamente existen dos tipos: los profesores y los administradores.

Estos primeros sólo podrán acceder parcialmente a la aplicación, permitiéndoles principalmente la gestión de los alumnos de los diferentes cursos, dónde podrán realizar altas, bajas y modificaciones de éstos. También controlarán las asistencias de los alumnos en los diferentes cursos que el profesor realice.

Finalmente se les permite dar de alta incidencias que puedan suceder en el aula en la cual están asignados, como también podrán utilizar el sistema de mensajería para comunicarse con los otros usuarios del sistema.

Por otra parte los administradores tendrán un acceso completo a la aplicación, permitiéndoles además de realizar las funciones permitidas a los profesores comentadas anteriormente, nuevas funcionalidades como la gestión de los diferentes tipos de cursos dónde podrán dar altas, bajas y modificar cualquiera de sus datos. Además de las ya comentadas gestiones de alumnos y tipos de cursos y control de asistencias, los administradores también pueden gestionar las programaciones de los diferentes centros, asignando un determinado tipo de curso y un profesor en unas fechas concretas. Por último en el tema de gestión de cursos, éste tipo de usuario podrá visualizar un historial con el estado de los cursos, el número de inscripciones que ha tenido y las asistencias de los alumnos.

La aplicación ofrece a los administradores extraer diferentes tipos de estadísticas en tablas y en forma gráfica. Se puede seleccionar dos determinadas fechas y extraer el total de cursos programados o separados por su estado, el número total de usuarios en cada curso, su sexo, su disponibilidad laboral o su disposición de un punto de acceso informático, entre éstas dos fechas seleccionadas.

Además de permitir el acceso al sistema de mensajería e incidencias como los profesores, los administradores en ésta última funcionalidad además de poder dar de alta, pueden consultar todas las incidencias de cualquier centro y modificarlas para poder cambiar su estado o su prioridad dependiendo de la gravedad de éstas.

Finalmente a los administradores se les permite gestionar los diferentes usuarios que van a utilizar la aplicación como también, realizar una gestión de los diferentes centros dónde se realizan cursos, permitiendo hacer altas, bajas y modificaciones de éstos y guardar información importante como el teléfono y la persona de contacto, el tipo de acceso a Internet, su IP pública, sus DNS, el login y password del router, etc...

243 Requerimientos nofuncionales

Compatibilidad: la aplicación debe ser compatible con los principales navegadores especialmente Internet Explorer y Mozilla Firefox ya que éstos son los más usados entre los usuarios. Además también debe tener compatibilidad con todo tipo de conexión: alta velocidad en la conexión con la base de datos y la carga de la página web.

Disponibilidad: se debe poder acceder a la página desde cualquier lugar con acceso a Internet y a cualquier hora del día por lo que el servicio de hosting debe ser un servicio fiable para no perder el acceso a la aplicación.

Mantenimiento: ha de tener un fácil mantenimiento de datos ya que debe proporcionar opciones que permitan realizar altas, bajas y modificaciones sin ninguna dificultad y de manera rápida

Facilidad de uso: la aplicación debe ser fácil de utilizar, con una buena navegabilidad y muy visual ya que los usuarios pueden tener diferentes niveles de informática, además de ser intuitiva y de fácil acceso.

Ser extensible: la página se debe poder actualizar con facilidad así como poder añadir nuevas funcionalidades de forma sencilla y poder realizar cambios en el código cómodamente.

Seguridad: tiene que tener la aplicación un sistema de seguridad para que no puedan acceder a ella cualquier tipo de usuario, sino que debe tener acceso permitido para poder utilizarla.

Estabilidad: el sistema ha de ser estable ya que se debe poder acceder a cualquier hora del día y debe estar siempre a punto para realizar cualquier operación.

2.5 Fuentes de información

Las fuentes de información necesarias para la creación de este proyecto serán:

- 1.** Manuales y libros relacionados con los entornos de programación utilizados (PHP), también con la creación y mantenimiento de Bases de datos (MySQL) y manuales de cómo configurar un servidor dentro de la plataforma utilizada (Apache).
- 2.** Página web oficial de ayuda sobre las diferentes funciones disponibles en PHP y con ejemplos reales (<http://www.php.net>).
- 3.** Documentos legales relativos a las leyes aplicables al proyecto.

2.6 Alternativas de solución

2.6.1 Descripción del sistema a realizar

- ? La herramienta que desarrollaremos tendrá una interfaz gráfica donde podremos escoger la actividad deseada. Para comenzar, habrá 2 niveles de seguridad (profesores y administrador) que nos permitirán acceder o no a determinadas actividades.
- ? Control actualizado de los alumnos que realizan los cursos del Puntxarxa.
- ? Historial con todos los cursos realizados, estado de éstos y sus asistencias.
- ? Crear nuevas programaciones describiendo el tipo de curso, el centro donde se realiza y el profesor asignado.
- ? Mostrar estadísticas actualizadas en tiempo real sobre los diferentes cursos.
- ? Tener un control de los tipos de cursos y poder actualizarlos en cualquier momento.
- ? Llevar al día los diferentes centros en los cuales se realizan cursos del Puntxarxa.
- ? Avisar rápidamente de alguna incidencia en un centro para una rápida solución.
- ? Permitir el contacto administrador/profesor y profesor/administrador para comentar dudas u otras cuestiones.

262Modelode desarrollo

El proceso de desarrollo de este software puede llevarse a cabo aplicando cualquiera de los modelos de estos modelos de desarrollo:

- ? Modelo Lineal secuencial
- ? Modelo de Prototipo
- ? Modelo de desarrollo rápido de aplicaciones
- ? Modelo evolutivo

El modelo que se usará en este proyecto será el Modelo evolutivo, ya que tenemos la estructura de nuestro proyecto dividida en módulos, y el funcionamiento de este modelo se basa en subdividir el problema en problemas más pequeños.

263Recursos

Los recursos necesarios para el proyecto son los siguientes:

1. El desarrollador de la aplicación deberá disponer de las herramientas imprescindibles para un correcto desarrollo de la aplicación y que éstas no sean un impedimento para la finalización de la página, ni genere ningún tipo de retraso en la planificación del proyecto. Por lo que deberá de disponer del software y hardware que mostramos a continuación para tener un pc preparado para una cómoda programación.

- ? Software:
 - SO: Windows XP Professional Edition
 - Programación: PHP, Dreamweaver MX, Bloc de notas.
 - Bases de Datos: MySQL , PHPmyAdmin.
 - Servidor: Apache.
- ? Hardware:
 - Procesador: Pentium IV 2.0 Ghz
 - Memoria RAM: 512 Mb
 - Disco Duro: 20 Gb
 - Unidad de CD-ROM/DVD-ROM
 - Tarjeta gráfica 64MB
 - Monitor SVGA
 - Router ADSL

2. Los usuarios que utilizan la aplicación no tienen que tener una máquina tan potente como la del desarrollador ya que para utilizarla solamente necesitan un pc con conexión a Internet y mediante un navegador accederán fácilmente, por lo que con un ordenador con las siguientes características ya podrán utilizar fácilmente y sin ningún tipo de restricciones la aplicación.

- ? Los usuarios a los que va destinado el proyecto necesitarán un PC que cumpla los siguientes requisitos mínimos:
 - Procesador: Pentium II 350 Mhz
 - Memoria RAM: 128 Mb
 - Disco Duro: 4 Gb
 - Modem 56K o superior.
 - Monitor VGA
- ? Además de todo esto, un recurso indispensable será el disponer de una conexión a Internet y también será necesario contratar un servicio de hosting y probablemente adquirir un dominio propio.

264Análisis costo-beneficio

Los costes variables que implicará este proyecto serán la contratación de un servicio de hosting (anual), la adquisición de un dominio propio y la habilitación de una conexión a Internet en cada centro y en la oficina (mensual), aunque éstas últimas ya están disponibles.

Además el gasto del total de horas empleadas en la realización del proyecto más el software y hardware necesario, pero en este caso el proyecto es de final de carrera y su valor de realización será nulo y ya se dispone de los recursos necesarios para realizarlo.

Como beneficio obtendremos un mayor número de estadísticas que permitirán obtener subvenciones de la Generalitat y también obtendremos como beneficio más facilidad en el trabajo diario.

265Evaluación de Riesgos

El riesgo principal de la aplicación propuesta es la seguridad de la ésta, ya que está desarrollada para su uso en Internet. Además esta herramienta debería ser compatible para todo tipo de navegadores.

Otros riesgos es el mantenimiento de la base de datos, encontrar pequeños bugs después de haberse realizado el proyecto o la necesidad de retardar la finalización de alguna tarea que repercutiría en la planificación del proyecto.

266 Planificación

La planificación de las tareas del proyecto se muestra en la tabla siguiente:

Tarea	Descripción	Duración (horas)	Predecesora
1	Estudio de viabilidad		
1.1	Definición de objetivos	4	
1.2	Análisis de requerimientos	10	1.1
1.3	Análisis de las tecnologías a utilizar	4	
1.4	Definir especificaciones	4	1.2
1.5	Diseñar diagramas de casos de uso	4	1.4
2	Preparación de las herramientas a utilizar		
2.1	Configuración de la plataforma y las herramientas	10	1.3
3	Diseño y codificación		
3.1	Diseñar el modelo entidad-relación de la BBDD	20	
3.2	Diseño y configuración de la BBDD en MySQL	10	3.1
3.3	Diseño de la estructura de la aplicación (modulación)	25	1.2
3.4	Diseño de las hojas de estilo CSS	30	
3.5	Diseño de las funciones JavaScript	20	
3.6	Diseño del menú DHTML	10	
3.7	Diseño de la página principal y de los diferentes módulos	50	3.3 – 3.4 – 3.5 – 3.6
3.8	Codificación de la página principal y proceso de identificación	20	3.2 – 3.7
3.9	Codificación de las rutinas de uso común (comprobaciones de usuario, permisos, etc...)	20	3.8
3.10	Codificación de los diferentes módulos	240	3.9
4	Pruebas		
4.1	Pruebas finales del sistema y corrección de errores	15	3.10
4.2	Pruebas finales del sistema con usuarios reales	15 días (no se contabiliza)	4.1
5	Memoria		
5.1	Redacción de la memoria	40	4.2
	Total	536	

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	■ Proyecto	159 días?	lun 16/10/06	jue 07/06/07	
2	■ Estudio de viabilidad	13 días?	lun 16/10/06	mié 01/11/06	
3	■ Definición de objetivos	1 día	lun 16/10/06	lun 16/10/06	
4	■ Análisis de requerimientos	3 días?	mar 17/10/06	jue 19/10/06	3
5	■ Análisis de las tecnologías a utilizar	2 días?	lun 23/10/06	mar 24/10/06	4
6	■ Definir especificaciones	1 día?	lun 30/10/06	lun 30/10/06	4;5
7	■ Diseñar diagramas de casos de uso	1 día?	mié 01/11/06	mié 01/11/06	6
8	■ Preparación de las herramientas a utilizar	3 días?	mié 08/11/06	vie 10/11/06	7
9	■ Configuración de la plataforma y las herramientas	3 días?	mié 08/11/06	vie 10/11/06	5
10	■ Diseño y codificación	116 días?	lun 13/11/06	lun 07/05/07	
11	■ Diseñar el modelo entidad-relación de la BBDD	5 días?	lun 13/11/06	vie 17/11/06	
12	■ Diseño y configuración de la BBDD en MySQL	3 días?	lun 20/11/06	mié 22/11/06	11
13	■ Diseño de la estructura de la aplicación (modulación)	7 días?	lun 27/11/06	mar 05/12/06	4;12
14	■ Diseño de las hojas de estilo CSS	8 días?	lun 11/12/06	mié 20/12/06	13
15	■ Diseño de las funciones JavaScript	5 días	lun 22/01/07	vie 26/01/07	14
16	■ Diseño del menú DHTML	3 días?	lun 29/01/07	mié 31/01/07	15
17	■ Diseño de la página principal y de los diferentes módulos	13 días?	jue 01/02/07	lun 05/03/07	13;14;15;16
18	■ Codificación de la página principal y proceso de identificación	5 días	mar 06/03/07	lun 12/03/07	12;17
19	■ Codificación de las rutinas de uso común (comprobaciones de usuario, permisos, etc...)	5 días	mar 13/03/07	lun 19/03/07	18
20	■ Codificación de los diferentes módulos	35 días	mar 20/03/07	lun 07/05/07	19
21	■ Pruebas	13 días?	mar 08/05/07	jue 24/05/07	
22	■ Pruebas finales del sistema y corrección de errores	1 día?	mar 08/05/07	mar 08/05/07	20
23	■ Pruebas finales del sistema con usuarios reales	12 días	mié 09/05/07	jue 24/05/07	22
24	■ Memoria	10 días	vie 25/05/07	jue 07/06/07	
25	■ Redacción de la memoria	10 días	vie 25/05/07	jue 07/06/07	23

Figura 1 – Realización de la planificación del proyecto con MsProject

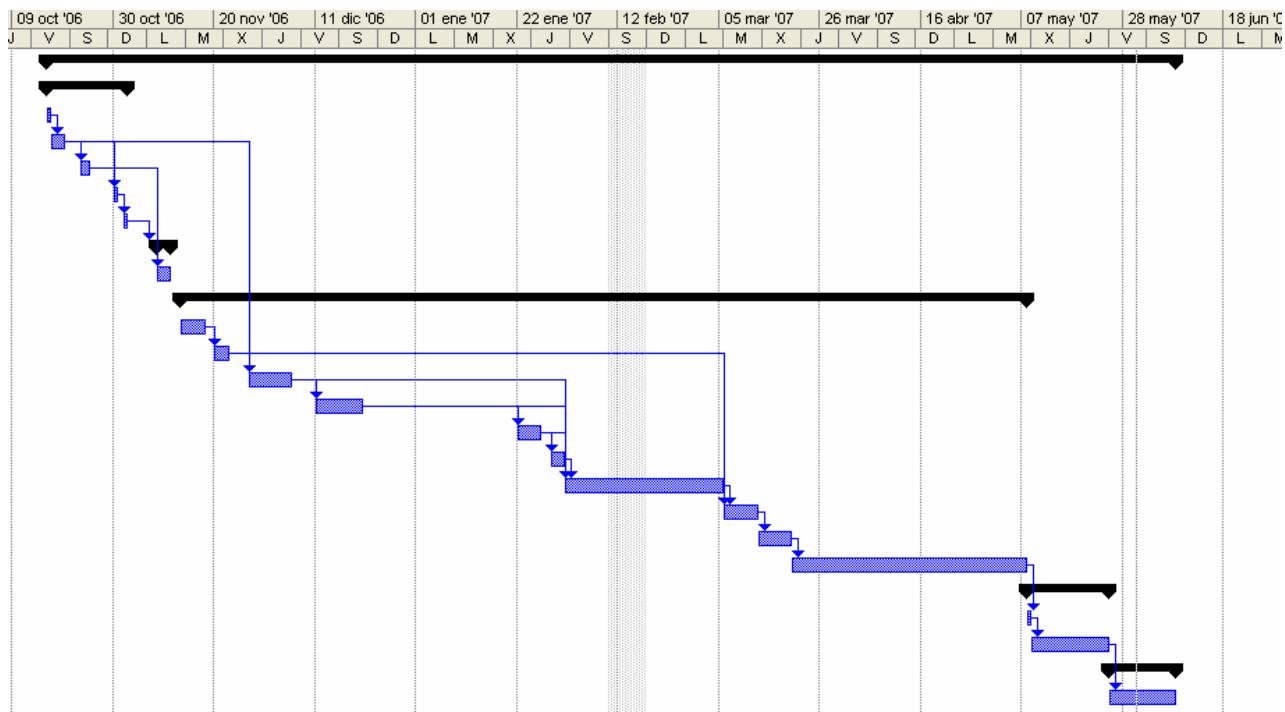


Figura 2 – Realización del diagrama de Gantt del proyecto con MsProject

2.7 Conclusiones

Después de realizar el estudio de viabilidad del proyecto podemos sacar como conclusión que éste es totalmente viable tanto técnicamente, ya que con los conocimientos de programación que se han ido adquiriendo durante toda la carrera se pueden utilizar en un proyecto de estas características, como económicoamente, como se puede observar en el apartado de análisis de costo - beneficio.

Este proyecto tendría un coste muy barato y obtendríamos unas grandes ventajas a la hora de mejorar las prestaciones de la gestión de cursos Puntxarxa, por lo tanto lo consideramos totalmente viable.

3 ANALISI DE LAS TECNOLOGIAS

3.1 HTML

3.1.1 Que es HTML?

El lenguaje HTML (HyperText Markup Language) es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

Este lenguaje puede ser creado y editado desde cualquier editor de textos como Gedit de Linux o el clásico Notepad del Windows o cualquier otro editor de texto sin formato.

Existen además, otros programas para la realización de sitios Web o edición de código HTML, como por ejemplo Microsoft FrontPage, el cual tiene un formato básico parecido al resto de los programas de Office. También existe el famoso software de Macromedia (que adquirió la empresa Adobe) llamado Dreamweaver, siendo uno de los más utilizados en el ámbito de diseño y programación Web. Estos programas se les conoce como editores WYSIWYG o What You See Is What You Get (en español: "lo que ves es lo que obtienes"). Esto significa que son editores los cuales van mostrando el resultado de lo que se está editando en tiempo real a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML la cual va generando todo el código a medida que se va trabajando.

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo; si estoy editando todo en HTML y de pronto olvido algún código o etiqueta, simplemente me dirijo al editor visual o WYSIWYG y continuo ahí la edición, o viceversa, ya que hay casos en que sale más rápido y fácil escribir directamente el código de alguna característica que queramos adherirle al sitio, que buscar la opción en el programa mismo.

Concretamente yo he utilizado solamente un editor de textos ya que los programas WYSIWYG generan a veces código innecesario y siempre que puedo intento no utilizarlos.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

3.1.2 Historia del HTML

En 1989 existían dos técnicas que permitían vincular documentos electrónicos, por un lado los hipervínculos (links) y por otro lado un poderoso lenguaje de etiquetas denominado SGML. Por entonces un usuario conocedor de ambas opciones, Tim Berners-Lee físico nuclear del Centro Europeo para la Investigación Nuclear da a conocer a la prensa que estaba trabajando en un sistema que permitiría acceder a ficheros en línea, funcionando sobre redes de computadoras o máquinas electrónicas basadas en el protocolo TCP/IP.

Principios de 1990, Tim Berners-Lee define por fin el HTML como un subconjunto del conocido SGML y crea algo más valioso aun, el World Wide Web. En 1991, Tim Berners-Lee crea el primer navegador de HTML que funcionaría en modo texto y para UNIX.

Los trabajos para crear un sucesor del HTML, posteriormente llamado 'HTML+', comenzaron a finales de 1993. El HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se le dio, por lo tanto, el número de versión 2.0 para distinguirla de esos "estándares no oficiales" previos. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar.

El borrador del estándar HTML 3.0 fue propuesto por el recién formado W3C en marzo de 1995. Con él se introdujeron muchas nuevas capacidades, tales como facilidades para crear tablas, hacer que el texto fluyese alrededor de las figuras y mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatible con HTML 2.0, era demasiado complejo para ser implementado con la tecnología de la época y, cuando el borrador del estándar expiró en septiembre de 1995, se abandonó debido a la carencia de apoyos de los fabricantes de navegadores web. El HTML 3.1 nunca llegó a ser propuesto oficialmente, y el estándar siguiente fue el HTML 3.2, que abandonaba la mayoría de las nuevas características del HTML 3.0 y, a cambio, adoptaba muchos elementos desarrollados inicialmente por los navegadores web Netscape y Mosaic. La posibilidad de trabajar con fórmulas matemáticas que se había propuesto en el HTML 3.0 pasó a quedar integrada en un estándar distinto llamado MathML.

El HTML 4.0 también adoptó muchos elementos específicos desarrollados inicialmente para un navegador web concreto, pero al mismo tiempo comenzó a limpiar el HTML señalando algunos de ellos como 'desaprobados'.

3.2 JAVASCRIPT

321 Que es el Javascript?

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web.

322 Historia del JavaScript

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la the European Computer Manufacturers' Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también lo fue como un estándar ISO.

JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen a ambas versiones con frecuencia incompatibles.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, ó Modelo de Objetos del Documento en castellano), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera versión 7, y Mozilla desde su primera versión.

3.2.3 Por qué JavaScript?

JavaScript es un lenguaje que no se debe abusar ya que requiere ser interpretado por el navegador aunque hoy en día cualquier navegador lo acepta. Aun así permite un gran potencial en algunos aspectos como la validación de los formularios de la página o también nos permite crear funciones de mucha utilidad. En la aplicación básicamente lo he utilizado para la ya comentada validación de formularios, como también para crear funciones que me cambie los valores de algún formulario o para que me refresque la página en algún momento concreto.

3.3 CSS

3.3.1 ¿Qué es el CSS?

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML en el caso de éste proyecto. El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- ? Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ? Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- ? Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- ? El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

3.3.2 Por qué CSS?

El motivo principal por el que he utilizado CSS en este proyecto tal y como explica anteriormente es para separar la estructura de su presentación. Así se pueden tener varias presentaciones para un mismo documento, en este caso página web, como también se pueden modificar con mucha mas facilidad.

3.4 PHP

3.4.1 Que es el PHP?

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones.... No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano al JavaScript o a C, para aquellos que conocen estos lenguajes.

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML la cual generalmente es dinámica.



Figura 3 - Procesamiento de una página PHP

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

PHP permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones web muy robustas.

También tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

342Usos de PHP

Los principales usos del PHP son los siguientes:

- ? Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- ? Programación en consola, al estilo de Perl o Shell scripting.
- ? Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

343Por qué PHP?

La principal ventaja de utilizar PHP es que es totalmente gratuito y además es un lenguaje apropiado para la realización de páginas web dinámicas y para conectar estas con una base de datos.

También de un tiempo a esta parte, se puede observar un ascenso imparable en la utilización del PHP, puesto que cada día son muchísimas más las páginas web que se apoyan en este lenguaje para su funcionamiento.

344Por qué PHP y no ASP?

Primero de todo debemos decir que ASP no es realmente un lenguaje como tal, es el acrónimo de Active Server Pages, el lenguaje usado en realidad para programar ASP es Visual Basic Script o JScript.

El mayor inconveniente de ASP es que se trata de un sistema propietario que es usado nativamente sólo por Microsoft Internet Information Server (IIS). Esto limita su disponibilidad a servidores basados en Win32, excepto un par de proyectos en desarrollo que permiten que ASP corra en otros entornos y servidores web: InstantASP de Halcyon, Chili!Soft ASP de Chili!Soft.

Se dice que ASP es un lenguaje más lento y pesado que PHP, y también menos estable. Algunas de las ventajas de ASP consisten en que debido a que usa principalmente VBScript, es relativamente simple tratar con el lenguaje si usted ya conoce cómo programar en Visual Basic. El soporte de ASP también se encuentra habilitado por defecto en el servidor IIS, facilitando su instalación y ejecución. Los componentes integrados en ASP son bastante limitados, de modo que si necesita usar características "avanzadas", como interactuar con servidores FTP, necesita comprar componentes adicionales.

Todos estos inconvenientes se pueden solucionar programando con PHP que además es un lenguaje gratuito, tal y como ya se ha comentado.

Algunas ventajas del PHP que podemos ver son:

<ul style="list-style-type: none">- Libre.- Abierto.- Código fuente disponible.- Diseñado para la web.- Multiplataforma HW.- Multisistema Operativo.- Soporte para varios servidores web.- Soporte nativo para prácticamente cualquier Base de Datos.- Buena documentación.- Miles de ejemplos y código fuente disponible.	<ul style="list-style-type: none">- Perfecta integración del Apache-PHP-MySQL.- Sintaxis clara y bien definida.- Bastante sencillo de aprender y utilizar.- Modulable.- Seguro (evidentemente tiene errores pero se solucionan mucho antes que otros sistemas propietarios).- Amplia base de usuarios (Ahora es el número 1, como lo es también Apache).- No dependes de un único proveedor de servicios.
---	---

Como desventaja sobre el asp a parte del tema del IIS que ya he comentado, tan solo se me ocurre que tiene una peor integración con el paquete Office de Microsoft, pero esto es totalmente lógico, puesto que Microsoft en su empeño de mantener su monopolio, no da las especificaciones del mismo e incluso crea artificialmente problemas entre sus propias versiones para forzar al usuario a comprar las nuevas versiones de su suite.

3.5 MySQL

351 Que es MySQL?

MySQL es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

La desarrolla y mantiene la empresa MySQL AB pero puede utilizarse gratuitamente y su código fuente está disponible.

Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.

Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

352 Historia de MySQL

SQL (Lenguaje de Consulta Estructurado) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael Widenius en la década de los 90 trató de usar mSQL para conectar las tablas usando rutinas de bajo nivel ISAM, sin embargo, mSQL no era rápido y flexible para sus necesidades. Esto lo llevó a crear una API SQL denominada MySQL para bases de datos muy similar a la de mSQL pero más portable.

La procedencia del nombre de MySQL no es clara. Por más de 10 años, las herramientas han mantenido el prefijo My. También, se cree que tiene relación con el nombre de la hija del cofundador Monty Widenius quien se llama My.

Por otro lado, el nombre del delfín de MySQL es Sakila y fue seleccionado por los fundadores de MySQL AB en el concurso "Name the Dolphin". Este nombre fue enviado por Ambrose Twebaze, un desarrollador de Opensource Africano, derivado del idioma SiSwate, el idioma local de Swaziland y corresponde al nombre de una ciudad en Arusha, Tanzania, cerca de Uganda la ciudad origen de Ambrose.

353 Características del MySQL

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- ? Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- ? Disponibilidad en gran cantidad de plataformas y sistemas.
- ? Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- ? Transacciones y claves foráneas.
- ? Conectividad segura.
- ? Replicación.
- ? Búsqueda e indexación de campos de texto.

354 Aplicaciones de MySQL

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas como la utilizada en este proyecto (LAMP), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

355Por qué MySQL?

He escogido esta base de datos debido a su sencillez y a la vez amplitud de ventajas que ofrece. Además es una base de datos libre, y debido a su compatibilidad con el lenguaje y el servidor escogido no he tardado mucho en decidir que base de datos utilizar.

Como ventajas tenemos que es muy rápido, confiable y fácil de usar. También tiene un práctico set de características desarrollado en cercana cooperación con nuestros usuarios.

MySQL Server fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápidamente que las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, esta ofrece hoy una rica variedad de funciones. Su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para accesar bases de datos en Internet.

Ahora veremos algunas comparaciones con otras bases de datos que también me podrían haber sido útiles, y las características que me hicieron decidir por MySQL.

356Por qué MySQL y no Microsoft SQL Server?

La principal característica que me hizo decidir por uno de estos dos es que MySQL es código libre mientras que Microsoft SQL Server es código propietario. A continuación veremos una clara diferencia, observando el precio de Hosting en ambas base de datos extraído de una página web¹ que muestra la comparativa:

Hosting BBDD SQL Server		Hosting BBDD MySQL	
100 MB	50 €/mes	100 MB	25 €/mes
200 MB	90 €/mes	200 MB	45 €/mes
500 MB	200 €/mes	500 MB	100 €/mes
1 GB	350 €/mes	1 GB	175 €/mes

Figura 4 – Comparación de precios entre MySQL y SQL Server

¹<http://www.fernandoplaza.com/2006/03/mysql-vs-microsoft-sql-server.asp>

357Por qué MySQL y no PostGreSQL?

Son muchos los benchmarks que se han publicado sobre estos gestores de bases de datos, aunque muchos de ellos tienen una clara tendencia hacia uno de los dos bandos. Es por esto que hay que saber extraer bien las conclusiones a partir de un benchmark.

A continuación se resumen las conclusiones obtenidas a partir de diversos benchmark's, intentando ser lo mas arbitrario posible.

	MySQL	PostGreSQL
Lo Mejor	<ul style="list-style-type: none">- Lo mejor de MySQL es su velocidad a la hora de realizar las operaciones.- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.- Las utilidades de administración de este gestor tienen una gran facilidad de configuración e instalación.- Tiene una probabilidad muy reducida de corromper los datos.- El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet.	<ul style="list-style-type: none">- Posee una gran escalabilidad.- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
Lo Peor	<ul style="list-style-type: none">- Carece de soporte para transacciones, rollback's y subconsultas.- El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación.- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.	<ul style="list-style-type: none">- Consumir gran cantidad de recursos.- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.- Es de 2 a 3 veces más lento que MySQL.

36APACHE

361 Que es Apache?

Es un servidor HTTP de código abierto, es decir, el encargado de construir y devolver las páginas web que solicitan los navegadores. Su nombre procede de "a patchy server", por ser una versión "parcheada" en 1995 de uno de los primeros servidores web, el NCSA HTTPD, y actualmente corre en muy diversas plataformas (Unix, Windows, etc.). El servidor Apache se desarrolla dentro del proyecto HTTP Server ([httpd](#)) de la Apache Software Foundation.

Debido a su licencia libre pero no copyleft, existen también versiones propietarias de Apache, aunque es desarrollado y mantenido por la comunidad del software libre a través de la Fundación Apache. Es la auténtica "kill app" del software libre en el ámbito de los servidores y el ejemplo de software libre más exitoso (por delante incluso del kernel Linux): desde hace años, más del 70% de los servidores web de Internet corren este magnífico software y esta cifra va creciendo aún su cuota en este mercado.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

362 Por qué Apache?

He escogido este servidor ya que es uno de los servidores compatibles con PHP que mejor rendimiento dan y además es un software libre por lo que no hay que realizar ningún gasto económico.

3.7 DEBIAN

3.7.1 Que es Debian?

Debian GNU/Linux es la principal distribución Linux del proyecto **Debian**, que basa su principio y fin en el software libre.

Creada por el proyecto Debian en el año 1993, la organización responsable de la creación y mantenimiento de la misma distribución, centrado en el kernel Linux y utilidades GNU. Éste también mantiene y desarrolla sistemas GNU basados en otros núcleos (Debian GNU/Hurd, Debian GNU/NetBSD y Debian GNU/kFreeBSD).

Nace como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

3.7.2 Características de Debian

Debian se caracteriza por:

- ? La disponibilidad en varias plataformas hardware. La versión 4.0 incluye soporte para 11 plataformas (alpha, amd64, arm, hppa, i386, ia64, mips, mipsel, powerpc, s390 y sparc).
- ? Una amplia colección de software disponible. La versión 4.0 viene con 18733 paquetes.
- ? Un grupo de herramientas para facilitar el proceso de instalación y actualización del software (APT, Aptitude, Dpkg, Synaptic, Dselect , /etc/sources.list, etc).
- ? Su compromiso con los principios y valores involucrados en el movimiento del Software Libre.
- ? No tiene marcado ningún entorno gráfico en especial, pudiéndose instalar, ya sean: GNOME, KDE, Xfce, Enlightenment u otro.

3.7.3 Por qué Debian?

Principalmente he escogido Linux Debian porque éste es un sistema operativo muy competitivo dentro del mundo del software libre. Además utilizar un sistema operativo linux nos permite junto al resto de las tecnologías elegidas, realizar una plataforma LAMP.

El acrónimo LAMP se refiere a un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un esfuerzo reducido.

Por este motivo, con todas estas tecnologías escogidas obtendremos de forma sencilla un servidor estable y potente para poder realizar nuestra aplicación.

4 BASE DE DATOS

4.1 Estructura de la base de datos

A continuación explicaré la función principal de cada tabla de la base de datos y la funcionalidad que tiene cada campo dentro de ésta. Primero de todo debo comentar que las tablas de la base de datos se encontrarán en formato InnoDB, ya que quiero plasmar las relaciones entre las diferentes tablas a través de la integridad referencial que ofrece ese tipo de tablas.

Tabla centros: contiene toda la información sobre los centros en los cuales se realizan los cursillos. Ésta tabla se relaciona con las tablas de programación e incidencias ya que en éstas dos se debe especificar en qué centro ha ocurrido una incidencia o se va a realizar un cursillo.

- ? cen_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? cen_identificador: este campo contiene el identificador del centro que será utilizado como abreviatura a su nombre. Es de tipo varchar y tendrá un tamaño máximo de 255 caracteres.
- ? cen_nombre: este campo contiene el nombre del centro y es de tipo text.
- ? cen_fechaalta: este campo de tipo int y es donde almacenamos la fecha en formato timestamp en la cual hemos dado de alta al centro.
- ? cen_direccion, cen_telefono, cen_pcontacto, cen_propietari: estos campos serán de tipo text y aquí mostraremos información relativa al centro (dirección, teléfono, persona de contacto, etc...)
- ? cen_tipologia: será un campo enum donde podremos escoger que tipo de centro es: distrito, barrio, biblioteca, otros.
- ? cen_linea: también sera un campo de tipo enum el cual nos permite escoger el tipo de línea que tiene contratada el centro para su conexión a Internet. Las posibilidades que nos muestra este campo son ADSL, RDSI, Cable de fibra óptica y otros.
- ? cen_penllaç, cen_ippublica, cen_submask, cen_dns1, cen_dns2, cen_gruptreballINT, cen_teladsl, cen_loginr, cen_passwdr: estos campos que también son de tipo text en éste caso nos muestran información relativa a la conexión a Internet que tiene el centro, como también almacena datos útiles como el teléfono de contacto de la línea contratada, el login y pass para acceder al router, etc...

- ? cen_pga: es un campo de tipo enum en el que guardaremos la información relativa a la instalación del programa de gestión de acceso (Puntxarxa) en el centro, ya que tenemos centro en que sí que se utiliza el programa de gestión de acceso, pero hay otros que no.
- ? cen_observaciones: campo de tipo text donde podremos insertar información adicional sobre el centro.

Tabla usuarios: contiene toda la información sobre los diferentes usuarios que utilizarán la aplicación. En éste caso la tabla usuarios se relaciona con la tabla programación ya que los monitores serán usuarios de ésta aplicación, también se relaciona con las tablas de mensajería para poder enviar mensajes entre los diferentes usuarios y finalmente con la tabla incidencias para registrar quién ha dado de alta una incidencia concreta.

- ? usu_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? usu_login: este campo contiene el login del usuario el cual lo utilizará para entrar en la aplicación. Es de tipo varchar y tendrá un tamaño máximo de 255 caracteres.
- ? usu_nombre: este campo contiene el nombre y apellidos del usuario y es de tipo text.
- ? usu_pass: este campo contiene el password del usuario, encriptado en MD5 para asegurar su confidencialidad, y será de tipo text.
- ? usu_fechaalta: este campo de tipo int es donde almacenamos la fecha en formato timestamp en la cual hemos dado de alta al usuario.
- ? usu_activo: campo de tipo enum donde tendremos o 0 o 1 y donde 0 indicará que no está activo el usuario y 1 que sí.
- ? usu_tipo: será un campo enum donde podremos escoger que tipo de usuario es: administrador o supervisor.
- ? usu_observaciones: campo de tipo text donde podremos insertar información adicional sobre el usuario.

Tabla incidencias: contiene toda la información sobre las diferentes incidencias en los centros gestionados por la aplicación para poder llevar una buena gestión de estas. Las incidencias tal y como he comentado antes se relacionan tanto con la tabla centros como con la tabla usuarios.

- ? inc_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? inc_idcentro: este campo contiene el identificador del centro en el cual se ha dado de alta la incidencia, será un campo de tipo int.

- ? inc_idusuario: este campo contiene el identificador del usuario que ha dado de alta la incidencia, será un campo de tipo int.
- ? inc_urgencia: este es un campo de tipo enum donde encontraremos las diferentes prioridades de la incidencia (Urgente, media, baja).
- ? inc_fechaalta: este campo de tipo int es donde almacenamos la fecha en formato timestamp en la cual hemos dado de alta la incidencia.
- ? inc_fechares: este campo de tipo int es donde almacenamos la fecha en formato timestamp en la cual hemos resuelto la incidencia.
- ? inc_resuelta: campo de tipo enum donde tendremos si o no, marcando el estado de resolución de la incidencia.
- ? inc_incidencia: campo de tipo text donde podremos insertar información relativa a la incidencia, es decir qué ha ocurrido.
- ? inc_resolucion: campo de tipo text donde podremos insertar información relativa a la resolución de la incidencia o en otras palabras, explicar la solución que se ha tomado para resolver la incidencia.

Tabla acciones: ésta es una tabla muy parecida a la anterior tabla de incidencias pero con una visión un tanto diferente, ya que en ésta tabla se almacenarán todas las acciones que se realicen en los diferentes centros, como actualizaciones, etc... sin ser una solución a ninguna incidencia. En este caso solamente estará relacionada con la tabla centro para saber en qué centro se ha realizado la acción.

- ? acc_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? acc_idcentro: este campo contiene el identificador del centro en el cual se ha realizado la acción, será un campo de tipo int.
- ? acc_fecha: este campo de tipo int es donde almacenamos la fecha en formato timestamp en la cual hemos realizado la acción.
- ? acc_texto: campo de tipo text donde podremos insertar toda la información acerca de la acción realizada.

Para el sistema de mensajería se han utilizado tres tablas, una referente al emisor, otra al receptor y otra que relaciona éstas dos. Las tablas de emisor y receptor serán iguales y guardarán la misma información pero se ha hecho de ésta manera para que si un usuario quiere borrar un mensaje de su bandeja de salida se eliminará de la tabla emisor pero el mensaje permanecerá en la tabla receptor para que éste pueda visualizarlo en su bandeja de entrada. Ahora explicaré un poco más la funcionalidad de cada tabla y cada campo dentro de éstas.

Tabla mensajeria_emisor: contiene todos los mensajes enviados de un usuario a otros almacenando además la fecha actual del envío del mensaje y el estado de éste. Ésta tabla está relacionada con la tabla mensajeria_ids que es la que relaciona las tablas del emisor y el receptor y con la tabla usuarios como ya hemos comentado anteriormente.

- ? msje_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? msje_idemisor: este campo contiene el identificador del usuario emisor, será un campo de tipo int.
- ? msje_idreceptor: este campo contiene el identificador del usuario receptor, será un campo de tipo int.
- ? msje_asunto: es un campo de tipo varchar con longitud máxima de 255 caracteres y es aquí donde almacenaremos el asunto del mensaje.
- ? msje_mensaje: este campo contiene el mensaje enviado y será de tipo text.
- ? msje_fecha: este es un campo int de longitud máxima de 11 números en el cual insertaremos la fecha del envío del mensaje en formato timestamp.
- ? msje_estado: es un campo de tipo enum donde tendremos guardado el estado del mensaje, es decir si ha estado leído o no.

Tabla mensajeria_receptor: ésta tabla como ya he comentado tiene la misma funcionalidad que la anterior con la única diferencia que la gestión se lleva desde la bandeja de entrada. Tiene relacionadas las mismas tablas que la tabla emisor y contiene los mismos campos por lo ya quedan explicados en la tabla anterior.

Tabla mensajeria_ids: esta tabla relaciona las tablas emisor y receptor. Principalmente esa es su funcionalidad, relacionar los mensajes de una tabla con la otra y cuando se elimina un mensaje de una de éstas dos tablas se pone un 0 en el campo correspondiente.

- ? msji_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? msji_idmsje: este campo contiene el identificador del mensaje guardado en la tabla emisor, será un campo de tipo int.
- ? msji_idmsjr: este campo contiene el identificador del mensaje guardado en la tabla receptor, será un campo de tipo int.

Tabla alumnos: contiene toda la información sobre los diferentes alumnos que realizan los cursillos. Esta tabla se relaciona con la tabla inscripciones ya que en ésta se tendrá constancia las inscripciones de los alumnos en el curso y con la tabla asistencias donde se guardarán las asistencias de estos en cada sesión.

- ? alu_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? alu_nombre: este campo contiene el nombre y apellidos del usuario y es de tipo text.
- ? alu_fechaalta: este campo de tipo int es donde almacenamos la fecha de alta del alumno en formato timestamp.
- ? alu_fechanac: este campo de tipo int es donde almacenamos la fecha de nacimiento del alumno en formato timestamp.
- ? alu_sexo: campo de tipo enum donde tendremos el sexo del alumno: mujer o hombre.
- ? alu_trabaja: campo de tipo enum donde tendremos la situación laboral del alumno: Trabajador autónomo, trabajador no autónomo, jubilado, parado, no trabaja o ama de casa.
- ? alu_extranjero: campo de tipo enum donde tendremos la información de si el alumno es inmigrante o no: si o no.
- ? alu_disponepc: campo de tipo enum donde tendremos información relativa a la utilización del ordenador del alumno: en casa con internet, en casa sin internet, en el trabajo con internet, en el trabajo sin internet, acceso esporádico con internet, acceso esporádico sin internet o si no dispone.
- ? alu_estudios: campo de tipo enum donde tendremos información básica acerca de los estudios del alumno: nivel elemental, medio o superior.

Tabla cursos: contiene toda la información acerca de los diferentes cursos que se podrán realizar en los centros de la base de datos. Esta tabla se relaciona con la tabla de programación que requerirá del tipo de curso ha realizar.

- ? cur_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? cur_titulo: este campo contiene el nombre del cursillo y es de tipo text.
- ? cur_nsesiones: este campo de tipo int es donde almacenamos el número de sesiones que se tarda en realizar el cursillo (cada sesión equivale a un día concreto).
- ? cur_observaciones: campo de tipo text donde podremos insertar información adicional sobre el cursillo.

Tabla programación: en esta tabla almacenamos la información relativa a la programación de los diferentes cursillos, en cada centro y con un monitor en concreto. Por lo que se relaciona como ya he comentado antes con los usuarios de la aplicación que serán los monitores de los cursos, con la tabla centros que no determinará el centro donde se realiza y con la tabla cursos que determina el tipo de curso ha realizar. Pero además también está relacionada con la tabla horarios donde se guardaran los horarios de la programación y con la tabla inscripciones que almacenará las inscripciones de los alumnos en cada programación.

- ? prog_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? prog_idcurso: en este campo tendremos el identificador del tipo de cursillo que se ha programado, es de tipo int.
- ? prog_idcentro: este campo también es de tipo int y se guardara el identificador del centro para el cual se ha realizado la programación.
- ? prog_idusuario: otro campo int que contiene el identificador del usuario que impartirá el cursillo, es decir el monitor.
- ? prog_estado: este es un campo de tipo enum donde guardaremos el estado del cursillo: pendiente, realizado o anulado.
- ? prog_observaciones: campo de tipo text donde podremos insertar información adicional sobre la programación.

Tabla inscripciones: aquí guardaremos la información relativa a las inscripciones de los alumnos en un determinado cursillo, almacenando los identificadores de ambos, por lo que la tabla inscripciones se relaciona como ya hemos visto con la tabla usuarios y la tabla programación. Los dos identificadores se utilizan como claves primarias por lo que nunca podrá haber más de una vez un alumno apuntado a una programación en concreto.

- ? ins_idprog: este campo será utilizado como clave primaria de la tabla y contendrá el identificador de la programación con el cual accederemos a un determinado cursillo, en un determinado centro.
- ? ins_idalu: este campo será utilizado también como clave primaria de la tabla y contendrá el identificador del alumno el cual se ha inscrito al cursillo determinado por el identificador anterior.

Tabla horarios: en esta tabla guardaremos el horario en el que se realiza el cursillo en un determinado centro. Cada sesión será insertada en esta tabla con el identificador de la programación, por lo que estará relacionada con esta tabla, y el inicio y fin de estas. También como veremos posteriormente esta tabla se relaciona también con la tabla de asistencias ya que utilizará los identificadores de los horarios para determinar las diferentes sesiones.

- ? hor_id: este campo será utilizado como clave primaria de la tabla. Es de tipo entero autoincrementable y es utilizado como índice.
- ? hor_idprog: en este campo de tipo int tendremos el identificador de la programación el cual relaciona un cursillo determinado, en un centro y en una fecha concreta.
- ? hor_inicio: campo donde guardaremos la fecha de inicio de una sesión del cursillo en formato timestamp, por lo que es un campo de tipo int.
- ? hor_final: campo donde guardaremos la fecha de final de una sesión del cursillo en formato timestamp, por lo que es un campo de tipo int.

Tabla asistencias: aquí guardaremos la información relativa a las asistencias de los alumnos en cada sesión de un determinado cursillo, almacenando los identificadores del alumno y de la sesión, por lo que se esta tabla se relaciona con la tabla de horarios y usuarios. Tal y como pasa con la tabla inscripciones los identificadores serán claves primarias ya que no podrá haber un alumno en una misma sesión más de una vez.

- ? asi_idhor: este campo será utilizado como clave primaria de la tabla y contendrá el identificador del horario con el cual accederemos a una determinada sesión de un cursillo, en un determinado centro.
- ? asi_idalu: este campo será utilizado también como clave primaria de la tabla y contendrá el identificador del alumno el cual ha asistido a la sesión determinada por el identificador anterior.

4.2 Modelo entidad-relación

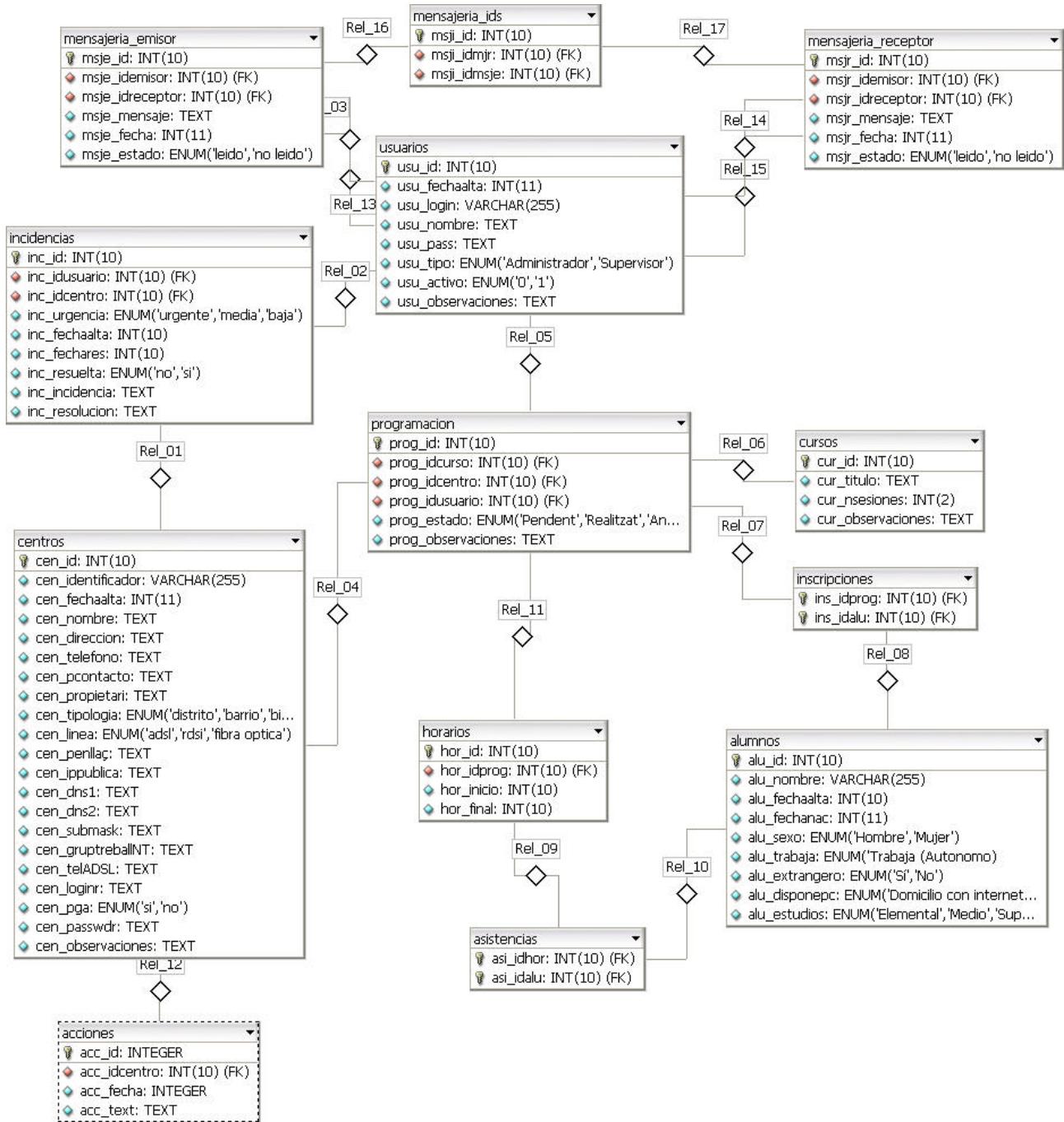


Figura 5 – Modelo Entidad-Relación

5 DISEÑO

5.1 Restricciones

Antes de empezar a diseñar la aplicación se deben tener en cuenta algunas restricciones que provienen de las tecnologías que se utilizan o de otros factores externos como puede ser el servicio de hosting que contratemos.

Empezando por éste último factor, se debe encontrar un servicio rápido que permita al usuario una conexión rápida con la aplicación y velocidad de procesamiento de las operaciones que se realice ya que no podemos permitir que se pierda demasiado tiempo en la utilización de la gestión a través de la página ya que el factor primordial es la formación de los alumnos.

Respecto a las tecnologías utilizadas tenemos algunas restricciones como puede ser en el caso de la base de datos. Al utilizar MySQL, en principio las tablas no disponen de integridad referencial pero como compensación sí que se permite el uso de índices para realizar las búsquedas complejas de texto. Todo y eso es posible realizar la base de datos con tablas con integridad referencial mediante el sistema InnoDB que permite realizar relaciones entre diversas tablas, tal y como ya he comentado en el anterior tema del diseño de la base de datos. InnoDB nos permite realizar la integridad referencial pero por el contrario, nos restringe la utilidad del uso de los índices para las búsquedas complejas de texto. Finalmente pensando en el factor que puede optimizar más a la aplicación se ha elegido el uso de InnoDB para poder tener integridad referencial entre las tablas y realizar con más facilidad los UPDATE i DELETE de los registros.

También otro aspecto a tener en cuenta es la seguridad de datos personales como las contraseñas de los usuarios, en éste caso será necesario encriptar ésta información para asegurar a la confidencialidad a los usuarios.

5.2 Objetivos

Tal y como ya se ha explicado anteriormente el objetivo básico del diseño se describe a través de la facilidad de navegación, compatibilidad con los principales navegadores, facilidad del manejo de la página y la correcta presentación de los diferentes datos que se muestran.

También se debe tener en cuenta como organizar las diferentes secciones o módulos dentro de la página para que la gestión, que es el principal objetivo de la aplicación, se pueda realizar de manera óptima sin que sea necesario tener una gran formación ni muchos conocimientos sobre la informática para la correcta manipulación de ésta, por lo que la organización de los diferentes módulos debe ser intuitiva.

Otro de los requerimientos comentados antes es la posibilidad de extensión de la aplicación en un futuro, por lo que debemos desarrollar la aplicación donde cada funcionalidad sea independiente y de ésta manera se permita añadir nuevas funcionalidades en función de las necesidades que vayan surgiendo en un futuro sin tener que modificar el código ya programado de la aplicación.

5.3 Diseño de la aplicación

Lo primero de todo antes de diseñar cualquier modulo particular es realizar el diseño de la estructura de la página, la cual será la base utilizada dónde se colgarán todos los módulos. La estructura se puede dividir en tres partes: la cabecera, el menú y el cuerpo. Ahora se realizará una breve explicación de cada una de éstas:

- Cabecera superior

La cabecera la podemos definir como la parte superior de la página, delimitada con el menú a través de una línea de 1 píxel de ancho. Dentro de ésta podemos encontrar el logo del departamento (Puntxarxa), el nombre de la aplicación con la versión actualizada y justo debajo el módulo particular en el cual nos encontramos. Finalmente contiene un recuadro dónde se especifica el nombre de usuario y el tipo que es (administrador o profesor, también llamado supervisor), el número de mensajes nuevos por el cual podremos acceder a la bandeja de entrada pulsando sobre él y un enlace para poder realizar el logout de la aplicación.

- Menú

El menú utilizado esta realizado mediante listas y animado con funciones Javascript. Éste contiene los enlaces a los diferentes módulos y su presentación se define a través de hojas de estilo en el que se especifican los tamaños de las listas, los márgenes, los colores, etc. Concretamente se utilizan dos hojas de estilo una general y otra particular para navegadores Internet Explorer de ésta manera se intenta homogeneizar el menú para cualquiera de los navegadores que se utilice.

Ejemplo:

Especificación del tamaño de la lista:

```
.horizontal li a{  
    text-decoration: none;  
    height: 1.2em;  
    float: left;  
    display: block;  
    width: 120px;  
    padding: 1px;  
    border: 1px solid;  
    border-color: white;  
}
```

Especificación del tipo de letra cuando el mouse esta encima:

```
.bluewhite li a:hover{  
    background-color: #5BB791;  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 12px;  
    font-weight: bold;  
    color: #000000;  
    text-align: center;  
}
```

- Cuerpo

El cuerpo de la aplicación es donde se mostrará cada módulo y donde se ejecutará cada funcionalidad requerida por el usuario, por lo que se detallará dentro del siguiente punto dónde se explicarán aspectos particulares del diseño de cada módulo.

Así es como finalmente queda la estructura de la aplicación para el usuario es muy sencilla en cuestión de diseño gráfico, pero muy intuitiva, fácil de manejar y rápida para cumplir con los requerimientos:

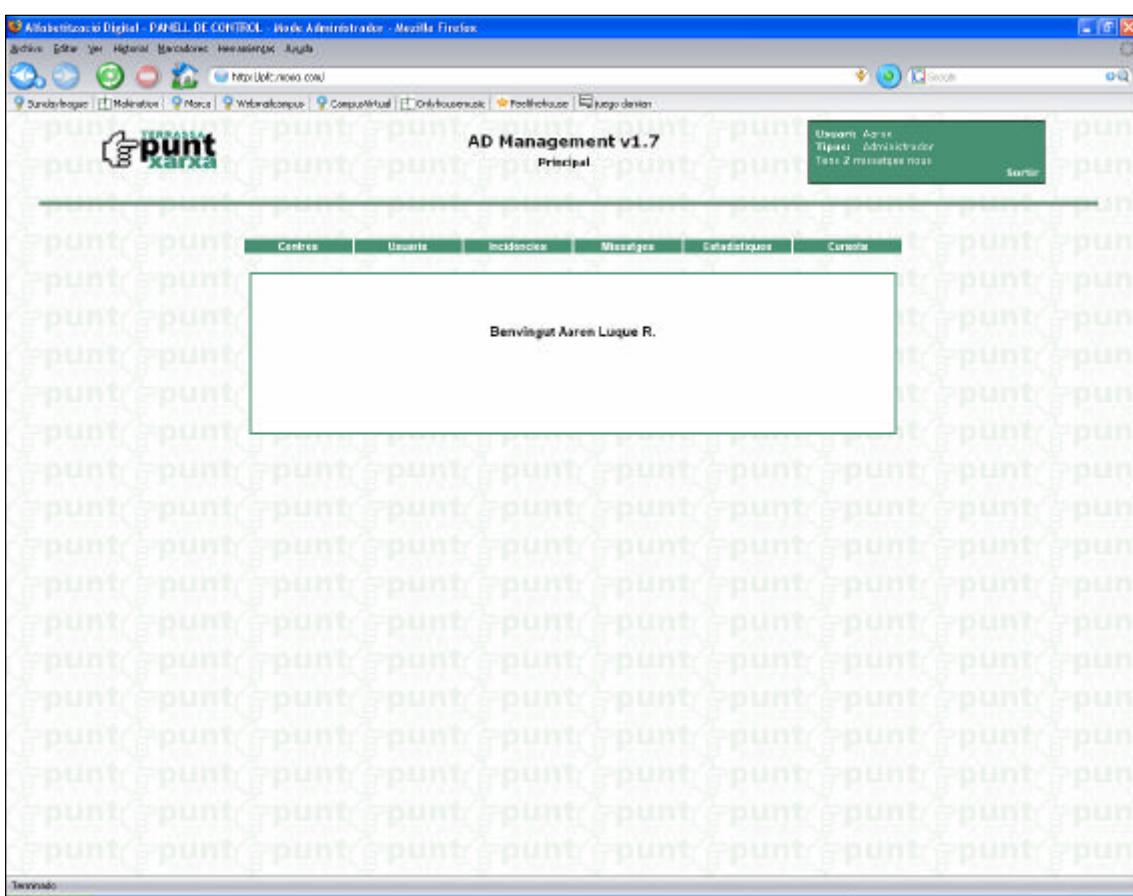


Figura 6 – Diseño de la página principal

5.4 Diseño de los módulos

Ahora mostraré los aspectos del diseño que se utilizan en los diferentes módulos y explicaré como están realizados y cuales son los módulos que lo utilizan.

- Tablas

La presentación de las tablas se define dentro de la hoja de estilo principal de la aplicación. En los diferentes módulos existen dos tipos de tablas que tienen una pequeña diferencia:

1. Las tablas utilizadas para la gestión de los centros y el mantenimiento de los usuarios. Este tipo de tablas presenta la información a partir de la primera columna de la izquierda, encontrando al final dos iconos, uno para la modificación del centro o usuario y otro para la eliminación de éste en la base de datos.

Ejemplo:

DATA ALTA	IDENTIF.	NOM GRUP	TIPOLOGIA	PGA		
05-02-2007	AVCR	Associació de Veïns Can Roca	Districte	X		
19-02-2007	AVVLL	Associació de Veïns de Vallparadís	Districte	✓		
05-02-2007	BBPR	Biblioaccés Pere Roca Dist.III	Biblioteca	✓		
05-02-2007	BBSU	Biblioaccés Salvador Utset Dist. III	Biblioteca	✓		
08-02-2007	BCTA	Biblioteca Central Adults	Biblioteca	✓		
10-02-2007	BCTRA	Biblioteca Central Ràpides	Biblioteca	✓		
01-02-2007	BCTRE	Biblioteca Central Reserves	Biblioteca	✓		
05-02-2007	BCTAU	Biblioteca Central Aula	Biblioteca	✓		
05-02-2007	BCTINF	Biblioteca Central Infantil	Biblioteca	✓		
05-02-2007	BD2A	Biblioteca Districte II Adults	Biblioteca	✓		

Figura 7 – Diseño de tablas (modelo 1)

2. Las tablas utilizadas por el resto de módulos (consulta de incidencias y acciones de mantenimiento, mensajería, programación, etc.). La tabla es como la anterior pero con una mejora, el primer cambio es el icono de modificación que además se coloca al principio antes de mostrar la información y finalmente se puede seleccionar más de un registro para su posterior eliminación. Ejemplo:

	DATA	CENTRE	ACCIÓ	ELIM.	
modificar		08/03/2007 Centre Cívic Alcalde Morera	S'ha actualitzat l'Acrobat Reader a la versió 7	<input type="checkbox"/>	eliminar
		07/03/2007 Biblioteca Central Aula	Instal·lació del programa de gestió remota	<input type="checkbox"/>	

Figura 8 – Diseño de tablas (modelo 2)

- Filtros

En los diferentes módulos siempre existe un filtro para facilitar la búsqueda de algún registro, ya sean centros, usuarios, incidencias, alumnos, etc. En cada caso hay más o menos características para poder filtrar los registros y estas se muestran de diferentes formas, a continuación describimos las formas básicas de representar los filtros:

1. Select: mediante una lista desplegable se muestran todas las características que se pueden seleccionar.

Filtres	Per tipologia	-- Tots --	<input type="button" value="Aplicar filtres"/>
	Per PGA	-- Tots --	

Figura 9 – Diseño de filtros (select)

2. Checkbox: las diferentes opciones se muestran mediante pequeñas cajas (checkbox) por lo que en este caso se permite seleccionar más de una.

Filtres				
Per resolució	Per urgència	Per usuari	Per centre	
<input type="checkbox"/> Pendents	<input type="checkbox"/> Urgent	-- Tots --	-- Tots --	<input type="button" value="Aplicar filtres"/>
<input type="checkbox"/> Resoltes	<input type="checkbox"/> Normal			
	<input type="checkbox"/> Baixa			
checkbox				

Figura 10 – Diseño de filtros (checkbox)

3. Textbox: en este caso se muestra una caja de texto por el cual podemos insertar la información para realizar el filtro.

The figure shows a user interface titled 'Filtres' (Filters). It includes dropdown menus for 'Per sexe' (Gender) with options 'Home' and 'Dona', and 'Per edat' (Age) with two empty text input fields. There is also a dropdown menu for 'Per data naixement' (Birth date) with a calendar icon. A red box highlights a text input field for 'Per nom' (Name). Below these fields is a button labeled 'Aplicar filters' (Apply filters).

Figura 11 – Diseño de filtros (textbox)

4. Calendario: para filtrar por fechas, se puede poner directamente la fecha en la caja de texto, pero en este caso se debe poner con el formato correcto. Si se desconoce éste, se puede utilizar el calendario desplegable para seleccionar la fecha deseada.

The figure shows a date range selector with 'Data inici' (Initial date) set to '03-06-2007' and 'Data fin' (Final date) set to '03-06-2007'. Below these are buttons for '?', 'Filtrar' (Filter), and a calendar icon. The calendar view for 'Junio, 2007' shows the days of the week from Sunday to Saturday. The date '03' is highlighted in blue, indicating it is selected. The calendar includes navigation buttons for months and years, and a 'Seleccionar fecha' (Select date) button at the bottom.

Figura 12 – Diseño de filtros (calendario)

- Formularios

Los formularios utilizados en los diferentes módulos de la aplicación son muy sencillos, ya que el usuario no tiene porqué tener grandes conocimientos acerca de la informática, todo y eso permite el correcto funcionamiento de la aplicación, cumpliendo con sus objetivos de manera fácil. Los elementos utilizados son principalmente textbox para poder insertar información libremente, radio en los casos en que tenemos dos opciones a escoger, select en el caso que se tenga más de dos opciones, checkbox utilizado por ejemplo en algunos filtros como hemos visto anteriormente o para controlar las asistencias de los alumnos, textarea para poder insertar gran cantidad de texto como puede ser en las observaciones y finalmente botones para enviar el formulario.

Ejemplo:

The form consists of several input fields: Nom, 1er Cognom, 2on Cognom, Data naixement (with a date input and calendar icon), Sexe (radio buttons for Home and Dona), Treball (dropdown menu for Treballa (Autònom)), Estranger (radio buttons for Sí and No), Disp. PC (dropdown menu for Domicili amb Internet), and Estudis (dropdown menu for Elemental). At the bottom is a green button labeled 'Donar d'alta'.

Figura 13 – Diseño de formularios

- Paginación

La paginación de los diferentes módulos es sencilla pero de muy fácil manejo. Se muestran los diferentes números de páginas ya que se limitan a mostrar un cierto número de registros por cada página, permitiendo el avanzar a la siguiente página, retroceder a la anterior e ir a una página en concreto.



Figura 14 – Diseño de paginación

- Hojas de inscripción y logs

Las hojas de inscripción y los logs se muestran en un diseño ideal para poder ser impresos en cualquier momento. Contienen el logotipo del ayuntamiento de Terrassa y muestran la información claramente resaltando en negrita los aspectos básicos tanto de la hoja de inscripciones como del log. En la hoja de inscripciones se utiliza una tabla bien estructurada dónde se permite escribir toda la información y que quede de manera clara y en el log cada actualización que claramente diferenciada mostrando correctamente todos sus resultados.

Ejemplo:

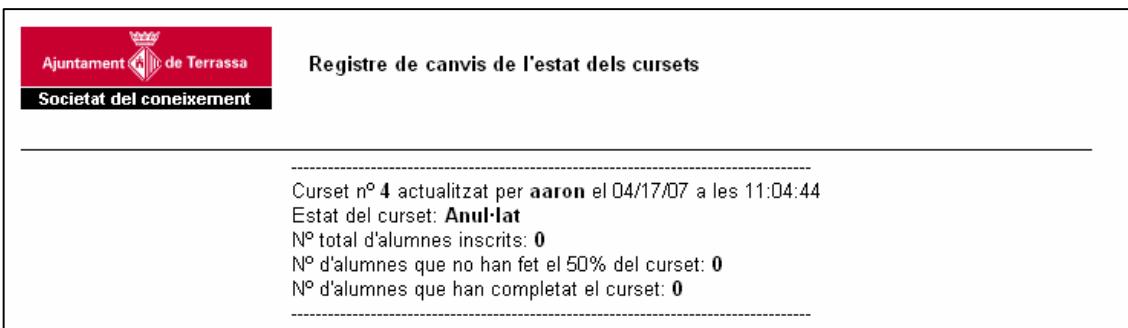


Figura 15 – Diseño de logs

Una vez hemos explicado los aspectos básicos del diseño utilizados en la aplicación, ahora detallaremos cada módulo en que consiste desde el punto de vista del diseño:

- Centros

El módulo principal de la gestión de centro se puede dividir en tres partes: una cabecera dónde tenemos un formulario con dos listas desplegables y un botón para poder filtrar, y otro que enlaza con el submódulo utilizado para añadir centros, un cuerpo en el cual tenemos la tabla en el que se muestran las características más importantes de los centros almacenados en la base de datos y donde se muestran dos imágenes: una que enlaza con el submódulo de modificación y otro para eliminar el registro de la base de datos, y finalmente la tercera parte sería el pie del módulo en el cual encontramos la paginación.

Tanto en el submódulo de alta de un nuevo centro como el de modificación, tenemos un formulario, con un diseño definido en la hoja de estilo de la aplicación, en el que tan solo utilizamos cajas de texto, listas desplegables, radios y un textarea para insertar observaciones.

- Usuarios – Datos personales

Este módulo consiste tan solo desde el punto de vista del diseño, en la representación de un formulario que permite al usuario cambiar sus datos personales.

- Usuarios – Mantenimiento usuarios

El módulo de mantenimiento de usuarios tal y como ocurre con la gestión de centros, se muestra una parte con un formulario de filtros, otra con la tabla que contiene los registros y la paginación. Además tanto el módulo utilizado para el alta como el utilizado para la modificación se basan básicamente en diseñar un formulario para poder realizar las operaciones pertinentes.

- Incidencias – Abrir incidencias

El diseño de éste módulo es simplemente un formulario con dos select y un textarea, que permiten dar de alta nuevas incidencias.

- Incidencias – Consulta de incidencias

Este módulo se puede dividir en cuatro partes: un enlace para abrir nuevas incidencias, un formulario de filtros un poco más complejo que los comentados anteriormente ya que además de listas desplegables se utilizan checkbox, la tabla que contiene los registros que a la vez es otro formulario ya que aquí se permite seleccionar más de un registro a través de un checkbox para poder eliminarlo y la paginación.

- Incidencias – Control del mantenimiento

Desde el punto de vista del diseño este módulo es como el anterior pero teniendo un formulario de filtros mucho más básico ya que solamente se dispone de un select. En este caso al abrir una nueva acción, se nos abre un popup muy simple sin barra de herramientas, ni barra de menú, ni barra de estado, con un tamaño de 620x350 píxeles y que se puede redimensionar y dispone del scrollbar.

- Mensajes – Enviar mensajes

Este módulo consta de un formulario básico con una caja de texto, un textarea y un select múltiple (para permitir más de una selección) con un tamaño para cinco registros.

- Mensajes – Mensajes recibidos y mensajes enviados

Consta de un formulario para realizar los filtros, la tabla que muestra los registros que a la vez contiene otro formulario que permite la eliminación de éstos, y la paginación. Además en la parte superior se encuentra un botón que enlaza con el módulo anterior.

- Mensajes – Mensajes recibidos y mensajes enviados

Consta de un formulario para realizar los filtros, la tabla que muestra los registros que a la vez contiene otro formulario que permite la eliminación de éstos, y la paginación. Además en la parte superior se encuentra un botón que enlaza con el módulo anterior. El popup utilizado para la visualización de los mensajes tiene un tamaño de 620x350 y no muestra las barras de herramientas, directorios, etc. Este popup consta de una tabla donde se muestra la información del mensaje y de un formulario utilizado para poder reenviarlo o responderlo.

- Estadísticas – En tablas

La selección de las estadísticas se muestra en una tabla que contiene los enlaces. Cuando se selecciona uno de ellos, se abre un popup de las características de los comentados anteriormente pero con un tamaño de 720x550. Este popup contiene un formulario para poder realizar los filtros que consiste en dos cajas de textos relacionadas con dos calendarios que nos permitirá escoger las fechas deseadas. Las estadísticas se muestran a través de una tabla generada dinámicamente dependiendo de la información que se vaya a mostrar. Finalmente tenemos un enlace para poder descargar el fichero en excel el cual se generará a partir de la tabla mostrada pero sin formato alguno.

- Estadísticas – En graficas

Las estadísticas en gráficas primero de todo tiene un formulario para escoger el centro, el cual al elegirlo abrirá un popup con las mismas características del comentado anteriormente. Este contendrá una imagen en formato .PNG, que habrá sido generada a través de la librería GD de php.

La imagen consiste en una gráfica en dos dimensiones en la cual tenemos el eje X (Tipos de curso) y el eje Y (Número total). Será un gráfico de barras en la cual para cada tipo de curso tendremos 2 barras uno para contar el número de inscripciones y otro para contar el número de cursos programados.

- Cursos – Gestión de cursos

La gestión de cursos contiene un enlace que abre un popup con el módulo que permite añadir nuevos cursos, el cual contiene un formulario, además también muestra una tabla con todos los registros que también tiene un enlace esta vez para la modificación de estos que abre un popup con el de añadir nuevos cursos y finalmente en el pie de la página tiene la paginación de los registros.

- Cursos – Programación

Contiene un formulario para realizar los filtros, una tabla para mostrar las diferentes programaciones que dentro lleva un formulario para poder realizar eliminación múltiple de los registros y poder generar las hojas de inscripción y finalmente tiene la paginación. Tanto el módulo de añadir nueva programación como el de modificar consta de un formulario generado dinámicamente dependiendo del número de sesiones que se vaya a realizar y el cual tiene unos campos (fechas de las sesiones) relacionados con un calendario tal y como he explicado en los filtros de las estadísticas. Para mostrar las hojas de inscripción explicadas anteriormente, se abre un popup sin un tamaño definido (ocupa toda la pantalla), el cual sí que tiene la barra de menú ya que se debe permitir la impresión de las hojas.

- Cursos – Inscripciones

El diseño de las inscripciones consta primero de todo de un formulario en el caso que el usuario sea un administrador, después mostrará una tabla con los diferentes cursos, en este caso los profesores también visualizan esta pantalla. Finalmente se muestra un enlace para poder inscribir los alumnos en el curso que abrirá un popup para realizar las búsquedas, una tabla que contiene los alumnos inscritos y un formulario que consta de varios checkbox para marcar sus asistencias, en el pie del módulo hay otro formulario que en el caso de los administradores tienen un select para cambiar el estado y en el caso de los profesores un botón para dar el curso por finalizado.

El popup que se abre para buscar a los alumnos tiene un link que enlaza con el módulo de añadir usuarios, un formulario para realizar búsquedas (dos cajas de texto, una de ellas relacionada con un calendario), y una tabla donde se muestran los resultados de la búsqueda y se puede seleccionar al alumno que queremos inscribir.

- Cursos – Gestión de alumnos

Como los otros módulos de gestión se puede dividir este módulo en 4 partes desde el punto de vista del diseño. Un enlace con el módulo de alta, un formulario de filtros (en este caso, combinación de checkbox, cajas de texto y calendario), una tabla que contiene un formulario donde se muestran los registros y se permite la eliminación de éstos y un enlace al módulo de modificación y finalmente la paginación. El módulo de añadir alumno contiene un mensaje que muestra la ley de protección de datos y un formulario, el cual también lo tiene el módulo de modificación, que consta de radios, selects, cajas de texto, un calendario y un botón.

- Cursos – Logs

El diseño de éste módulo es muy básico ya que simplemente utiliza un formulario para elegir la opción deseada y si se escoge visualizar el log se abre un popup como el de las hojas de inscripciones ya que se debe permitir también la impresión del log y se muestra una página como se ha explicado en el apartado anterior.

5.5 Problemas y soluciones

El primer problema surgió a la hora de realizar la estructura de la página, al principio las tablas que definían la estructura de la aplicación eran de tamaño fijo cosa que hacia que fuera homogénea para los diferentes navegadores y que si se utilizaba un tamaño de pantalla apropiado se visualizaba correctamente, pero para hacer la página más estándar y que también se pudiera visualizar con otras configuración se planteó cambiar el tamaño fijo por porcentajes cosa que hacia que la página se adaptase al tamaño que tuviera la pantalla, permitiendo así una correcta visualización en pantallas con diferentes tamaños como puede ser 800x600, 1024x768, 1024x1024, etc.

Otro problema respecto al diseño lo tenemos con el menú DHTML, éste a veces al llamarse desde el exterior si la conexión no funciona rápidamente, se carga mostrando todo el menú desplegado, además cuando seleccionas algún módulo el menú se cambia y en vez de mostrarse como una sola línea se muestra en dos. Este problema se ha intentado solucionar modificando las características del menú, longitud de celdas, etc. Pero finalmente se ha decidido que se realizará un nuevo menú estático aunque no por eso menos vistoso, ya que así será mucho más fácil su carga y solucionará todos los problemas anteriores. Además se creará este nuevo menú con iconos para que sea más visual y más intuitivo de cara a los usuarios.

Finalmente existe un problema con la paginación ya que divide todos los registros entre el número de registros por página y muestra todas las páginas disponibles. Si tenemos pocos registros no hay ningún problema pero si empezamos a tener un número considerable de registros, tendremos muchas más páginas y no podemos permitir que se muestren toda una fila de números en el pie del módulo para poder seleccionar la página deseada. Para solucionar este problema se ha decidido cambiar la paginación y en vez de mostrar las páginas como enlaces, introducir una select (lista desplegable) que muestre todas las páginas y que al seleccionar una de la lista se acceda a ella. Con esto solucionaremos el problema ya que no tendremos una fila de números sino un select de formulario donde podremos desplegar toda la lista.

6 CODIFICACIÓN

6.1 Codificación de los módulos

En este apartado se explicarán los requisitos y objetivos de cada módulo como también los aspectos técnicos de la implementación de éstos, profundizando un poco más en el aspecto técnico de programación en vez del diseño ya explicado en el apartado anterior.

- Página principal
 - o Conexión con la base de datos

Al cargar la página principal primero de todo se realiza un include a la rutina que conecta con la base de datos, ésta conecta con la base de datos correcta introduciendo el login y password y finalmente hace la llamada al fichero funcions que incluye todas las funciones php utilizadas por los diferentes módulos para que puedan ser llamadas en cualquier momento.

- o Proceso de log-in

El primer paso que haremos al visitar la página será autenticarnos, ya que antes de mostrar la página principal nos saldrá una ventana para poder introducir el nombre de usuario y el password. Una vez hemos introducido los datos, la aplicación comprobará que son correctos realizando la búsqueda en la base de datos y que es un usuario válido para utilizar la aplicación. Cuando se ha realizado las comprobaciones y todo está correcto se crea una nueva sesión y se guardan variables para recuperar rápidamente datos del usuario (nombre de usuario, nivel de acceso) y para que cada módulo compruebe que el usuario que intenta acceder ha estado logado correctamente.

Una vez validado el usuario se mostrará en el índice los módulos al cual puede acceder dependiendo del tipo de usuario que sea y los permisos que tenga, esta comprobación se realiza a través de un fichero de permisos dónde cada módulo tiene su nivel de acceso definido y se compara con el nivel de acceso del usuario.

Una vez dentro de la página principal el usuario podrá escoger el módulo al cual quiere acceder a través del menú o podrá realizar log-out.

- o Proceso de log-out

Cuando un usuario realiza el log-out de la aplicación simplemente se destruye la sesión que se había creado mediante el comando `session_destroy()` y muestra por pantalla un mensaje conforme se ha salido correctamente.

- Comprobaciones

Antes de empezar a explicar cada módulo en particular explicaré como se realizan las comprobaciones ya que todos los módulos los primero que hacen es realizar una llamada a ésta rutina para que compruebe que el usuario que intenta acceder sea el correcto. Lo primero que se comprueba es que el módulo es llamado desde la página principal, para esto se utiliza la variable `$_SERVER['PHP_SELF']` que es una variable predefinida por PHP que guarda el nombre de archivo del script ejecutándose actualmente, relativo a la raíz de documentos, por lo que miramos que el valor de esta variable sea `index.php`.

Una vez se finalizada esta comprobación, se comprueba que el usuario está logado y se recoge su nivel de acceso, finalmente se comprueba que el nivel de acceso al módulo deseado es el mismo que el nivel de acceso que tiene el usuario y si todo es correcto se permite la carga de la página, y si no está autorizado se incluye la página de error enviando un mensaje que muestra que el usuario no está autorizado para visualizar esa página.

- Centros

El principal objetivo de este módulo es la correcta gestión de los centros en los cuales se impartirán los diferentes cursos de formación. Lo primero que se realiza es acceder a la base de datos para recoger toda la información de los centros que tenemos almacenada, mostrando un resumen en la tabla principal con las características principales. En esta tabla podemos ordenar los registros por fecha o por la tipología del centro, es decir, si son bibliotecas, centros de distrito o de barrio etc. Lo que hacemos al pulsar a la columna para ordenar es insertar una nueva condición a la consulta para que nos ordene los registros.

Además podemos filtrar los centros seleccionando el tipo de centro que queremos y si dispone del programa de gestión de acceso o no, tal y como hacemos al ordenar, los filtros añadirán condiciones a la consulta para restringir ésta con los parámetros seleccionados, podemos observar en la figura 16 como está estructurada la página principal del módulo centros.

GESTIÓ DE CENTRES							
Filtres	Per tipologia	-- Tots --		Aplicar filters	Afegir nou grup		
	Per PGA	-- Tots --					
	DATA ALTA	IDENTIF.	NOM GRUP	TIPOLOGIA	PGA		
	05-02-2007	AVCR	Associació de Veïns Can Roca	Districte	X		
	19-02-2007	AWLL	Associació de Veïns de Vallparadís	Districte	✓		
	05-02-2007	BBPR	Biblioaccés Pere Roca Dist.III	Biblioteca	✓		
	05-02-2007	BBSU	Biblioaccés Salvador Utset Dist. III	Biblioteca	✓		
	08-02-2007	BCTA	Biblioteca Central Adults	Biblioteca	✓		
	10-02-2007	BCTRA	Biblioteca Central Ràpides	Biblioteca	✓		
	01-02-2007	BCTRE	Biblioteca Central Reserves	Biblioteca	✓		
	05-02-2007	BCTAU	Biblioteca Central Aula	Biblioteca	✓		
	05-02-2007	BCTINF	Biblioteca Central Infantil	Biblioteca	✓		
	05-02-2007	BD2A	Biblioteca Districte II Adults	Biblioteca	✓		

Pàgina 1 2 3 4 >>
32 grups

Figura 16 – Gestión de centros

Los centros los podremos eliminar uno a uno y no varios de golpe como se puede realizar en otros módulos, al pulsar al icono de eliminar, recargamos la página pero con nuevos parámetros que son la operación de borrar en este caso y el identificador del registro que queremos eliminar, para realizar la consulta (DELETE).

También podemos dar de alta un usuario o modificarlo, si lo damos de alta debemos llenar el formulario y enviarlo para su posterior almacenamiento en la base de datos. Y si lo queremos modificar captaremos la información del centro de la base de datos y la mostraremos en el formulario, el mismo tipo que el de dar de alta, para poder modificar algún campo y realizar el UPDATE en la base de datos.

Los formularios serán validados mediante una función JavaScript que obliga a que ciertos campos sean obligatorios y no puedan estar en blanco mostrando un mensaje si no se llenan.

- Usuarios – Datos personales

Este módulo ha estado diseñado con una funcionalidad muy clara y es que el usuario se pueda personalizar su nombre de usuario y contraseña. Al visualizar la página de datos personales, figura 17, no se visualiza el password por lo que si no se rellena este campo en la base de datos no se modificará. Si se rellena el contenido no se muestra ya que se utiliza un campo password para que los caracteres escritos no salgan por pantalla, los caracteres se muestran con un *. Finalmente se realiza el UPDATE adecuado dependiendo si se ha actualizado el campo login y contraseña o solo el campo login.

MODIFICACIÓ DADES USUARI	
Nom i cognoms	Aaron Luque R.
Nom usuari	aaron
Password	*
<input type="button" value="Modificar"/> <input type="button" value="Esborrar"/>	
* Si el password no està indicat, no serà modificat	

Figura 17 – Modificación de datos personales

- Usuarios – Mantenimiento de usuarios

Tal y como pasa con el módulo de gestión de centros, el mantenimiento de usuarios tiene como finalidad la gestión de los usuarios que van a utilizar la aplicación. Lo primero que hace es recoger de la tabla usuarios de la base de datos toda la información y muestra las características principales sobre una tabla. En este caso tenemos que en la primera columna sale una imagen que indica si el usuario está activo o no, si la imagen es un circulo verde quiere decir que el usuario está activo y si el circulo es rojo que no. Se puede activar o desactivar a los usuarios pulsando sobre la imagen del circulo la cual recargará la página pasando la operación (activar o desactivar) y el identificador del usuario para que se pueda realizar la modificación en la base de datos.

En este caso, tal y como pasa en la gestión de centros sólo se pueden eliminar los usuarios uno a uno.

El alta y modificación de los usuarios también funciona de la misma manera que los centros, se ha de rellenar el siguiente formulario y finalmente se realiza la consulta de inserción en el caso de alta de un nuevo usuario o de actualización en el caso de la modificación. El formulario, también es validado por una función JavaScript que obliga a que algunos campos no se puedan dejar vacíos, a continuación podemos ver el bucle for que valida los campos obligatorios:

```
for (i=1; i<(args.length); i++)
{
    test=args[i];
    if (((test.name=='tipus') || (test.name=='tipologia') ||
(test.name=='linea')) && (test.value=='0'))
        error+=- La selecció de '+test.name+' és errònea\n';
    else if (test.value == "")
        error+=- El camp '+test.name+ ' és obligatori\n';
}
```

En el caso de los campos que en la base de datos es un Enum, se utiliza una función realizada en php, podemos ver a continuación, que coge los valores de una columna concreta de alguna tabla (a la función se le llama pasándole el nombre del campo y de la tabla, además del identificador de la conexión) y genera un array con los posibles resultados del enum, finalmente se compara con el valor que tiene el usuario a modificar y éste queda seleccionado mientras los otros valores también se insertan en la lista del select.

```
function GetEnumValues($Table,$Column,$IdSQL)
{
    $dbSQL = "SHOW COLUMNS FROM ".$Table." LIKE '".$Column."'";
    $dbQuery = mysql_query($dbSQL,$IdSQL);

    $dbRow = mysql_fetch_assoc($dbQuery);
    $EnumValues = $dbRow["Type"];

    $EnumValues = substr($EnumValues, 6, strlen($EnumValues)-8);
    $EnumValues = str_replace(",","",",$EnumValues);

    return explode(",",$EnumValues);
}
```

- Incidencias – Abrir incidencias

Este es un módulo muy sencillo que consiste en mostrar un formulario para que los diferentes usuarios de la aplicación puedan informar de las incidencias que ocurren en el centro que están ubicados. Simplemente se debe seleccionar el centro de la lista, elegir la prioridad de la incidencia según el usuario crea y describir la incidencia. Después al enviar el formulario se realizará el INSERT en la base de datos y ésta podrá ser consultada por los administradores. El aspecto más importante de abrir incidencias es la sencillez del módulo, pero todo y eso es uno de los módulos que más importancia tienen ya que es una forma muy rápida de comunicación.

- Incidencias – Consulta de incidencias

Este apartado tiene como funcionalidad la gestión de las incidencias que se van dando de alta en la base de datos. Primero se muestran todas las incidencias sobre una tabla mostrando la fecha de alta, la prioridad, el centro, una breve descripción y si está resuelta o no. Si está resuelta se muestra una imagen con un “Visto bueno” y justo debajo aparece la fecha en la cual se ha resuelto la incidencia y si no está resuelta aparece una imagen con una cruz roja. Como se puede apreciar en la figura 18, en este caso se pueden seleccionar más de un registro para su eliminación, de ésta manera podemos “limpiar” la base de datos con todas las incidencias antiguas que ya no nos sirvan para mantener la base de datos actualizada. Para realizar esta operación se muestran unos checkbox que como valor tienen el identificador del registro al que acompañan y que al enviar el formulario se envía una lista con todos los valores seleccionados, entonces una vez se ha enviado se realiza un implode de la lista y se hace una consulta que realiza un DELETE del registro si se encuentra el identificador en la lista (DELETE FROM incidencies WHERE inc_id IN (".\$lista.")).

En este módulo se permite filtrar los registros aplicando una serie de características. En este caso se muestran filtros por resolución y por urgencia que son de tipo checkbox por lo que podremos seleccionar más de un aspecto y filtros por centro y por usuario que podremos escoger un elemento de la lista. Se puede seleccionar más de un filtro a la vez ya que se comprueba si han estado seleccionados y se añaden condiciones a la búsqueda mediante AND si son condiciones para campos diferentes u OR si son para poner más de una condición para el mismo campo.

CENTRE	PRIORITAT	DATA ALTA	INCIDÈNCIA	RESOLTA	ELIM.
AVCR	Urgent	08/03/2007	Ratoli trençat.		<input type="checkbox"/>
CCAM	Urgent	08/03/2007	L'ordinador n°1 té un...		<input type="checkbox"/>

Figura 18 – Consulta de incidencias

Como se aprecia en la figura anterior la paginación también indica además del número de incidencias totales, cuantas de éstas están sin resolver. Finalmente a través de la consulta se puede acceder al módulo comentado anteriormente para poder dar de alta una nueva incidencia y se pueden modificar pulsando el icono de la izquierda, por el cual a través de una función programada en JavaScript, que hace una llamada window.open con los parámetros deseados (altura, anchura, etc.), abre un popup con un formulario, con la información de la incidencia y dos nuevos campos que en formulario de alta no estaban: un checkbox para poder resolver la incidencia y otro textarea para poder explicar la solución tomada. Además también a parte de la fecha de alta también se muestra la fecha de la última modificación que se ha realizado.

- Incidencias – Control del mantenimiento

Este módulo es muy parecido al anterior, por no decir prácticamente idéntico, con la diferencia que en éste no se dan prioridades ya que se utiliza para insertar actualización y acciones de mantenimiento que se han realizado en algún aula. Por lo que el diseño y la funcionalidad será la misma pero únicamente tendremos tres campos para llenar e insertar en la base de datos: la fecha del mantenimiento, el centro y la acción que se ha desempeñado. Para el alta y modificación de acciones de mantenimiento se utiliza un sencillo formulario que se abre a través de la función abrir del JavaScript explicada en el apartado anterior que nos permite la visualización de páginas como popups.

- Mensajes – Enviar mensaje

El sistema de mensajería es sencillo, por lo que los usuarios lo podrán utilizar de manera rápida y eficaz pero es la forma más rápida de comunicación entre diferentes usuarios de la aplicación. El módulo de enviar mensaje primero comprueba si ha sido llamado desde la bandeja de entrada o salida para poder poner el enlace que vuelva a ésta y tener así una mejor navegación, si no es así entonces ha sido llamado directamente desde el menú por lo que no pondremos ningún enlace.

El formulario consiste básicamente en seleccionar los destinatarios, ponerle un asunto y escribir el cuerpo del mensaje. Los destinatarios se muestran en un select múltiple que permite la selección de más de un usuario y que guarda los seleccionados en un array que después se pueden recuperar fácilmente mediante un bucle for que recorre todo el array y va realizando los INSERT pertinentes.

- Mensajes – Mensajes recibidos

Este módulo es considerado como la bandeja de entrada de cualquier cuenta de correo electrónico. Aquí se muestran los mensajes recibidos, recaptando la información de la base de datos mediante SELECT, y muestra la fecha del mensaje, el remitente, el asunto y el estado. El estado se muestra mediante una imagen: un sobre cerrado si aún no ha sido leído y un sobre abierto si ya se ha consultado. Tal y como pasa con las incidencias, los mensajes se pueden borrar varios a la vez con el mismo método utilizado anteriormente y se puede filtrar por remitente y por estado del mensaje. La eliminación en este caso se hace de la tabla mensajeria_receptor, que se relaciona con la bandeja de entrada y así al eliminar un mensaje de esta tabla el emisor lo sigue teniendo en su bandeja de salida ya que el guarda los mensajes enviados en otra tabla.

CONSULTA DE MISSATGES REBUTS					
Enviar un nou missatge					
Filtres					
Per usuari	Per estat			Aplicar filtres	
baron	– Tots –				
	DATA	REMITENT	ASSUMpte	ESTAT	ELIM.
	23/03/2007	Aaron Luque R.	RE:looooooolaaaaaaaaaaaaaaaaaaaaaaa asifasfasd...		<input type="checkbox"/>
	23/03/2007	Aaron Luque R.	RE:Prova1		<input type="checkbox"/>
	23/03/2007	Aaron Luque R.	asifasfasf		<input type="checkbox"/>
	23/03/2007	Aaron Luque R.	asifasfasdf		<input type="checkbox"/>
	23/03/2007	Aaron Luque R.	Prova1		<input type="checkbox"/>

Figura 19 – Consulta de mensajes recibidos

Los mensajes como se puede deducir, no se pueden modificar ya que no tiene sentido, por lo que el icono utilizado en otros módulos como un enlace para modificar, en este caso se utiliza para visualizar el mensaje elegido. Se utiliza la función “obrir” para abrir un popup con el mensaje en el que se muestra el remitente, la fecha, el asunto y el mensaje y donde se permite la opción de reenviar o responder el mensaje.

Para reenviar el mensaje únicamente lo que se hace es en la misma pantalla del popup mostrar un formulario como el de enviar mensaje copiando el cuerpo del mensaje, poniendo el mismo asunto pero con las siglas FWD: delante y dejando al usuario escoger a los destinatarios.

Por otro lado si se desea responder se mostrará como destinatario el remitente del mensaje y no se podrá modificar ni elegir nuevos destinatarios y el asunto se copiará con las siglas RE: delante, por lo que el usuario podrá escribir el cuerpo del mensaje como respuesta. Este sistema intenta emular el sistema utilizado por cualquier correo electrónico para que al usuario le sea familiar.

- Mensajes – Mensajes enviados

Este módulo no lo comentaré ya que simplemente se diferencia del anterior en que se recoge la información de la tabla mensajeria_emisor y que la eliminación de mensajes también la realiza en esta tabla. El funcionamiento es el mismo y los métodos también, solo que en este caso se muestran los mensajes enviados (bandeja de salida) en vez de los mensajes recibidos.

- Estadísticas – En tablas

Las estadísticas en tablas tal y como indica su nombre, son tablas que muestran la información deseada por el usuario. Existen nueve tipos de estadísticas diferentes para mostrar, las cuales las podemos escoger mediante su enlace en la página principal de éste módulo. Los tipos de estadísticas los podemos observar en la siguiente figura:

ESTADÍSTIQUES EN TAULES	
	Total cursos programats
	Total cursos per estat
	Total usuaris per cursos
	Total usuaris per sexe
	Promig d'edat dels cursos
	Total edat per franja
	Total usuaris per disposició laboral
	Total usuaris per disposició de PCs
	Total anuari

Figura 20 – Estadísticas sobre tablas

Antes de explicar cada módulo particular se debe prestar atención en la cabecera utilizada por cada uno de éstos. Se trata de una tabla que contiene un formulario que permite filtrar las estadísticas entre dos fechas seleccionadas y extraer la información pertinente entre la fecha de inicio y fecha final. Se utiliza un calendario realizado en DHTML¹, en el cual mediante una función JavaScript se configura el formato de la fecha, del tiempo y permite relacionarlo con un campo input para que al seleccionar la fecha del calendario se muestre en éste. Una vez se envíe el formulario se añadirán las restricciones de fecha a la búsqueda SQL y si no se aplican filtros se realizar una búsqueda completa.

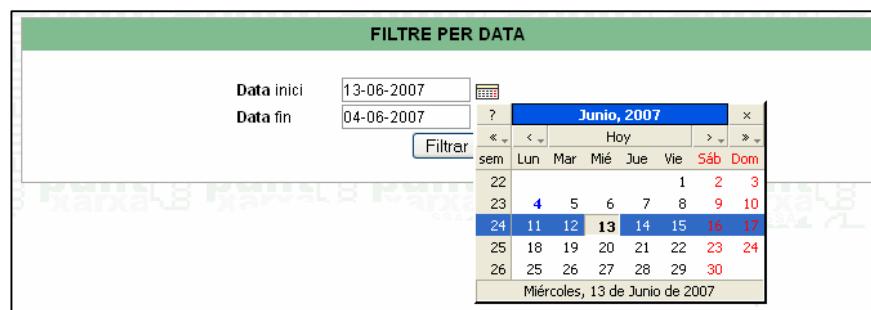


Figura 21 – Filtro de las estadísticas

Ahora explicaremos cada módulo en particular y como han sido programados:

1. Total cursos programados: se muestra el número total de cursos programados divididos por centros y dentro de cada uno de éstos se clasifican por mes. Se permite filtrar entre dos fechas, tal y como se ha comentado.

Primero se buscan los tipos de cursillos existentes en la base de datos y después los centros los cuales tienen algún registro en la tabla de programación. Entonces se busca la fecha de las programaciones encontradas en la tabla horarios y se utiliza un array dónde cada mes se contabiliza el número de cursos de ese tipo de cursos en ese centro.

¹ <http://www.dynarch.com/projects/calendar/>

También existe la opción de descargar las estadísticas, esta opción está en todos los módulos y consiste en la generación de las estadísticas en plano (sin formato) para que mediante unas cabeceras (header) se pueda descargar un archivo excel con la tabla que se estaba visualizando. Aquí podemos ver las cabeceras utilizadas para la descarga del archivo excel:

```
header("Content-Disposition: attachment; filename=\"\".$arxiu_descarrega.\"\"");  
header("Content-Type: application/vnd.ms-excel");  
header("Content-Name: \"\".$arxiu_descarrega.\"\"");
```

2. Total cursos por estado: estas estadísticas, son como la anterior pero añadiendo una nueva clasificación ya que además cada curso se divide en tres estados (pendiente, realizado o anulado). En este caso se utiliza las mismas consultas que en el apartado anterior, pero en vez de utilizar solamente un array, se utiliza uno para cada estado. Entonces una vez tenemos el identificador de la programación en vez de aumentar el array miramos su estado y aumentados el array correspondiente al estado en el que se encuentra. También se puede realizar filtros y descargar un archivo excel.
3. Total usuarios por curso: éste módulo muestra el número de usuarios que tiene cada centro para cada tipo de curso clasificados por meses. Los primeros pasos son como las estadísticas de total de cursos programados, pero una vez tenemos el identificador de la programación lo buscamos en la tabla de inscripciones realizando una agrupación (GROUP BY en sql) y contando el número total de usuarios (COUNT en sql). Una vez hemos encontrado el valor lo guardamos en el array que posteriormente mostraremos en la tabla.
4. Total usuarios por sexo: tal y como pasa con las estadísticas de total de cursos por estado respecto a la de total de cursos programados, pasa con éstas respecto a la anterior de total de usuarios por curso. En este caso obtenemos la misma información que en el módulo anterior pero la clasificamos dependiendo del sexo del usuario (hombre o mujer) utilizando un array para cada sexo que vamos aumentando mientras comparamos el sexo del usuario con hombre o con mujer.

5. Promedio de edad de los usuarios: en este módulo se realizan algunas operaciones más que en los módulos anteriores. Tal y como especifica su nombre su finalidad es la de mostrar el promedio de edad de los usuarios en cada centro para cada tipo de curso y clasificados por meses. El principio es prácticamente igual que el módulo de total de usuarios (las mismas consultas), pero en vez de ir aumentando el array cuando encontramos un usuario lo que hacemos es extraer su edad de nacimiento. Mediante un explode de la función mk_getdate() implementada en php que devuelve una fecha de timestamp a un formato estándar, recogemos el año, mes y día de la fecha de nacimiento. Entonces realizamos la resta con la fecha actual para obtener la edad, comprobando que si la resta del mes da negativo es que ya tiene un año más y si la resta del mes da cero pero la resta de los días da negativo se debe sumar un año más. Ejemplo:

```
$hoy[2]=(int)$hoy[2]-(int)$ro['cusr_dnaixement'][2];
$hoy[1]=(int)$hoy[1]-(int)$ro['cusr_dnaixement'][1];
$hoy[0]=(int)$hoy[0]-(int)$ro['cusr_dnaixement'][0];
if($hoy[1]<0) $hoy[2]--;
else if($hoy[1]==0) { if((int)$hoy[0]<0) $hoy[2]--; }
$ro['cusr_edat']=$hoy[2];
```

Finalmente para cada usuario se realiza esta operación y se va calculando la media y se guarda en el array correspondiente dependiendo del identificador del curso (buscándolo en el array cursos mediante la función array_search() de php).

6. Total de usuarios por franja de edad: éste módulo genera las estadísticas del número total de usuarios clasificados por diferentes franjas de edad (<15 años, de 15 a 34, etc.). Hasta obtener la edad del usuario se realizan las mismas operaciones que el módulo anterior pero una vez obtenemos la edad, se compara para ver en qué franja está y se aumenta el array que realiza de contador de esa franja en concreto, como podemos observar en el siguiente ejemplo. Por lo que en este caso tenemos un array para cada franja tal y como pasa en algunos de los módulos anteriores.

```
if($ro['cusr_edat']>14 && $ro['cusr_edat']<35)
$arraycontadorcurssetsf1[array_search($datah['prog_idcurset'],$arrayidcurses)]++;
if($ro['cusr_edat']>34 && $ro['cusr_edat']<55)
$arraycontadorcurssetsf2[array_search($datah['prog_idcurset'],$arrayidcurses)]++;
if($ro['cusr_edat']>54 && $ro['cusr_edat']<75)
$arraycontadorcurssetsf3[array_search($datah['prog_idcurset'],$arrayidcurses)]++;
if($ro['cusr_edat']>74 && $ro['cusr_edat']<85)
$arraycontadorcurssetsf4[array_search($datah['prog_idcurset'],$arrayidcurses)]++;
if($ro['cusr_edat']>84)
$arraycontadorcurssetsf5[array_search($datah['prog_idcurset'],$arrayidcurses)]++;
```

7. Total de usuarios por disposición laboral: este módulo es como el módulo de total de usuarios por sexo pero clasificando a los usuarios por su disposición laboral, aunque se elabora de la misma manera.
8. Total de usuarios por disposición de PCs: tal y como pasa con el módulo anterior, se buscan el número total de usuarios pero esta vez clasificados por su disposición de PCs y acceso a Internet.
9. Anuario: en este caso se hace como un resumen de las principales estadísticas mostradas anteriormente para poder hacer balance de la formación realizada entre las dos fechas seleccionadas. Estas estadísticas son las más importantes y las más utilizadas ya que demuestran la cantidad de afluencia de alumnos que ha asistido a la formación en los diferentes centros. Se utiliza una sentencia SQL para extraer el número de cursos totales en cada centro clasificados por cada tipo de curso y otra para extraer el número de inscripciones total en estos cursos. Para extraer el número de inscripciones si se ha utilizado el filtro al tener que comparar la fecha de las programaciones, se busca en la tabla horarios y como se tienen más de un registro (1 para cada sesión) el resultado de las inscripciones no era el correcto. Por lo que se decidió utilizar una tabla temporal mediante la instrucción sql CREATE TEMPORARY TABLE para obtener primero las inscripciones del curso y acto seguido realizar otra consulta utilizando el GROUP BY que hiciera la cuenta total. El resultado final se puede ver en la figura 22 y a continuación podemos ver la consulta sql explicada anteriormente:

ANUARI PUNTXARXA				
	Iniciació a internet		Iniciació a la informàtica	
	Cursets	Inscripcions	Cursets	Inscripcions
Centre Cívic Alcalde Morera	1	1	1	4
Centre Cívic Avel·lí Estrenjer	1	0	2	0
Centre Cívic Maria Aurèlia Capmany	2	1	2	2

Figura 22 – Estadísticas (Anuario)

```

mysql_query("CREATE TEMPORARY TABLE lol (
    SELECT ins_idprog, cen_nom, cur_titol, prog_idcentre, prog_idcurset
        FROM crs_programacio, centre, crs_curssets, crs_inscripcions, crs_horaris
        WHERE prog_idcentre = cen_id AND prog_idcurset = cur_id
        AND ins_idprog = prog_id AND (hor_idprog=prog_id
        AND hor_inici >= ".mk_getts($_POST['i_not_fecha_i'])." 00:00")."
        AND hor_inici <= ".mk_getts($_POST['i_not_fecha_f'])." 00:00")."
        GROUP BY ins_idusuari ORDER BY cen_nom, cur_titol)", $sqlhandle);

$query_inscr="SELECT COUNT(ins_idprog) AS contador_ins, cen_nom, cur_titol
    FROM lol GROUP BY prog_idcentre, prog_idcurset ORDER BY cen_nom, cur_titol";

```

- Estadísticas – En gráficas

Las estadísticas en gráficas son una representación gráfica del anuario para el centro deseado. Primero se debe seleccionar que centro queremos para generar sus estadísticas y automáticamente se nos abrirá un popup con la gráfica generada.

Las sentencias sql para extraer la información son las mismas que el anuario pero añadiendo la restricción del centro seleccionado. Pero esta en vez de mostrar la información por la tabla primero se define un objeto gráfico que se generará a través de la ya comentada generación de gráficas PHPCharts. Se le pondrá un nombre a la gráfica, y se le pasará el nombre al eje X y al eje Y. Finalmente mediante un bucle cogaremos el valor de los arrays y se lo pasaremos al objeto, crearemos la gráfica y la mostraremos por pantalla. A continuación veremos el bucle utilizado, como se añade los valores al objeto creado y como queda finalmente la gráfica generada:

```

for($j=0;$j<count($cursillos);$j++)
{
    $var=(int)$numeroocc[1][$j+1];
    $var1=(int)$numeroins[1][$j+1];
    $var2=$cursillos[$j+1];
    $objChart->AddValue($var2, $var, "Curssets");
    $objChart->AddValue($var2, $var1, "Inscripcions");
}

```

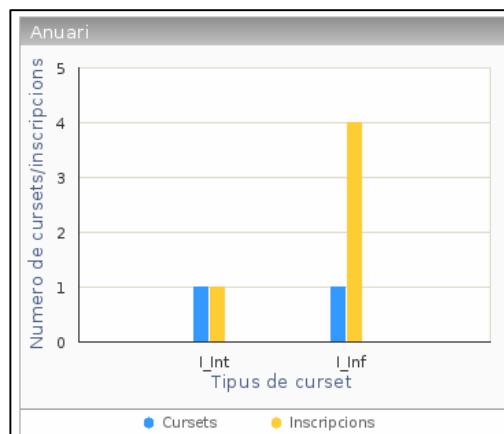


Figura 23 – Gráfica de estadísticas (Anuario de un centro)

- Cursos – Gestión de cursos

Este módulo tiene como principal funcionalidad la gestión de los diferentes tipos de curso. Tal y como ocurre con los centros, usuarios y incidencias lo primero que se muestra es una tabla con todos los cursos que están almacenados en la base de datos. En este módulo se puede eliminar los cursos uno a uno, al pulsar sobre la papelera, enviando la operación de eliminar y el identificador del registro.

Para realizar el alta y la modificación tal y como pasa en módulos anteriores simplemente se muestra un formulario en una nueva ventana (popup) que es validado mediante una función JavaScript y una vez se ha recogido los valores del formulario (a través de POST) se realiza la sentencia sql: INSERT para insertar un nuevo curso en la base de datos o UPDATE para actualizar un registro.

- Cursos – Programación

El módulo de programación es utilizado para programar diferentes cursos en los centros almacenados en la base de datos. El curso se divide en diferentes sesiones por lo que al realizar el alta de una programación se debe especificar el tipo de curso, los días y horario de inicio y final de las diferentes sesiones, el centro en el cual se va a impartir y el profesor que dará el curso. Además existen dos campos más uno con el estado del curso, que si se ha dado de alta un curso que no se ha realizado estará en estado pendiente, y un campo observaciones para poder poner información adicional. En el formulario de alta lo primero que se muestra es un select con los diferentes tipos de curso para poder elegir el tipo de curso deseado, una vez se ha seleccionado se consulta en la base de datos el número de sesiones para así poder mostrar el correcto número de campos a llenar para las fechas de éstas. Los campos son de tipo input pero se puede utilizar el calendario para insertar la fecha, tal y como hemos visto en los filtros de las estadísticas. Finalmente se puede elegir el centro, el profesor y el estado seleccionando la opción correcta de cada select. Una vez se pulsa el botón enviar se valida el formulario para que los campos obligatorios estén llenados y para que las fechas de las sesiones tengan sentido y no se ponga por ejemplo, que la sesión 2 se realiza antes que la 1.

GESTIÓ DE PROGRAMACIONS >> AFEGIR PROGRAMACIÓ

Curs	<input type="text" value="Iniciació a internet"/>												
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Sessió 1</td> <td style="width: 15%; text-align: center;">04-06-2007 00:00</td> <td style="width: 15%; text-align: center;">00:00</td> </tr> <tr> <td>Sessió 2</td> <td style="text-align: center;">04-06-2007 00:00</td> <td style="text-align: center;">00:00</td> </tr> <tr> <td>Sessió 3</td> <td style="text-align: center;">04-06-2007 00:00</td> <td style="text-align: center;">00:00</td> </tr> <tr> <td>Sessió 4</td> <td style="text-align: center;">04-06-2007 00:00</td> <td style="text-align: center;">00:00</td> </tr> </table>		Sessió 1	04-06-2007 00:00	00:00	Sessió 2	04-06-2007 00:00	00:00	Sessió 3	04-06-2007 00:00	00:00	Sessió 4	04-06-2007 00:00	00:00
Sessió 1	04-06-2007 00:00	00:00											
Sessió 2	04-06-2007 00:00	00:00											
Sessió 3	04-06-2007 00:00	00:00											
Sessió 4	04-06-2007 00:00	00:00											
Centres	<input type="text" value="Associació de Veïns Can Roca"/>												
Monitors	<input type="text" value="Becari 1"/>												
Estat	<input type="text" value="Pendent"/>												
Observacions	<input type="text"/>												
<input type="button" value="Afegir"/>													

Figura 22 –Añadir programación

En la página principal del módulo de programación se muestra una tabla con todas las programaciones almacenadas en la base de datos, por la cuales se pueden modificar abriendo un popup con un formulario como el visto anteriormente pero con la información de la programación seleccionada y se pueden eliminar varias a la vez tal y como ocurre con las incidencias y la mensajería. Podemos aplicarle varios filtros que restringirán la sentencia sql de búsqueda, añadiendo restricciones sobre el tipos de curso, el estado, el centro donde se imparte o el mes que se imparte, además podemos buscar una programación en concreto filtrando a través de su identificador, que es único.

Finalmente igual que se pueden seleccionar varias programaciones para su posterior eliminación de la base de datos, se pueden seleccionar para generar sus hojas de inscripción. Las hojas de inscripción se generan para ser impresas y llevadas a los diferentes centros para que se utilicen en la preinscripción de los diferentes cursos. El diseño de esta parte ya se ha explicado en el apartado de diseño, ahora explicaré como se genera. Se recoge la información acerca del curso de la base de datos a través del identificador que se le ha pasado a la página. Los horarios además de mostrar la fecha completa se calcula mediante unos arrays para poder escribir el nombre en catalán y su abreviatura. Finalmente mediante un bucle for generamos las 11 celdas para poder escribir la preinscripción y dibujamos la tabla.

- Cursos – Inscripciones

El módulo inscripciones es un poco diferente para los administradores que para los profesores. Los administradores lo primero que verán es una página, con un formulario que contiene 3 select para poder seleccionar los cursos de un centro concreto, con un estado concreto y en un mes concreto.

Una vez han seleccionado estos parámetros, se mostrarán todos los cursos que cumplan los requisitos deseados. En el caso de los profesores se mostrarán los cursos que tienen que impartir ellos y que tienen su estado como pendiente. Para ambos casos se utiliza una consulta sql que dependiendo el caso tendrá unas u otras restricciones y que acabará mostrando los cursos que las cumplan.

Cuando se ha seleccionado el curso deseado se muestra otra pantalla donde se muestran los alumnos inscritos al curso, mostrando sus características principales, las asistencias en ese curso y permitiendo la modificación de cualquiera de sus datos y la eliminación del usuario del curso. Para modificar sus datos se abre un popup con el módulo de modificar alumno que se explicará posteriormente. Las asistencias tal y como se puede observar en la figura 23, están divididas en el número de sesiones que tenga el curso y con un checkbox para poder marcarlas, cada vez que se pulse una asistencia se recargará la página actualizando la base de datos. Además a través de esta pantalla el administrador puede cambiar el estado del curso a través de otro formulario que tiene al final y el profesor en vez de éste formulario le aparece un botón que puede pulsar para dar el curso por finalizado y cambiarlo a estado “Realizado” en la base de datos. Cada vez que un administrador cambia el estado del curso o un profesor finaliza uno, se inserta en el fichero toda la información acerca de éste (esta información se puede ver en el apartado de diseño). En una variable se va concatenando toda la información que queremos guardar en el formato correcto, incluyendo \n, etc. Y finalmente se abre el fichero con la función fopen() y como parámetro “ab”, que si no está el fichero creado te lo crea y si está te inserta toda la información al final.

INSCRIPCIONS A CURSETS								
					Centre Civic Maria Aurèlia Capmany Dimarts, 23-01-2007 de 18:00 a 21:00 Dimecres, 24-01-2007 de 18:00 a 21:00 Dijous, 25-01-2007 de 18:00 a 21:00 Divendres, 26-01-2007 de 18:00 a 21:00			
Inscrire usuari(s)								
	NOM	EDAT	SEXЕ	TREB.	ESTR.	DISP.PC	ASSISTÈNCIES	ELIM.
<input checked="" type="checkbox"/>	Maria del pilar Casas	61	Dona	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	A la feina sense Internet	<input type="checkbox"/> 23-01-2007 de 18:00 a 21:00 <input type="checkbox"/> 24-01-2007 de 18:00 a 21:00 <input type="checkbox"/> 25-01-2007 de 18:00 a 21:00 <input checked="" type="checkbox"/> 26-01-2007 de 18:00 a 21:00	
<input checked="" type="checkbox"/>	Manel Lopez Rubio	68	Home	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Domicili sense Internet	<input checked="" type="checkbox"/> 23-01-2007 de 18:00 a 21:00 <input checked="" type="checkbox"/> 24-01-2007 de 18:00 a 21:00 <input checked="" type="checkbox"/> 25-01-2007 de 18:00 a 21:00 <input checked="" type="checkbox"/> 26-01-2007 de 18:00 a 21:00	

CANVIAR ESTAT	
Realitzat	<input type="button" value="Canviar estat"/>

[Tornar enrera](#)

Figura 23 – Inscripciones

Finalmente explicaré como se puede inscribir un nuevo usuario pulsando el botón “Inscrire usuari(s)”. Al pulsar este botón se abre un popup que permite realizar búsquedas complejas por nombre y búsquedas de alumnos por su fecha de nacimiento o ambas juntas. Para realizar la búsqueda por nombre primero se utiliza la función trim() para eliminar los posibles espacios de principio y final de cadena ya que después utilizaremos la función explode(“ ”,) para separar las palabras que se deseen utilizar para la búsqueda y mediante un bucle for y el comando OR de sql se realiza una búsqueda con todas las palabras introducidas en el campo. La fecha en cambio es mucho más sencilla ya que se pasa a formato timestamp y se añade la restricción a la consulta. Finalmente se muestran los usuarios que cumplen con las condiciones expuestas, y se podrá seleccionar cualquiera de la lista.

Una vez se ha seleccionado se le pasa a la página “padre” del popup, el identificador a través de la función

```
window.opener.enviaform('add', ".$datau['cusr_id']."'")
```

que insertará en el campo hidden addalumne el identificador del alumno a inscribir, se insertará en la base de datos la inscripción y se recargará la página para poder mostrar al nuevo alumno.

Además a parte de poder buscar alumnos, se pueden añadir de nuevos desde éste módulo, que nos abrirá un nuevo popup con la ventana de añadir alumno que se explicará a continuación, y está si ha sido llamada desde la ventada de buscar alumnos para realizar inscripciones, pasará a la ventana “padre” el identificador del registro insertado:

```
window.opener.sendtoopener('.mysql_insert_id($sqlhandle).');
```

Para así que éste se lo pueda enviar a la página principal tal y como hemos explicado antes y se pueda realizar la inscripción correctamente.

- Cursos – Gestión de alumnos

Este módulo es prácticamente igual al módulo de gestión de usuarios pero se comunica con la tabla alumnos que tiene más información por lo que los formularios de añadir y modificar serán diferentes. Además en este caso se permite la eliminación de varios alumnos a la vez, añadiendo la funcionalidad de poder seleccionar toda la lista (1 página de la paginación) y des-seleccionarla. Esta nueva funcionalidad se realiza mediante una función JavaScript que recorre todos los checkbox del formulario y los pone a true si se quiere seleccionar y a false si se quiere des-seleccionar.

A parte de los cambios en el módulo de añadir alumnos descritos ya en el apartado anterior, el modificar incluye una nueva información ya que realiza la consulta de que cursos ha realizado este alumno y en que sitios y los muestra en la ventana de modificar.

- Cursos – Logs

En el módulo de logs podemos visualizarlo o vaciarlo. Para éste último lo único se realiza es abrir el fichero con la función fopen() pasándole como parámetro w+ que abre el archivo como escritura y machaca todo lo que hay y entonces escribimos un espacio en blanco. En cambio si lo que deseamos es visualizarlo mediante la función JavaScript ya comentada anteriormente window.open() se abre un popup mostrando la información del log. La visualización del log ya se ha comentado en el tema de diseño, como parte de programación lo único que se puede comentar es que se abre el fichero con la función file() que nos devuelve todo el fichero en líneas y mediante un bucle for lo vamos escribiendo por pantalla (véase figura 15 de la página 46).

6.2 Problemas y soluciones

Como sucede en cualquier proyecto han surgido durante éste muchos problemas que se han tenido que solucionar, algunos de más importancia y otros de menos.

Uno de los principales problemas encontrados ha sido en la generación de las estadísticas, ya que al principio ha costado encontrar la sentencia sql que permitiese extraer la información deseada. En algunos casos la solución ha sido realizar varias consultas en vez de intentar extraer la información en una sola y en otros casos como el anuario la solución fue como ya se ha comentado, la creación de una tabla de uso temporal que una vez extraímos los datos la eliminábamos de la base de datos. Este problema ha significado un pequeño retraso en el desarrollo de esta parte pero como otros módulos se han realizado en menos tiempo del previsto la planificación ha quedado compensada y no ha habido ningún retraso en la entrega del mismo.

Otro problema encontrado durante la codificación fue en la parte de realizar las programaciones, ya que en un principio directamente se mostraba un formulario en el que se escogía el tipo de curso, el centro, el profesor y la fecha de las sesiones. Esto en un principio no funcionaba correctamente ya que el número de sesiones podía variar entre un curso y otro, por lo que se decidió que primero se debía escoger solamente el curso y entonces se consulta en la base de datos su número de sesiones y se genera el nuevo formulario dinámicamente. Además en la modificación de programaciones para no tener el problema comentado, se decidió bloquear el campo tipo de curso para que hubieran conflictos con el número de sesiones del curso.

En el apartado de inscripciones al principio también hubo algún tipo de conflicto. No era un aspecto que hiciese dejar de funcionar a la aplicación pero no era óptimo por lo que se decidió mejorarlo. A la hora de dar de alta nuevos alumnos a través de la página de inscripciones, al principio se llenaba el formulario y cuando se enviaba la ventana se cerraba dejando sólo la ventana de búsqueda de alumnos y no se inscribía el alumno directamente en el curso, por lo que se debía volver a buscar al alumnos que se acababa de insertar en la base de datos para inscribirlo en el curso. Después de comentar este tema se decidió optimizar la aplicación para que todas las altas realizadas a través de la página de inscripciones, una vez se inserte en la base de datos también se inscribiese al alumno al curso en el cual estábamos inscribiendo a los alumnos.

Un problema más que nos encontramos mientras se desarrollaba la aplicación fue el conflicto entre mensajes enviados y mensajes recibidos. La idea era realizar un sistema de mensajería que fuera similar a cualquier buzón de correo electrónico y que fuera sencillo de utilizar, por lo que se decidió realizar una bandeja de entrada y otra de salida. El problema lo encontramos ya que primeramente utilizábamos solo una tabla para almacenar los mensajes, y al querer eliminar un usuario un mensaje de la bandeja de entrada o de salida no podíamos eliminarlo de la base de datos ya que el otro u otros usuarios implicados tal vez no quieran eliminar el mensaje. La solución fue crear una base de datos para los mensajes de entrada y otro para los de salida y cada vez que se enviaba un mensaje se insertaba en las dos tablas, y éstas se relacionan a través de otra tabla que contiene los dos identificadores.

Entonces si se quiere eliminar un mensaje de salida, solamente se borra de la tabla de mensajes de salida y se actualiza la tabla de relaciones ya que el mensaje ha sido eliminado, por lo que el usuario que ha sido destinatario aun tiene el mensaje en la tabla de mensajes de entrada.

Finalmente un último problema encontrado ha sido en el sistema de logs, ya que no se ha conseguido insertar la información más reciente al principio del fichero. Se ha probado con varios tipos de apertura de ficheros (`w+`, `w`, etc.) pero todos estos machacaban el contenido del fichero por lo que al final se ha dejado la apertura del fichero con el parámetro `"ab"` que permite la inserción al final. Se podría haber recogido el contenido del fichero y concatenarlo a la nueva información, que se quiere almacenar, y entonces volverlo a escribir en el fichero, pero al final se decidió dejar así. Esta solución podría ser una posible mejora de la aplicación.

6.3 Pruebas

Durante la codificación se han ido realizando las pruebas particulares de cada módulo, ya que cada vez que se ha desarrollado se ha testado de varias formas, realizando las operaciones correctamente y equivocándome a propósito para ver como respondía la aplicación. Normalmente todas esas pruebas particulares han salido satisfactoriamente, excepto algunos pequeños errores que se han corregido sin ningún tipo de problema.

Una vez se ha tenido la aplicación totalmente completada, se insertaron varios usuarios y se hicieron pruebas añadiendo un par o tres de programaciones a cada usuario, con diferentes centros, y con pocos alumnos en la base de datos. La respuesta de la aplicación fue la correcta y las estadísticas se generaban correctamente, mostrando la aplicación un comportamiento satisfactorio.

Una vez probamos que la aplicación funcionaba correctamente, se aumentaron el número de usuarios hasta tener unos ocho. Se insertaron nuevos centros y se probó con un gran número de programaciones (unas 250) repartidas entre los diferentes usuarios, con un gran volumen de alumnos (unos 500), se insertaron unas 100 incidencias, etc. El resultado de la prueba fue también satisfactorio ya que seguía funcionando todo correctamente y no hubo ningún problema.

Finalmente se probó la aplicación con usuarios reales, se escogieron tres profesores que actualmente dan la formación y se les explicó un poco el funcionamiento de la aplicación. Entonces durante dos semanas, los cursos que impartían lo gestionaban desde la aplicación y yo como administrador realizaba mi parte también. Al final de las pruebas los profesores mostraban el apoyo a la aplicación ya que para ellos no suponía mucha faena el tener que gestionar las inscripciones y las asistencias a través de la aplicación y por parte de la administración se valoró muy positivamente los resultados obtenidos ya de ésta forma se pueden obtener mucha más información acerca de los alumnos y de forma automática.

6.4 Jerarquía de la aplicación

La jerarquía en que se divide los ficheros que comprenden todo el proyecto es la siguiente, una carpeta raíz donde se tienen todos los módulos, el css principal, algún fichero más de configuración y cinco carpetas en las cuales se guardan ficheros respecto a la generación de gráficas, los logs, las imágenes, las fuentes y por último las rutinas de uso común.

Los diferentes módulos, excepto los popups, no se podrán ejecutar a no ser que sean llamados desde la página principal, ya que es ésta la que contiene todas las comprobaciones. Además todos los módulos tienen acceso restringido dependiendo del nivel de acceso, ahora a continuación se detallan los diferentes ficheros de cada carpeta, explicando brevemente su objetivo, el tipo de fichero que es y el nivel de acceso que tienen.

Nos podremos encontrar ficheros de datos como puede ser documentos de texto que almacenan información que se muestra en la página web o dónde se va almacenando los diferentes cambios (log), ficheros de programación como pueden ser los diferentes módulos, rutinas, o ficheros javascript o ficheros que tratan el diseño de la aplicación como son las hojas de estilo.

Niveles de acceso como ya se ha comentado en puntos anteriores tenemos dos, los administradores, que pueden acceder a cualquier archivo de la aplicación, y los profesores.

Dentro de la carpeta raíz aparte de contener las carpetas class, font, imatges, includes i log que veremos más adelante contiene los siguientes ficheros, los cuales son la base de la aplicación dónde tenemos los diferentes módulos, hojas de estilo, etc.:

acciones.php Muestra los detalles sobre las acciones de mantenimiento realizadas.

Tipo de fichero: **Módulo**.

Nivel de acceso: **Administrador**.

centres.php Muestra los detalles sobre los centros que están dados de alta.

Tipo de fichero: **Módulo**.

Nivel de acceso: **Administrador**.

centres_add.php Rutina que añade un centro a la base de datos.

Tipo de fichero: **Módulo**.

Nivel de acceso: **Administrador**.

centres_mod.php	Rutina que se encarga de la modificación del centro seleccionado. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
chartclasses.php	Define el tipo de gráfico y realiza los includes pertinentes. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_addprog.php	Rutina que realiza el alta en la base de datos de una programación. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_curadd.php	Rutina que añade a la base de datos un nuevo curso. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_curmod.php	Módulo que permite modificar el curso seleccionado. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_curssets.php	Muestra los diferentes cursos dados de alta en la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_gestusu.php	Muestra los diferentes alumnos que están dados de alta en la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
crs_inscripcions.php	Realiza la rutina de control de las inscripciones de los alumnos en algún curso. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
crs_logs.php	Muestra las diferentes opciones que se puede realizar al log. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_notesalbrowser.txt	Contiene la información que se muestra como ayuda para la búsqueda de alumnos. Tipo de fichero: Datos . Nivel de acceso: Administrador / Profesor .

crs_prog.php	Muestra la programación almacenada en la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_progmod.php	Rutina para modificar la programación seleccionada. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
crs_usradd.php	Rutina para añadir alumnos a la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
crs_usrmod.php	Permite modificar los datos del alumno seleccionado. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
css.css	Diseño y presentación del menú DHTML. Tipo de fichero: Diseño . Nivel de acceso: Administrador .
dades.php	Rutina para modificar el login y el password del usuario. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
detail_acc.php	Permite añadir acciones de mantenimiento a la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
detail_albrowser.php	Permite la búsqueda de alumnos para la inscripción en el cursillo. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
detail_inc.php	Muestra los detalles de la incidencia seleccionada para su posterior modificación. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
detail_msj.php	Muestra los detalles del mensaje seleccionado. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
detail_send.php	Rutina que permite responder o reenviar un mensaje. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .

error.php	Muestra un mensaje de error por pantalla. Tipo de fichero: Control . Nivel de acceso: Administrador / Profesor .
est_curssets1.php	Muestra las estadísticas, sobre una tabla, del total de cursos programados. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets2.php	Muestra las estadísticas, sobre una tabla, del total de cursos divididos por su estado. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets3.php	Muestra las estadísticas, sobre una tabla, del total de usuarios por cada curso programado. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets4.php	Muestra las estadísticas, sobre una tabla, del número de usuarios en los diferentes cursos clasificados por su sexo. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets5.php	Muestra las estadísticas, sobre una tabla, del promedio de edad en los diferentes cursos. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets6.php	Muestra las estadísticas, sobre una tabla, del número de usuarios en cada curso clasificados por diferentes franjas de edad. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets7.php	Muestra las estadísticas, sobre una tabla, del número de cursos totales y número de inscripciones en cada centro. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets7g.php	Muestra gráficamente el anuario del centro seleccionado. Tipo de fichero: Módulo . Nivel de acceso: Administrador .

est_curssets8.php	Muestra las estadísticas, sobre una tabla, del número de usuarios en cada curso clasificados por su disposición laboral. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_curssets9.php	Muestra las estadísticas, sobre una tabla, del número de usuarios en cada curso clasificados por su acceso a la informática. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_formcurset.php	Rutina de filtros utilizado por todas las estadísticas anteriores. Tipo de fichero: Rutina . Nivel de acceso: Administrador .
est_graficamenu.php	Muestra los diferentes centros que se pueden seleccionar para su posterior generación del gráfico sobre el anuario del centro. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
est_taulamenu.php	Muestra los diferentes tipos de estadísticas que se pueden generar. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
full.php	Genera la hoja de inscripción que será repartida en los diferentes centros. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
hack.css	Diseño y presentación del menú DHTML sobre el navegador Internet Explorer. Tipo de fichero: Diseño . Nivel de acceso: Administrador / Profesor .
incidencies.php	Módulo que muestra todas las incidencias dadas de alta en la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
incidencies_add.php	Rutina que añade una nueva incidencia en la base de datos. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .

index.php	Módulo principal que muestra el menú, llama a las funciones generales como la conexión con la base de datos, proceso de autenticación, etc... y es de donde se cuelgan todos los módulos. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
js.js	Funciones JavaScript que animan el menú DHTML. Tipo de fichero: Javascript . Nivel de acceso: Administrador / Profesor .
log.php	Muestra por pantalla el fichero de log almacenado. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
main.css	Diseño y presentación de la aplicación web. Tipo de fichero: Diseño . Nivel de acceso: Administrador / Profesor .
manteniment_usuaris.php	Muestra los diferentes usuarios que están dados de alta para poder utilizar la aplicación. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
msg_fw.php	Muestra los mensajes enviados, es decir la bandeja de salida. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
msg_rec.php	Muestra los mensajes recibidos, es decir la bandeja de entrada. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
msg_send.php	Módulo que permite enviar un mensaje a otros usuarios de la aplicación. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
principal.php	Pantalla principal de la aplicación que muestra un mensaje de bienvenida. Tipo de fichero: Módulo . Nivel de acceso: Administrador / Profesor .
usuaris_add.php	Permite añadir nuevos usuarios de la aplicación. Tipo de fichero: Módulo . Nivel de acceso: Administrador .

usuaris_mod.php	Muestra los detalles del usuario seleccionado y permite su modificación. Tipo de fichero: Módulo . Nivel de acceso: Administrador .
La carpeta includes también es una carpeta importante ya que contiene las rutinas generales utilizadas por los diferentes módulos, el proceso de autenticación, etc. Dentro de ésta carpeta además de otra subcarpeta jscalendar dónde tenemos todos los ficheros utilizados para la visualización del calendario que se utiliza en varios puntos de la aplicación, podemos encontrar los siguientes ficheros:	
processlogin.php	Rutina que realiza la autenticación y validación del usuario. Tipo de fichero: Rutina . Nivel de acceso: Administrador / Profesor .
functions.js	Contiene todas las funciones javascript utilizadas en la aplicación. Tipo de fichero: Javascript . Nivel de acceso: Administrador / Profesor .
funcions.php	Contiene funciones implementadas en php utilizadas en varios módulos de la aplicación. Tipo de fichero: Rutina . Nivel de acceso: Administrador / Profesor .
controlacces.php	Realiza las comprobaciones conforme el usuario que intenta acceder a la página está autenticado, que todos sus datos son correctos y que tiene los permisos necesarios. Tipo de fichero: Rutina / Control . Nivel de acceso: Administrador / Profesor .
connectdb.php	Realiza la conexión con la base de datos. Tipo de fichero: Rutina . Nivel de acceso: Administrador / Profesor .
checkp.php	Comprueba que el módulo es llamado a través de la página principal (index.php). Tipo de fichero: Rutina / Control . Nivel de acceso: Administrador / Profesor .
adodb-time.inc.php	Contiene funciones para tratar las fechas de diferentes formas y realizar conversiones de manera fácil. Tipo de fichero: Rutina . Nivel de acceso: Administrador / Profesor .

El resto de carpetas que contiene la raíz no analizaremos su contenido ya que la única que contiene ficheros con código implementado es la carpeta class que pertenece a la aplicación phpcharts utilizada para la generación de gráficas mediante la librería gd. A continuación detallaré una breve explicación del uso de las siguientes carpetas en la aplicación:

<u>Carpeta</u>	<u>Función</u>
Log	Contiene el fichero de log (log.txt) dónde se almacenan todos los cambios en los estados de los cursos, su número de inscripciones, etc... para su posterior visualización a través del módulo log de la aplicación.
Font	Contiene todos los tipos de letra adicionales que no se encuentran en el sistema operativo y que son utilizados por la aplicación.
Imatges	Contiene todas las imágenes que se muestran en la aplicación.
Class	Contiene los diferentes ficheros utilizados para la generación de gráficas que utiliza el módulo de estadísticas gráficas.

7. Conclusiones

Ahora se comentarán las conclusiones finales sobre el proyecto, explicando si los objetivos propuestos en el análisis han sido realizados satisfactoriamente, las posibles mejoras que se pueden realizar en la aplicación y una pequeña valoración personal.

7.1 Objetivos alcanzados

Los objetivos que se propusieron al iniciar el proyecto han sido alcanzados con éxito hasta llegar a la finalización de éste.

Al principio del proyecto se tiene unas ideas de lo que se quiere realizar pero luego durante la programación van surgiendo de nuevas para mejorar la aplicación, éstos son unos nuevos objetivos que se han añadido durante la implementación y que también se han ido cumpliendo con éxito.

Los objetivos finalizados no han sido tan solo la realización de los diferentes requerimientos propuestos al comenzar el proyecto, sino también el saber poner en práctica los conocimientos adquiridos durante la carrera, etc...

La aplicación se ha finalizado, como ya he comentado, correctamente ya que los requerimientos se han cumplido con éxito por lo que ahora mismo tenemos una buena aplicación para gestionar los cursos de alfabetización digital, aun así con el tiempo se verá si la aplicación es suficiente o se debe ampliar, cosa que se puede realizar con total facilidad debido a la estructuración que tiene.

7.2 Posibles mejoras

Se pueden mejorar varios aspectos de la aplicación, incluso ya se está trabajando sobre algunos para ponerlos en funcionamiento cuanto antes mejor. Primero de todo se podría realizar un diseño mucho mas trabajado aunque este no sea el objetivo principal de la aplicación ya que el diseño no importa demasiado, eso sí, con un nuevo diseño un poco más visual ayudaría a los usuarios que no tienen muchos conocimientos sobre la informática a aprender a utilizar la aplicación con mucha más facilidad.

Otro aspecto, en el que ya se está trabajando, es en la incorporación de un módulo de inventario. En este módulo se podrán inventariar todos los ordenadores disponibles, introduciendo sus características, en el centro en que están ubicados, etc. Además también se permitirá el inventario de los diferentes periféricos: impresora, escáner, etc., y la gestión de éstos para poder añadir nuevos tipos de periféricos sin tener que modificar código.

Se pondrán varias funcionalidades a estos módulos, como la posibilidad de copiar datos de otro ordenador u otro periférico ya que muchos son iguales y de esta manera nos ahorraremos tener que volver a introducir los datos, además podremos introducirles propiedades específicas a cada hardware para poder tenerlos bien clasificados y especificados.

Esta ampliación de la aplicación ya está casi realizada completamente, ahora comentaré otras posibles mejoras que se podrían aplicar:

- Insertar una nueva tabla y un pequeño módulo para que gestione los diferentes servicios de formación, ya que así tendríamos los tipos de cursos y a través de qué servicio se están impartiendo.
- Generación de un pdf de estadísticas: este módulo podría ser muy interesante ya que le podrías pasar el intervalo de fechas en que quieras extraer las estadísticas y te generaría un pdf con todas ellas.
- Generación de la programación a través de tablas que representen un calendario, de esta manera sería mucho más sencillo y visual insertar las programaciones.
- Generar nuevas estadísticas.
- Realizar una página web pública con un diseño agradable para mostrar toda la información de los cursos.
- Poner un sistema de puntuación en esta nueva web pública que he comentado en el punto anterior para que los alumnos puntuaran la formación que han obtenido.

7.3 Valoración personal

Personalmente he disfrutado mucho realizando este proyecto, ya que es un proyecto real, que se va a aprovechar a partir de septiembre. Además es un proyecto que se utilizará para mejorar la forma de trabajar en el puesto de trabajo que estoy actualmente por lo que me siento muy orgulloso de haber podido realizar una aplicación de este tipo, que me permite trabajar con más facilidad con una herramienta desarrollada por mi mismo.

Como he podido comprobar no es nada fácil realizar un proyecto como este, lo más complicado no viene a la hora de programar los diferentes módulos, ni de crear la base de datos, ya que realizar una página web tampoco es excesivamente complicada aunque tienes que trabajar las mismas o mas horas que otro proyecto en otro lenguaje de programación. Sino que viene antes de todo eso, cuando tienes que pensar que especificaciones tiene el proyecto que vas a realizar, que tipo de herramientas convendría utilizar, que tablas se usarán para el correcto funcionamiento, es decir todo el análisis. Todo y eso, las especificaciones y los requerimientos no son pensados todos antes de comenzar a programar sino que también a veces van surgiendo ideas, y es necesario crear una tabla más o introducir un nuevo campo.

Otra de las partes más difíciles es la parte de diseño, ya que como siempre se ha comentado, y por mi parte lo puedo corroborar es que los programadores no son grandes diseñadores, por lo que en un proyecto de mayor magnitud dónde se requiera un gran diseño lo mejor es combinar a ambos.

Como conclusión puedo decir que ha sido bastante amena la realización del proyecto, y que no se ha hecho pesada en ningún momento. Además he tenido la oportunidad de poner a prueba los conocimientos adquiridos en algunas de las asignaturas de la carrera, que sin ellos fuese sido imposible la implementación de esta herramienta.

BIBLIOGRAFIA

- Libros

- ? **PHP 4**, Pedro Pablo Fabrega, Editorial Pearson Educación

- Teoria

- ? Enciclopedia utilizada para recopilar información teórica

- o <http://es.wikipedia.org>

- Manuales

- ? Manuales sobre como instalar PHP, todo tipo de funciones, etc...

- o <http://www.php.net/manual/es>
- o <http://www.aprenderphp.com/index.html>
- o <http://www.desarrolloweb.com/manuales/12/>

- ? Manuales sobre como montar un servidor APACHE con PHP y MySQL

- o <http://www.desarrolloweb.com/articulos/1094.php?manual=12>
- o <http://www.desarrolloweb.com/articulos/1380.php?manual=12>
- o [http://es.tldp.org/Manuales_LuCAS/manual_PHP/manual_PHP\(instalacion/instalar_windows.htm](http://es.tldp.org/Manuales_LuCAS/manual_PHP/manual_PHP(instalacion/instalar_windows.htm)

- Ayuda

- ? Ayuda sobre cualquier tipo función en PHP

- o <http://www.php.net>
- o <http://www.programacion.com/php/>

- ? Ayuda sobre dudas en la base de datos

- o <http://www.programacion.com/bbdd/>



ANEXO 1:

MANUAL DE USUARIO

Proceso de identificación

El primer paso que haremos al visitar la página será autenticarnos, ya que antes de mostrar la página principal nos saldrá una ventana, como el de la siguiente figura, para poder introducir el nombre de usuario y el password.

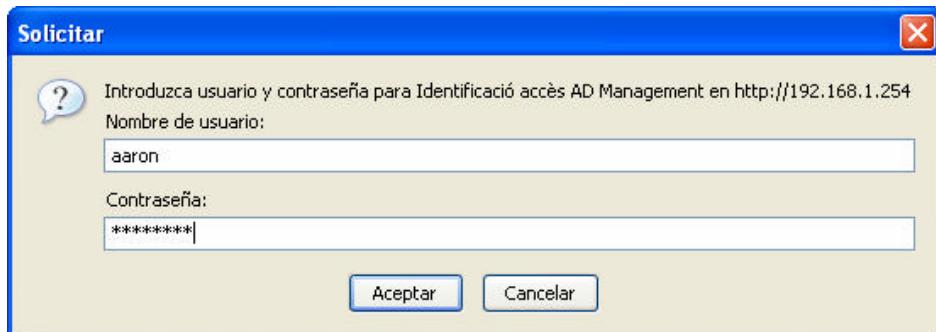


Figura 1 - Ventana de autenticación

Página principal

Una vez dentro de la página principal se podrá escoger el módulo al cual quiere acceder a través del menú que podemos ver en la figura 2.

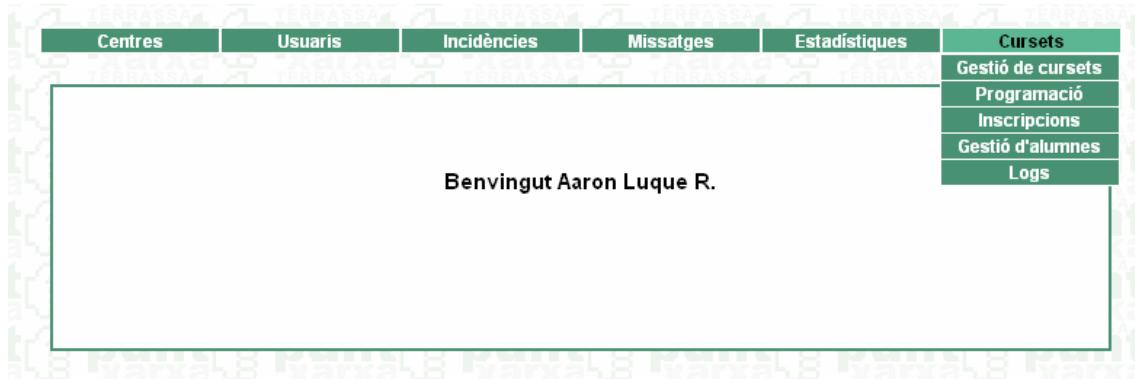


Figura 2 – Menú de la aplicación

Además tendremos un recuadro con nuestro identificador dónde se podrá realizar log-out pulsando sobre el enlace y dónde podremos ver si tenemos mensajes nuevos y visitar el módulo de mensajes recibidos pulsando sobre esta información del recuadro.



Figura 3 – Opciones de la página principal

Al pulsar sobre el enlace “Sortir”, realizaremos el proceso de log-out de la aplicación, y una vez se ha completado correctamente nos mostrará un mensaje como éste:

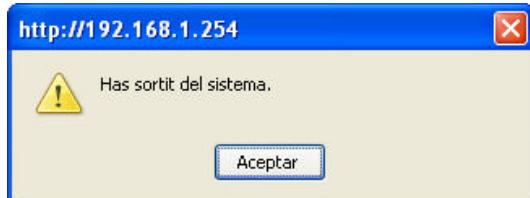


Figura 4 – Ventana de confirmación de logout

Sección Centros

Permite gestionar de forma rápida y sencilla todos los centros, mostrándolos todos los que están almacenados en la base de datos y permitiendo que se ordenen y se filtren a través de diferentes criterios. Además de poder añadir de nuevos, modificar los ya existentes o poderlos eliminar. En la página principal nos encontraremos con una pantalla como la de la siguiente figura. Aquí podremos ordenar pulsando sobre la columna deseada los centros a través de dos criterios: la fecha de alta y su tipología, la ordenación se realiza de mayor a menor y viceversa, podemos escogerla pulsando más de una vez sobre la misma columna.

El filtraje se puede escoger entre dos aspectos: filtro por tipología o filtro dependiendo si tiene programa de gestión de acceso o no. Se pueden combinar los dos filtros seleccionando las características deseadas y pulsando sobre el botón “Aplicar filtres”.

En esta tabla también podemos enlazar con la página que nos permite modificar los datos del centro pulsando sobre el ícono de la penúltima columna, y el ícono de la papelera se utiliza para eliminar el centro.

Además podemos añadir nuevos grupos pulsando sobre el botón “Afegir nou grup” y podremos ver todos los centros si vamos cambiando de página a través de la paginación que se encuentra en el pie del módulo.

GESTIÓ DE CENTRES							
Filtres	Per tipologia	-- Tots --		Aplicar filters	Afegeix nou grup		
	Per PGA	-- Tots --					
	DATA ALTA	IDENTIF.	NOM GRUP	TIPOLOGIA	PGA		
	05-02-2007	AVCR	Associació de Veïns Can Roca	Districte	X		
	19-02-2007	AVLL	Associació de Veïns de Vallparadís	Districte	✓		
	05-02-2007	BBPR	Biblioaccés Pere Roca Dist.III	Biblioteca	✓		
	05-02-2007	BBSU	Biblioaccés Salvador Utset Dist. III	Biblioteca	✓		
	08-02-2007	BCTA	Biblioteca Central Adults	Biblioteca	✓		
	10-02-2007	BCTRA	Biblioteca Central Ràpides	Biblioteca	✓		
	01-02-2007	BCTRE	Biblioteca Central Reserves	Biblioteca	✓		
	05-02-2007	BCTAU	Biblioteca Central Aula	Biblioteca	✓		
	05-02-2007	BCTINF	Biblioteca Central Infantil	Biblioteca	✓		
	05-02-2007	BD2A	Biblioteca Districte II Adults	Biblioteca	✓		

Pàgina 1 2 3 4 >>
32 grups

Figura 5 – Mòdul de centres

Para añadir un nuevo grupo una vez hemos pulsado sobre el enlace correspondiente, se debe completar el siguiente formulario, mostrado en la figura 6. No todos los campos son obligatorios, sólo lo son el identificador, el nombre, la dirección y el teléfono. Una vez se ha rellenado correctamente el formulario, se debe pulsar el botón “Guardar” para completar la operación correctamente.

GESTIÓ DE GRUPS >> NOU GRUP	
Data Alta	06 / 06 / 2007
Identificador	
Identificador (Puntxarxa)	
Nom Grup	
Adreça	
Telèfon	
Persona Contacte	
Tipologia	Districte
PGA	<input checked="" type="radio"/> Si <input type="radio"/> No
Tipus Llinia	ADSL
IP pública	
Porta d'enllaç	
Màscara de Subxaxa	
DNS 1	
DNS 2	
Grup de Treball NT	
Telèfon ADSL/Línia	
Login Router	
Password Router	
Observacions	
<input type="button" value="Guardar"/> <input type="button" value="Esborrar"/>	

Figura 6 – Formulario de añadir nuevo centro

La sección de modificar centro nos dirigirá a un formulario como el anterior pero con los campos rellenados con la información del centro seleccionado. Por lo que solamente debemos modificar los campos que deseemos y volver a guardar el formulario.

Sección Usuarios

Este módulo es utilizado para la modificación de datos personales de acceso y para la gestión de usuarios y permisos.

Datos personales

Permite modificar nuestro login de usuario y la contraseña, con tal solo modificar los campos del formulario. El campo contraseña no muestra la información ya que esta es confidencial por lo que si no se rellena, la contraseña no será cambiada, pero si por lo contrario se escribe algo en el campo, este será la nueva contraseña.

MODIFICACIÓ DADES USUARI	
Nom i cognoms	Aaron Luque R.
Nom usuari	aaron
Password	*
<input type="button" value="Modificar"/> <input type="button" value="Esborrar"/> <small>* Si el password no està indicat, no serà modificat</small>	

Figura 7 – Módulo de datos personales

Mantenimiento de usuarios

Esta sección permite llevar una gestión de los usuarios que van a utilizar la aplicación, es decir realizar las consecuentes altas, bajas y modificaciones. Además los lista utilizando una paginación para mayor comodidad en la búsqueda de usuarios.

Sobre la página principal de este módulo podemos navegar a través de la paginación para encontrar al usuario deseado, este lo podemos activar y desactivar para que pueda utilizar o no la aplicación pulsando la imagen de la columna de la izquierda. También podemos eliminar al usuario pulsando sobre la papelera o ir a la página de modificación o alta a través de los enlaces ya comentados en el módulo anterior.

MANTENIMENT D'USUARIS						
<input type="button" value="Afegir nou usuari"/>						
	DATA ALTA	NOM I COGNOMS	NOM USUARI	TIPUS	OBSERVACIONS	
●	22-12-2006	Elisa Heymann	elisa	Administrador		
●	08-12-2006	Aaron Luque R.	aaron	Administrador		
●	14-12-2006	Becari 1	bec1	Supervisor	Psswd de tots els becaris:....	
●	14-12-2006	Becari 2	bec2	Supervisor		
●	14-12-2006	Becari 3	bec3	Supervisor		

Pàgina 1
5 usuaris

Figura 8 – Módulo mantenimiento de usuarios

El alta y la modificación se realizan completando el formulario mostrado en la figura 9, todos los campos son obligatorios de llenar excepto el campo Observaciones. Una vez se ha completado el formulario o se han modificado los campos deseados se deben guardar los datos para completar la operación correctamente.

MANTENIMENT D'USUARIS >> MODIFICACIÓ USUARI	
Data Alta	08 <input type="button" value="▼"/> / 12 <input type="button" value="▼"/> / 2006 <input type="button" value="▼"/>
Nom i cognoms d'usuari	Aaron Luque R.
Nom usuari	aaron
Password	<input type="password"/> *
Tipus	<input type="button" value="Administrador"/>
Actiu	<input type="radio"/> No actiu <input checked="" type="radio"/> Actiu
Observacions	<input type="text"/>
<input type="button" value="Modificar"/> <input type="button" value="Valors inicials"/>	
<small>* Si el password no està indicat, no serà modificat</small>	

Figura 9 – Formulario de modificación de usuarios

Sección Incidencias

Se utiliza para gestionar las incidencias dentro de los puntos de acceso. Se permite la eliminación de estas en grupo, es decir eliminar varias incidencias a la vez, seleccionando las que queremos eliminar a través del checkbox, y pulsando al botón eliminar (figura 10). Cada incidencia está ligada a un centro al usuario que la redacta. En la tabla del módulo se muestran las principales características de la incidencia, mostrando si están resueltas y la fecha de resolución, o no. Las incidencias se muestran paginadas para mejor organización y además se pueden filtrar a partir de varias características seleccionando la opción deseada a través del formulario de filtros.

CENTRE	PRIORITAT	DATA ALTA	INCIDÈNCIA	RESOLTA	ELIM.
CCAM	Normal	06/06/2007	S'han importat la bola d'un...	X	<input type="checkbox"/>
BCTAU	Urgent	06/06/2007	El pc nº 10 esta...	✓	<input type="checkbox"/>

Figura 10 – Módulo de consulta de incidencias

Para dar de alta una incidencia, como podemos ver en la figura 11, únicamente se ha de seleccionar la prioridad de ésta y redactarla, para que los administradores la puedan consultar. En cambio en la modificación de éstas (figura 12), tenemos dos campos más: uno para dar por resuelta la incidencia y otra para redactar la solución tomada.

Figura 11 – Formulario para dar de alta una incidencia

DETALL D'INCIDÈNCIA

Usuari	Aaron Luque R.
Centre	Biblioteca Central Aula
Urgència	Urgent
Data Alta / Modificació	06/06/2007, a les 12:49 (Modificat: 06/06/2007, a les 12:50)
Incidència	El pc nº 10 esta infectat amb 4 virus!!
Resolta	<input checked="" type="checkbox"/>
Resolució	
Guardar canvis	

Figura 12 – Formulario para la modificación de una incidencia

La tercera posibilidad del módulo de incidencias es la sección de control del mantenimiento. Esta sección puede parecer que tenga la misma finalidad que las incidencias pero no es así, ya que este módulo se utiliza para registrar todas las actualizaciones y acciones de mantenimiento que se realicen. La forma de gestionar estas acciones se realiza de la misma manera que las incidencias, permitiéndonos realizar altas, bajas y modificaciones con toda facilidad. A continuación podemos ver que el módulo se estructura de la misma manera que el anterior por lo que su manejo es el mismo:

REGISTRE D'ACTIONS DE MANTENIMENT A CENTRES

[Afegeix nova acció](#)

Filtres

Per centre:

	DATA	CENTRE	ACCIÓ	ELIM.
	06/06/2007	Centre Cicle Montserrat Roig	Actualització del Acrobat Reader a la versió 8	<input type="checkbox"/>

[Eliminar registre](#)

Pàgina 1 de 1

DETALL DE LA ACCIÓ

Data	06-06-2007
Centre	Associació de Veïns Can Rocío
Acció	

Guardar canvis

Terminado

Figura 13 – Módulo de control del mantenimiento

Sección mensajes

Este módulo consiste en un sistema de mensajería entre usuarios. La primera opción que nos aparece es la de enviar mensaje, lo único que se debe hacer para enviar un mensaje es primero seleccionar los destinatarios, se puede seleccionar mas de uno si seleccionan manteniendo pulsada la tecla control. Después solo se ha de llenar el asunto y el cuerpo del mensaje y enviar el formulario.

Figura 14 – Formulario para enviar un nuevo mensaje

En las dos siguientes opciones del módulo mensajes (Mensajes recibidos y mensajes enviados) se nos mostrará una pantalla similar, la única diferencia es que en el primero encontraremos los mensajes que nos han enviado y en el segundo se nos mostrará los mensajes que hemos enviado. Los mensajes los podemos filtrar, como se puede observar en la figura 15, por el estado del mensaje (leído o no) y por el usuario que no han enviado el mensaje en el apartado de recibidos o por el usuario al que hemos enviado el mensaje en el apartado de enviados.

Además podemos eliminar los mensajes seleccionando los deseados y pulsando el botón eliminar. También para mejor organización se muestran los mensajes paginados por lo que para buscarlos podremos navegar a partir de la paginación que se encuentra debajo de la tabla. Finalmente podremos leer el mensaje pulsando sobre el icono de la columna de la izquierda el cual se nos mostrará en una nueva ventana (figura 16), este mensaje lo podremos responder y reenviar en el caso que sea un mensaje recibido o solamente reenviar en el caso que sea un mensaje enviado. Al responder un mensaje se nos pondrá el asunto con un RE: delante y el destinatario será la persona que ha enviado el mensaje por lo que éste no se podrá cambiar y tan solo se ha de escribir la respuesta. En cambio para reenviar el asunto se muestra con un FWD: delante y se ha de seleccionar de la lista los destinatarios y en el cuerpo del mensaje aparecerá el mensaje ya escrito.

CONSULTA DE MISSATGES REBUTS

Enviar un nou missatge

Filtres

Per usuari	Per estat	Aplicar filtres
– Tots –	– Tots –	

	DATA	REMITENT	ASSUMpte	ESTAT	ELIM.
	07/06/2007	Aaron Luque R.	Mensaje leido		<input type="checkbox"/>
	07/06/2007	Aaron Luque R.	Prueba		<input type="checkbox"/>

Actualitzar canvis

Pàgina 1
2 missatges

Figura 15 – Módulo de consulta de mensajes recibidos



Figura 16 – Ventana de visualización del mensaje

Sección Estadísticas

En esta sección podemos ver diferentes tipos de información sobre los cursos realizados, sus usuarios, etc... Lo primero que debemos hacer es elegir si queremos ver las estadísticas en tabla o gráficamente.

En las estadísticas sobre tablas tenemos nueve tipos que veremos a continuación, una vez hemos seleccionado el tipo que queremos visualizar se nos abrirá un popup con las estadísticas y un filtro por el cual podemos filtrar entre dos fechas que escojamos. Lo único que se debe hacer es seleccionar la fecha a través del calendario o introducirla a mano con el formato Dia-Mes-Año y pulsar el botón filtrar.

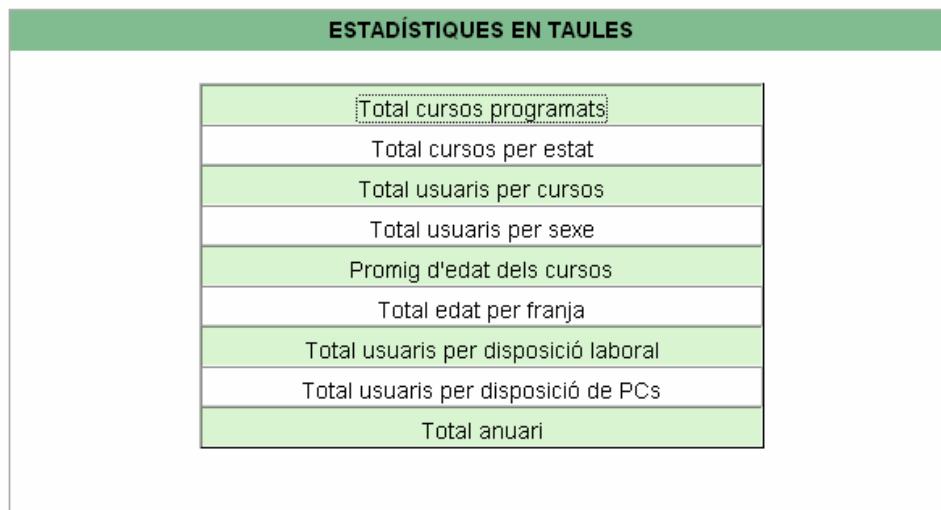


Figura 17 – Módulo de estadísticas en tablas

Entonces se nos generarán las estadísticas entre las dos fechas elegidas y posteriormente podemos descargar las estadísticas en formato excel mediante el enlace que tenemos al final de la ventana (Descargar).

A continuación veremos un ejemplo de las estadísticas, con su filtro y su enlace para descargar:

	Iniciació a la informàtica	Iniciació a internet	Iniciació a la Informàtica PL
Centre Cívic Alcalde Morera			
Juny 2007	1	0	0
Biblioteca Central Aula			
Juny 2007	0	1	0

Figura 18 – Estadísticas: total cursos programados

Las estadísticas también se pueden mostrar gráficamente. Se selecciona, el centro sobre el que queremos generar las estadísticas y se nos abrirá una ventana con una imagen que contiene la gráfica deseada.

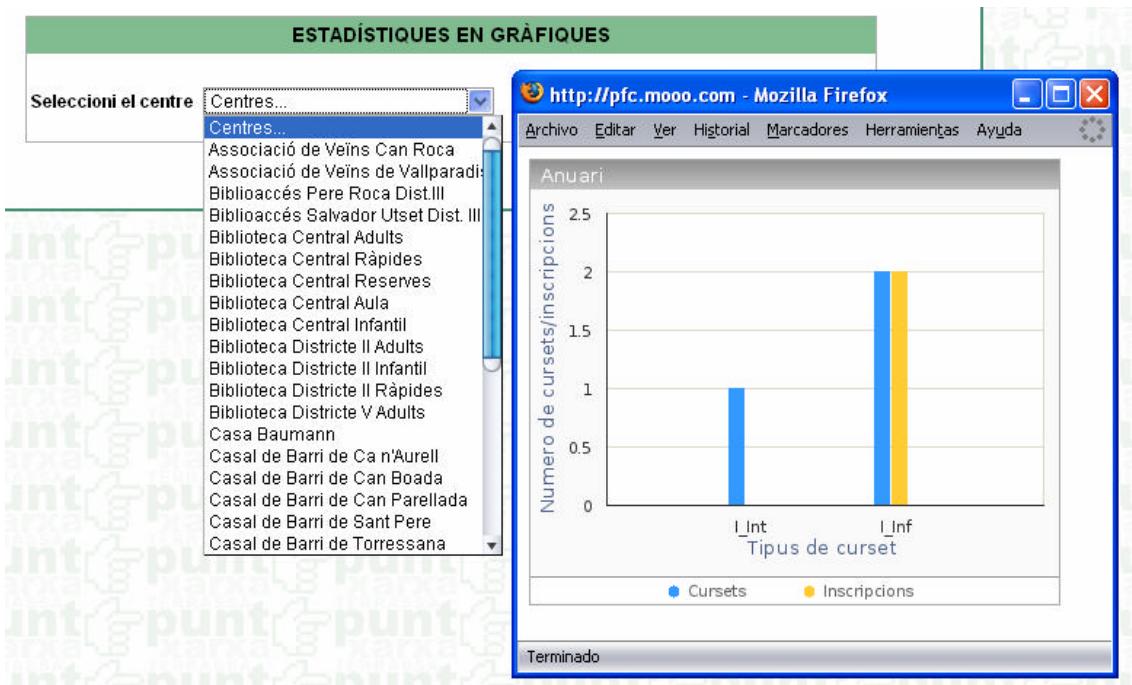


Figura 19 – Módulo de estadísticas en gráfica

Sección cursos

Esta sección permite llevar un control sobre los cursos realizados en los diferentes centros, a través de un seguimiento de los alumnos que los realizan. Además permite llevar un buen mantenimiento de los cursos e ir creando nuevas programaciones cada mes.

Gestión de cursos

En este módulo se permite eliminar, modificar o añadir nuevos cursos. Los primero que se muestra es una tabla con todos los cursos que están dados de alta, éstos se pueden eliminar uno a uno mediante el icono de la papelera. Los cursos se muestran paginados por lo que para buscarlos deberemos navegar a través de la paginación. Para modificar alguna característica del curso solamente se debe pulsar sobre el icono que acompaña a éste y si queremos dar de alta un nuevo curso se ha de pulsar sobre el enlace “Afegir nou curset”.

GESTIÓ DE CURSETS					
<input type="button" value="Afegir nou curset"/>					
	IDENTIF.	NOM CURSET	CENTRES ON S'IMPARTEIX	Nº SESIONS	
<input type="checkbox"/>	2	Iniciació a internet	Centre Cívic Alcalde Morera	4	<input type="button" value="Borrar"/>
<input type="checkbox"/>	1	Iniciació a la informàtica	Centre Cívic Alcalde Morera	4	<input type="button" value="Borrar"/>
<input type="checkbox"/>	3	Iniciació a la Informàtica PL	N/A	4	<input type="button" value="Borrar"/>

Pàgina 1
3 curselets

Figura 20 – Módulo de gestión de cursos

Para añadir el nuevo curso o modificarlo solo se ha de rellenar el formulario correspondiente o modificar los campos deseados y enviar el formulario. Los campos correspondientes son, como podemos observar en la siguiente figura, el título del curso, el identificador (ha de ser un diminutivo del título), el número de sesiones y se opcionalmente se pueden insertar observaciones acerca de éste.

MODIFICAR CURSET	
Títol	Iniciació a la informàtica
Identificador	I_Inf
N' Sesions	4
Observacions	Iniciació + Processador de textos
<input type="button" value="Modificar"/>	

Terminado

Figura 21 – Formulario de modificación del curso

Programación

Dentro de este módulo es donde se llevará todo el control de las programaciones, así aquí se podrán crear de nuevas, modificarlas y eliminar las ya creadas.

Para realizar una búsqueda más rápida de la programación deseada tenemos unos filtros donde podremos buscar por tipos de curso, por centro, por estado, por mes y por identificador. Una vez hemos realizado la búsqueda en la tabla sólo se nos mostrarán los resultados obtenidos y a partir de aquí las podremos eliminar o generar sus hojas de inscripción, seleccionando las deseadas o modificar alguna programación en concreto.

GESTIÓ DE PROGRAMACIONS						
Afegir nova programació						
Filtres						
Per curset		Per centre				
Tipus de curset...		Centres...				
Per estat		Per mes				
Estat...		Tots				
Per ID		Aplicar filtres				

ID	TIPUS CURSET	CENTRE	ESTAT	DATES	
13	Iniciació a internet	Centre Cívic Alcalde Morera	Pendent	06-06-2007 de 00:00 a 00:00 06-06-2007 de 00:00 a 00:00 06-06-2007 de 00:00 a 00:00 06-06-2007 de 00:00 a 00:00	<input type="checkbox"/>
12	Iniciació a la informàtica	Centre Cívic Alcalde Morera	Pendent	16-06-2007 de 10:00 a 12:00 17-06-2007 de 10:00 a 12:00 18-06-2007 de 10:00 a 12:00 19-06-2007 de 10:00 a 12:00	<input type="checkbox"/>
11	Iniciació a la informàtica	Centre Cívic Alcalde Morera	Pendent	10-06-2007 de 10:00 a 12:00 11-06-2007 de 10:00 a 12:00 12-06-2007 de 10:00 a 12:00 13-06-2007 de 10:00 a 12:00	<input type="checkbox"/>

Figura 22 – Módulo de gestión de la programación

Para dar de alta una nueva programación, primero de todo hemos de seleccionar el tipo de curso.

GESTIÓ DE PROGRAMACIONS >> AFEGIR PROGRAMACIÓ	
Curset	<input type="button" value="Tipus de curset..."/> Tipus de curset... Iniciació a internet Iniciació a la informàtica Iniciació a la Informàtica PL

Figura 23 – Selección de curso

Una vez hemos seleccionado el tipo de curso nos saldrá un formulario como el de la figura 24, que hemos de llenar introduciendo las fechas y la hora en que se harán las sesiones (pulsando el calendario o escribiéndolas a mano en el formato Día-Mes-Año Hora:Minuto), el centro donde se impartirá y el profesor que dará la formación. Además hay un campo con el estado del curso, pero supuestamente al crear una nueva programación el estado será Pendiente por lo que se debería dejar por defecto, también hay un campo observaciones utilizado para insertar información adicional.

GESTIÓ DE PROGRAMACIONS >> AFEGIR PROGRAMACIÓ

Curset	Iniciació a internet
Sessió 1	07-06-2007 00:00 <input type="button" value="Calendari"/>
Sessió 2	07-06-20 ? <input type="button" value="Calendari"/> Junio, 2007 <input type="button" value="x"/>
Sessió 3	07-06-20 00:00 <input type="button" value="Calendari"/> sem Lun Mar Mié Jue Vie Sáb Dom 22 1 2 3 23 4 5 6 7 8 9 10 24 11 12 13 14 15 16 17 25 18 19 20 21 22 23 24 26 25 26 27 28 29 30
Centres	Associació <input type="button" value="Selecció de centre"/> Selecció de centre
Monitors	Becari 1
Estat	Pendent
Observacions	<input type="text"/>
<input type="button" value="Afegir"/>	

Figura 24 – Formulario para añadir programación

Para la modificación se nos abrirá un formulario igual pero con la información de la programación seleccionada, la cual podremos cambiar los campos que nos interese excepto el tipo de curso.

Inscripciones

A través de este módulo se pueden ver todos aquellos cursos que están programados para controlar las inscripciones de los alumnos. Para realizar una búsqueda más rápida por parte de los administradores hay un filtro, como el de la figura 25, que permite encontrar los cursos buscando por centro, estado y mes de realización. En el caso de los profesores no existe este filtro ya que a ellos solo se les mostrarán las programaciones de los cursos que están ellos asignados como profesores y que además el estado del curso es pendiente.

INSCRIPCIONS A CURSETS

Menú Administrador - Cercar cursets

Centres	Tots <input type="button" value="Cerca"/>
Estat del curset	Pendent <input type="button" value="Cerca"/>
Mes de realització	Tots <input type="button" value="Cerca"/>
<input type="button" value="Cerca"/>	

Figura 25 – Filtros para mostrar los cursos programados

Una vez se ha realizado la búsqueda, nos aparecerán los cursos que cumplen con los requisitos seleccionados y solamente se ha de pulsar sobre el curso deseado:

INSCRIPCIONS A CURSETS					
Cursos disponibles amb estat Pendent					
ID	CENTRE	CURSET	MONITOR	HORARIS	SELECCIONAR
11	Centre Cívic Alcalde Morera	Iniciació a la informàtica	Becari 2	10-06-2007 de 10:00 a 12:00 11-06-2007 de 10:00 a 12:00 12-06-2007 de 10:00 a 12:00 13-06-2007 de 10:00 a 12:00	
12	Centre Cívic Alcalde Morera	Iniciació a la informàtica	Becari 2	16-06-2007 de 10:00 a 12:00 17-06-2007 de 10:00 a 12:00 18-06-2007 de 10:00 a 12:00 19-06-2007 de 10:00 a 12:00	
13	Centre Cívic Alcalde Morera	Iniciació a internet	Becari 1	06-06-2007 de 00:00 a 00:00 06-06-2007 de 00:00 a 00:00 06-06-2007 de 00:00 a 00:00 06-06-2007 de 00:00 a 00:00	

Tornar enrera

Figura 26 – Selección del curso deseado

Dentro de este apartado se podrá:

- Modificar los datos del alumno mediante el botón de modificar.
- Eliminar el alumno del curso seleccionado pulsando sobre la papelera.
- Marcar o desmarcar las asistencias de los alumnos pulsando sobre el recuadro que acompaña a la sesión deseada.
- Inscribir alumnos con el botón “Inscrire Usuari(s)”.

Además de estas opciones el administrador a partir de aquí podrá cambiar el estado del curso mientras que el profesor solamente podrá dar el curso por finalizado.

INSCRIPCIONS A CURSETS																								
Inscripcions - ID 11 Iniciació a la informàtica				Centre Cívic Alcalde Morera Diumenge, 10-06-2007 de 10:00 a 12:00 Dilluns, 11-06-2007 de 10:00 a 12:00 Dimarts, 12-06-2007 de 10:00 a 12:00 Dimecres, 13-06-2007 de 10:00 a 12:00																				
<input type="button" value="Inscrire usuari(s)"/> <table border="1"> <tr> <td>NOM</td> <td>EDAT</td> <td>SEXÈ</td> <td>TREB.</td> <td>ESTR.</td> <td>DISP.PC</td> <td>ASSISTÈNCIES</td> <td>ELIM.</td> </tr> <tr> <td>Luis Lopez Rubio</td> <td>0</td> <td>Home</td> <td>X</td> <td>X</td> <td>Domicili amb Internet</td> <td> <input checked="" type="checkbox"/> 10-06-2007 de 10:00 a 12:00 <input checked="" type="checkbox"/> 11-06-2007 de 10:00 a 12:00 <input type="checkbox"/> 12-06-2007 de 10:00 a 12:00 <input type="checkbox"/> 13-06-2007 de 10:00 a 12:00 </td> <td></td> </tr> </table>									NOM	EDAT	SEXÈ	TREB.	ESTR.	DISP.PC	ASSISTÈNCIES	ELIM.	Luis Lopez Rubio	0	Home	X	X	Domicili amb Internet	<input checked="" type="checkbox"/> 10-06-2007 de 10:00 a 12:00 <input checked="" type="checkbox"/> 11-06-2007 de 10:00 a 12:00 <input type="checkbox"/> 12-06-2007 de 10:00 a 12:00 <input type="checkbox"/> 13-06-2007 de 10:00 a 12:00	
NOM	EDAT	SEXÈ	TREB.	ESTR.	DISP.PC	ASSISTÈNCIES	ELIM.																	
Luis Lopez Rubio	0	Home	X	X	Domicili amb Internet	<input checked="" type="checkbox"/> 10-06-2007 de 10:00 a 12:00 <input checked="" type="checkbox"/> 11-06-2007 de 10:00 a 12:00 <input type="checkbox"/> 12-06-2007 de 10:00 a 12:00 <input type="checkbox"/> 13-06-2007 de 10:00 a 12:00																		
<input type="button" value="CANVIAR ESTAT"/> <table border="1"> <tr> <td>Pendent</td> <td><input type="button" value="Canviar estat"/></td> </tr> </table>									Pendent	<input type="button" value="Canviar estat"/>														
Pendent	<input type="button" value="Canviar estat"/>																							
Tornar enrera																								

Figura 27 – Módulo de inscripciones

Inscribir alumnos

En este apartado se podrán realizar búsquedas por nombre o cualquier apellido del alumno o por la fecha de nacimiento introduciendo a mano la fecha o seleccionándola a través del calendario. Ésta búsqueda se utiliza para ver si el alumno ya ha estado dado de alta anteriormente y no se ha de volver a introducir los datos. Una vez hemos encontrado el alumno pulsando sobre el icono de la mano se inscribirá al curso.



Figura 28 – Búsqueda de alumnos

Si el alumno no ha sido dado de alta previamente, se podrá dar de alta presionando sobre el botón “Afegir nou usuari” y rellenando el formulario que contiene algunos datos personales y unas pocas preguntas de interés. Una vez dado de alta a través de esta ventana, también se añadirá automáticamente al curso.

Nome	<input type="text" value="Maria"/>
ter	<input type="text"/>
Cognom	<input type="text"/>
Zen	<input type="text"/>
Cognom	<input type="text"/>
Data	07-06-2007 <input type="button" value="..."/>
intervenció	
Sexe	<input checked="" type="radio"/> Home <input type="radio"/> Dona
Treball	<input type="radio"/> Treball (Autònom) <input type="radio"/> No
Estranger	<input checked="" type="radio"/> Si <input type="radio"/> No
Disp. PC	<input type="radio"/> Domicili amb Internet <input type="radio"/>
Estudis	<input type="radio"/> Elemental <input type="radio"/>
<input type="button" value="Dades d'alta"/>	

Figura 29 – Añadir alumno

Gestión de alumnos

Dentro de este módulo se podrá dar de alta alumnos, modificarlos y eliminarlos. Para dar de alta un nuevo alumno se tiene que presionar sobre el botón “Afegir nou alumne” y llenar el formulario como el mostrado en la figura 29 del apartado anterior.

Para modificar primero de todo hemos de encontrar el alumno el cual queremos modificar sus datos. Esta búsqueda se puede hacer mediante:

- Su sexo: hombre o mujer
- Su fecha de nacimiento
- Su edad
- Su nombre o apellidos

Una vez hemos encontrado los alumnos deseados, se pueden eliminar seleccionándolos a través del recuadro de la derecha, existe la posibilidad de seleccionarlos todos pulsando sobre el botón “seleccionar tots” que se convertirá en “des-seleccionar tots” si se han seleccionado todos.

GESTIÓ D'ALUMNES																																												
<input type="button" value="Afegir nou alumne"/> Filtres <table border="1" style="margin-bottom: 5px; width: 100%;"> <tr> <td style="width: 25%;">Per sexe</td> <td style="width: 25%;">Per edat</td> <td style="width: 25%;">Per data naixement</td> <td style="width: 25%;">Per nom</td> </tr> <tr> <td><input type="checkbox"/> Home</td> <td><input type="text"/></td> <td><input type="text"/> </td> <td><input type="text"/></td> </tr> <tr> <td><input type="checkbox"/> Dona</td> <td></td> <td></td> <td></td> </tr> </table> <input type="button" value="Aplicar filters"/> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>NOM</th> <th>EDAT</th> <th>SEXЕ</th> <th>TREBALL</th> <th>ESTR.</th> <th>DISP. PC</th> <th>ESTUDIS</th> <th>ELIM.</th> </tr> </thead> <tbody> <tr> <td>Luis Lopez Rubio</td> <td>0</td> <td>Homes</td> <td>Treballa (Autònoma)</td> <td>No</td> <td>Domicili amb Internet</td> <td>Elemental</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Marta Miranda Limones</td> <td>47</td> <td>Dona</td> <td>Treballa (Autònoma)</td> <td>No</td> <td>Domicili amb Internet</td> <td>Elemental</td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <input type="button" value="Seleccionar tot"/> <input type="button" value="Eliminar"/> Pàgina 1 2 Alumnes									Per sexe	Per edat	Per data naixement	Per nom	<input type="checkbox"/> Home	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Dona				NOM	EDAT	SEXЕ	TREBALL	ESTR.	DISP. PC	ESTUDIS	ELIM.	Luis Lopez Rubio	0	Homes	Treballa (Autònoma)	No	Domicili amb Internet	Elemental	<input type="checkbox"/>	Marta Miranda Limones	47	Dona	Treballa (Autònoma)	No	Domicili amb Internet	Elemental	<input type="checkbox"/>
Per sexe	Per edat	Per data naixement	Per nom																																									
<input type="checkbox"/> Home	<input type="text"/>	<input type="text"/>	<input type="text"/>																																									
<input type="checkbox"/> Dona																																												
NOM	EDAT	SEXЕ	TREBALL	ESTR.	DISP. PC	ESTUDIS	ELIM.																																					
Luis Lopez Rubio	0	Homes	Treballa (Autònoma)	No	Domicili amb Internet	Elemental	<input type="checkbox"/>																																					
Marta Miranda Limones	47	Dona	Treballa (Autònoma)	No	Domicili amb Internet	Elemental	<input type="checkbox"/>																																					

Figura 30 – Gestión de alumnos

Una vez hemos encontrado el alumno que queremos modificar, se debe pulsar sobre el recuadro que se encuentra a la izquierda de su nombre para que se nos abra una ventana con sus datos, los cuales los podremos modificar y volver a guardar el formulario:

MODIFICAR USUARI

Nom	Luis
1er Cognom	Lopez
2on Cognom	Rubio
Data naixement	06-06-2007
Sexe	<input checked="" type="radio"/> Home <input type="radio"/> Dona
Treball	Treballa (Autònom)
Estranger	<input checked="" type="radio"/> Sí <input type="radio"/> No
Disp. PC	Domicili amb Internet
Estudis	Elemental

Guardar canvis

Cursets que ha realitzat
Iniciació a la informàtica - Centre Cívic Alcalde Morera

Terminado

Figura 31 – Formulario para la modificación de alumnos

Además también se visualizarán los cursos que ha realizado este alumno y dónde los ha impartido.

Logs

En este módulo se puede visualizar el log o vaciarlo. El funcionamiento es muy sencillo ya que solo se debe escoger la opción deseada tal y como se ve en la siguiente figura:

GESTIÓ DELS LOGS

Visualitzar Log
En aquest log es podran visualitzar els cursets finalitzats pels dinamitzadors, així com els canvis que poden fer qualsevol administrador als cursets.

Buidar Log
Ens permet buidar el log del registre de canvis als cursets esmentat anteriorment.

Visualitzar log

Buidar log

Figura 32 – Módulo de logs

Si seleccionamos la opción de vaciar log aparecerá un mensaje de confirmación, como el de la figura 33, para poderlo vaciar.

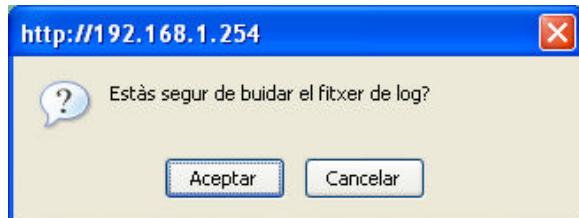


Figura 33 – Mensaje de confirmación

En cambio si queremos visualizarlo el resultado será el siguiente:

Curset nº	Actualitzat per	El dia	A les	Estat del cursset	Nº total d'alumnes inscrits	Nº d'alumnes que no han fet el 50% del cursset	Nº d'alumnes que han completat el cursset
4	aaron	04/17/07	11:04:44	Anul·lat	0	0	0
3	aaron	06/05/07	00:06:34	Anul·lat	2	1	1
1	aaron	05/06/07	00:06:25	Realitzat	1	1	0

Figura 34 –Ventana de logs

Memoria realizada por:
Aarón Luque Rubio

Resumen de la memoria

La memoria de este proyecto consiste, primero de todo en realizar un análisis del proyecto, de la situación actual y redactar un estudio de viabilidad que determinará si el proyecto es viable o no. Despues se describen las tecnologías que se han utilizado y se comparan con otras similares para demostrar el por qué se han elegido éstas.

Después se empieza a describir los aspectos técnicos de la aplicación, realizando un análisis de la base de datos, una explicación del diseño de la aplicación y finalmente se comentan los aspectos más técnicos de la programación.

Para acabar la memoria se realiza una valoración personal y se explican posibles mejoras y los objetivos alcanzados.

Resum de la memòria

La memòria d'aquest projecte consisteix, primerament en realitzar un anàlisi del projecte, de la situació actual i redactar un estudi de viabilitat que determinarà si el projecte es viable o no. Després es descriuen les tecnologies que s'han utilitzat y es comparen amb d'altres similars per demostrar el per què s'han escollit aquests.

Després es comença a descriure els aspectes tècnics de l'aplicació, realitzant un anàlisi de la base de dades, una explicació del disseny de l'aplicació i finalment es comenten els aspectes mes tècnics de la programació.

Per acabar la memòria es realitza una valoració personal i s'expliquen possibles millors i els objectius assolits.

Summary of the memory

The memory of this project consists, first of all to carry out an analysis of the project, the current situation and writing a viability study that determine if the project is viable or not. Next it is described the technologies that were used and they are compared with other similar technologies to demonstrate why they have been chosen. Then it is described the technical aspects of the application, it is carried out an analysis of the database, an explication of the application design and finally it is explained the more technical aspects of the code.

To finish the memory, it is done a personal valuation and it is explained other possible improves and the objectives reached.