



UNIVERSIDAD
POLITECNICA
DE VALENCIA



etsinf

Escuela Técnica
Superior de Ingeniería
Informática

PROYECTO FINAL DE CARRERA:

Diseño e implementación de un
sistema de evaluación docente

Autor: Víctor Mateo Torres
Director: Josep Francesc Silva Galiana

Valencia, Septiembre 2010

A mis padres y a mi hermano.

Este proyecto no ha sido trabajo de una sola persona, y por ello agradezco a todas las personas que han colaborado en la realización del mismo.

Primero de todo, quiero agradecer a mi director Josep Francesc Silva Galiana, por haber hecho posible este proyecto, así mismo, también quiero agradecerle su enorme esfuerzo, dedicación y tiempo que ha invertido. También mencionar su apoyo y confianza depositada en mí, y haberme brindado la posibilidad de participar en un proyecto interesante tanto en el plano profesional como en el educativo.

Por último, agradecer el apoyo ofrecido por cada uno de mis compañeros de la escuela al igual que nuestros familiares.

Muchas gracias a cada uno de ellos.

Índice

1	Introducción.....	10
1.1	Motivaciones.....	11
1.2	Planteamiento Técnico.....	11
2	Presentación del proyecto	12
3	Metodologías empleadas	14
3.1	Modelo en espiral	14
3.2	OOWS: Un método de producción de aplicaciones Web	15
3.2.1	Proceso de desarrollo	16
3.2.1.1	Modelo de navegación	16
3.2.1.2	Modelo de presentación	16
3.2.1.3	Mecanismos de acceso a la información.....	17
4	Especificación de requerimientos software	18
4.1	Introducción	18
4.1.1	Propósito	18
4.1.2	Ámbito.....	18
4.1.3	Definiciones, acrónimos y abreviaturas	21
4.1.4	Referencias	22
4.1.5	Visión global.....	22
4.2	Descripción general.....	23
4.2.1	Perspectiva del producto	23
4.2.2	Funciones del producto.....	23
4.2.3	Características del usuario	25
4.2.4	Restricciones generales.....	25
4.2.5	Supuestos y dependencias.....	26
4.3	Requerimientos específicos.....	26
4.3.1	Requerimientos funcionales rol alumno	26
4.3.1.1	Consultar información personal	26
4.3.1.2	Cambiar la contraseña de acceso	26
4.3.1.3	Insertar nota media del expediente.....	26
4.3.1.4	Consultar los resultados de sus exámenes.....	26
4.3.1.5	Confirmar asistencia insertando la posición en el aula.....	27
4.3.1.6	Realizar un examen.....	27
4.3.1.7	Enviar preguntas durante el examen.....	27
4.3.1.8	Finalizar el examen.....	27
4.3.1.9	Consultar docencia.....	27
4.3.1.10	Asignar docencia	28
4.3.1.11	Revocar docencia.....	28
4.3.1.12	Asignar compañero.....	28
4.3.1.13	Modificar compañero	28
4.3.1.14	Borrar compañero.....	28
4.3.2	Requerimientos funcionales rol profesor.....	29
4.3.2.1	Consultar información personal	29
4.3.2.2	Cambiar la contraseña de acceso	29
4.3.2.3	Alta de un tema.....	29
4.3.2.4	Baja de un tema.....	29
4.3.2.5	Consulta de temas.....	29
4.3.2.6	Modificación de un tema	29
4.3.2.7	Alta de una pregunta	29
4.3.2.8	Baja de una pregunta	30

4.3.2.9	Consulta de preguntas	30
4.3.2.10	Modificación de una pregunta	30
4.3.2.11	Consultar respuestas de una pregunta.....	30
4.3.2.12	Asignar solución a una pregunta	30
4.3.2.13	Alta de una respuesta.....	30
4.3.2.14	Baja de una respuesta.....	31
4.3.2.15	Consulta de respuestas.....	31
4.3.2.16	Modificación de una respuesta	31
4.3.2.17	Consultar a que pregunta pertenece una respuesta.....	31
4.3.2.18	Alta de asignatura.....	31
4.3.2.19	Baja de asignatura.....	31
4.3.2.20	Consulta de asignaturas.....	31
4.3.2.21	Modificación de una asignatura	32
4.3.2.22	Consultar asignaturas relacionadas.....	32
4.3.2.23	Asignar asignaturas relacionadas.....	32
4.3.2.24	Revocar asignaturas relacionadas.....	32
4.3.2.25	Alta de asignaturas relacionadas.....	32
4.3.2.26	Baja de asignaturas relacionadas.....	32
4.3.2.27	Consulta de asignaturas relacionadas.....	33
4.3.2.28	Modificación de una asignatura relacionada	33
4.3.2.29	Consultar a que asignatura está relacionada	33
4.3.2.30	Consultar alumnos que cursaron la asignatura relacionada.....	33
4.3.2.31	Alta de grupo.....	33
4.3.2.32	Baja de grupo.....	33
4.3.2.33	Consulta de grupos.....	34
4.3.2.34	Modificación de un grupo	34
4.3.2.35	Consultar el profesor del grupo.....	34
4.3.2.36	Consultar los alumnos del grupo.....	34
4.3.2.37	Alta de profesor	34
4.3.2.38	Baja de profesor	34
4.3.2.39	Consulta de profesores	34
4.3.2.40	Modificación de un profesor	35
4.3.2.41	Consultar grupos de un profesor.....	35
4.3.2.42	Asignar docencia a un profesor	35
4.3.2.43	Revocar docencia a un profesor	35
4.3.2.44	Alta de alumno	35
4.3.2.45	Baja de alumno	35
4.3.2.46	Consulta de alumnos.....	36
4.3.2.47	Modificación de un alumno.....	36
4.3.2.48	Consultar grupos de un alumno	36
4.3.2.49	Consultar exámenes de un alumno	36
4.3.2.50	Consultar asignaturas de un alumno	36
4.3.2.51	Consultar asignaturas relacionadas de un alumno	36
4.3.2.52	Asignar docencia a un alumno.....	36
4.3.2.53	Revocar docencia a un alumno.....	37
4.3.2.54	Asignar compañero.....	37
4.3.2.55	Modificar compañero	37
4.3.2.56	Borrar compañero.....	37
4.3.2.57	Asignar asignaturas relacionadas a un alumno	38
4.3.2.58	Revocar asignaturas relacionadas a un alumno	38

4.3.2.59	Crear lista de asistencia	38
4.3.2.60	Consultar lista de asistencia.....	38
4.3.2.61	Modificar lista de asistencia	38
4.3.2.62	Validar alumnos	38
4.3.2.63	Alta aula	39
4.3.2.64	Baja aula	39
4.3.2.65	Consulta de aulas	39
4.3.2.66	Modificación de un aula.....	39
4.3.2.67	Consulta de grupos que se imparten en un aula.....	39
4.3.2.68	Alta de examen	39
4.3.2.69	Baja de examen	39
4.3.2.70	Consulta de exámenes.....	40
4.3.2.71	Modificación de un examen.....	40
4.3.2.72	Consultar preguntas de un examen.....	40
4.3.2.73	Generar preguntas para un examen.....	40
4.3.2.74	Consultar alumnos que han realizado un examen.....	40
4.3.2.75	Iniciar un examen	40
4.3.2.76	Parar un examen	41
4.3.2.77	Predecir probabilidad de copias.....	41
4.3.2.78	Generar listados de notas	41
4.3.2.79	Generar estadísticas	41
4.3.3	Requisitos de interfaces externas	41
4.3.3.1	Interfaces de usuario	41
4.3.3.2	Interfaces hardware.....	41
4.3.3.3	Interfaces software	42
4.3.3.4	Interfaces de comunicaciones.....	42
4.3.4	Requerimientos de eficiencia	42
4.4	Restricciones de diseño	42
4.4.1	Estándares cumplidos.....	42
4.4.2	Atributos del sistema.....	42
4.4.2.1	Seguridad.....	42
4.4.2.2	Fiabilidad.....	43
4.4.2.3	Portabilidad	43
5	Análisis del problema.....	44
5.1	Estudio de la situación actual de portales Web orientados a la docencia.....	44
5.1.1	Soluciones comerciales	44
5.1.2	Conclusiones del estudio.....	45
5.2	Análisis del diseño Web	45
5.2.1	Análisis del diseño de página	46
5.2.1.1	Espacio de pantalla	46
5.2.1.2	Diseño multiplataforma.....	47
5.2.1.3	Diseño independiente de la resolución.....	47
5.2.1.4	Separar el significado de la presentación.....	48
5.2.1.5	Tiempos de respuesta	48
5.2.1.6	Vinculación.....	49
5.2.1.7	Plantillas de estilo	50
5.2.1.8	Marcos	50
5.2.1.9	Credibilidad	51
5.2.1.10	Conclusión	51

5.2.2	Análisis del diseño del contenido	51
5.2.2.1	Escribir para la Web	51
5.2.2.1.1	Mantener los textos cortos.....	52
5.2.2.1.2	Textos legibles.....	52
5.2.2.1.3	Explorabilidad	52
5.2.2.1.4	Lenguaje plano	52
5.2.2.2	Títulos de página	53
5.2.2.3	Legibilidad.....	53
5.2.2.4	Multimedia.....	53
5.2.2.5	Conclusión	53
5.2.3	Análisis del diseño del sitio.....	54
5.2.3.1	La página de inicio.....	54
5.2.3.1.1	Pantalla de bienvenida.....	54
5.2.3.2	Páginas interiores.....	54
5.2.3.3	Navegación	55
5.2.3.3.1	¿Dónde estoy?	55
5.2.3.3.2	¿Dónde he estado?.....	55
5.2.3.3.3	¿A dónde puedo ir?.....	55
5.2.3.3.4	La estructura del sitio	56
5.2.3.3.5	Extenso vs. Profundo	56
5.2.3.4	Capacidades de búsqueda.....	57
5.2.3.5	Diseño de URL.....	57
5.2.3.6	JavaScript	58
5.2.3.7	Conclusiones	58
6	Diseño del sistema.....	59
6.1	Diseño de la base de datos.....	59
6.1.1	Modelo Entidad-Relación.....	60
6.1.1.1	Diagrama Entidad Relación.....	61
6.1.1.2	La importancia de los procedimientos almacenados	65
6.1.1.3	Especificar restricciones de integridad	67
6.1.1.4	Diseño de transacciones	67
6.1.1.4.1	Inserción	68
6.1.1.4.2	Borrado	69
6.1.1.4.3	Modificación.....	70
6.1.1.5	Transformación del esquema conceptual.....	70
6.1.1.6	MySQL WorkBench	70
6.2	Diseño de la aplicación Web.....	73
6.2.1	Modelos OOWS	73
6.2.1.1	Modelos de navegación y presentación	74
6.2.1.1.1	Diagrama de usuarios	74
6.2.1.1.2	Mapas navegacionales	74
6.2.1.1.3	Contextos navegacionales.....	76
6.2.1.2	Mecanismos de acceso a la información.....	84
6.3	Diseño del sitio Web.....	84
6.3.1	Diseño de la página	85
6.3.1.1	Distribución en pantalla (Layout).....	85
6.3.1.2	Colores.....	86
6.3.1.3	Sistema de navegación	86
6.3.1.4	Menú lateral	87
6.3.1.4.1	La importancia de los iconos	87

6.3.1.5	Ruta de acceso o migas	88
6.3.1.6	Enlazar siempre al índice	89
6.3.2	Estructura del sitio.....	89
6.3.2.1	Página de inicio	89
6.3.2.2	Página de usuario Profesor y Alumno	90
6.3.3	Diseño del contenido	90
6.3.3.1	Gráficas con jQuery	90
6.3.3.2	Ejemplo de uso	90
6.3.3.2.1	Tooltip Labels	93
6.3.3.3	Tablas para listados de información	93
6.3.3.3.1	Mejoras proporcionadas.....	94
6.3.3.4	Ventanas modales	95
6.3.3.5	Exámenes	96
6.3.3.6	Predicción de copias	96
7	Implementación e Integración	99
7.1	Tecnologías	99
7.1.1	HTTP	99
7.1.2	CSS	99
7.1.3	PHP	100
7.1.4	MySQL.....	101
7.1.5	AJAX.....	101
7.1.6	Javascript	102
7.2	Herramientas	103
7.2.1	Adobe Dreamweaver CS4	103
7.2.2	Adobe Fireworks	104
7.2.2.1	MySQL WorkBench	104
7.2.3	Microsoft Visio	106
7.3	PhpMyAdmin	107
7.4	Librería PHP para crear PDFs	108
7.5	Highcharts.....	108
7.6	Calendario en JavaScript	109
7.7	Framework jQuery	109
8	Conclusiones.....	111
9	Ampliaciones y mejoras futuras	113
9.1	Mejoras a nivel de persistencia.....	113
9.2	Mejoras a nivel de funcionalidad.....	113
Bibliografía.....		115
10	Anexo.....	118
10.1	Instalación de AWAD	118

Índice de figuras

Figura 1. Modelo en espiral.	14
Figura 2. Gráfica asistencia nota.	19
Figura 3. Grafica nota media.	20
Figura 4. Asistencia de no presentados.	20
Figura 6. Tabla de porcentajes de resoluciones utilizadas.	47
Figura 7. Ejemplo de Entidad-Relación.	60
Figura 8. Diagrama Entidad-Relación.	61
Figura 9. Diagrama ER MySQL Workbench.	62
Figura 10. MySQL Workbench	72
Figura 11. Diagrama de usuarios.	74
Figura 12. Mapa navegacional para student.	75
Figura 13. Mapa navegacional para teacher.	75
Figura 14. Layout.	86
Figura 15. Menú de Navegación y ejemplo de iconos.	88
Figura 16. Estás en:	88
Figura 17. Interfaz página de inicio.	89
Figura 18. Gráfica estadísticas examen.	93
Figura 19. Gráfica: Tooltip Labels.	93
Figura 20. Tabla de listados de información.	94
Figura 21. Tabla: Resultados de búsqueda.	94
Figura 22. Tabla: Paginación variable.	95
Figura 23. Ventana modal.	95
Figura 24. Exámenes iniciados.	96
Figura 25. Exámenes.	96
Figura 26. Interfaz de predecir copias.	97
Figura 27. Ver asistencia	98
Figura 28. Esquema de funcionamiento de PHP.	100
Figura 29. Dreamweaver CS4.	104
Figura 30. MySQL Workbench.	106
Figura 31. Microsoft Visio 2003	107
Figura 32. PhpMyAdmin.	108
Figura 33. Ejemplo de uso de JSCalendar.	109
Figura 34. Pantalla de login de la aplicación Web.	119

CAPÍTULO 1

1 Introducción

Este documento trata de explicar las soluciones adoptadas a la hora de crear un portal Web para asistir en la evaluación de los alumnos en asignaturas universitarias. La herramienta será una aplicación Web que pretende generar contenido dinámico personalizado a cada usuario (tanto alumnos como profesores). La información del sistema, que incluirá preguntas, exámenes, aulas, profesores, alumnos, etc, será almacenada en una base de datos ya diseñada.

En los últimos años, se ha notado un incremento considerable del uso de nuevas tecnologías en la educación. Este hecho ha sido promovido principalmente por la introducción de Internet en las aulas. De este modo, se han popularizado el uso de portales orientados a la docencia, como “PoliformaT” en el caso de la Universidad Politécnica de Valencia.

Mediante estas tecnologías, es posible dar un mayor apoyo al personal docente con respecto a los medios tradicionales. A través de los portales orientados a la docencia el personal docente puede entre otras cosas: difundir material a los alumnos, crear foros para debatir cuestiones, llevar a cabo cursos online, realizar actividades de evaluación, etc.

Aunque actualmente existen proyectos funcionando como “Poliformat” en la Universidad Politécnica de Valencia, este tipo de portales no ofrecen la posibilidad de capturar y procesar información derivada del proceso de aprendizaje y evaluación comentada anteriormente, es por eso que surge la necesidad de desarrollar un portal más especializado y dedicado para estas tareas.

Teniendo en cuenta lo comentado anteriormente el objetivo de este proyecto es el de mejorar y adaptar la funcionalidad existente, desarrollar nuevos módulos que sumen funcionalidad y convertir la aplicación en una herramienta totalmente útil, funcional y usable tanto para el personal docente como para los alumnos, para ello se cambiara totalmente la interfaz grafica de modo que sea más amigable y intuitiva, en los siguientes apartados se detallan con claridad los requisitos y funcionalidades.

1.1 Motivaciones

Para explicar las motivaciones que me llevaron a la elección y realización de este proyecto no hay más que mirar la situación de expansión y crecimiento en la que se encuentran los portales Webs docentes.

Como usuarios activos en este tipo de páginas y con compañeros y amigos que hacen uso de ellas a diario, me sentí atraído por la posibilidad de aportar novedades y mejoras que pudieran satisfacer las necesidades requeridas.

Además de todo esto también quería poner en práctica el uso de las herramientas y lenguajes de programación que se utilizan en la creación de páginas Web dinámicas como PHP y MySQL (o AJAX) para acercarme a las demandas del mundo laboral.

1.2 Planteamiento Técnico

En lo que se refiere a los aspectos técnicos de la Web hemos intentado centrarnos en varias características para intentar lograr que la Web se acomodara de la mejor forma tanto a nuestras necesidades como a las necesidades de los usuarios. Por ello, entre muchas otras características, encontramos:

- Uso de tecnologías libres.
- Un diseño fácil y entendible para el usuario.
- Una interfaz Web accesible.

Estas características fueron establecidas desde el principio del proyecto, por ello vamos a desglosarlas a continuación.

El uso de tecnologías libres permite la realización de proyectos con un coste mínimo, casi nulo. En nuestro caso hemos hecho uso de la tecnología de desarrollo XAMPP que se trata de la unión de Windows, Apache, MySQL y PHP. Otra característica de esta tecnología es el hecho que para hacer correr XAMPP no se necesita de una maquina potente.

Por otro lado, el diseño de la Web lo queríamos realizar de forma que para el usuario fuera lo mas intuitivo posible a la hora de realizar cada una de las funcionalidades de la que consta el proyecto. Para ello, detuvimos el proceso de desarrollo del proyecto para visitar un gran número de portales de docencia y webs donde se permitía la opción de realizar exámenes para basarnos y observar cuales eran las ventajas y desventajas de cada una de sus interfaces, para así conseguir un patrón común.

CAPÍTULO 2

2 Presentación del proyecto

El presente documento expone todo el proceso de desarrollo—incluyendo las técnicas, metodologías y herramientas utilizadas—del proyecto final de carrera. El objetivo de esta memoria es servir de ayuda para la comprensión y evaluación del proyecto, así como ser un documento de referencia para futuros proyectos, de modo que la experiencia adquirida en la realización de este proyecto pueda ser aprovechada por otros profesionales.

El sistema posee varias partes bien diferenciadas.

- Una parte centrada en la evaluación, donde es posible crear y corregir automáticamente los exámenes realizados por los alumnos, dando la posibilidad de predecir posibles copias.
- Otra parte enfocada a la asistencia del alumno, donde es posible llevar un control sobre el alumno durante el proceso de aprendizaje y evaluación.
- Otra parte importante es la creación de tablas y graficas estadísticas con la información almacenada en el sistema, tanto de asistencia como evaluaciones por parte de los alumnos.

Todas estas partes y sus respectivas funcionalidades están explicadas con más detalle en la especificación de requisitos software IEEE830 del presente documento.

Para la realización del proyecto, se ha utilizado el **ciclo de vida software en espiral**. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgo. Nos hemos decidido por este modelo ya que muchos de los requisitos y necesidades de la aplicación fueron surgiendo a medida que avanzaba el proyecto. Para cada ciclo habrá cuatro actividades:

Determinar o fijar objetivos, análisis del riesgo, desarrollar, verificar y validar (probar) y planificar.

Finalmente, se definirán las correspondientes pruebas para garantizar el correcto funcionamiento del sistema. Tras la finalización del proyecto, se planea la utilización del sistema en una prueba real.

CAPÍTULO 3

3 Metodologías empleadas

En el presente apartado se presentan las metodologías empleadas para la realización del proyecto. Las metodologías se comentan y explican brevemente aunque con más detalle que cuando fueron expuestas en la sección 2 *Presentación del proyecto* del presente documento. El objetivo de este apartado es clarificar conceptos para el lector de este documento. En ningún momento se va a entrar a explicar en profundidad estas metodologías, ya que no es el propósito de esta memoria.

3.1 Modelo en espiral

Propuesto inicialmente por Boehm en 1988. Consiste en una serie de ciclos que se repiten. Cada uno tiene las mismas fases y cuando termina da un producto ampliado con respecto al ciclo anterior. En este sentido es parecido al modelo incremental, la diferencia importante es que tiene en cuenta el concepto de riesgo. Un riesgo puede ser muchas cosas: requisitos no comprendidos, mal diseño, errores en la implementación, etc. Una representación típica de esta estructura se muestra en la figura.



Figura 1. Modelo en espiral.

En cada iteración Boehm recomienda recopilar la siguiente lista de informaciones:

Determinar o fijar objetivos

- Fijar también los productos definidos a obtener: requerimientos, especificación, manual de usuario.
- Fijar las restricciones.
- Identificación de riesgos del proyecto y estrategias alternativas para evitarlos.
- Hay una cosa que solo se hace una vez: planificación inicial o previa.

Análisis del riesgo

Desarrollar, verificar y validar (probar)

- Tareas de la actividad propia y de prueba.
- Análisis de alternativas e identificación resolución de riesgos.
- Dependiendo del resultado de la evaluación de los riesgos, se elige un modelo para el desarrollo, el que puede ser cualquiera de los otros existentes, como formal, evolutivo, cascada, etc. Así si por ejemplo los riesgos en la interfaz de usuario son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de protección son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado.

Planificar

- Revisamos todo lo hecho, evaluándolo, y con ello decidimos si continuamos con las fases siguientes y planificamos la próxima actividad.[WIKI- Modelo]

3.2 OOWS: Un método de producción de aplicaciones Web

OOWS es un método de producción de aplicaciones Web, una definición de este método puede ser:

“Aproximación para el modelado orientado a objetos de soluciones Web que proporciona mecanismos para la especificación de sistemas de información dinámicos hipermediales y de comercio electrónico. Propone una extensión de un método OO para la generación automática de código usando modelos conceptuales (OO-Method) que permite capturar y representar la semántica navegacional y de presentación de información usando primitivas de abstracción de alto nivel. [OOWS2, 2002]

De la definición se pueden resaltar las siguientes características:

- Un método de desarrollo Web.
- Introduce nuevos conceptos OO para dar la noción de semántica navegacional y de presentación (primitivas de modelado).
- Define un proceso de producción sistemático adaptado al Web.

- Usa una estrategia de generación de código basada en modelos (patrones software).
- Extiende adecuadamente OO-Method (un método OO de desarrollo de software).

3.2.1 Proceso de desarrollo

El proceso de desarrollo de aplicaciones Web que propone OOWS tiene dos grandes pasos: [OOWS, 2004]

- Especificación conceptual de la aplicación.
 - Uso de modelos conceptuales.
- Desarrollo de la solución final.

En este apartado nos centraremos en la especificación conceptual de la aplicación concretamente en el uso de modelos conceptuales ya que para la realización del presente proyecto se han generado modelos OOWS en la sección 6.2 “Diseño de la aplicación Web”.

En la especificación conceptual encontramos dos partes bien diferenciadas:

Clásica. Descripción de las necesidades de las aplicaciones convencionales (estructura y comportamiento) usando modelos OO (OOMethod).

Web. Descripción de las características Web mediante modelos Web. Modelo Navegacional, de Presentación, mecanismos de acceso a la información, etc.

Se puede decir que la parte Web extiende a la clásica, es decir, los modelos Web son construidos a partir de la información de los modelos disponibles en la parte clásica. En los siguientes apartados se explican brevemente los modelos Web.

3.2.1.1 Modelo de navegación

El modelo de navegación especifica las características navegacionales de las aplicaciones Web para el acceso a la información como para la ejecución de servicios mediante **mapas navegacionales** para cada tipo de usuario. Se construye mediante primitivas gráficas de abstracción navegacional. Además define y estructura el acceso al sistema para los diferentes tipos de usuarios (gestión de usuarios) mediante el **diagrama de usuarios**. [OOWS2, 2002]

3.2.1.2 Modelo de presentación

El Modelo de presentación de OOWS es el encargado de definir ciertos aspectos básicos de apariencia de las aplicaciones. Su objetivo consiste en definir aspectos de presentación de información de las páginas, para ello utiliza un conjunto de patrones de presentación que asocia a elementos existentes del

modelo navegacional. Se puede ver este modelo de presentación como una extensión del modelo de navegación al que se enriquece con un conjunto de patrones básicos de presentación. [OOWS2, 2002]

3.2.1.3 Mecanismos de acceso a la información

Cuando un usuario se conecte al sistema, podrá explorarlo usando su mapa de navegación y cuando acceda a un contexto navegacional, se recuperará *toda* la información que el contexto define. En algunos casos esta información puede ser demasiada, millones de datos, por esto, se propone siguiendo una estrategia de modelado conceptual:

“extender las capacidades expresivas de los contextos navegacionales para introducir mecanismos que permitan restringir la información a recuperar con el objetivo de ayudar y facilitar la búsqueda y selección de información”. [OOWS, 2004]

Existen dos tipos de mecanismos de acceso a la información:

- Índices: definen una estructura de acceso en base a unas propiedades predefinidas.
- Filtros: permiten establecer, en base a unas propiedades, una restricción de filtrado ó selección de información.

CAPÍTULO 4

4 Especificación de requerimientos software

4.1 *Introducción*

4.1.1 Propósito

En esta sección identificaremos y definiremos tanto los actuales como son los nuevos funcionamientos que debe tener la herramienta para dar soporte a la docencia y satisfacer las necesidades del usuario final.

Esta especificación de requerimientos está destinada a ser leída por los usuarios de la herramienta, así como, cualquier persona que desee conocer el funcionamiento del producto.

4.1.2 *Ámbito*

El portal Web Aplicación Web de Apoyo a la Docencia (a partir de este momento abreviado como AWAD) consistirá en una aplicación que permitirá al alumno y personal docente interactuar y realizar diversas operaciones con el sistema de forma muy diferente.

El producto dará soporte al personal docente para poder generar actividades de evaluación, además de captar y procesar información útil para el proceso de evaluación del alumno. Esta aplicación Web será utilizada para la evaluación de los alumnos de varias asignaturas de la Ingeniería en Informática.

El sistema a desarrollar posee dos partes bien diferenciadas, una parte centrada en la evaluación y otra enfocada a la asistencia del alumno.

En la parte encargada de la evaluación, con el fin de evaluar a los alumnos, el personal docente indicará al sistema un tema y un número de preguntas, la aplicación se encargará de generar automáticamente una prueba a partir de la información almacenada en la base de datos de preguntas. A continuación, se le presentará al alumno la prueba mediante una interfaz Web.

Una de las ventajas de la aplicación es la corrección automática de los exámenes. El alumno podrá ver la nota obtenida cuando finalice el examen.

Por otra parte, en lo referente a la asistencia del alumno, se realizará un control de la asistencia del alumno así como su posición en el aula. A partir de dicha información, se realizará un estudio de la relación entre la nota obtenida por el alumno con su asistencia y posición en el aula. Para ello el sistema generará una serie de graficas y tablas que comentamos a continuación:

Esta tabla muestra diversa información útil para el personal docente, muestra la posición donde se sentaron los alumnos del grupo cada uno de los días que hubo clase y la nota que obtuvieron en los exámenes de la asignatura.

Nombre	Días																									Media	Exámenes		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	F,C	13	
Victor Mateo Torres													1,3	2,3	2,1	2,2				1,3							1,6,2,4	2,86	
Omar Pastor Nomeacuero															2,4	2,3	2,1	2,3										2,2,8	0,00
Lucia Revert Loquesea	1,5	1,1	1,2	1,2	2,2	2,4	1,2	2,1	2,6	1,3	1,1	1,1	1,3	2,2	2,4	1,4	3,2	1,2		1,1				2,2	3,1	1,1	1,5,2,4		
Fran Alonso Jaen															1,1			1,2										1,1,5	
Diego Insa Lopez											1,4	1,4	1,2	2,2	1,2	1,1	1,1	1,1	1,2	2,1	1,1	1,2	1,2				1,2,2	5,71	
David Pastor Bayo													1,1	4,4	3,3	3,2											2,8,2,5	1,43	
Carlos Pastor Bayo												3,1	1,2	2,1	4,2	1,5	3,2										2,3,2,2	2,86	

Figura 3 Tabla asistencia y nota.

Para presentar la información de forma más clara la siguiente gráfica mostrará el número de veces que el alumno asistió a clase y la nota para los distintos exámenes de la asignatura, de esta forma se facilitará al personal docente información para deducir si la asistencia influye en la nota de alumno.

La asistencia está representada por las barras verticales moradas y la nota de examen por los puntos rojos.



Figura 2. Gráfica asistencia nota.

Otra posibilidad que ofrece la aplicación es determinar si la nota va relacionada con la posición en la que el alumno se sienta en clase, para ello el sistema genera una tabla en la que se muestra la nota media de la silla, es decir se calcula la nota de todos los alumnos que se sentaron en la silla, de esta forma se ve de forma clara si los alumnos que se sientan en sillas cercanas a la primera fila obtuvieron mejor nota.



Figura 3. Grafica nota media.

Un estudio interesante que debemos realizar es saber donde se sientan los alumnos que deciden no presentarse al examen. Para ello se mostrará el aula y el número de veces que se sentaron en cada silla los alumnos que no realizaron el examen.

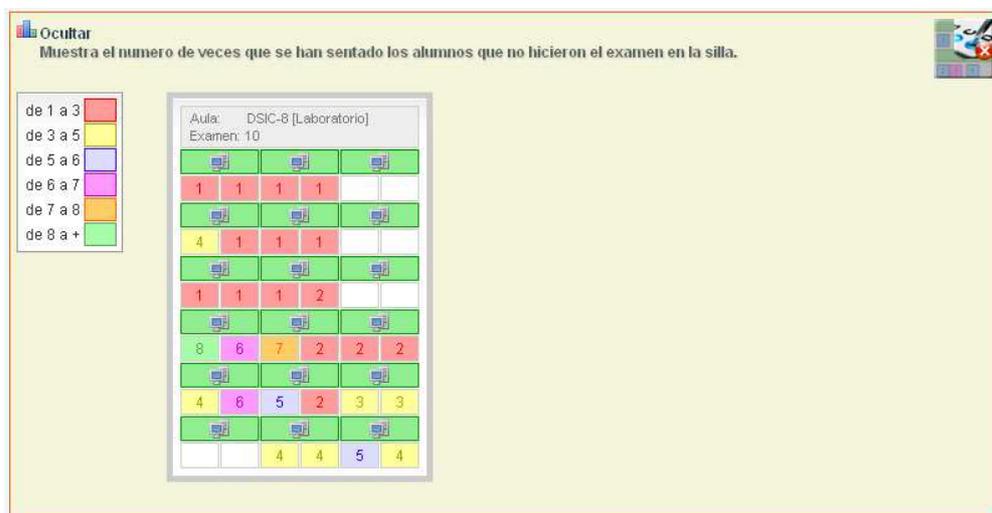


Figura 4. Asistencia de no presentados.

Por ultimo se estudiará el posicionamiento de los alumnos distinguiendo por sexo, el sistema mostrará el número de veces que se sentaron en la silla.

4.1.3 Definiciones, acrónimos y abreviaturas

En este apartado se presentan las definiciones, acrónimos y abreviaturas de términos usados en este documento.

- **AWAD:** Acrónimo usado para hacer referencia al nombre de nuestra aplicación, Aplicación Web de Apoyo a la Docencia.
- **PHP:** (PHP Hypertext Pre-processor). Lenguaje de programación usado generalmente en la creación de contenidos para sitios Web. Es un lenguaje interpretado especialmente usado para crear contenido dinámico Web y aplicaciones para servidores. Permite la conexión a todo tipo de servidores de base de datos como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.
- **MySQL:** MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.
- **HTML:** *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. [WIKI-HTML, 2009].
- **Apache:** Es un servidor HTTP (Web) de código abierto, está disponible para la mayoría de los sistemas operativos.
- **HTTP:** *HyperText Transfer Protocol*, es el protocolo de transferencia de hipertexto, es usado en cada transacción de la Web. [WIKI-HTTP, 2009].
- **IDE:** *Integrated Development Environment*, es un entorno de desarrollo de software compuesto por un conjunto de herramientas para programadores. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. [WIKI-IDE, 2009].
- **Framework:** Es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, incluye soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. [WIKI-FRA, 2009].

- **Framework: Ajax**, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dados que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

4.1.4 Referencias

Para la redacción del presente documento se han tenido en cuenta los siguientes documentos y referencias:

[IEEE830]. IEEE std 830 Recommended Practice for Software Requirements Specifications.

[WIKI-HTML, 2009]. Wikipedia la enciclopedia libre, artículo sobre el lenguaje de marcas HTML, <http://es.wikipedia.org/wiki/Html>.

[WIKI-HTTP, 2009]. Wikipedia la enciclopedia libre, artículo sobre el protocolo HTTP, <http://es.wikipedia.org/wiki/Http>.

[WIKI-IDE, 2009]. Wikipedia la enciclopedia libre, artículo sobre entornos de desarrollo integrados IDE, <http://es.wikipedia.org/wiki/>.

[WIKI-FRA, 2009]. Wikipedia la enciclopedia libre, artículo sobre Frameworks, <http://es.wikipedia.org/wiki/Framework>.

4.1.5 Visión global

Este documento consta de tres secciones. Esta es la primera sección y es la introducción al documento de ERS y proporciona una visión general. La segunda sección trata de hacer una descripción general de la aplicación, describiendo los factores que afectan a la aplicación y el contexto en el que se encuentra, todo ello sin entrar en excesivos detalles, lo que hará más comprensible la descripción de los requisitos específicos. En la última sección se definen detalladamente los requisitos de la aplicación.

4.2 Descripción general

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

4.2.1 Perspectiva del producto

Nuestra aplicación se desarrollara en el lenguaje de programación PHP para generar contenido dinámico personalizado a cada usuario. Esta personalización se basa en datos del usuario almacenados en una base de datos, MySQL en nuestro caso. Usaremos Apache como servidor Web y lo ejecutaremos en Windows, haciendo uso de la pila XAMPP (Windows, Apache, MySQL y PHP) disponible en multitud de distribuciones.

La elección del lenguaje se justifica en que es un lenguaje libre y muy popular para el desarrollo de este tipo de aplicaciones. Debido a ello, existe gran cantidad de información (libros, manuales, comunidades, etc.), Frameworks, librerías e IDEs dedicados a PHP.

Usaremos Apache como servidor Web y al tratarse de una Aplicación Web será totalmente independiente del sistema operativo.

Para acceder a la aplicación Web será necesario un navegador Web (Mozilla Firefox, Opera, Safari, Internet Explorer, etc) por parte del usuario. En cuanto a la resolución, es recomendable el uso 1024x768, ya que es una de las resoluciones más usadas por los usuarios, no obstante se puede visualizar sin problemas en otras resoluciones como 800x600 o 1280x1024.

El sistema ofrecerá distintas interfaces dependiendo del rol del usuario que inicie sesión en la aplicación. Se establece como requisito que la aplicación funcione en las últimas versiones de los navegadores Internet Explorer, Firefox y Safari.

Las comunicación entre los cliente (navegador Web) y el servidor de la aplicación Web se realizará a través de Internet mediante TCP/IP y haciendo uso del protocolo de aplicación HTTP.

Para la base de datos se debe utilizar MySQL, ya que popularmente está muy ligada a PHP, existiendo también gran cantidad de documentación para su uso ya sea aparte o conjuntamente con PHP. MySQL es un sistema de gestión de base de datos relacional, multiusuario y multihilo, está disponible para la mayoría de sistemas operativos y al igual que PHP es libre y su uso es muy popular.

4.2.2 Funciones del producto

El sistema podrá ser usado por alumnos y profesores. Las funciones que va a realizar el producto dependerán del rol del usuario que haya iniciado sesión. A continuación se listan las funciones que podrá realizar cada uno de ellos.

Alumno:

- Consultar información personal.
- Cambiar la contraseña de acceso.
- Insertar nota media del expediente.
- Alta / Baja / Consulta / Modificación de docencia.
- Alta / Baja / Consulta / Modificación de compañero de practicas.
- Consultar los resultados de sus exámenes.
- Confirmar asistencia insertando la posición en el aula.
- Realizar un examen.
- Enviar preguntas durante el examen.
- Finalizar el examen.

Profesor:

- Consultar información personal.
- Cambiar la contraseña de acceso.
- Alta / Baja / Consulta / Modificación de temas.
- Alta / Baja / Consulta / Modificación de preguntas.
- Consultar respuestas de una pregunta.
- Asignar solución a una pregunta.
- Alta / Baja / Consulta / Modificación de respuestas.
- Consultar a que pregunta pertenece una respuesta.
- Alta / Baja / Consulta / Modificación de asignaturas.
- Consultar asignaturas relacionadas.
- Asignar asignaturas relacionadas.
- Revocar asignaturas relacionadas.
- Alta / Baja / Consulta / Modificación de asignaturas-relacionadas.
- Consultar a que asignatura está relacionada.
- Consultar alumnos que cursaron la asignatura relacionada.
- Alta / Baja / Consulta / Modificación de grupos.
- Consultar el profesor del grupo.
- Consultar los alumnos del grupo.
- Alta / Baja / Consulta / Modificación de profesores.
- Consultar grupos de un profesor.
- Asignar docencia a un profesor.
- Revocar docencia a un profesor.
- Alta / Baja / Consulta / Modificación de alumnos.
- Consultar grupos de un alumno.
- Consultar exámenes de un alumno.
- Consultar asignaturas de un alumno.
- Consultar asignaturas relacionadas de un alumno.
- Asignar docencia a un alumno.
- Revocar docencia a un alumno.
- Alta / Baja / Consulta / Modificación de compañero de practicas.
- Asignar asignaturas relacionadas a un alumno.
- Revocar asignaturas relacionadas a un alumno.
- Alta / Baja / Consulta de lista de asistencia.
- Alta / Baja / Consulta / Modificación de aulas.
- Consulta de grupos que se imparten en un aula.

- Alta / Baja / Consulta / Modificación de exámenes.
- Consultar preguntas de un examen.
- Generar preguntas para un examen.
- Consultar alumnos que han realizado un examen.
- Iniciar examen.
- Parar examen.
- Predecir probabilidad de copias.
- Generar listados de notas.
- Generar estadísticas.
- Estadísticas por grupo.

4.2.3 Características del usuario

Este producto será usado por personal docente y alumnos. En principio, la herramienta está pensada para ser usada en un ámbito universitario, pero también podría ser usada en otros ámbitos como el de la enseñanza secundaria. Esto deberá tenerse en cuenta a la hora de diseñar la herramienta para no limitar su posible uso en ámbitos no universitarios.

Para poder utilizar la aplicación no será necesario ningún tipo de nivel de estudios o experiencia profesional. Los usuarios simplemente tendrán que tener los mínimos conocimiento de informática a nivel de usuario, saber usar el ratón, teclado y utilizar un navegador Web. Además, se tratará de implementar la interacción del producto para que las interfaces de usuario sean claras e intuitivas.

4.2.4 Restricciones generales

Teniendo en cuenta que el producto es una aplicación Web, en lo referente a la parte cliente, ya se ha comentado anteriormente que la aplicación podrá ser utilizada desde cualquier sistema operativo simplemente utilizando un navegador Web y una conexión a Internet.

En lo referente a la parte de servidor, la aplicación Web tendrá que estar alojada en un servidor Web que admita funcionalidad con PHP y MySQL.

Al ser nuestra aplicación dinámica debemos usar un lenguaje de programación dinámico, entre las opciones disponibles nos decantamos por usar el lenguaje PHP por las ventajas que supone a la hora de trabajar con una base de datos (MySQL).

Otro aspecto importante es la eficiencia, MySQL suele ser utilizado en aplicaciones Web. Su popularidad está ligada a su combinación con el lenguaje PHP. MySQL es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. Inicialmente, nuestro sistema, así como en la mayoría de aplicaciones Web, posee baja concurrencia en la modificación de datos, mientras que, es intensivo en lectura de datos. Por tanto, MySQL es el sistema ideal para este tipo de aplicación.

Respecto a la seguridad de la aplicación debemos tener en cuenta que al ser una aplicación Web hay multitud de amenazas, por lo que debemos cuidar la programación para mantener la integridad del sistema al igual que la confidencialidad de los usuarios.

4.2.5 Supuestos y dependencias

Este producto va a ser desarrollado como proyecto final de carrera para la Facultad de informática de la Universidad Politécnica de Valencia. Por tal razón, es posible que los requerimientos de esta especificación de requisitos se modifiquen por exigencias del director del proyecto o por factores de la propia universidad.

4.3 Requerimientos específicos

4.3.1 Requerimientos funcionales rol alumno

4.3.1.1 Consultar información personal

El sistema mostrará al alumno su información personal.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.

4.3.1.2 Cambiar la contraseña de acceso

El sistema deberá permitir al alumno cambiar la contraseña de acceso a la aplicación Web mediante una ventana modal.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.

4.3.1.3 Insertar nota media del expediente

El alumno debe poder insertar la nota media de su expediente.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.

4.3.1.4 Consultar los resultados de sus exámenes

El sistema deberá permitir a los alumnos consultar los resultados de sus exámenes, se mostrará la nota del examen y un resumen de las preguntas acertadas, fallos cometidos y preguntas no enviadas.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- Previamente el alumno debe haber realizado algún examen.

4.3.1.5 Confirmar asistencia insertando la posición en el aula

El alumno debe poder confirmar su asistencia en el aula, ya sea para la realización de una sesión de prácticas o para un examen. El alumno insertará su posición en el aula que será capturada por el sistema. Si la asistencia ya está confirmada para el grupo seleccionado, el sistema nos avisará.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- El alumno debe estar dado de alta en la asignatura y en el grupo.
- La posición introducida debe ser válida para el aula.
- La posición introducida no puede coincidir con la de otro alumno.

4.3.1.6 Realizar un examen

El sistema deberá permitir realizar un examen a los alumnos.

Como precondiciones, tenemos:

- El alumno debe estar dado de alta en la asignatura y en el grupo.
- El alumno deberá confirmar la asistencia para el grupo del examen.
- El alumno debe de haber iniciado sesión.
- Previamente el profesor debe de haber iniciado un examen.

4.3.1.7 Enviar preguntas durante el examen

El sistema permitirá que el alumno envíe al sistema las preguntas con las respuestas elegidas durante la realización de un examen.

Como precondiciones, tenemos:

- El alumno debe estar dado de alta en la asignatura y en el grupo.
- El alumno debe de haber iniciado sesión.
- Previamente el profesor debe de haber iniciado un examen.
- El alumno debe de haber marcado una respuesta.

4.3.1.8 Finalizar el examen

El sistema permitirá que el alumno finalice un examen que este realizando. El sistema deberá corregir automáticamente el examen.

Como precondiciones, tenemos:

- El alumno debe estar dado de alta en la asignatura y en el grupo.
- El alumno debe de haber iniciado sesión.
- Previamente el profesor debe de haber iniciado un examen.
- El alumno debe de haber enviado todas las preguntas del examen previamente.

4.3.1.9 Consultar docencia

El sistema permitirá que el alumno pueda consultar su docencia, podrá visualizar los grupos en los que esté matriculado, ver su compañero y si es repetidor o no.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.

- El alumno debe de estar matriculado en al menos un grupo.

4.3.1.10 Asignar docencia

El sistema permitirá al usuario darse de alta en un grupo e indicar si es repetidor, para ello se mostrarán los grupos en los que no está registrado y se indicará al sistema en que grupos se desea asignar docencia.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- El grupo tiene que estar dado de alta en el sistema.
- La asignatura debe de estar dada de alta en el sistema.

4.3.1.11 Revocar docencia

El sistema permitirá al usuario darse de baja en un grupo y asignatura, para ello se mostrarán los grupos en los que está registrado el alumno y se indicará al sistema los grupos en los que se desea darse de baja.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- El alumno debe de estar matriculado en el grupo.

4.3.1.12 Asignar compañero

El sistema permite asignar al alumno su compañero para un grupo de prácticas. La asignación se realizara en ambos sentidos.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- El grupo debe ser de prácticas.
- El alumno que queremos asignar como nuestro compañero debe de ser del grupo.
- El alumno que queremos asignar como nuestro compañero debe de estar libre.

4.3.1.13 Modificar compañero

El sistema permitirá al usuario cambiar el compañero de prácticas.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- Debe ser un grupo de prácticas.
- Debe tener compañero asignado.

4.3.1.14 Borrar compañero

El sistema permitirá al usuario borrar el compañero de prácticas.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- Debe ser un grupo de prácticas.
- Debe tener compañero asignado.

4.3.2 Requerimientos funcionales rol profesor

4.3.2.1 Consultar información personal

El sistema mostrará al profesor su información personal.

Como precondiciones, tenemos:

- El profesor debe de haber iniciado sesión.

4.3.2.2 Cambiar la contraseña de acceso

El sistema deberá permitir al profesor cambiar la contraseña de acceso a la aplicación Web mediante una ventana modal.

Como precondiciones, tenemos:

- El profesor debe de haber iniciado sesión.

4.3.2.3 Alta de un tema

El sistema permitirá dar de alta un nuevo tema, deberá de indicarse el nombre del tema.

Como precondiciones, tenemos:

- El tema no puede coincidir con algún tema ya existente en el sistema para una determinada asignatura.

4.3.2.4 Baja de un tema

El sistema debe permitir dar de baja un tema.

Como precondiciones, tenemos:

- El tema indicado debe de existir.

4.3.2.5 Consulta de temas

El sistema debe permitir consultar los temas registrados en el sistema, para ello el profesor podrá hacer una búsqueda por nombre.

Como precondiciones, tenemos:

- Deben de existir temas registrados en el sistema.

4.3.2.6 Modificación de un tema

El sistema debe permitir modificar la información de un tema.

Como precondiciones, tenemos:

- El tema indicado debe de existir.

4.3.2.7 Alta de una pregunta

El sistema debe permitir dar de alta una nueva pregunta para ello se indicará la pregunta y a que tema pertenece. Para facilitar al profesor esta tarea es posible asignar tantas respuestas como se quiera para la pregunta que estamos creando.

Como precondiciones, tenemos:

- La pregunta no puede coincidir con otra ya existente en el sistema.
- El tema relacionado debe de existir en el sistema.

4.3.2.8 Baja de una pregunta

El sistema debe permitir dar de baja una pregunta, para ello se indicará el código de pregunta a eliminar.

El sistema debe permitir dar de baja una pregunta, para ello se desplegará el menú de opciones y se escogerá Eliminar.

Como precondiciones, tenemos:

- La pregunta indicada debe de existir.

4.3.2.9 Consulta de preguntas

El sistema debe permitir consultar las preguntas registradas en el sistema, para ello el profesor podrá realizar búsquedas por texto o por temas.

Como precondiciones, tenemos:

- Deben de existir preguntas registradas en el sistema.

4.3.2.10 Modificación de una pregunta

El sistema debe permitir modificar la información de una pregunta.

Como precondiciones, tenemos:

- La pregunta indicada debe de existir.

4.3.2.11 Consultar respuestas de una pregunta

El sistema debe permitir mostrar las respuestas de una pregunta para ello se indicará la pregunta.

Como precondiciones, tenemos:

- La pregunta indicada debe de existir.
- La pregunta indicada debe de tener respuestas relacionadas.

4.3.2.12 Asignar solución a una pregunta

El sistema debe permitir asignar soluciones a una pregunta para ello se mostrarán las respuestas de la pregunta y se indicará al sistema cuales son solución.

Como precondiciones, tenemos:

- La pregunta indicada debe de existir.
- La pregunta indicada debe de tener respuestas relacionadas.

4.3.2.13 Alta de una respuesta

El sistema debe permitir dar de alta una nueva respuesta para ello se indicará la respuesta, la pregunta a la que pertenece y si es solución.

Como precondiciones, tenemos:

- La respuesta no puede coincidir con otra ya existente para esa pregunta.

- La pregunta relacionada debe de existir en el sistema.

4.3.2.14 Baja de una respuesta

El sistema debe permitir dar de baja una respuesta, para ello se indicará el código de respuesta a eliminar.

Como precondiciones, tenemos:

- La respuesta indicada debe de existir.

4.3.2.15 Consulta de respuestas

El sistema debe permitir consultar las respuestas registradas en el sistema, para ello el profesor podrá realizar búsquedas por texto o por código.

Como precondiciones, tenemos:

- Deben de existir respuestas registradas en el sistema.

4.3.2.16 Modificación de una respuesta

El sistema debe permitir modificar la información de una respuesta. Se podrá modificar la opción de si la respuesta es solución o no.

Como precondiciones, tenemos:

- La respuesta indicada debe de existir.

4.3.2.17 Consultar a que pregunta pertenece una respuesta

El sistema debe de permitir mostrar la información de la pregunta a la que pertenece la respuesta.

Como precondiciones, tenemos:

- La respuesta indicada debe de existir.

4.3.2.18 Alta de asignatura

El sistema debe de permitir dar de alta una asignatura, para ello se indicará el nombre.

Como precondiciones, tenemos:

- La asignatura no puede coincidir con otra ya existente en el sistema.

4.3.2.19 Baja de asignatura

El sistema debe de permitir dar de baja una asignatura, para ello se indicará el código de la asignatura a borrar.

Como precondiciones, tenemos:

- La asignatura indicada debe de existir.

4.3.2.20 Consulta de asignaturas

El sistema debe permitir consultar las asignaturas registradas en el sistema, para ello el profesor para ello el profesor podrá realizar búsquedas por nombre o por código.

Como precondiciones, tenemos:

- Deben de existir asignaturas registradas en el sistema.

4.3.2.21 Modificación de una asignatura

El sistema debe permitir modificar la información de una asignatura.

Como precondiciones, tenemos:

- La asignatura indicada debe de existir.

4.3.2.22 Consultar asignaturas relacionadas

El sistema debe permitir la consulta de las asignaturas relacionadas a una asignatura seleccionada por el usuario.

Como precondiciones, tenemos:

- La asignatura seleccionada debe de existir.
- La asignatura debe de tener asignadas asignaturas relacionadas.

4.3.2.23 Asignar asignaturas relacionadas

El sistema permitirá asignar asignaturas relacionadas a una asignatura, para ello mostrará las asignaturas relacionadas y se indicará al sistema cuales se quieren asignar.

Como precondiciones, tenemos:

- Deben de existir asignaturas relacionadas en el sistema.

4.3.2.24 Revocar asignaturas relacionadas

El sistema permitirá revocar asignaturas relacionadas de una asignatura, para ello mostrarán las asignaturas relacionadas de la asignatura y se indicará al sistema cuales se quieren revocar.

Como precondiciones, tenemos:

- La asignatura deberá tener asignadas asignaturas relacionadas.

4.3.2.25 Alta de asignaturas relacionadas

El sistema debe de permitir dar de alta asignaturas relacionadas con las asignaturas que se están evaluando a través del sistema, para ello se indicará el nombre de la asignatura relacionada y con que asignatura a ser evaluada tiene relación.

Como precondiciones, tenemos:

- La asignatura relacionada introducida no debe de existir en el sistema.

4.3.2.26 Baja de asignaturas relacionadas

El sistema debe de permitir dar de baja asignaturas relacionadas con las asignaturas a ser evaluadas con el sistema, para ello se indicará el código de la asignatura relacionada a borrar.

Como precondiciones, tenemos:

- La asignatura relacionada indicada debe de existir en el sistema.

4.3.2.27 Consulta de asignaturas relacionadas

El sistema debe permitir consultar las asignaturas relacionadas registradas en el sistema, para ello el profesor podrá realizar búsquedas por nombre o por código.

Como precondiciones, tenemos:

- Deben de existir asignaturas relacionadas registradas en el sistema.

4.3.2.28 Modificación de una asignatura relacionada

El sistema debe permitir modificar la información de una asignatura relacionada.

Como precondiciones, tenemos:

- La asignatura relacionada indicada debe de existir.

4.3.2.29 Consultar a que asignatura está relacionada

El sistema permitirá mostrar la información de la asignatura con la que esta relacionada una asignatura relacionada. En este requerimiento funcional no existen precondiciones.

4.3.2.30 Consultar alumnos que cursaron la asignatura relacionada

El sistema permitirá mostrar los alumnos que cursaron una asignatura relacionada.

Como precondiciones, tenemos:

- Deben de existir alumnos en el sistema que cursaron la asignatura relacionada.

4.3.2.31 Alta de grupo

El sistema permitirá dar de alta nuevos grupos, para ello se introducirá el nombre del grupo, la asignatura a la que pertenece ese grupo y el aula.

Como precondiciones, tenemos:

- El grupo no debe de existir previamente en el sistema.
- La asignatura indicada debe de existir.
- El aula del grupo debe de existir en el sistema.

4.3.2.32 Baja de grupo

El sistema debe de permitir dar de baja grupos, para ello se indicará al sistema que grupo se desea eliminar.

Como precondiciones, tenemos:

- El grupo indicado debe de existir en el sistema.

4.3.2.33 Consulta de grupos

El sistema debe permitir consultar los grupos registrados en el sistema, para ello el profesor podrá realizar búsquedas en el sistema por nombre o por código.

Como precondiciones, tenemos:

- Deben de existir grupos registrados en el sistema.

4.3.2.34 Modificación de un grupo

El sistema debe permitir modificar la información de un grupo.

Como precondiciones, tenemos:

- El grupo indicado debe de existir.

4.3.2.35 Consultar el profesor del grupo

El sistema debe de permitir consultar la información del profesor asignado a un grupo.

Como precondiciones, tenemos:

- El grupo indicado debe de existir.
- El grupo indicado debe de tener un profesor asignado.

4.3.2.36 Consultar los alumnos del grupo

El sistema debe de permitir consultar los alumnos registrados en un grupo.

Como precondiciones, tenemos:

- El grupo indicado debe de existir.
- El grupo indicado debe de tener alumnos registrados.

4.3.2.37 Alta de profesor

El sistema debe de permitir dar de alta nuevos profesores, para ello se introducirá toda la información necesaria del profesor. El sistema asignará automáticamente como contraseña el DNI del profesor, pudiendo posteriormente ser modificado.

Como precondiciones, tenemos:

- El profesor introducido no debe de existir en el sistema.

4.3.2.38 Baja de profesor

El sistema debe de permitir dar de baja profesores, para ello se indicará al sistema que profesor se desea eliminar.

Como precondiciones, tenemos:

- El profesor indicado debe de existir en el sistema.

4.3.2.39 Consulta de profesores

El sistema debe permitir consultar los profesores registrados en el sistema, para ello el profesor podrá realizar búsquedas por apellido, email o DNI.

Como precondiciones, tenemos:

- Deben de existir profesores registrados en el sistema.

4.3.2.40 Modificación de un profesor

El sistema debe permitir modificar la información de un profesor.

Como precondiciones, tenemos:

- El profesor indicado debe de existir.

4.3.2.41 Consultar grupos de un profesor

El sistema debe permitir consultar los grupos asignados a un profesor (docencias).

Como precondiciones, tenemos:

- El profesor debe de tener grupos asignados.

4.3.2.42 Asignar docencia a un profesor

El sistema permitirá asignar docencia para un profesor, para ello se mostrarán los grupos que aún no tienen docencia asignada y se indicará al sistema con que grupos se desea relacionar al profesor.

Como precondiciones, tenemos:

- No debe de existir en el sistema una docencia igual.
- El profesor debe de existir en el sistema.
- El grupo debe de existir en el sistema.

4.3.2.43 Revocar docencia a un profesor

El sistema debe permitir revocar docencia a un profesor, para ello se mostrarán los grupos que tiene asignados el profesor y se indicará al sistema los grupos que se quieren revocar.

Como precondiciones, tenemos:

- El profesor debe de tener docencia asignada en el sistema.

4.3.2.44 Alta de alumno

El sistema debe de permitir dar de alta nuevos alumnos, para ello se introducirá toda la información necesaria del alumno. El sistema asignará automáticamente como contraseña el DNI del alumno, pudiendo posteriormente ser modificado. El alumno creado por el profesor pasara inmediatamente a ser validado.

Como precondiciones, tenemos:

- El alumno introducido no debe de existir en el sistema.

4.3.2.45 Baja de alumno

El sistema debe de permitir dar de baja alumnos, para ello se indicará al sistema que alumno se desea eliminar.

Como precondiciones, tenemos:

- El alumno indicado debe de existir en el sistema.

4.3.2.46 Consulta de alumnos

El sistema debe permitir consultar los alumnos registrados en el sistema, para ello el profesor podrá realizar búsquedas por apellido, email o DNI.

Como precondiciones, tenemos:

- Deben de existir alumnos registrados en el sistema.

4.3.2.47 Modificación de un alumno

El sistema debe permitir modificar la información de un alumno.

Como precondiciones, tenemos:

- El alumno indicado debe de existir.

4.3.2.48 Consultar grupos de un alumno

El sistema debe permitir consultar los grupos en los que esta registrado un alumno (docencia).

Como precondiciones, tenemos:

- El alumno debe de tener grupos asignados.

4.3.2.49 Consultar exámenes de un alumno

El sistema debe permitir consultar los exámenes que ha realizado un alumno.

Como precondiciones, tenemos:

- El alumno debe de haber realizado algún examen previamente.

4.3.2.50 Consultar asignaturas de un alumno

El sistema debe permitir consultar las asignaturas en las que esta registrado un alumno (docencia).

Como precondiciones, tenemos:

- El alumno debe de tener asignaturas asignadas.

4.3.2.51 Consultar asignaturas relacionadas de un alumno

El sistema debe permitir consultar las asignaturas relacionadas que ha cursado un alumno.

Como precondiciones, tenemos:

- El alumno debe de haber cursado asignaturas relacionadas.

4.3.2.52 Asignar docencia a un alumno

El sistema permitirá asignar docencia a un alumno, para ello se mostrarán los grupos en los que no está registrado el alumno y se indicará al sistema en que grupos se desea asignar docencia.

Como precondiciones, tenemos:

- No debe de existir en el sistema una docencia igual.
- El alumno debe de existir en el sistema.
- La asignatura debe de existir en el sistema.
- El grupo debe de existir en el sistema.

4.3.2.53 Revocar docencia a un alumno

El sistema debe permitir revocar docencia a un alumno, para ello se mostrarán los grupos del alumno y se indicará al sistema que grupos se desean revocar.

Como precondiciones, tenemos:

- Debe de existir una docencia para ese alumno.

4.3.2.54 Asignar compañero

El sistema permite asignar al alumno su compañero para un grupo de prácticas. La asignación se realizara en ambos sentidos.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- El grupo debe ser de prácticas.
- El alumno que queremos asignar como nuestro compañero debe de ser del grupo.
- El alumno que queremos asignar como nuestro compañero debe de estar libre.

4.3.2.55 Modificar compañero

El sistema permitirá al usuario cambiar el compañero de prácticas. La modificación se realizará en ambos sentidos.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- Debe ser un grupo de prácticas.
- Debe tener compañero asignado.

4.3.2.56 Borrar compañero

El sistema permitirá al usuario borrar el compañero de prácticas. La modificación se realizará en ambos sentidos.

Como precondiciones, tenemos:

- El alumno debe de haber iniciado sesión.
- Debe ser un grupo de prácticas.
- Debe tener compañero asignado.

4.3.2.57 Asignar asignaturas relacionadas a un alumno

El sistema permitirá asignar asignaturas relacionadas a un alumno, para ello se mostrarán las asignaturas relacionadas del sistema y se indicará al sistema que asignaturas relacionadas asignar.

Como precondiciones, tenemos:

- El alumno debe de existir en el sistema.
- La asignatura relacionada debe de existir en el sistema.

4.3.2.58 Revocar asignaturas relacionadas a un alumno

El sistema debe permitir revocar asignaturas relacionadas a un alumno, para ello se mostrarán las asignaturas relacionadas asignadas al alumno y se indicará al sistema que cuales se quiere revocar.

Como precondiciones, tenemos:

- Deben de existir asignaturas relacionadas para ese alumno.

4.3.2.59 Crear lista de asistencia

El profesor podrá crear la lista de asistencia para un grupo; esté se seleccionará de una lista, y para el grupo indicado se mostrará una lista con los alumnos del grupo y el aula asignada. El profesor seleccionará el alumno de la lista y posteriormente indicará su posición en el aula.

Como precondiciones, tenemos:

- El profesor debe seleccionar la fecha y al menos un alumno.

4.3.2.60 Consultar lista de asistencia

El profesor podrá consultar la lista de asistencia para un grupo y fecha determinada. El sistema mostrará una lista con los alumnos y su posición, también se visualizará un dibujo del aula, donde se mostrarán las sillas de la misma y el posicionamiento de los alumnos en el aula.

Como precondiciones, tenemos:

- El profesor debe seleccionar el grupo y fecha a consultar.

4.3.2.61 Modificar lista de asistencia

El profesor podrá modificar la lista de asistencia para un grupo, el funcionamiento es el mismo que para crear la asistencia, se selecciona el alumno al que se desea modificar y se asignará la nueva posición.

4.3.2.62 Validar alumnos

El profesor podrá validar a los alumnos, es decir permitir o denegar el acceso del usuario a la aplicación. El sistema muestra una lista de los usuarios y su estado de validación.

Como precondiciones, tenemos:

- Deben de existir alumnos en el sistema.

4.3.2.63 Alta aula

El sistema debe de permitir dar de alta un aula, para ello se definirá como es el aula indicando el número de filas y columnas, también se introducirá un nombre y si se trata de un aula de prácticas o de teoría.

Como precondiciones, tenemos:

- No debe de existir en el sistema un aula con el mismo nombre.
- El número de filas y columnas no puede ser cero o negativo.

4.3.2.64 Baja aula

El sistema debe de permitir dar de baja un aula, para ello se indicará el código del aula a eliminar.

Como precondiciones, tenemos:

- El aula a eliminar debe de existir en el sistema.

4.3.2.65 Consulta de aulas

El sistema debe permitir consultar las aulas registradas en el sistema, para ello el profesor podrá realizar búsquedas por nombre o código.

Como precondiciones, tenemos:

- Deben de existir aulas registradas en el sistema.

4.3.2.66 Modificación de un aula

El sistema debe permitir modificar la información de un aula.

Como precondiciones, tenemos:

- El aula indicada debe de existir.

4.3.2.67 Consulta de grupos que se imparten en un aula

El sistema permitirá consultar los grupos que se imparten en un aula.

Como precondiciones, tenemos:

- El aula debe de existir en el sistema.
- El aula debe de tener grupos asignados.

4.3.2.68 Alta de examen

El sistema debe de permitir dar de alta un examen, para ello se indicará la fecha, para que grupo y en que aula se realizará el examen. En este requerimiento funcional no existen precondiciones.

4.3.2.69 Baja de examen

El sistema debe de permitir dar de baja un examen, para ello se indicará el código del examen a eliminar.

Como precondiciones, tenemos:

- El examen a eliminar debe de existir en el sistema.

4.3.2.70 Consulta de exámenes

El sistema debe permitir consultar los exámenes registrados en el sistema; para ello el profesor podrá realizar búsquedas por fecha o código.

Como precondiciones, tenemos:

- Deben de existir exámenes registradas en el sistema.

4.3.2.71 Modificación de un examen

El sistema debe permitir modificar la información de un examen.

Como precondiciones, tenemos:

- El examen indicado debe de existir.

4.3.2.72 Consultar preguntas de un examen

El sistema debe permitir consultar las preguntas asignadas a un examen.

Como precondiciones, tenemos:

- Se deben haber generado preguntas previamente para dicho examen.

4.3.2.73 Generar preguntas para un examen

El sistema debe permitir generar preguntas para un examen, para ello se indicará el número de preguntas y a qué tema pertenecen.

Como precondiciones, tenemos:

- Deben de existir en el sistema suficientes preguntas sobre dicho tema.
- El tema indicado debe de existir en el sistema.
- Las preguntas deben tener al menos dos soluciones.

4.3.2.74 Consultar alumnos que han realizado un examen

El sistema debe permitir consultar los alumnos que han realizado un examen.

Como precondiciones, tenemos:

- El examen debe de haber sido realizado previamente.

4.3.2.75 Iniciar un examen

El sistema permitirá al profesor iniciar un examen. El sistema mostrará el examen a los alumnos teniendo en cuenta la posición de ellos con el fin de evitar copias.

Como precondiciones, tenemos:

- Se deben de haber generado preguntas para el examen previamente.

4.3.2.76 Parar un examen

El sistema permitirá al profesor parar un examen. El sistema ocultará el examen a los alumnos.

Como precondiciones, tenemos:

- El examen debe de estar iniciado previamente.

4.3.2.77 Predecir probabilidad de copias

El sistema debe permitir predecir la probabilidad de copia de un alumno en un examen junto con los alumnos adyacentes a él. Debe de mostrar la información en una interfaz clara para ayudar al profesor a detectar las posibles copias.

Como precondiciones, tenemos:

- El examen debe de estar corregido por el sistema.
- El sistema no debe de estar iniciado.

4.3.2.78 Generar listados de notas

El sistema debe de permitir generar un listado de notas de un examen, para ello el profesor indicará el examen.

Como precondiciones, tenemos:

- El examen debe de estar corregido por el sistema.

4.3.2.79 Generar estadísticas

El sistema generará estadísticas a partir de todos los datos obtenidos durante todo el proceso de evaluación (probabilidad de copia, asistencia, posición del alumno, etc.). El sistema presentará estos datos de forma amigable para que el profesor pueda analizarlos.

4.3.3 Requisitos de interfaces externas**4.3.3.1 Interfaces de usuario**

Para acceder a AWAD será necesario un navegador Web con conexión a Internet, ya que se trata de una aplicación Web. La interfaz de usuario será idéntica para todos los navegadores que cumplan los estándares de XHTML 1.0 estricto que establece la W3C.

4.3.3.2 Interfaces hardware

Los usuarios que deseen acceder a AWAD deberán hacerlo desde un ordenador que disponga de conexión a Internet y un monitor con una resolución mínima recomendada de 800x600.

4.3.3.3 Interfaces software

No se necesita ningún sistema operativo en concreto para que el usuario pueda acceder a AWAD, simplemente mediante un navegador Web actualizado.

4.3.3.4 Interfaces de comunicaciones

El usuario siempre se conectara a LF desde Internet, por lo que las conexiones entre cliente y servidor se realizaran totalmente por Internet a través de TCP/IP y mediante el protocolo HTTP (típicamente por el puerto 80).

4.3.4 Requerimientos de eficiencia

Como el producto a desarrollar va a ser utilizado en la Facultad de informática de Valencia se podrán utilizar los recursos que posee la Universidad Politécnica de Valencia.

AWAD es una aplicación Web que por su naturaleza está diseñada para que estén conectados simultáneamente multitud de usuarios y el número de peticiones que se realicen sea exponencial al número de usuarios conectados, lo que supone una carga considerable y el número de conexiones a la base de datos será un punto a tener en cuenta que deberemos minimizar en los casos que sea posible. El número de conexiones a la base de datos es tan elevado ya que toda la información se encuentra en ella.

4.4 Restricciones de diseño**4.4.1 Estándares cumplidos**

La aplicación Web deberá cumplir con los estándares Web seguidos por el Consorcio World Wide Web (W3C).

4.4.2 Atributos del sistema**4.4.2.1 Seguridad**

La seguridad es uno de los puntos fundamentales de nuestra aplicación Web, para que un usuario pueda hacer uso de la aplicación debe estar autenticado, antes de realizar cualquier acción se comprueba que el usuario se haya acreditado, esto se hace en el formulario inicial de login con un correo y una contraseña previamente fijada en el momento en que se registró.

Las contraseñas son almacenadas cifradas en la base de datos, por lo que en ningún momento se guardan en claro.

4.4.2.2 Fiabilidad

Hay determinadas acciones que cuando las ejecuta el usuario implican una actualización de la base de datos, estas acciones suponen que el usuario queda pendiente de saber si se han ejecutado correctamente, para ello se informará al usuario con un mensaje del estado de su acción.

4.4.2.3 Portabilidad

En cuanto al servidor, nuestra aplicación será autocontenida, lo que significa que es independiente de la plataforma.

CAPÍTULO 5

5 Análisis del problema

Para la realización del análisis del problema, se ha dividido el trabajo en dos partes de estudio independientes.

La primera parte es un estudio de la situación actual de los portales Web orientados a la docencia. En él se saca información de las herramientas actuales que existen, cómo y con qué tecnologías han sido construidas, así como las características que ofrecen. Con esta información se pueden captar ideas y funcionalidades para nuestro portal, además de tomar decisiones basadas en análisis realizados sobre casos reales a la hora de desarrollar un portal.

En la segunda parte se realiza un estudio de los estándares clásicos de diseño Web, de manera que nos sea de ayuda para obtener mejores resultados a la hora de crear la aplicación Web. Este estudio es la base para crear un sitio Web de calidad, de forma que se consiga la satisfacción del usuario. Esta fase de análisis es muy importante ya que las aplicaciones Web tienen que ofrecer al usuario una navegación intuitiva, fácil, ágil y de calidad.

5.1 Estudio de la situación actual de portales Web orientados a la docencia

Actualmente existen varios portales Web orientados a la docencia. Para la realización de este estudio vamos a clasificarlos en dos grupos: los de código libre y los de uso comercial. En los siguientes apartados comentaremos las soluciones más utilizadas en ambos grupos.

5.1.1 Soluciones comerciales

En la parte de soluciones comerciales, podemos encontrar la herramienta *WebCT*, la cual es actualmente utilizada en la Universidad de Alcalá y en la Universidad de Sevilla entre otras universidades. La empresa encargada de desarrollar y distribuir esta herramienta se fusionó en el año 2006 con la compañía Blackboard, la cual también cuenta con un producto de las mismas características *Blackboard Academic Suite*.

5.1.2 Conclusiones del estudio

Actualmente existen herramientas muy completas, tanto libres como propietarias. Las soluciones comerciales parecen ofrecer más funcionalidades, aunque como hemos visto proyectos como *Sakai* pueden ofrecer funciones similares y cada vez están compitiendo más fuerte con las soluciones comerciales, sobretodo en la enseñanza universitaria.

Los productos comerciales que podemos encontrar hoy en día ofrecen muchas funciones, como hemos visto en los apartados anteriores. El objetivo de nuestro proyecto no es ese, ya que nuestro proyecto se centra en poder generar exámenes o pruebas y ser corregidas automáticamente, todo ello llevando un seguimiento del alumno y sobre todo recopilando y procesando toda la información derivada de estos procesos, donde luego será utilizada para otras funciones o mostrada de forma eficiente a los usuarios de la herramienta. De este modo dejamos de lado toda la parte de gestión de contenido, comunicación, etc., ya que puede conseguirse con herramientas libres como *Sakai* o *Moodle*.

Como conclusión podemos decir que nuestro proyecto, es una herramienta más específica y no tan general como las soluciones vistas, de ahí que su uso conjuntamente con otras plataformas libres podría ofrecer funciones y características tan completas como las soluciones comerciales.

5.2 Análisis del diseño Web

Como se ha comentado anteriormente en esta segunda parte se realiza un análisis del diseño Web. El objetivo de este análisis es que nuestro portal Web sea diseñado e implementado con calidad, para que se consiga una satisfacción del usuario. Este análisis está enfocado desde un punto de vista de la usabilidad Web, como uno de los gurús de la Web Jakob Nielsen propone en sus libros.

Para empezar definiremos los conceptos de diseño Web y usabilidad.

*El diseño Web es una actividad que consiste en la planificación, diseño e implementación de sitios Web y páginas Web. No es simplemente una aplicación del diseño convencional sobre Internet ya que requiere tener en cuenta cuestiones tales como **navegabilidad, interactividad, usabilidad, arquitectura** de la información y la **interacción** de medios como el audio, texto, imagen y video [WIKI-DW, 2009].*

Usabilidad es un atributo de calidad, relacionado en cómo es de fácil utilizar algo. Más concretamente, se refiere a la rapidez con que la gente puede aprender a utilizar algo, cómo de eficientes son mientras lo usan, cómo de memorable es, cómo de propenso a errores es, y cuánto les gusta a los usuarios usarlo. Si la gente no puede o no usa una función, es como si no existiera [NIELSEN-LORANGER, 2006].

Jakob Nielsen en [NIELSEN, 2000] basa su libro en observaciones obtenidas en test de usabilidad con unos 400 usuarios de diversas procedencias

usando un gran número de sitios Web. Los resultados muestran que los errores más frecuentes a todos los niveles del diseño de Web son:

- El modelo de negocio, no aceptar la innovación que supone la economía en la red.
- Gestión del proyecto, como un proyecto corporativo tradicional enfocado hacia adentro y no hacia un único interfaz con el cliente.
- Arquitectura de la información, como un espejo de la estructura de la compañía y no en función de las necesidades y tareas del usuario.
- Diseño de páginas, creadas para las demos internas más que para ser eficientes en las condiciones reales del usuario.
- Redacción de contenidos, escribiendo en el estilo lineal de siempre, y no pensando en los modos de lectura particulares de la Web.
- Estrategia de links, mirándose al ombligo y sin pensar en los sitios ajenos interesantes ni en los múltiples puntos de entrada con que pueden enlazar al nuestro.

Todos estos errores se resumen en uno, *La gente plantea proyectos Web basándose en su experiencia ajena a la Web.*

A continuación se tratan los factores más importantes que influyen en la usabilidad de un sitio Web.

5.2.1 Análisis del diseño de página

En esta sección se van a analizar los factores a considerar a la hora de abordar el diseño de las páginas de un portal Web.

5.2.1.1 Espacio de pantalla

Las páginas deberían estar dominadas por el contenido que interesa al usuario. Desgraciadamente en muchos casos se dedica casi más espacio a la navegación que a la información en sí. También encontramos publicidad que añaden a los sitios Web, teniendo en cuenta la usabilidad es totalmente recomendable que las páginas carezcan de esta publicidad.

Como regla general el contenido debería ocupar al menos un 50% del área de la página, y preferiblemente acercarse al 80% (perdemos cerca de un 16% con el propio navegador o sistema operativo). Las opciones de navegación deberían mantenerse en un porcentaje cercano al 20% en las páginas destino, aunque lógicamente tendrán más peso en las páginas de inicio o en las páginas intermedias.

Simplicidad es mejor que complejidad, especialmente en la Web, donde cada cinco bytes guardados es un milisegundo menos de tiempo de descarga [NIELSEN, 2000].

5.2.1.2 Diseño multiplataforma

En el diseño tradicional GUI (Graphic User Interface) para aplicaciones de escritorio, el diseñador o programador controla cada píxel en la pantalla, pone las cosas en el punto deseado y al tamaño requerido y puede prever el modo de acceso a cada punto de su programa.

En cambio en la Web es el usuario el que controla la navegación, pudiendo entrar a cualquier parte del sitio Web desde múltiples dispositivos y configuraciones de pantalla. Normalmente los usuarios vienen a través de computadoras tradicionales pero también hay que considerar la posibilidad de que accedan a través de dispositivos móviles. Por ello el diseño de la Web debe de estar pensado para que sea correcto el acceso desde distintos dispositivos.

5.2.1.3 Diseño independiente de la resolución

Para que el diseño sea independiente de la resolución de la pantalla hay que pensar y diseñar para cualquier resolución. Para conseguir esto no hay que usar un ancho de píxel fijo para tablas, marcos o elementos de diseño, habrá que trabajar siempre con porcentajes y tamaños relativos. Con ello conseguiremos no sólo la correcta visualización del contenido en pantalla, sino que facilitaremos una correcta impresión del mismo.

Además tendremos que tener en cuenta que los usuarios pueden tener distintas configuraciones respecto al tamaño de la fuente, ya sea por gustos o por incapacidad visual, por lo que hay que asegurarse de que la visualización sea correcta tanto con fuentes pequeñas como grandes. También es recomendable utilizar el texto para el texto es decir, no incluir texto en los gráficos.

Con respecto al uso del contenido no estándar, se debe intentar en la medida de lo posible no ser usado. De no ser así ha de cuidarse su inclusión, para evitar resultados no apetecidos o al menos, usar valores razonables por defecto. [CUEVA, 2004].

Como es imposible diseñar para todas las resoluciones, hay que centrarse en las más utilizadas. En la siguiente tabla se muestra en porcentajes las resoluciones utilizadas en los últimos años: [W3S, 2009].

Fecha	Más alta	1024x768	800x600	640x480	Desconocida
Enero 2008	38%	48%	8%	0%	6%
Enero 2007	26%	54%	14%	0%	6%
Enero 2006	17%	57%	20%	0%	6%
Enero 2005	12%	53%	30%	0%	5%
Enero 2004	10%	47%	37%	1%	5%
Enero 2003	6%	40%	47%	2%	5%
Enero 2002	6%	34%	52%	3%	5%
Enero 2001	5%	29%	55%	6%	5%
Enero 2000	4%	25%	56%	11%	4%

Figura 5. Tabla de porcentajes de resoluciones utilizadas.

Podemos observar que a medida que pasan los años la tendencia es tener resoluciones más altas.

5.2.1.4 Separar el significado de la presentación

En los orígenes de la Web, tanto su diseño como la manera de codificar los contenidos se basaron en codificar el significado de la información, no su presentación.

Hay que separar el contenido de las instrucciones específicas de visualización, para conseguir esto se utilizan las hojas de estilo en cascada (CSS). Donde por una parte están los archivos de contenido que sólo se incluye marcado semántico y por otra parte archivos CSS con información sobre la visualización.

5.2.1.5 Tiempos de respuesta

En la Web el tiempo de respuesta es el tiempo transcurrido desde la petición de una página hasta la completa visualización de la misma. Evidentemente los usuarios solicitan que se acelere la descarga de páginas, en sistemas de hipertexto el usuario necesita tiempos de respuesta inferiores al segundo cuando navega libremente entre páginas para que no tenga sensación de espera.

Recomendación normal en torno a los tiempos: [NIELSEN, 2000]

- 0.1 segundos es el límite para que el usuario perciba la descarga como inmediata.
- 1 segundo es el límite que hay para que el usuario piense que no hay interrupción.
- 10 segundos es el límite máximo para mantener la atención del usuario centrada en el diálogo.

Normalmente, los tiempos de respuesta suelen ser variables, debido a que están influidos por numerosos factores. Es importante intentar que los tiempos de respuesta no varíen mucho, ya que la satisfacción de los usuarios viene determinada por sus expectativas. Si la misma acción siempre dura lo mismo, los usuarios sabrán a que atenerse, y se acostumbrarán a ello.

También es recomendable indicar el tamaño de los archivos cuya descarga vaya a durar más de diez segundos tomando como referencia el ancho de banda medio de los usuarios.

Los tiempos de respuesta en la Web están sujetos a numerosos factores en función del volumen de información a transferir, como son: [CUEVA, 2004]

- El rendimiento del servidor.
- La conexión del servidor.
- La propia Internet, especialmente los cuellos de botella intercontinentales.

- La conexión del usuario a Internet.
- La velocidad de render del navegador y computador del usuario.

Estos factores de retardo son acumulativos, lo que significa que no conseguiremos tiempos rápidos de respuesta modificando uno solo de los elementos de la cadena.

Ante todo esto podemos decir que las páginas Web deben de ser diseñadas pensando en la rapidez. Para que el tamaño de las páginas sea pequeño, los gráficos deben ser reducidos a la mínima expresión, se deben de reutilizar en la medida de lo posible y sólo hay que emplear efectos multimedia cuando verdaderamente contribuyan a la comprensión de la información de usuario.

Lo más importante relacionado con el tiempo de respuesta es que el usuario vea una pantalla de información útil, además han de crearse las páginas de tal manera que se enmascare el tiempo de carga. Hay varios trucos para ello, como son:

- La parte superior debe decir algo aún antes de cargar las imágenes, (más texto y menos imágenes).
- Usar siempre el atributo ALT para que aparezca la descripción en texto de lo que enseñará la imagen que está llegando.
- Para recrear rápidamente la parte superior de la imagen deben incluirse WIDTH y HEIGHT en todos los elementos tales como imágenes y tablas.
- Las tablas complejas tardan en calcularse. Es más eficiente en términos de descarga crear tablas separadas, especialmente las que aparezcan en la parte superior.

5.2.1.6 Vinculación

Los links son los componentes más importantes del hipertexto ya que conectan las páginas y permiten a los usuarios ir a nuevos sitios de la Web. Hay tres grandes tipos de links:

- Links de navegación estructural.
- Links asociativos, información adicional a términos.
- Links referenciales, sugerencias listadas.

Debido al tiempo de descarga, y a la necesidad de no marear al usuario, es necesario informar claramente del alcance y contenido del link. Por ello es importante la descripción de los links, frente a la vieja regla de incluir el "pinche aquí" como link, es mejor declarar como link la frase o el término de manera que el usuario sepa adonde le lleva ese link.

Aunque el vínculo de hipertexto no debe contener más de cuatro palabras, es muy conveniente incluir información adicional que explique el vínculo. El tag HREF permite incluir un parámetro TITLE, el literal aquí incluido se presentará cuando el ratón se detenga sobre él. Existen algunas recomendaciones para su redacción: [NIELSEN, 2000].

- Incluir el nombre del sitio si es diferente del actual.
- Incluir el nombre de la sección del sitio en que estamos si es diferente de la actual.
- Detallar la clase de información que el usuario encontrará, así como la manera en que se relaciona con el texto del enlace y con el contenido de la página.
- Avisar de los problemas del destino (p.e., sólo para usuarios registrados, etc.).
- Deben ser de tamaño inferior a 80 caracteres, pero no deberían exceder los 60. Los cortos son los mejores.
- Si el texto marcado como enlace es obvio no incluir un *title* para decir lo mismo.
- Pensar que el título es una ayuda, el usuario no tendría por que utilizarlo para tener una primera noción del tipo de link de que se trata.

Con respecto a los colores de los enlaces, no utilizar colores no estándar, generalmente los vínculos de páginas que el usuario no ha visto son azules y las que ya ha visitado aparecen en rojo o morado. Con esto se ayuda al usuario a formarse una rápida idea de lo que puede ver y de lo que ya ha visto.

Los vínculos externos (salientes) deben seleccionarse cuidadosamente, siempre en el sentido de resultar útil al usuario y no generar insatisfacción en él. Hay quien tiende a pensar que es mejor tener al usuario en el propio sitio y se resiste a facilitar enlaces a otros sitios Web. Esto es inútil y además innecesario.

Los vínculos internos (entrantes) son una de las mejores maneras de generar tráfico hacia nuestro sitio Web. Desde este punto de vista es importante:

- Tener URLs permanentes para nuestras páginas.
- Tener páginas con contenidos concretos y no mezclar muchos temas en una misma página.

5.2.1.7 Plantillas de estilo

El estándar CSS es una de las mayores esperanzas de recuperar la separación entre la presentación y el contenido. Es importante usar un único estilo para el sitio, incluso cuando se personalice por áreas de contenido, ya que da coherencia y unidad al sitio. Es mejor insertar las plantillas como vinculadas que como código en la página, salvo que solamente afecten a una de ellas.

5.2.1.8 Marcos

Es recomendable no utilizar marcos por las siguientes razones:

- Rompen el modelo único.
- Difuminan el concepto de URL única.

- Distorsionan la mecánica de los links, favoritos, navegación hacia atrás...
- Dificultan la visión en pantallas pequeñas.
- Plantean problemas a la hora de imprimir con algunos navegadores.
- Son más complicados para los autores.
- Producen efectos no deseados en algunos buscadores.

5.2.1.9 Credibilidad

Hay tal cantidad de contenidos en la Web que es importante dar elementos al usuario para que nuestro sitio tenga credibilidad. Es importante que la página luzca profesional, hay que evitar los fondos pesados, las animaciones para enlazar con el correo, etc. Los gráficos bonitos no siempre inciden en aumentar la usabilidad de una página, pero también es cierto que debe cuidarse el aspecto visual ya que es lo primero que verá el usuario.

También es posible que el concepto de credibilidad se haga objetivo mediante ratings o rankings más o menos oficiales.

5.2.1.10 Conclusión

Como conclusiones del análisis del diseño de página para nuestro proyecto podemos señalar:

- Nuestro portal Web es para uso educativo y no tendrá publicidad, esto aumenta la usabilidad del sitio.
- Después de comprobar las estadísticas sobre las resoluciones de pantalla utilizadas hoy en día, nos centraremos en un diseño para 1024x768 o superior.
- Nuestro proyecto utiliza hojas de estilo CSS, para separar el contenido de la presentación.
- No utilizar marcos.
- Seguir las principales recomendaciones de usabilidad para los links y para reducir tiempos de respuesta.

5.2.2 Análisis del diseño del contenido

5.2.2.1 Escribir para la Web

Con pocas excepciones, la gente visita los sitios Web por su utilidad, no por su aspecto. Tener un buen aspecto es bueno, pero lo realmente importante es el contenido.

Una correcta escritura hace una gran diferencia en el aspecto de página, el tiempo dedicado a un sitio, y ventas. La esencial de la usabilidad Web consi-

dera el contenido escrito tanto como el diseño: Los clientes eligen claridad en la confusión. [NIELSEN-LORANGER, 2006].

Aunque es importante ser gramaticalmente correcto es importante presentarlo en función de la manera en que será leído. En esa línea es conveniente:

- Escribir no más del 50% de lo que se escribiría en una publicación impresa.
- Escribir de forma que las cosas puedan ser encontradas fácilmente, párrafos cortos, listas con bullets, subcabeceras, etc.
- Usar el hipertexto para romper la información extensa en varias páginas.
- Contratar editores de Web, los profesionales darán el mejor tratamiento a los contenidos.

5.2.2.1.1 Mantener los textos cortos.

Como se ha comentado anteriormente escribir no más del 50% de lo que se escribiría en una publicación impresa, ya que leer de la pantalla es cerca de un 25% más lento que leer de papel. También se sabe que a los usuarios no les gusta hacer scroll.

5.2.2.1.2 Textos legibles

Hay que tener cuidado con los “palabros” y con las faltas de ortografía, ya que crean desconfianza en el usuario. Siempre se debería usar un corrector ortográfico y gramático antes de publicar un texto, pero es además conveniente revisar los textos ya que los correctores automáticos no son perfectos.

5.2.2.1.3 Explorabilidad

Los usuarios tienden a no leer grandes cadenas de texto. La mayoría de los usuarios da un vistazo a la página según llega, y sólo unos pocos la leen letra a letra. Para intentar no perder al usuario el escritor para Web debe:

- Estructurar los artículos con dos o hasta tres niveles de titular.
- Utilizar títulos y subtítulos significantes.
- Usar listas bulleteadas y elementos similares para romper la uniformidad de los bloques de texto.
- Utilizar remarcas, negritas, o lo que sea para captar la atención del usuario.

5.2.2.1.4 Lenguaje plano

Se debe de escribir claro, conviene empezar la página con la conclusión y las cosas mas importantes, al revés de cómo se hace habitualmente, así se consigue que el lector capte el contenido aunque lleve poco tiempo leyendo. Se debe intentar poner una idea por párrafo, utilizar frases con estructura simple y ser cuidadosos con el humor y el sarcasmo.

5.2.2.2 Títulos de página

Escribir para la Web es a menudo escribir para ser encontrado. Hay que tener en cuenta que muchos buscadores presentarán nuestra página con el contenido del tag TITLE, por lo que puede ser interesante seguir las siguientes indicaciones: [NIELSEN, 2000]

- El título óptimo tiene que tener entre 2 y seis palabras.
- A diferentes páginas deben corresponder diferentes títulos.
- Eliminar los artículos del inicio.
- Coloca los elementos claves al inicio del título.

Además no solo los buscadores tienen en cuenta los títulos, los usuarios también pueden guiarse a través de ellos.

5.2.2.3 Legibilidad

Todo de lo que estamos hablando no sirve de mucho si el usuario no puede leer bien el texto. Hay unas cuantas reglas básicas para asegurar la legibilidad: [NIELSEN-LORANGER, 2006].

- Usar colores de gran contraste para el texto y el fondo. Lo óptimo es texto negro sobre fondo blanco.
- No utilizar fondos gráficos que puedan interferir la capacidad del ojo humano para interpretar líneas y letras.
- Usar tamaños de letra suficientemente grandes.
- Dejar el texto quieto, moverlo, parpadearlo o escalarlo dificultarán su lectura. Justificar el texto a la izquierda, naturalmente centrar textos es conveniente para algunos efectos.
- El texto pequeño es más legible si usamos fuentes de la familia sans-serif o verdana.
- Evita el uso de textos en mayúsculas. Un texto en mayúsculas se lee más lento que uno en minúsculas.

5.2.2.4 Multimedia

Actualmente hay una creciente aparición de recursos multimedia en los sitios Web ofreciendo mayores posibilidades de diseño. El uso de estos recursos requiere una buena disciplina de diseño, ya que el uso no restringido de multimedia confunde a los usuarios y dificulta la comprensión de la información.

5.2.2.5 Conclusión

El contenido es el centro de atención de los usuarios Web. La calidad de contenidos es uno de los determinantes de la usabilidad Web. Los usuarios son impacientes y guiados por sus objetivos, los contenidos deben orientarse a dar respuestas rápidas y ser útil al usuario.

Como conclusiones podemos decir que para la realización del proyecto se seguirán todas las recomendaciones comentadas en los puntos anteriores sobre escritura para la Web, legibilidad y títulos de página, además se evitará insertar recursos multimedia a no ser que sean necesarios para un correcto funcionamiento del portal Web.

5.2.3 Análisis del diseño del sitio

Como se ha comentado en el apartado anterior el contenido de calidad es uno de los determinantes par la usabilidad, otro determinante aún más importante es que los usuarios puedan encontrar la página que deseen. Desde el punto de vista de la usabilidad el diseño del sitio es más importante que el diseño de página. El diseño de los sitios debe de enfocarse a la simplicidad, con una arquitectura de información muy clara, sin distracciones.

5.2.3.1 La página de inicio

La página de inicio es la más importante del sitio y por ello debe ser diseñada de forma distinta al resto, siempre respetando el mismo estilo que las páginas interiores. La página de inicio debe responder a las dos principales cuestiones del usuario: ¿Dónde estoy? y ¿Qué hace este sitio?

Otras funciones que debe ofrecer la página de inicio son: [CUEVA, 2004].

- Presentar en la parte superior izquierda el nombre y el logo de la organización.
- Ofrecer una idea clara del esquema de navegación en el sitio.
- Ofrecer acceso directo a las funcionalidades más comúnmente usadas del sitio.
- Anunciar las novedades o promociones.
- Proporcionar una herramienta de búsqueda.

5.2.3.1.1 Pantalla de bienvenida

Algunos sitios utilizan páginas de bienvenida, este tipo de página es mostrada al usuario antes de la página de inicio. El objetivo de esta página es simplemente dar la bienvenida al usuario y mostrar el nombre y el logotipo de la organización. Lo cierto es que los usuarios las saltan tan rápido como pueden ya que son molestas y una perdida de tiempo, ya que ralentizan al usuario hasta llegar a la página de inicio.

5.2.3.2 Páginas interiores

Las páginas interiores deben centrarse en los contenidos y dejar las bienvenidas e información de lo que ofrece el sitio para la página de inicio. Aunque lógicamente información como el nombre y el logo de la organización debería repetirse en todas las páginas interiores, especialmente porque desconocemos el punto de entrada del usuario. Para facilitar el acceso y reubicar al usuario, debe colocarse siempre un enlace directo a la página de inicio.

5.2.3.3 Navegación

En el diseño de páginas Web los elementos de navegación son considerados como uno de los puntos fuertes e importantes. No deben de considerarse simplemente como enlaces hipertexto ya que son los que permiten al usuario moverse a través de las diferentes páginas del sitio Web y un mal uso de ellos puede provocar que el usuario no encuentre lo que busca, se pierda o este totalmente confundido.

Un interface de navegación debe de responder al usuario las siguientes preguntas: ¿Dónde estoy?, ¿Dónde he estado? Y ¿A dónde puedo ir?

5.2.3.3.1 ¿Dónde estoy?

Es la pregunta más importante para que el usuario no este confundido y no se pierda. La respuesta, en términos de situación dentro de un sitio debe proporcionarse a dos niveles: En relación a la Web como un todo y en relación a la estructura del sitio.

La primera es importante ya que el usuario a menudo percibe la Web como un todo. Los cambios drásticos en mecanismos de navegación no son bien asumidos dentro de esta perspectiva. Esto implica así mismo, que el usuario no sabrá en que sitio está a menos que se lo digas, ya se ha comentado anteriormente que se debe colocar el nombre y el logo de la organización en todas las páginas.

La situación en cuanto a la estructura del sitio se consigue mediante la inclusión de cabeceras que muestran a grandes rasgos la estructura del sitio Web y resaltan la situación actual. Otro elemento a tener en cuenta es el propio título de la página, que debe ser significativa e individualizado.

5.2.3.3.2 ¿Dónde he estado?

Con la tecnología Web actual resulta difícil contestar a esta pregunta. Existen sin embargo algunos recursos que nos permiten afrontarla:

- El botón Atrás del navegador, que nos proporciona una historia secuencial de las páginas visitadas.
- El historial del navegador, donde se pueden consultar las páginas visitadas durante el periodo de tiempo que tenga configurado el navegador.
- Los links ya visitados aparecen de color diferente, hay que respetar la convención en cuanto a la representación de los links, (azul para los activos y rosa para los visitados).

5.2.3.3.3 ¿A dónde puedo ir?

Esta cuestión se responde con las opciones visibles de navegación y con el resto de links dentro de la página. Es imposible mostrar todos los destinos posibles en una sola página, una buena estructura del sitio fácil de comprender y asimilar por el usuario será de gran utilidad a la hora de conocer posibles puntos de destino desde un sitio dado.

Hay tres clases de hyperenlaces que pueden ser usados en una página:

- Links embebidos: El tradicional texto subrayado.
- Links estructurales: Apuntan a otro nivel de la estructura, por lo que diferirán de página a página, si bien es necesario que mantengan una coherencia comprendida y esperada por el usuario.
- Links asociativos: Proporcionan información adicional o colateral.

5.2.3.3.4 La estructura del sitio

Para una correcta navegación en nuestro portal, es de vital importancia tener en cuenta la estructura del mismo. En ella se muestra al usuario la situación actual y los posibles movimientos alternativos que puede realizar.

Las dos reglas más importantes acerca de la estructura de un sitio son: Que la propia estructura exista y hacer que refleje la visión del usuario con respecto al sitio, su información o sus servicios. La primera parece obvia, pero muchos sitios evolucionan sin una estructura original clara hasta terminar en el caos más absoluto.

Existen distintos tipos de estructuras Web, la mayoría de los sitios tienen una estructura jerárquica, con niveles de información progresivamente más detallados. Otros sitios tienen una estructura tabular en la que las páginas se clasifican de acuerdo a cierto número de atributos o parámetros, el usuario puede seleccionar uno o varios de ellos para agrupar la información según sus criterios. Finalmente hay estructuras lineales que son de aplicación en sitios que refleja una sucesión de pasos.

La estructura del sitio debe de construirse en función de la percepción del usuario, un error muy frecuente es plantearla como una réplica de la estructura departamental de la organización a la que pertenece el sitio Web sin tener en cuenta para nada al usuario. La estructura debe estar determinada por las tareas que los usuarios quieren desarrollar, incluso si ello supone mezclar en una página información de distintas áreas de la organización.

Esta concepción de la estructura, orientada en función de la percepción del usuario presenta respecto a aquella que pretende imponerles la decisión de la organización un porcentaje de éxito superior al 80% frente al 9% de la segunda opción. [NIELSEN, 2000].

5.2.3.3.5 Extenso vs. Profundo

El diseño de navegación se puede clasificar en dos tipos: extenso y profundo, aunque en algunos casos se utiliza una combinación de los dos.

El diseño más utilizado es el extenso, en el cual se listan todos los elementos de alto nivel del sitio, a menudo en la banda izquierda de la pantalla, aunque también puede ir en la banda derecha. El beneficio de este diseño es que el usuario es continuamente informado de los servicios disponibles en el sitio haya entrado por el punto que haya entrado. Su inconveniente es que se pierde mucho espacio de pantalla en todas las páginas.

En el diseño profundo se presenta la información en profundidad, es decir, en cada página se informa de los diferentes elementos jerárquicos que la ante-

ceden. Este segundo escenario tiene la ventaja de ocupar poco espacio y ser muy sencillo lo que deja al usuario concentrarse en el área de contenidos que es lo que generalmente hace. Su inconveniente radica en la dificultad de utilizarlo en estructuras no jerárquicas.

5.2.3.4 Capacidades de búsqueda

Se puede distinguir tres grupos de usuarios en función de su forma de llegar a la información: usuarios búsqueda, usuarios link, usuarios mixtos. Los primeros utilizan directamente las opciones de búsqueda del sitio Web. Los segundos prefieren seguir los links para acceder a la información y si fracasan usarán las opciones de búsqueda. Finalmente los mixtos utilizan uno de los dos caminos en función de las expectativas que se le presentan en cada momento.

Para satisfacer todos los escenarios es conveniente que todas las páginas transmitan una idea clara de su situación estructural. Además la búsqueda debería ser fácilmente disponible desde cualquier página del sitio.

En lo referente a búsquedas avanzadas, la utilización de búsquedas booleanas y parámetros de búsqueda avanzados solo deben de poder utilizarse en el apartado de búsqueda avanzada y no en la primera instancia de búsqueda.

La página de resultados de búsqueda es importante. Los resultados se deben mostrar de forma clara y ser ordenados en función de su importancia, los más importantes primero.

5.2.3.5 Diseño de URL

Las URL han de considerarse como un elemento de diseño, ya que en la práctica se exponen a los usuarios. En principio, lo correcto es indicar las URL's completas, pero hoy en día los servidores Web están preparados para responder tan sólo con el dominio.

El nombre de nuestro dominio es importante, ya que muchos usuarios nos buscarán directamente a través de él. Para su selección ha de pensarse en la sencillez, en la facilidad de recordarlo y en la facilidad de escribirlo correctamente. En caso de términos compuestos hay varias posibilidades razonables: juntar las palabras, usar abreviaturas, usar un guión para intercalar. Lo más recomendado es la primera opción.

Igualmente, a la hora de especificar una URL en nuestro código es recomendable hacerlo de forma completa, incluyendo una barra al final para facilitar al servidor la distinción entre archivos y directorios.

Es importante tener URLs estables, la URL es el camino por el que otras páginas generarán tráfico hacia nosotros y este tráfico debe de encontrar su destino. Cuando se reorganiza un sitio deben mantenerse un mínimo de seis meses las urls antiguas, y redireccionar al usuario a la nueva dirección. La estabilidad de las URL nos permitirá publicitarlas por diferentes medios, buscando siempre el incremento del tráfico hacia nuestro sitio.

5.2.3.6 JavaScript

JavaScript se utiliza en el lado del cliente, es decir en el navegador Web, esto significa que todo lo que podamos hacer con este lenguaje de programación evitará carga al servidor Web. Es muy utilizado para comprobar datos en los formularios antes de ser enviados al servidor, también con su uso podemos conseguir una correcta visualización de las páginas en pantalla, ya que podemos detectar las resoluciones de pantalla empleadas, versión del navegador, etc. Hay que tener cuidado con el código JavaScript, ya que existen diferentes versiones y es muy dependiente del navegador.

5.2.3.7 Conclusiones

Como conclusiones del diseño del sitio, tendremos en cuenta y destacaremos para la realización de nuestro portal Web en el proyecto:

- El diseño de los sitios debe de enfocarse a la simplicidad, con una arquitectura de información muy clara, sin distracciones.
- Con respecto a la página de inicio:
 - Presentar en la parte superior izquierda el nombre y el logo de la organización.
 - Ofrecer una idea clara del esquema de navegación en el sitio.
 - Ofrecer acceso directo a las funcionalidades más comúnmente usadas del sitio.
- No utilizaremos página de bienvenida.
- Las páginas interiores tendrán un contenido adecuado y un enlace a la página de inicio.
- Estructura del sitio orientada al usuario.
- Inicialmente nuestro portal no tendrá opciones de búsqueda de información, ya que el objetivo no es que el usuario busque información. Hay que recordar que nuestro portal ofrecerá unos específicos y claros servicios que estarán al alcance de los usuarios.
- Utilizar JavaScript en la medida de lo posible para no sobrecargar el servidor.

CAPÍTULO 6

6 Diseño del sistema

La aplicación que vamos a desarrollar en este proyecto tiene como elemento central a la base de datos. Es decir, las funciones principales del sistema se basan en el acceso a la base de datos. Por esta razón, en la primera parte del diseño del sistema nos centramos en el diseño de la base de datos, ya que las otras partes del sistema consisten básicamente en el procesado de la información extraída de la misma.

En la segunda parte del diseño del sistema se ha utilizado un método de producción de aplicaciones Web (OOWS) basado en modelado conceptual y estrategias de implementación. De esta forma se ha utilizado ingeniería del software aplicada a la Web, evitando así la implementación del portal Web sin seguir ningún tipo de soporte para desarrollo de software. [OOWS, 2004] [OOWS2, 2002].

6.1 Diseño de la base de datos

En el diseño del sistema, la primera parte que se va a exponer es el diseño de la base de datos. Dentro del sistema, la base de datos es una parte muy importante y debe de diseñarse correctamente teniendo en cuenta el análisis de requisitos previo.

El diseño conceptual es la fase del diseño de una base de datos cuyo objetivo es *“obtener una representación de la realidad que capture las propiedades estáticas y dinámicas de la misma que son necesarias para satisfacer los requisitos; esta representación debe suponer una imagen fiel del comportamiento del mundo real”* [DBD, 2004].

Los modelos de datos conceptuales son las herramientas que se utilizan para realizar este diseño. En este proyecto se utiliza uno de estos modelos, el Modelo Entidad-Relación (ER a partir de ahora), así como una metodología de diseño basada en este modelo. Hay que destacar que algunas de las extensiones que se pueden encontrar de este modelo incluyen un lenguaje para la representación de las propiedades dinámicas, sin embargo, en este proyecto estas propiedades serán tratadas de manera informal.

6.1.1 Modelo Entidad-Relación

Un **diagrama o modelo entidad-relación** (a veces denominado por su siglas, *E-R* "Entity relationship", o, "DER" Diagrama de Entidad Relación) es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades. [WIKI-MER, 2010].

En el siguiente ejemplo se muestra un diagrama Entidad – Relación para la realización de pedidos por parte de un cliente. Los cuadros Cliente, Pedido y Artículo son entidades y representan tablas en la base de datos. Dichas tablas se componen por atributos y se representan como círculos unidos a las entidades. Por último las relaciones se representan por rombos y según su cardinalidad se transformaran en tablas a la hora de transformar el diagrama E-R al esquema lógico.

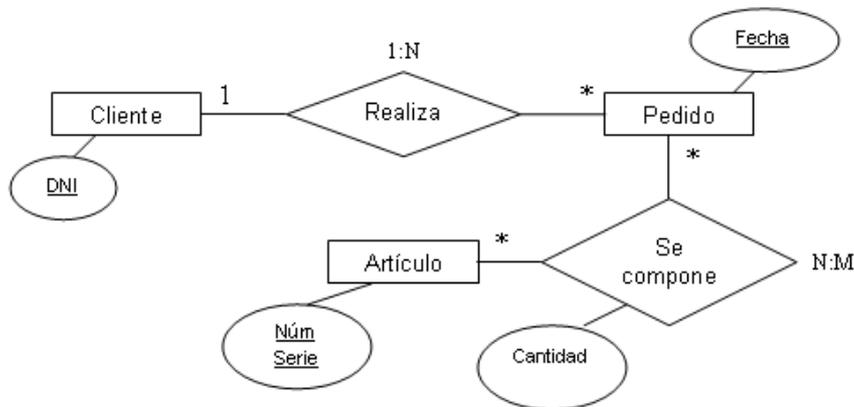


Figura 6. Ejemplo de Entidad-Relación.

6.1.1.1 Diagrama Entidad Relación

El diagrama ER de la base de datos para el proyecto se muestra en la figura 3.

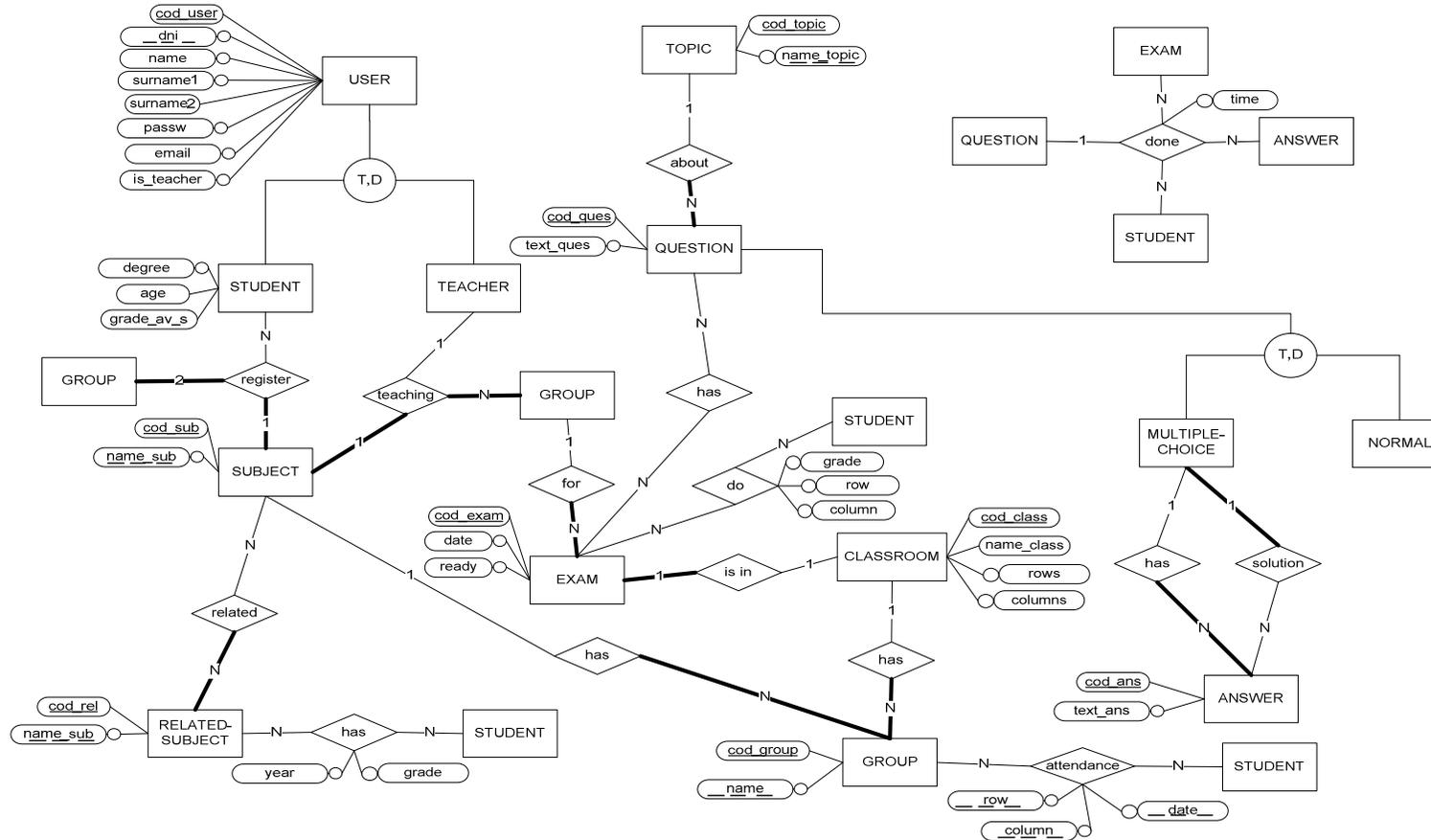


Figura 7. Diagrama Entidad-Relación.

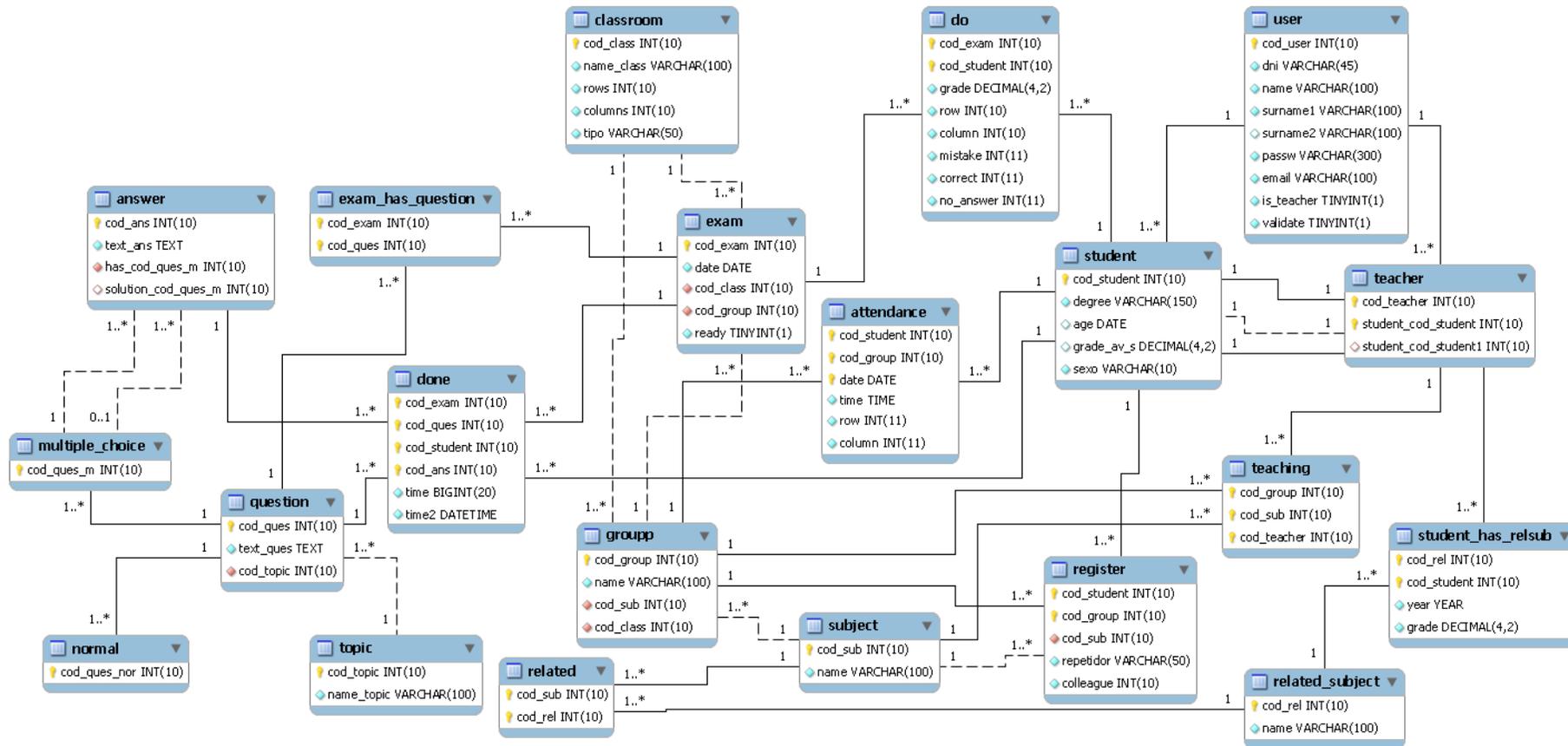


Figura 8. Diagrama ER MySQL Workbench.

El diagrama está representado con la notación *crow's foot* y no con la notación *Chen* empleada anteriormente para crear el diagrama ER manualmente.

Ya que el diagrama ER puede resultar críptico, a continuación se especifica el significado de cada entidad, los atributos y las relaciones entre las entidades.

Las entidades del diagrama ER son las siguientes:

USER: Representa un usuario del sistema, esta entidad se especializa en dos entidades especializadas o subclases STUDENT y TEACHER. Esta especialización es total (T) ya que todas sus ocurrencias deben participar en alguna subclase y disjunta (D) puesto que una ocurrencia de la entidad no puede pertenecer a distintas subclases. Los atributos de la entidad son:

- cod_user: Código del usuario en la base de datos.
- dni: Dni del usuario o equivalente en caso de usuarios extranjeros.
- name: Nombre del usuario.
- surname1: Primer apellido del usuario.
- surname2: Segundo apellido del usuario.
- passw: Palabra clave del usuario para acceder al sistema.
- email: Correo electrónico del usuario.
- is_teacher: Campo *boolean* que indica si el usuario es profesor o no. Aunque este campo no es necesario por la especialización puede amenizar bastante la programación a la hora de implementar.
- Validate: Determina si el alumno podrá acceder a la aplicación.

STUDENT: Representa a un usuario del sistema tipo estudiante. Los atributos de la entidad son:

- degree: Estudios que esta realizando el estudiante.
- age: Edad del estudiante.
- grade_av_s: Nota media del expediente del estudiante.
- sexo: Sexo del estudiante.

LECTURER: Representa a un usuario del sistema tipo profesor.

GROUP: Representa un grupo de una asignatura. Los atributos de la entidad son:

- cod_grup: Código del grupo en la base de datos.
- name: Nombre del grupo.

SUBJECT: Representa una asignatura. Los atributos de la entidad son:

- cod_sub: Código de la asignatura en la base de datos.
- name_sub: Nombre de la asignatura.

RELATED_SUBJECT: Representa una asignatura la cual tiene contenidos relacionados con otra asignatura registrada en el sistema. Los atributos de la entidad son:

- cod_rel: Código de la asignatura en la base de datos.
- name_sub: Nombre de la asignatura.

EXAM: Representa un examen. Los atributos de la entidad son:

- **cod_exam:** Código del examen en la base de datos.
- **date:** Fecha en la que se realiza el examen.
- **ready:** Campo *boolean* indica si un examen esta listo para que los alumnos empiecen a realizarlo.

QUESTION: Representa una pregunta de examen, esta entidad se especializa en dos entidades especializadas o subclases **MULTIPLE_CHOICE** y **NORMAL**. Esta especialización es total (T) ya que todas sus ocurrencias deben participar en alguna subclase y disjunta (D) ya que una ocurrencia de la entidad no puede pertenecer a distintas subclases. Los atributos de la entidad son:

- **cod_ques:** Código de la pregunta en la base de datos.
- **text_ques:** Texto de la pregunta.

MULTIPLE_CHOICE: Representa una pregunta de examen de tipo test.

NORMAL: Representa una pregunta de examen de tipo normal.

TOPIC: Representa un tema de las preguntas. Los atributos de la entidad son:

- **cod_topic:** Código de tema en la base de datos.
- **name_topic:** nombre del tema.

ANSWER: Representa una respuesta para una pregunta de tipo test. Los atributos de la entidad son:

- **cod_ans:** Código de la respuesta en la base de datos.
- **text_ans:** Texto de la respuesta.

CLASSROOM: Representa un aula. Los atributos de la entidad son:

- **cod_class:** Código del aula en la base de datos.
- **name_class:** Nombre del aula.
- **rows:** Número de filas del aula.
- **columns:** Número de columnas del aula.

Las relaciones del diagrama ER son las siguientes:

REGISTER (student-group-subject): Representa un alumno dado de alta en una asignatura y para un grupo.

TEACHING (teacher-group-subject): Representa la docencia de un profesor, la asignatura que dará y el grupo.

RELATED (related_subject-subject): Representa la relación entre una asignatura y una asignatura relacionada, es decir que tiene contenidos relacionados con la otra asignatura.

HAS (subject-group): Representa que todas las asignaturas tienen grupos.

FOR (exam-group): Representa que un examen es para un grupo.

HAS (related_subject-student): Representa que un alumno ha cursado alguna asignatura relacionada. Los atributos de la relación son:

- year: Año en que el alumno curso la asignatura.
- grade: Nota obtenida por el alumno en la asignatura relacionada.

ATTENDANCE (group-student): Representa la asistencia del alumno a los grupos donde esta dado de alta. Los atributos de la relación son:

- row: Fila donde ha estado sentado el alumno en clase.
- column: Columna donde ha estado sentado el alumno en clase.
- date: Fecha en la que ha asistido el alumno a la clase.

HAS (multiple_choice-answer): Representa que una pregunta de tipo test tiene respuestas.

SOLUTION (multiple_choice-answer): Representa que una pregunta de tipo test tiene respuestas que son solución a la pregunta.

DONE (exam-question-student-answer): Representa una pregunta con su respectiva contestación por un alumno en un examen. Los atributos de la relación son:

- time: Tiempo exacto en el que el alumno envía la respuesta.

HAS (exam-question): Representa que un examen tiene preguntas.

DO (exam-student): Representa que un alumno hace exámenes. Los atributos de la relación son:

- grade: Nota del alumno en el examen.
- row: Fila donde esta sentado el alumno en el examen.
- column: Columna donde esta sentado el alumno en el examen.

IS IN (exam-classroom): Representa la clase donde se hace un examen.

HAS (classroom-group): Representa la clase que tiene asignada cada grupo.

ABOUT (question-topic): Representa que las preguntas son sobre un tema.

6.1.1.2 La importancia de los procedimientos almacenados

Los procedimientos almacenados son subrutinas que están disponibles para las aplicaciones que acceden a una base de datos relacional. Los procedimientos almacenados están, como su nombre lo indica, almacenados en el diccionario de datos de la base de datos.

Los procedimientos usados en nuestra aplicación los explicamos a continuación:

Insertar_alumno: Se usa cuando el profesor inserta un alumno en el sistema.

```
DROP PROCEDURE `insertar_alumno`//
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertar_alumno`(IN
v_dni TEXT, IN v_nombre TEXT, IN v_ape1 TEXT, IN v_ape2 TEXT,
IN v_email TEXT, IN v_estudios TEXT, IN v_edad DATE, IN v_sexo TEXT)
BEGIN
INSERT INTO project.user (cod_user, dni, name, surname1,
surname2, passw, email, is_teacher, validate)
VALUES (NULL, v_dni, v_nombre, v_ape1, v_ape2,
PASSWORD(v_dni), v_email, 0, 1);

INSERT INTO project.student (cod_student, degree, age,
grade_av_s, sexo)
VALUES (LAST_INSERT_ID(), v_estudios, v_edad, NULL, v_sexo);
END
```

Insertar_alumno_registro: Se usa cuando el alumno se registra en la aplicaci3n.

```
DROP PROCEDURE `insertar_alumno_registro`//
CREATE DEFINER=`root`@`localhost` PROCEDURE
`insertar_alumno_registro`(IN v_dni TEXT, IN v_nombre TEXT, IN v_ape1 TEXT, IN
v_ape2 TEXT,
IN v_email TEXT, IN v_estudios TEXT, IN v_edad DATE, IN v_media INT, IN
v_pass TEXT, IN v_sexo TEXT)
BEGIN
INSERT INTO project.user (cod_user, dni, name, surname1,
surname2, passw, email, is_teacher, validate)
VALUES (NULL, v_dni, v_nombre, v_ape1, v_ape2,
PASSWORD(v_pass), v_email, 0, 0);

INSERT INTO project.student (cod_student, degree, age,
grade_av_s, sexo)
VALUES (LAST_INSERT_ID(), v_estudios, v_edad, v_media, v_sexo);
END
```

Insertar_examen: Se usa cuando el profesor inserta un nuevo examen en el sistema.

```
DROP PROCEDURE `insertar_examen`//
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertar_examen`(IN
v_date TEXT, IN v_class INT, IN v_group INT)
BEGIN
INSERT INTO project.exam (cod_exam, `date`, cod_class,
cod_group, ready)
VALUES (NULL, STR_TO_DATE(v_date, '%d/%m/%Y'), v_class,
v_group, 0);
END
```

Insertar_profesor: Se usa cuando el profesor inserta un profesor.

```

DROP PROCEDURE `insertar_profesor`//
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertar_profesor` (IN
v_dni TEXT, IN v_nombre TEXT, IN v_ape1 TEXT, IN v_ape2 TEXT,
IN v_email TEXT)
BEGIN
INSERT INTO project.user (cod_user, dni, name, surname1,
surname2, passw, email, is_teacher, validate)
VALUES (NULL, v_dni, v_nombre, v_ape1, v_ape2,
PASSWORD(v_dni), v_email, 1, 1);
|
INSERT INTO project.teacher (cod_teacher)
VALUES (LAST_INSERT_ID());

END

```

Insertar_question: Se usa cuando el profesor se inserta una pregunta.

```

DROP PROCEDURE `insertar_question`//
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertar_question` (IN
v_text TEXT, IN v_cod_topic INT)
BEGIN
INSERT INTO `project`.`question` (`text_ques`,
`cod_topic`)VALUES (v_text, v_cod_topic);
INSERT INTO `project`.`multiple_choice` (`cod_ques_m`) VALUES
(LAST_INSERT_ID());

END

```

6.1.1.3 Especificar restricciones de integridad

Las restricciones de integridad sobre los atributos, las entidades y las relaciones están expresadas en el diagrama ER. Ahora bien, hay que tener en cuenta que para terminar con el diseño, todas aquellas propiedades de la realidad que no hayan quedado expresadas en el diagrama ER deben incluirse. Dado que no se ha especificado ningún lenguaje formal para su representación, hay que intentar que la frase o frases que definan estas restricciones no sean ambiguas. En el sistema de nuestro proyecto no tenemos ninguna restricción de integridad por expresar.

6.1.1.4 Diseño de transacciones

Una transacción es una secuencia de operaciones de acceso a los datos que constituye una unidad lógica de ejecución. [DBD, 2004].

Para modelar la evolución (dinámica) del sistema hay que diseñar un conjunto de transacciones mínimas que permitan actualizar la información almacenada. Se incluirá, para cada entidad y para cada relación del diagrama una transacción de inserción, una de borrado y varias de modificación. Las transacciones mínimas deben considerarse como la única forma de modificar la información almacenada, y por tanto, podrán ser utilizadas en transacciones más complejas diseñadas a medida para la aplicación.

Las transacciones deben ser diseñadas de forma que permitan realizar estas tareas por lo que las restricciones de integridad tendrán que ser tenidas en cuenta para evitar estados inconsistentes en el sistema.

Las transacciones mínimas diseñadas para el proyecto se listan a continuación, junto con el nombre de las relaciones con las que se relaciona:

6.1.1.4.1 Inserción

- Transacción para insertar en USUARIO de tipo STUDENT.
 - Insertar en USUARIO.
 - Insertar en STUDENT.
- Transacción para insertar en USUARIO de tipo TEACHER.
 - Insertar en USUARIO.
 - Insertar en TEACHER.
- Transacción para insertar en GROUP.
 - Insertar en GROUP.
- Transacción para insertar en SUBJECT.
 - Insertar en SUBJECT.
- Transacción para insertar en RELATED_SUBJECT.
 - Insertar en RELATED_SUBJECT.
 - Insertar en RELATED (related_subject-subject).
- Transacción para insertar en TOPIC.
 - Insertar en TOPIC.
- Transacción para insertar en CLASSROOM.
 - Insertar en CLASSROOM.
- Transacción para insertar en QUESTION de tipo MULTIPLE_CHOICE.
 - Insertar en QUESTION.
 - Insertar en MULTIPLE_CHOICE.
 - Insertar en SOLUTION (multiple_choice-answer).
- Transacción para insertar en QUESTION de tipo NORMAL.
 - Insertar en QUESTION.
 - Insertar en NORMAL.
- Transacción para insertar en EXAM.
 - Insertar en EXAM.
 - Insertar en DO (exam-student).
- Transacción para insertar en ANSWER.
 - Insertar en ANSWER.
- Transacción para insertar en REGISTER (student-group-subject).
 - Insertar en REGISTER (student-group-subject).
- Transacción para insertar en TEACHING (teacher-group-subject).
 - Insertar en TEACHING (teacher-group-subject).
- Transacción para insertar en RELATED (related_subject-subject).
 - Insertar en RELATED (related_subject-subject).
- Transacción para insertar en HAS (related_subject-student).
 - Insertar en HAS (related_subject-student).
- Transacción para insertar en ATTENDANCE (group-student).
 - Insertar en ATTENDANCE (group-student).
- Transacción para insertar en DONE (exam-question-student-answer).
 - Insertar en DONE (exam-question-student-answer).

- Transacción para insertar en HAS (exam-question).
 - Insertar en HAS (exam-question).
- Transacción para insertar en DO (exam-student).
 - Insertar en DO (exam-student).

6.1.1.4.2 Borrado

Se pueden definir operaciones de borrado de varios tipos, pero para el proyecto sólo se considerarán dos: restrictivo y en cascada respecto a cada relación en la que participe la entidad o el objeto agregado.

- Transacción para borrar en USUARIO.
 - Borrar en USUARIO.
 - Borrar en STUDENT (cascada respecto a REGISTER (student-group-subject), HAS (related_subject-student), ATTENDANCE (group-student), DONE (exam-question-student-answer), DO (exam-student)).
 - Borrar en TEACHING (cascada respecto a TEACHING (teacher-group-subject)).
- Transacción para borrar en GROUP (cascada respecto a FOR (group-exam), HAS (group-subject), ATTENDANCE (group-student)).
 - Borrar en GROUP.
 - Borrar en REGISTER (student-group-subject).
 - Borrar en TEACHING (teacher-group-subject).
- Transacción para borrar en SUBJECT (cascada respecto a RELATED (subject-related_subject)).
 - Borrar en SUBJECT.
 - Borrar en REGISTER (student-group-subject).
 - Borrar en TEACHING (teacher-group-subject).
- Transacción para borrar en RELATED_SUBJECT.
 - Borrar en RELATED-SUBJECT.
 - Borrar en RELATED (subject-related_subject).
 - Borrar en HAS (related_subject-student) (cascada).
- Transacción para borrar en TOPIC (cascada respecto a ABOUT (topic-question)).
 - Borrar en TOPIC.
- Transacción para borrar en CLASSROOM (restrictivo respecto a IS IN (exam-classroom), HAS (classroom-group)).
 - Borrar en CLASSROOM.
- Transacción para borrar en QUESTION (cascada respecto a HAS (question-exam), DONE (exam-question-answer-student)).
 - Borrar en QUESTION.
 - Borrar en MULTIPLE_CHOICE (cascada respecto a HAS (multiple_choice-answer)).
 - Borrar en SOLUTION (multiple_choice-answer).
 - Borrar en NORMAL.
- Transacción para borrar en EXAM (cascada respecto a DONE (question-exam-student-answer), DO (exam-student), HAS (exam-question)).

- Borrar en EXAM.
- Transacción para borrar en ANSWER (cascada respecto a DONE (question-exam-student-answer), SOLUTION (answer-multiple_choice)).
 - Borrar en ANSWER.
- Transacción para borrar en REGISTER (student-group-subject) (restrictivo respecto a GROUP, SUBJECT).
 - Borrar en REGISTER (student-group-subject).
- Transacción para borrar en TEACHING (teacher-group-subject) (restrictivo respecto a GROUP, SUBJECT).
 - Borrar en TEACHING (teacher-group-subject).
- Transacción para borrar en RELATED (related_subject-subject) (restrictivo respecto a RELATED_SUBJECT).
 - Borrar en RELATED (related_subject-subject).
- Transacción para borrar en HAS (related_subject-student) (restrictivo respecto a SUBJECT).
 - Borrar en HAS (related_subject-student).
- Transacción para borrar en ATTENDANCE (group-student).
 - Borrar en ATTENDANCE (group-student).
- Transacción para borrar en DONE (exam-question-student-answer).
 - Borrar en DONE (exam-question-student-answer).
- Transacción para borrar en HAS (exam-question).
 - Borrar en HAS (exam-question).
- Transacción para borrar en DO (exam-student) (restrictivo respecto a EXAM).
 - Borrar en DO (exam-student).

6.1.1.4.3 Modificación

Dependiendo de las necesidades del proyecto, puede haber muchas transacciones de modificación diferentes. Deben diseñarse las transacciones que posibiliten modificar cualquier atributo de cualquier objeto que sea necesario.

6.1.1.5 Transformación del esquema conceptual

Una vez diseñada la base de datos, el siguiente paso consiste en transformar el esquema conceptual en estructuras y transacciones descritas en términos del modelo de datos en el cual se base el sistema de gestión de bases de datos que se vaya a utilizar, en nuestro caso MySQL. Para llevar a cabo esta tarea, existen herramientas de software automáticas para el diseño de base de datos. En nuestro caso, es decir, para la base de datos MySQL, existen varias herramientas que generan automáticamente el esquema físico de la base de datos. Después de instalar y probar varias opciones nos decantamos a usar MySQL Workbench, ya que es fácil de usar y por su licencia GNU.

6.1.1.6 MySQL WorkBench

MySQL Workbench es un software creado por la empresa informática Sun Microsystems, esta herramienta permite modelar diagramas de entidad-relación para bases de datos MySQL. Puede utilizarse para diseñar el esquema de una base de datos nueva, documentar una ya existente o realizar una migración compleja.

La aplicación elabora una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real, ingeniería inversa para importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySql Administrator.

MySQL Workbench también puede generar el guión necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5. [WIKI MySQL_Workbench]

Las principales características son:

- Representación visual de tablas, vistas, procedimientos, funciones y claves ajenas.
- Soporta ingeniería inversa.
- Sincronización del modelo con la base de datos.
- Generación de scripts SQL.
- Arquitectura extensible a través de plugins y scripts.
- Exportar como SQL CREATE script.
- Importar modelos de DBDesigner 4.
- Soporta características de MySQL 5.
- Soporta diferentes notaciones para el diagrama entidad relación.
- Generar documentación.

MySQL Workbench se distribuye en dos versiones distintas:

- *MySQL Workbench Community Edition*. Bajo licencia GPL.
- *MySQL Workbench Standard Edition*. Disponible con suscripción de pagos. Incluye características adicionales.

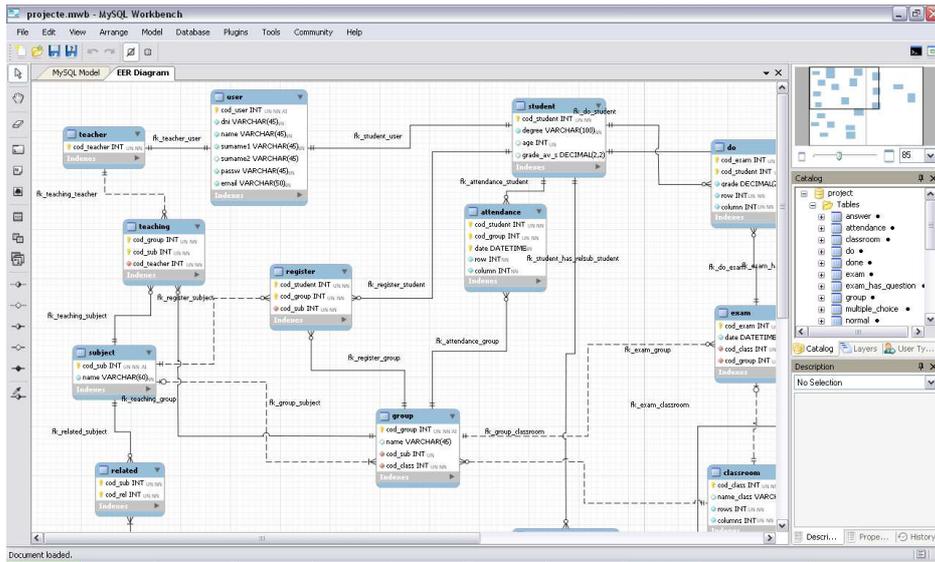


Figura 9. MySQL Workbench

6.2 Diseño de la aplicación Web

Como se ha comentado anteriormente esta segunda parte del diseño del sistema se centra en el diseño de la aplicación Web a través del método OOWS explicado en la sección 3 “Metodologías empleadas” del presente documento.

Es frecuente confundir y no distinguir adecuadamente entre los siguientes sistemas Web: estáticos, estáticos con formularios de entrada, con acceso dinámico a datos, construidos dinámicamente y finalmente las aplicaciones software para la Web o simplemente aplicaciones Web.

Las aplicaciones Web no son sencillas ya que necesitan gestionar esencialmente: [OOWS, 2004]

- Requisitos de Navegación dentro de la aplicación.
- Presentación de información.
- Gestión de información (filtrados, búsquedas,...).
- Tratamiento de usuarios:
 - Gran cantidad de tipos de usuarios.
 - Adaptación / Personalización.
- Ejecución de servicios.
- Además de los requisitos de las aplicaciones “clásicas”.

Por esta razón es necesario seguir entornos de desarrollo Web que:

- Definan un proceso sistemático y estructurado de producción de software.
- Faciliten la especificación de los sistemas usando técnicas basadas en el modelado conceptual.
- Permitan la rápida prototipación.

En definitiva es necesario aplicar Ingeniería Web. La Ingeniería Web propone el desarrollo a partir de modelos conceptuales de la Web, el establecimiento de procesos de desarrollo adecuados, el uso de métodos y técnicas específicas para representar las características Web.

“La Ingeniería Web es la rama de la Ingeniería del Software que estudia los procesos, métodos, técnicas, recursos esenciales para el desarrollo de aplicaciones Web de calidad”. [OOWS, 2004].

6.2.1 Modelos OOWS

En las siguientes secciones se van a presentar los modelos conceptuales del método OOWS

6.2.1.1 Modelos de navegación y presentación

6.2.1.1.1 Diagrama de usuarios

Se definen en el sistema dos clases de usuarios, profesores (teacher) y estudiantes (student). Ambas clases pertenecen al tipo de usuario registrado, por eso son representados con un candado en la cabeza. Estos usuarios necesitan identificarse al conectarse al sistema, habitualmente gestionan la funcionalidad del sistema y la información sensible. Se pueden establecer políticas de personalización individuales. Como se muestra en la siguiente figura no existe ninguna relación de especialización entre ellos.

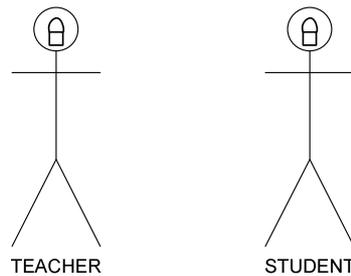


Figura 10. Diagrama de usuarios.

6.2.1.1.2 Mapas navegacionales

Se define un mapa navegacional para cada clase de usuario. Proporcionan la vista global del sistema para cada de usuario (descripción global) definiendo la estructura del Web sitio. Están compuestos de:

- Nodos Navegacionales.
 - Contextos navegacionales: definen contenido de información.
 - Subsistemas: permiten estructurar la navegabilidad dentro del sistema.
- Enlaces Navegacionales: definen una relación de alcanzabilidad entre nodos de navegación.
 - De secuencia: (flechas continuas) definidos implícitamente por las relaciones navegacionales de contexto (en definición de contexto). Llevan información contextual.
 - De exploración: (flechas discontinuas) definidos implícitamente por los contextos de Exploración. No llevan información contextual.

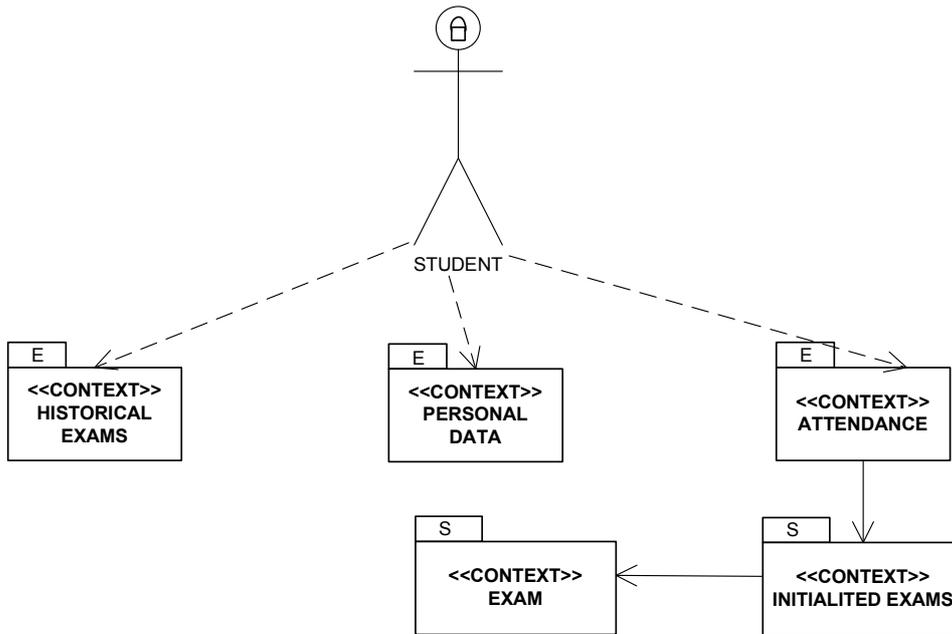


Figura 11. Mapa navegacional para student.

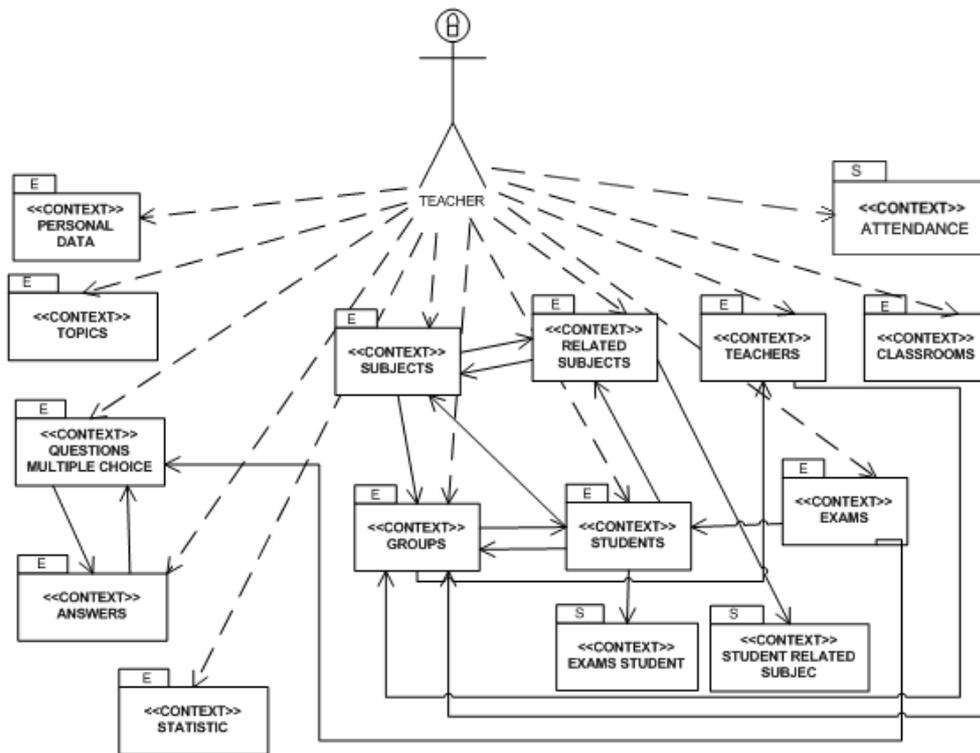


Figura 12. Mapa navegacional para teacher.

6.2.1.1.3 Contextos navegacionales

Hay dos tipos de contextos:

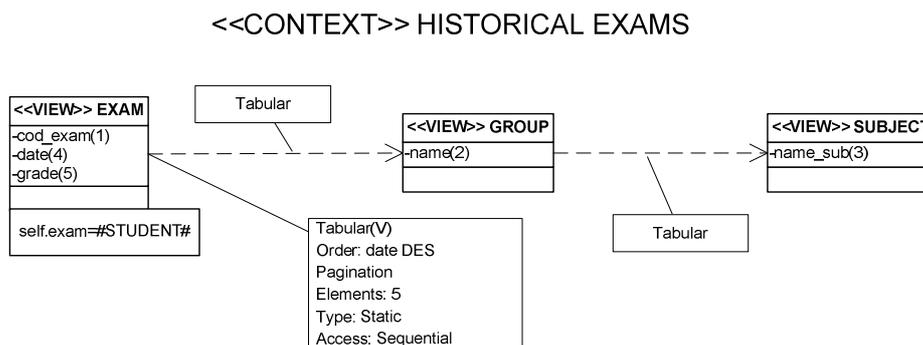
- Exploración: definen los enlaces de exploración del mapa de navegación, son siempre accesibles y son marcados con una E.
- Secuencia: sólo accesibles a través de caminos de navegación preestablecidos, son marcados con una S.

Los contextos navegacionales están compuestos por:

- Clases navegacionales. especifican una recuperación de información (atributos) y acceso a una funcionalidad (operaciones).
 - Directora: clase principal del contexto, toda la exploración de información parte de ella, sólo puede haber una por contexto.
 - Complementarias: “complementan” la información de la clase directora, usan relaciones navegacionales para definirse, pueden haber varias.
- Relaciones navegacionales. enlazan las clases navegacionales, representan un requisito para recuperar información complementaria relacionada.
 - Relaciones de Dependencia Contextual: no definen navegación (flechas discontinuas).
 - Relaciones de Contexto: definen una navegabilidad a un contexto destino llevando información contextual, (flechas continuas).

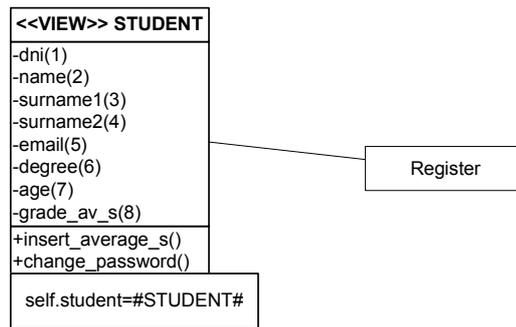
A continuación se muestran los contextos navegacionales para nuestro proyecto.

Contexto *historial exams*: muestra un histórico de los exámenes realizados por el alumno conectado en ese momento, además de para que grupo y asignatura.



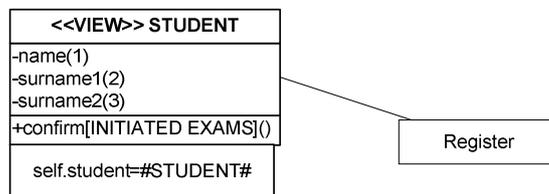
Contexto *personal data*: muestra información personal del alumno conectado. Ofrece los servicios de insertar la nota media del expediente y cambiar la contraseña.

<<CONTEXT>> PERSONAL DATA



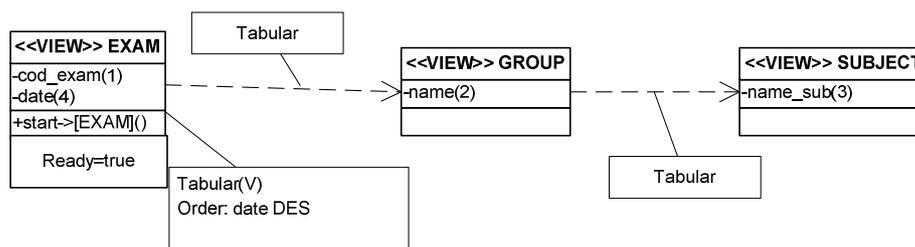
Contexto attendance: este contexto es de entrada de datos, sirve para que el alumno conectado pueda confirmar su asistencia en el aula. Al confirmar la asistencia hay una navegación al contexto *initialited exams*.

<<CONTEXT>> ATTENDANCE



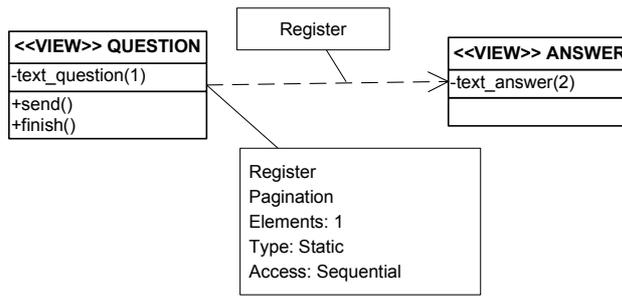
Contexto initialited exams: este contexto es solo accesible a través del contexto asistencia. Una vez el alumno ha confirmado su asistencia llegará a este contexto donde podrá realizar un examen o simplemente navegar a otro contexto en el caso de que su intención solo fuera confirmar la asistencia a alguna clase. Muestra los exámenes iniciados por algún profesor, para que el alumno empiece la realización del examen.

<<CONTEXT>> INITIATED EXAMS



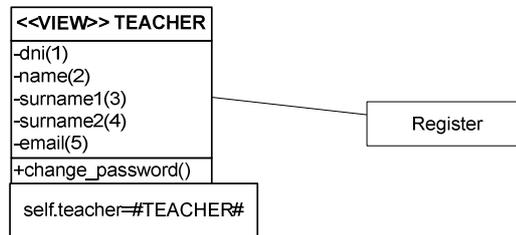
Contexto exam: muestra las preguntas de test del examen y sus respectivas respuestas. Ofrece los servicios de enviar la respuesta elegida y finalizar el examen.

<<CONTEXT>> EXAM



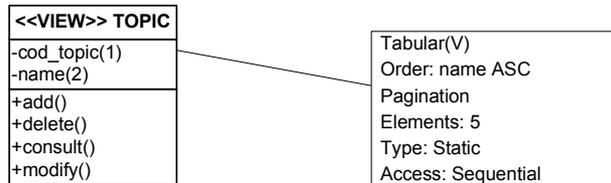
Contexto personal data: muestra información personal del profesor conectado. Ofrece el servicio de cambiar la contraseña.

<<CONTEXT>> PERSONAL DATA



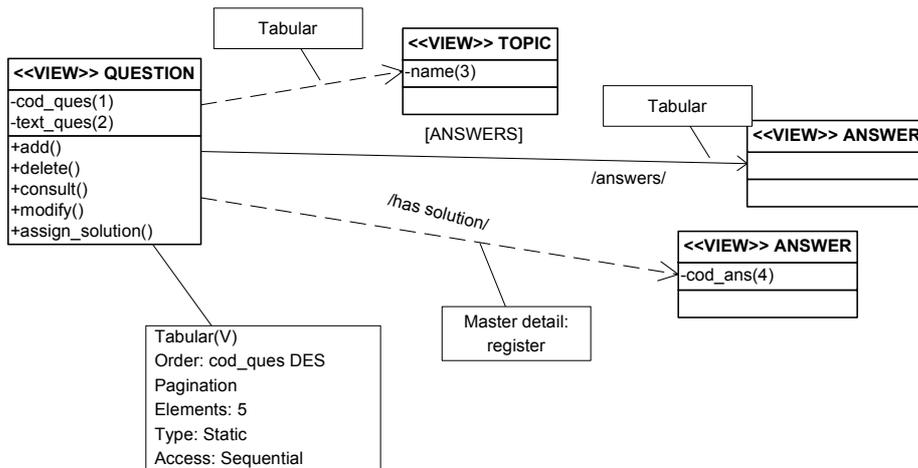
Contexto topics: muestra la información de los temas de las preguntas disponibles en el sistema. Ofrece los servicios de añadir, eliminar, consultar y modificar temas.

<<CONTEXT>> TOPICS



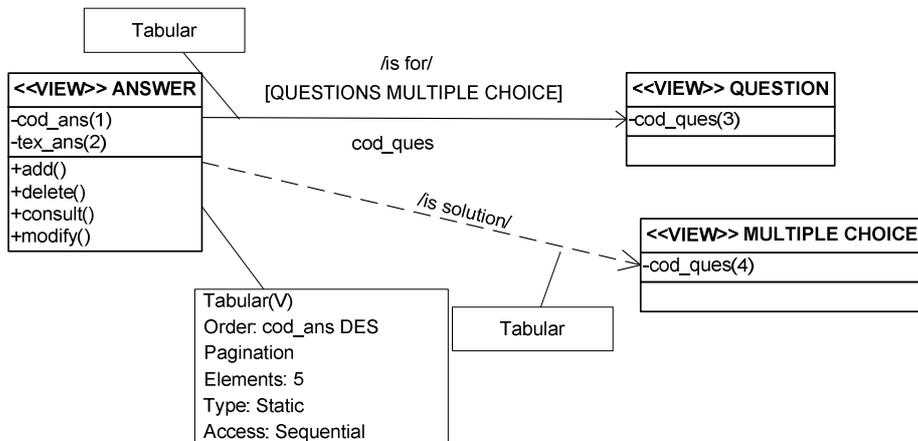
Contexto questions multiple choice: muestra las preguntas de test disponibles en el sistema, tema y la respuesta solución de cada pregunta. Ofrece un enlace (answers) de modo que al seleccionarlo se produce una navegación al contexto answers donde se muestran las respuestas de esa pregunta. Ofrece los servicios de añadir, eliminar, consultar, modificar preguntas de test y asignar solución a la pregunta.

<<CONTEXT>> QUESTIONS MULTIPLE CHOICE



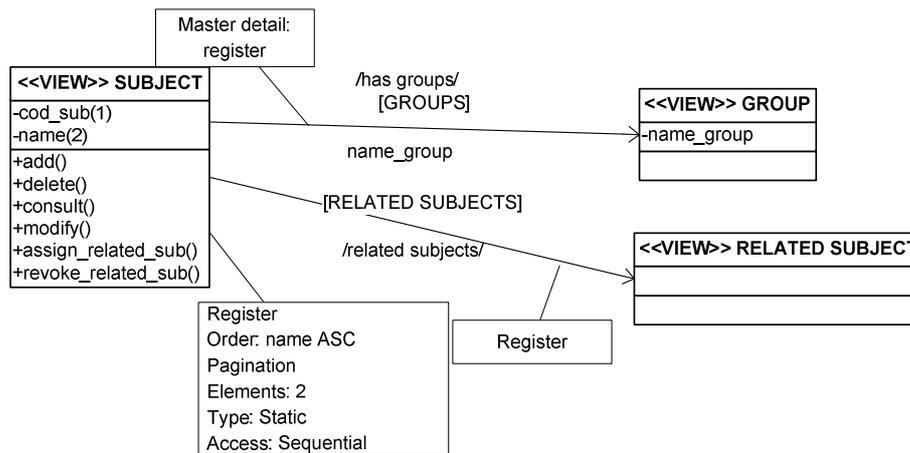
Contexto answers: muestra las respuestas disponibles y si son solución de alguna pregunta. Se puede navegar al contexto *question multiple Choice* seleccionando el código de pregunta para ver la información de la pregunta a la que pertenece la respuesta. Ofrece los servicios de añadir, eliminar, consultar y modificar respuestas.

<<CONTEXT>> ANSWERS



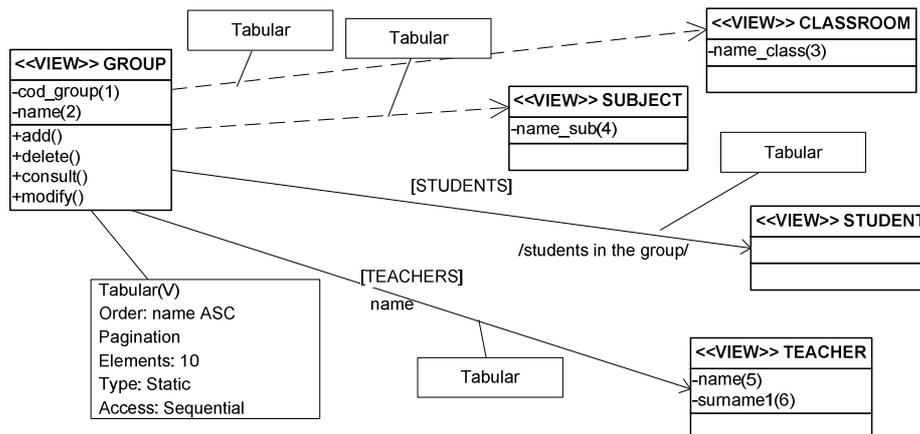
Contexto subjects: muestra la información de las asignaturas disponibles, para que grupos y si tienen asignaturas relacionadas. De modo que al seleccionar un grupo se navegará al contexto *groups* donde se muestra información de ese grupo, también se puede navegar al contexto *related subjects* donde se pueden ver las asignaturas relacionadas (en el caso de que tenga). Ofrece los servicios de añadir, eliminar, consultar y modificar asignaturas, además de asignar y revocar asignaturas relacionadas.

<<CONTEXT>> SUBJECTS



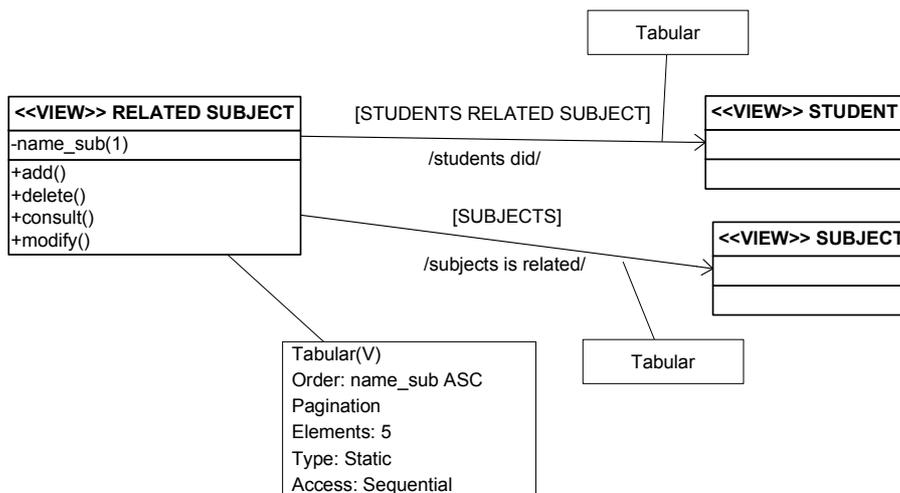
Contexto groups: muestra la información de los grupos, que clase tienen asignada, para que asignatura, que profesor imparte el grupo. Además se puede navegar al contexto *students* donde se pueden ver los alumnos pertenecientes a ese grupo y al contexto *teachers* donde se puede ver la información del profesor. Ofrece los servicios de añadir, eliminar, consultar y modificar grupos.

<<CONTEXT>> GROUPS



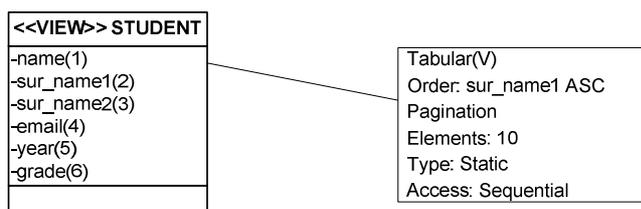
Contexto related subjects: muestra la información de las asignaturas relacionadas con las asignaturas del sistema. Se puede navegar a los contextos *students related subjects* y *subjects* donde se muestra respectivamente la información de los alumnos que cursaron la asignatura relacionada y la información de la asignatura a la que esta relacionada. Ofrece los servicios de añadir, eliminar, consultar y modificar asignaturas relacionadas.

<<CONTEXT>> RELATED SUBJECTS

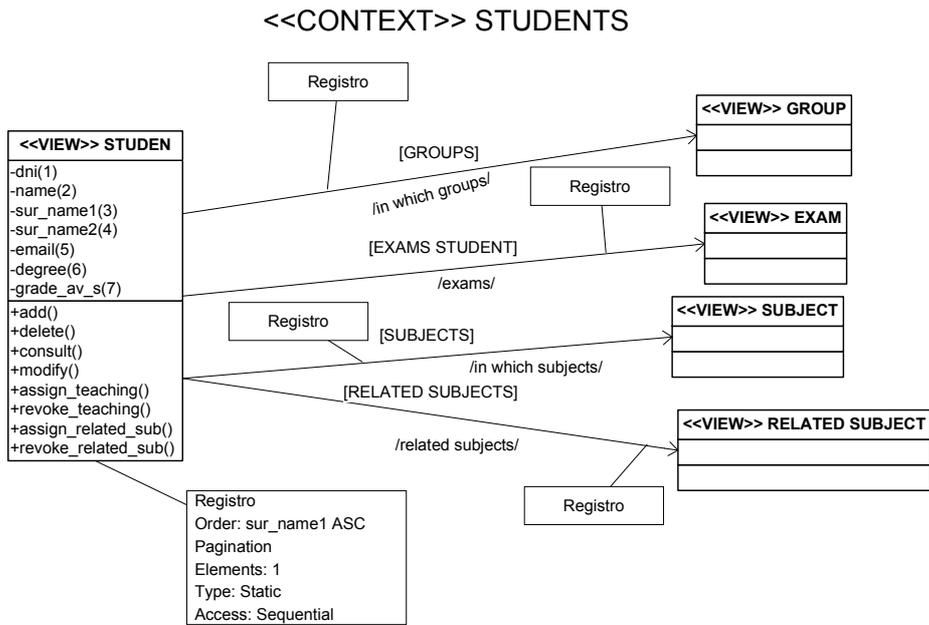


Contexto students related subject: contexto solo accesible desde el contexto *related subjects*, muestra la informaci3n del alumno que curso la asignatura relacionada.

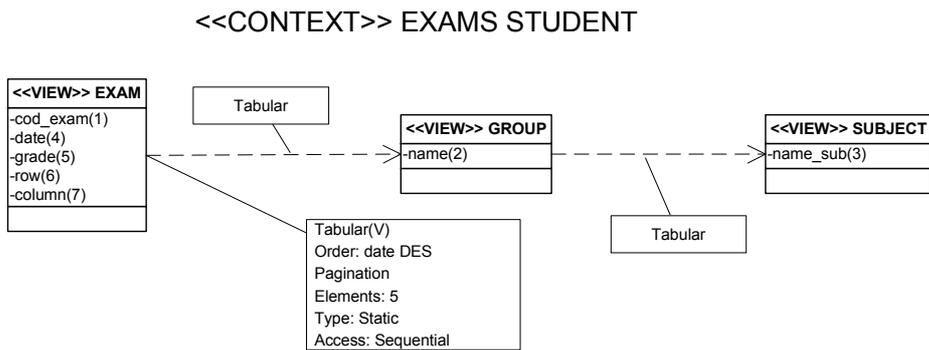
<<CONTEXT>> STUDENTS RELATED SUBJECT



Contexto students: muestra los alumnos en el sistema, se puede navegar al contexto *groups* donde se ven los grupos del alumno, al contexto *exams student* donde se ven los exámenes que ha realizado el alumno, al contexto *subjects* donde se ven las asignaturas que tiene el alumno y al contexto *related subjects* donde se ven las asignaturas relacionadas del alumno (si tiene). Ofrece los servicios de ańadir, eliminar, consultar y modificar alumnos, adem1s de asignar, revocar docencia para los alumnos y asignar, revocar asignaturas relacionadas.

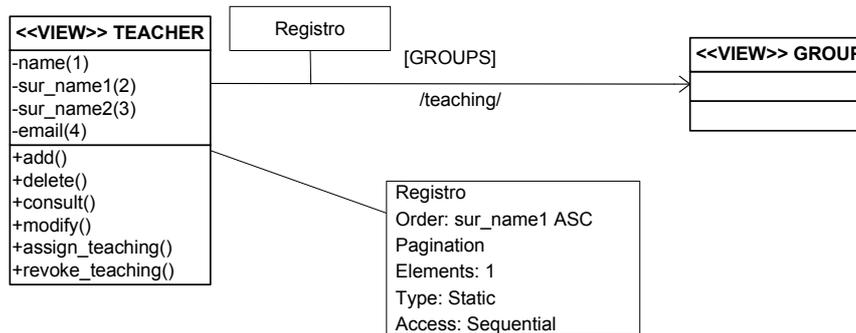


Contexto exams student: contexto solo accesible desde el contexto *students*, muestra un histórico de los exámenes realizados por el alumno seleccionado, además de para que grupo y asignatura.



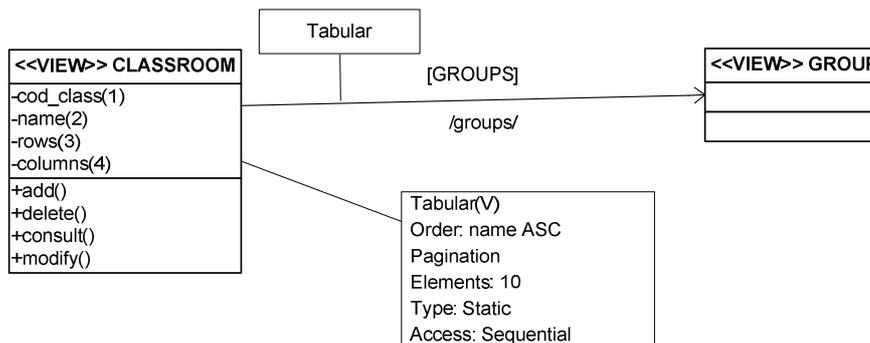
Contexto teachers: muestra los profesores en el sistema, se puede navegar al contexto *groups* donde se ven los grupos del profesor. Ofrece los servicios de añadir, eliminar, consultar y modificar profesores, además de asignar y revocar docencia para los profesores.

<<CONTEXT>> TEACHERS



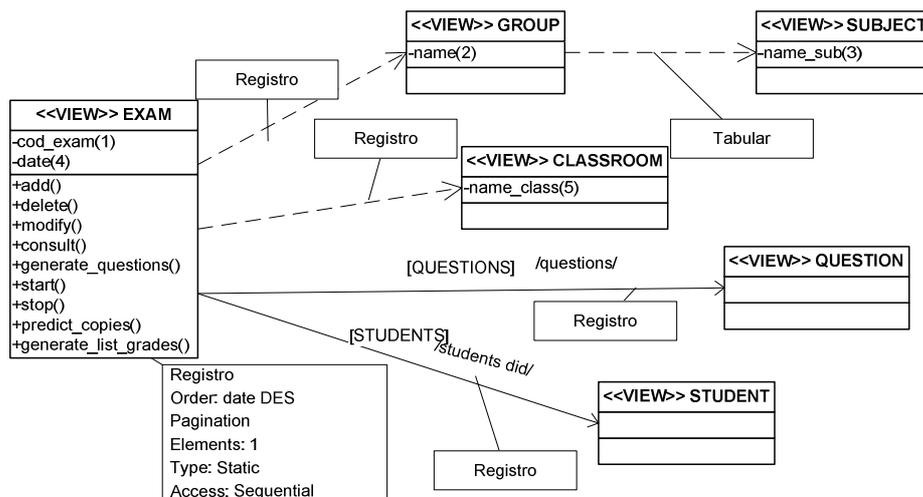
Contexto classrooms: muestra la información de las aulas, se puede navegar al contexto *groups* para ver los grupos asignados a cada clase. Ofrece los servicios de añadir, eliminar, consultar y modificar aulas.

<<CONTEXT>> CLASSROOMS



Contexto exams: en este contexto el profesor puede generar preguntas, iniciar exámenes, parar exámenes, además de predecir probabilidad de copias y generar listados de notas de los exámenes ya terminados. Muestra información de los exámenes, en que clase se realizaron, para que grupo, para que asignatura. Además se puede navegar al contexto *students* para ver los alumnos que han realizado ese examen y al contexto *questions* para ver que preguntas tiene el examen. Ofrece los servicios de añadir, eliminar, consultar y modificar exámenes.

<<CONTEXT>> EXAMS



6.2.1.2 Mecanismos de acceso a la información

Como ya se ha comentado en la sección 3 “Metodologías empleadas” del presente documento, existen dos mecanismos de acceso a la información: índices y filtros. Para la realización del presente proyecto se ha decidido no tener en cuenta mecanismos de acceso a la información en los modelos, ya que no se dispone de suficiente información en la especificación de requerimientos software.

Aunque si serán considerados e implementados cuando sea necesario en la fase de implementación. Para orientarnos un poco hemos de tenerlos en cuenta cuando: el contexto sea de exploración y la población del contexto sea elevada, es decir, existen muchas instancias de objetos de la clase directora. Teniendo en cuenta esto para nuestro proyecto podríamos pensar en los siguientes contextos: historical exams, questions multiple choice, answers, students, exams.

6.3 Diseño del sitio Web

Una vez realizado el análisis del sitio Web a desarrollar en el capítulo 5 del documento, procederemos a aplicar las conclusiones extraídas de dicho análisis a nuestro sitio Web y realizaremos la toma de decisiones que afectará a la implementación de nuestro sitio Web.

El objetivo principal es construir un sitio Web que obedezca de la mejor manera posible a los criterios de usabilidad estudiados.

Una de las cosas que más ha cambiado en AWAD es la interfaz gráfica, puesto que se ha intentado crear una nueva interfaz que resulte simple, sencilla, y sobre todo funcional. Al contrario que en el caso de la base de datos, la interfaz gráfica ha sido totalmente rediseñada.

A continuación se muestran las decisiones tomadas en la creación del nuevo portal Web.

6.3.1 Diseño de la página

6.3.1.1 Distribución en pantalla (Layout)

El layout es el plano o distribución de la información a través de la pantalla, esta distribución tiene dos objetivos importantes:

- Facilitar la lectura.
- Hacer la página atractiva a la vista y organizada.

El diseño de páginas Web está dividido en cinco bloques: cabecera, navegación, menú lateral, contenidos, y pie de página. Visualmente, los bloques se disponen en varias filas y columnas. Por este motivo, hace varios años la estructura de las páginas HTML se definía mediante tablas.

El desarrollo de CSS nos permite realizar los mismos diseños utilizando divs en vez de tablas HTML. Las principales ventajas de diseñar la estructura de las páginas web con CSS son las siguientes:

- **Mantenimiento:** una página diseñada exclusivamente con CSS es mucho más fácil de mantener que una página diseñada con tablas. Cambiar el aspecto de una página creada con CSS es tan fácil como modificar unas pocas reglas en las hojas de estilos. Sin embargo, realizar la misma modificación en una página creada con tablas supone un esfuerzo muy superior y es más probable cometer errores.
- **Accesibilidad:** las páginas creadas con CSS son más accesibles que las páginas diseñadas con tablas. De hecho, los navegadores que utilizan las personas discapacitadas (especialmente las personas invidentes) pueden tener dificultades con la estructura de las páginas complejas creadas con tablas HTML. No obstante, diseñar una página web exclusivamente con CSS no garantiza que la página sea accesible.
- **Velocidad de carga:** el código HTML de una página diseñada con tablas es mucho mayor que el código de la misma página diseñada exclusivamente con CSS, por lo que tarda más tiempo en descargarse. En cualquier caso, si el usuario accede al sitio con una conexión de banda ancha y la página es de un tamaño medio o reducido, las diferencias son casi imperceptibles.
- **Semántica:** aunque resulta obvio, las tablas HTML sólo se deben utilizar para mostrar datos cuya información sólo se entiende en forma de filas y columnas. Utilizar tablas para crear la estructura completa de una página es tan absurdo como utilizar por ejemplo la etiqueta `<u1>` para crear párrafos de texto.

El layout usado por AWAD se muestra a continuación.

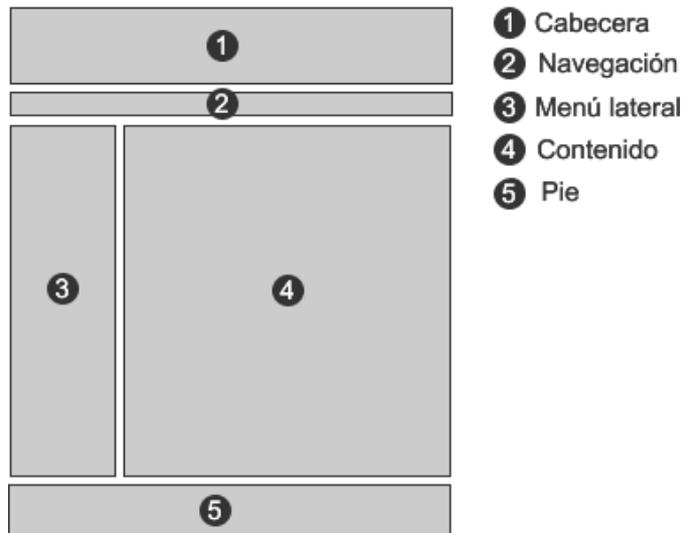


Figura 13. Layout.

6.3.1.2 Colores

Los internautas utilizan sistemas operativos diferentes y navegadores diferentes, y en lo que al color respecta, cada uno de estos posee una paleta de colores distinta.

A la hora de realizar el diseño se ha procurado utilizar los mismos colores en todas las páginas para que el diseño sea más uniforme y no causar asombro en el usuario. Si de una página a otra cambian los colores completamente, un usuario puede pensar que ha sido reenviado a otra web, es decir, puede sentirse perdido. En nuestro caso, hemos trabajado siempre con tonos blancos y azules.

6.3.1.3 Sistema de navegación

Uno de los requisitos era dotar a la página de un buen sistema de navegación ya que es imprescindible en toda página Web. Es una parte importante de la organización de la estructura de la Web.

ISO/IEC 9126 define la usabilidad como:

“La capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso”

Principios básicos en los que se basa la usabilidad:

- **Facilidad de Aprendizaje:** facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. Está rela-

cionada con la sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.

- **Flexibilidad:** relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
- **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.

Con este nuevo sistema de navegación nuestro objetivo es:

- Que el usuario sea capaz de moverse con soltura y facilidad por las distintas páginas del sitio.
- Que encuentre lo que busca rápidamente.
- Que no se pierda yendo de un enlace a otro sin saber donde está.
- Que no quede colgado en una página concreta sin poder navegar por otras o volver atrás



6.3.1.4 Menú lateral

El menú de una página Web es la principal herramienta de navegación que le podemos facilitar al visitante para que encuentre lo que busca. Es imprescindible para que las personas encuentren las demás páginas de la Web

En nuestro caso el menú lateral de navegación vertical que es básicamente una lista de enlaces a las diferentes páginas o secciones de la Web.

6.3.1.4.1 La importancia de los iconos

Como se trata de un menú con muchas opciones lo dotaremos de una serie de iconos intuitivos en cada una de las opciones.

Los iconos son esos pequeños gráficos que nos ayudan a entender la información más fácilmente.

Nos decantamos por el uso de iconos por dos motivos, por una parte las ventajas de usabilidad que nos da el tener una buena iconografía en un sitio

Web y por otro lado, también tienen un papel muy importante en la estética del sitio, hacen que el sitio se vea más diseñado más completo, le dan personalidad al sitio y forman parte de la identidad gráfica del mismo.



Figura 14. Menú de Navegación y ejemplo de iconos.

6.3.1.5 Ruta de acceso o migas

Es frecuente que aunque el usuario pueda navegar con facilidad por las páginas de tu sitio no sepa exactamente donde está. Para evitar esto es muy útil incluir el camino que se sigue desde la página principal, pasando por las secciones (principales) hasta la página concreta en la que se encuentra el usuario. Un ejemplo es:

Estás en: Inicio > Grupos > Alumnos para el grupo: PL- 2

Figura 15. Estás en:

Es conveniente que cada parte de la ruta tenga su enlace correspondiente, de forma que el usuario pueda ir a esa categoría si lo desea. Sin el enlace a cada sitio, la ruta de acceso pierde sentido ya que le dice al usuario donde está pero no le da alternativas.

6.3.1.6 Enlazar siempre al índice

Una de las formas de evitar páginas huérfanas y que mejora sin duda la navegabilidad de una Web es incluir en cada página del sitio un enlace al índice.



Parece bastante extendido el enlazar a la página principal del sitio mediante el banner o logo de cabecera. Así, el usuario siempre tiene un recurso para volver al principio y buscar lo que necesita.

6.3.2 Estructura del sitio

6.3.2.1 Página de inicio

La página principal de AWAD esta diseñada para que un usuario pueda identificarse como usuario ya existente o registrarse en la aplicación. En este apartado, todos los usuarios son considerados visitantes hasta que realizan el inicio de sesión.

Las opciones de inicio de sesión y registro aparecen en este caso en el centro de la pantalla ya que no se puede empezar a usar la aplicación Web sin realizar una de estas dos acciones. En este menú central se muestran los campos de usuario y contraseña para poder iniciar sesión un enlace a la página de registro.

Figura 16. Interfaz página de inicio.

6.3.2.2 Página de usuario Profesor y Alumno

Una vez iniciada la sesión de un usuario se puede contemplar la estructura de la página indicada anteriormente, en la que en la cabecera encontramos el logo en la parte izquierda y el nombre del usuario en la izquierda.

En el cuerpo encontramos en un lado el menú, que variará en función de si el inicio de sesión fue por parte de un profesor o alumno, y en el lado izquierdo el contenedor con la información de la página.

6.3.3 Diseño del contenido

AWAD dispone de una interfaz gráfica agradable e intuitiva para el usuario, pudiendo fácilmente navegar entre las distintas secciones mediante un menú de pestañas, además el uso de la tecnología AJAX hace que mejore la experiencia del usuario.

A continuación se muestran las decisiones tomadas en relación del contenido:

6.3.3.1 Gráficas con jQuery

Como una parte importante de la aplicación se basa en la creación de gráficas, nuestra intención era incluir una biblioteca de gráficos que ofreciese una forma fácil de añadir gráficos interactivos.

Después de buscar, instalar y probar muchas opciones, finalmente me decante por *Highcharts* ya que soportaba muchos tipos de gráficas y su aspecto es elegante y a la vez sencillo y agradable. [Highcharts, 2010]

Después de usar y trabajar con graficas Highcharts, mi opinión es que a sido una elección perfecta para este tipo de portal.

A continuación veremos un ejemplo de uso de esta librería, el ejemplo muestra una grafica lineal que corresponde a la página de estadísticas de examen.

6.3.3.2 Ejemplo de uso

Lo primero es obtener la información que mostrara la grafica, para ello realizaremos las consultas necesarias a la base de datos mediante PHP e iremos construyendo un string con dicha información.

Veamos un ejemplo de cómo obtener la información de las preguntas de un examen y como vamos formando el string con el formato necesario.

Primero hacemos la consulta con la query necearía para obtener los datos y ejecutamos la consulta:

```
//preguntas del examen
$query_registros_preguntas = sprintf("SELECT q.cod_ques, q.text_ques, t.name_topic FROM
question q NATURAL JOIN topic t JOIN exam_has_question e ON (e.cod_ques = q.cod_ques)
WHERE e.cod_exam = %s ORDER BY q.cod_ques", GetSQLValueString($cod_exam, "int"));

$registros_preguntas = mysql_query($query_registros_preguntas, $conexion) or
die(mysql_error());

$totalRows_registros_preguntas = mysql_num_rows($registros_preguntas);
```

Creamos un array asociativo con los datos que nos devolvió la consulta. Dicho array contiene las tuplas de las preguntas del examen e iremos recorriéndolo para crear un variable con la información en el formato necesario.

```
while($row_registros_preguntas = mysql_fetch_assoc($registros_preguntas))
{
    ...
    $a_preguntas="P".$row_registros_preguntas['cod_ques'].",
".$a_preguntas;
    ...
}
```

Este es el código javascript que se encargará de dibujar la gráfica.

```

$(document).ready(function() {
  var chart;
  chart = new Highcharts.Chart({
    chart: {
      renderTo: 'containerpreguntas',
      defaultSeriesType: 'column',
      borderWidth: 1
    },
    title: {
      text: 'Preguntas del examen'
    },
    subtitle: {
      text: 'Número de acierto y fallos'
    },
    xAxis: {
      categories: [php echo $a_preguntas ?&gt;]</b
    },
    yAxis: {
      min: 0,
      title: {
        text: 'Cantidad'
      }
    },
    legend: {
      layout: 'horizontal',
      borderWidth: 1,
      backgroundColor: '#FFFFFF'
    },
    tooltip: {
      formatter: function() {
        return "+ this.x +": "+ this.y +"" + this.series.name;
      }
    },
    plotOptions: {
      column: {
        pointPadding: 0.2,
        borderWidth: 0
      }
    },
    series: [
      {
        name: 'Aciertos',
        data: [php echo $a_aciertos ?&gt;]</b
      }, {
        name: 'Fallos',
        data: [php echo $a_fallos ?&gt;]
      }, {
        name: 'No contestadas',
        <bdata: [php echo $a_no_contestadas ?&gt;]</b
      }
    ]
  });
});

```

Por ultimo hay que especificar la ubicación que tendrá la grafica en la página, para ello crearemos un *div* y le asignaremos el *id* que previamente hemos indicado en la opción **renderTo: 'containerpreguntas'**.

```
<div id = 'containerpreguntas'></div>
```

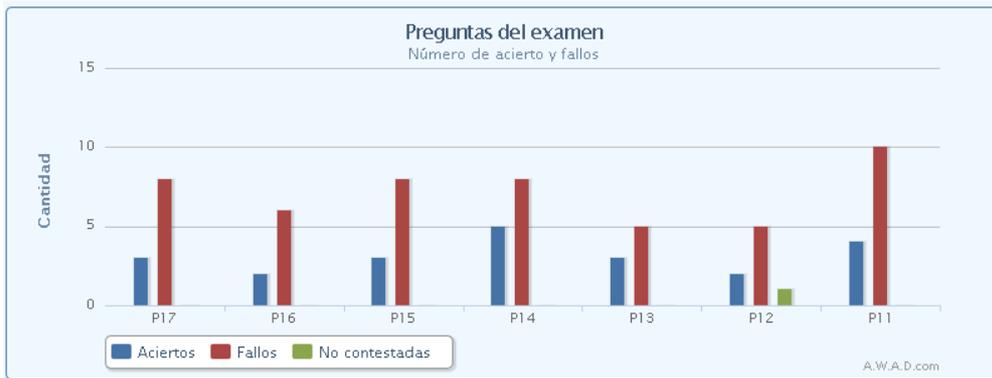


Figura 17. Gráfica estadísticas examen.

6.3.3.2.1 Tooltip Labels

Una ventaja que tiene este tipo de graficas es que pueden mostrar un texto tooltip con información sobre cada punto y series. Observamos en la siguiente imagen que cuando se coloca el ratón sobre una barra se muestra la información pertinente a esta.

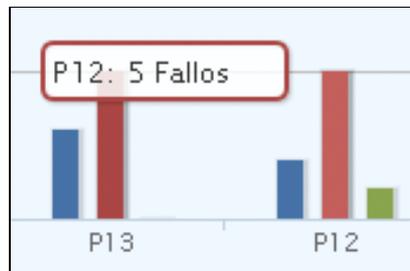


Figura 18. Gráfica: Tooltip Labels.

6.3.3.3 Tablas para listados de información

Un punto fuerte y muy usado en nuestra aplicación son los listados de información, por este motivo decidimos cambiar por completo el aspecto de las tablas en las cuales se muestran los datos, de esta forma se intenta conseguir una tabla interactiva mejorada, añadiendo controles avanzados de interacción con las tablas HTML.

The screenshot shows the A.W.A.D. (Aplicación Web de Apoyo a la Docencia) interface. At the top, the user is logged in as Francisco Bosch. The main content area displays a table of student information with the following data:

Cód	Dni	Nombre	Email	Consultas	Opciones
10	11111111-D	Diego Insa Lopez	diego@ei.upv.es	Consultas	Más Opciones
12	11111111-L	Luis Noseq Nomeacuerdo	luis@ei.upv.es	Consultas	Más Opciones
14	11111111-M	Javier Motos Noseq	motos@ei.upv.es	Consultas	Más Opciones
15	11111111-O	Omar Pastor Nomeacuerdo	omar@ei.upv.es	Consultas	Más Opciones
16	11111111-F	Fran Alonso Jaen	franky@ei.upv.es	Consultas	Más Opciones
17	11111111-Fa	Fabio Carmona Antoñanzas	Fabio@ei.upv.es	Consultas	Más Opciones
18	11111111-cc	Carlos Pastor Bayo	carlos@ei.upv.es	Consultas	Más Opciones
19	11111111-Dd	David Pastor Bayo	david@ei.upv.es	Consultas	Más Opciones
3	1111111V	Victor Mateo Torres	victor@ei.upv.es	Consultas	Más Opciones
5	1111111P	Paco Pardo Vesces	paco@ei.upv.es	Consultas	Más Opciones

The interface also includes a search bar at the top right and a sidebar menu on the left with options like 'Inicio', 'Temas', 'Preguntas', 'Respuestas', 'Asignaturas', 'Asignaturas relacionadas', 'Grupos', 'Profesores', 'Alumnos', 'Atlas', 'Exámenes', 'Estadísticas', and 'Asistencias'.

Figura 19. Tabla de listados de información.

6.3.3.3.1 Mejoras proporcionadas

En este apartado vamos a ver un ejemplo de las tres principales ventajas por las que nos decantamos a incorporar este tipo de graficas en nuestra aplicación.

Ordenación por campos: En un principio nos pareció interesante la opción de poder ordenar la información por los diferentes campos, este tipo de graficas nos permite la ordenación dinámica de forma sencilla y por cualquiera de sus campos, basta con pinchar sobre cabecera.

Búsquedas dinámicas: En esta captura de pantalla se puede observar la zona de búsqueda marcada por cuadro rojo, las búsquedas se ejecutan de forma dinámica y a medida que vamos escribiendo la tabla se va autocompletando con los nuevos resultado. Es una forma sencilla, eficaz y sobretodo rápida de hacer búsquedas.

The screenshot shows the search results for the query 'vic'. The search bar is highlighted in red. The table displays the following result:

Cód	Dni	Nombre	Email	Consultas	Opciones
3	1111111V	Victor Mateo Torres	victor@ei.upv.es	Consultas	Más Opciones

The interface also shows the search bar with the text 'Buscar: vic' and a status message at the bottom: 'Mostrando 1 a 1 de 1 entradas (filtered from 13 total entries)'.

Figura 20. Tabla: Resultados de búsqueda.

Paginación de longitud variable: Otra opción que nos brindan estas tablas es la paginación de longitud variable, muy útil cuando tenemos que mostrar gran cantidad de datos, basta con seleccionar el número de ítems que deseamos mostrar en el desplegable que se muestra a continuación.



The image shows a table with a pagination control at the top. The control includes a dropdown menu labeled 'Mostrar' with the value '10' selected, followed by the text 'entradas'. Below the table, a dropdown menu is open, showing options for 10, 25, 50, and 100. The table itself has columns for 'Cód', a numerical value, and a date-like string.

Cód		
10	10	11-D
12	25	11-L
14	50	11-M
15	100	11-O
16	11111111	11-F

Figura 21. Tabla: Paginación variable.

6.3.3.4 Ventanas modales

Las ventanas modales son una nueva tendencia de diseño de 2010 y son una buena opción para no interrumpir la interacción del usuario con la Web, posibilitando ofrecerle otra información sin dejar la que estaba viendo en ese momento. Son, básicamente, la segunda generación de los pop-ups.

Me pareció interesante añadir esta funcionalidad a la aplicación no tanto por la funcionalidad que desempeñan en la Web, si no por motivos de aprendizaje ya que me parecía útil aprender a usar esta nueva modalidad cada vez más en uso.

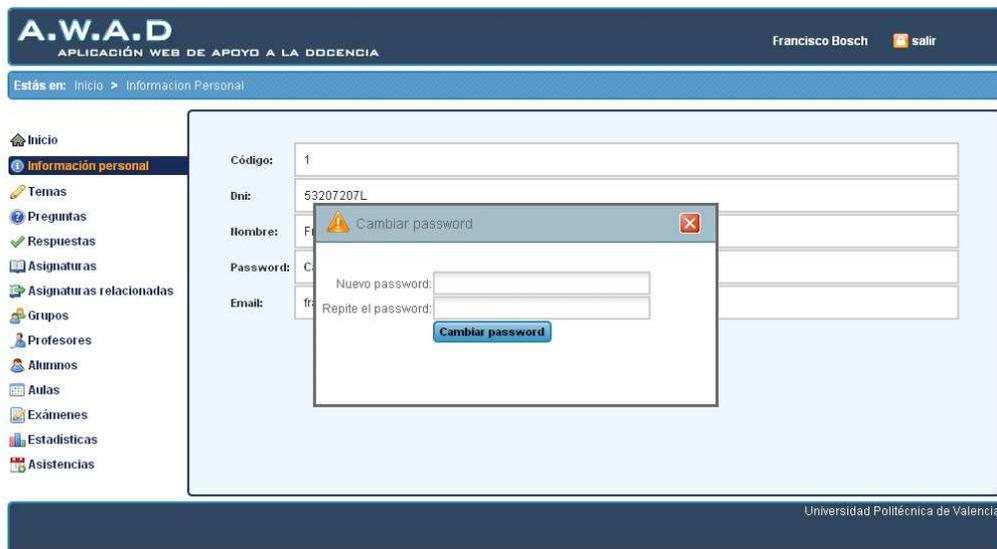


Figura 22. Ventana modal.

6.3.3.5 Exámenes

La parte de evaluación ha sido mejorada en la aplicación para ofrecer al alumno una forma más sencilla para la correcta realización del examen, cuando se confirma la asistencia a un grupo, el sistema buscará si existen exámenes iniciados para dicho grupo y se le comunicará al alumno mediante la página exámenes iniciados.

EXÁMENES INICIADOS				
<p>Nota: Para poder acceder al examen primero debes confirmar tu asistencia en el grupo.</p>				
Código	Fecha	Grupo	Asignatura	Opciones
13	11/08/2010	PL- 2 8	HAD	Iniciar examen

Figura 23. Exámenes iniciados.

Una vez el alumno decida empezar el examen observará que se ha incorporado un cuadro informativo con información sobre el estado actual del examen, el alumno sabrá en todo momento el número de preguntas enviadas y el estado actual de examen.

The screenshot shows the exam interface for the group 'PL-2 - HAD'. It features an information panel on the left and a question box in the center. The information panel displays: Preguntas del examen: 7, Preguntas enviadas: 1, and Estado: Sin terminar. The question asks: '¿CUÁL DE LAS SIGUIENTES NO ES UNA RESTRICCIÓN INHERENTE AL MODELO RELACIONAL?' with four multiple-choice options (A, B, C, D). Below the question is an 'Enviar pregunta' button. At the bottom of the interface, there is a 'Finalizar examen' button and a navigation bar showing '2 a 2 de 7'.

Figura 24. Exámenes.

6.3.3.6 Predicción de copias

En este apartado hablaremos de la funcionalidad que ofrece el sistema de predicción de copias.

Con la finalidad de evitar posibles copias, las preguntas de examen serán ordenadas adecuadamente dependiendo de la posición del alumno en el aula, de esta forma cada alumno tendrá las preguntas ordenadas en diferentes posiciones. Además, el sistema intentará predecir la probabilidad de copia de un alumno con respecto a los alumnos adyacentes, teniendo en cuenta las respuestas seleccionadas y el instante de tiempo en el que se han seleccionado las respuestas, de esta forma se sabrá que alumno copio de los dos.

Para que el profesor tenga la mayor información a la hora de predecir copias, el sistema comparará los exámenes de una forma exhaustiva. Se comprobarán los fallos, aciertos y preguntas no contestadas que coincidan en ambos exámenes.

Los porcentajes que el sistema tiene en cuenta para avisar de una posible copia son los siguientes:

- Si los fallos idénticos son del 60% o superior.
- Si la totalidad del examen es del 70% o superior.
- Si los fallos idénticos + las preguntas no contestadas con del 65% o superior.

Estos parámetros podrán ser modificados desde código, queda pendiente hacer una interfaz para cambiar dichos valores.

Se pretende que la forma en la que el sistema muestre las predicciones sea sencilla y eficaz para ayudar al profesor. A continuación se muestra una captura de la interfaz.

En la parte izquierda de la figura 27 el cuadro rojo indica que se ha detectado una posible copia, es decir se han superado los valores mencionados anteriormente. Por otro lado observamos en la parte derecha los fallos idénticos que han tenido ambos alumnos y quien contestó más tarde.



Figura 25. Interfaz de predecir copias.

Para evitar que un alumno realice el examen desde otra aula o desde su propia casa, el profesor podŕ ver en todo momento la disposici3n del aula y d3nde esta sentado cada alumno, de esta forma el profesor tendŕ constancia de los alumnos que hay en el aula y los que est́n haciendo el examen.

Colocando el rat3n sobre la silla aparece un *tooltip* con el nombre del alumno sentado en dicha silla, es una forma ŕpida de ver donde est́ sentado cada alumno.

The screenshot shows a software interface for classroom management. At the top, it displays the course '8 - PL- 2 - HAD' and the date '2010/08/23'. The interface is divided into two main sections:

- Student List:** A table with columns 'Alumno', 'Fila', and 'Columna'. It lists eight students with their respective row and column numbers. A 'Confirmar asistencia' button is located at the bottom of this list.
- Classroom Grid:** A 6x6 grid representing the classroom layout. The grid is titled 'Aula: DSIC-8 [Laboratorio]'. Each cell in the grid contains a small computer icon. Some cells are highlighted in red, indicating where students are seated: (1,1), (1,2), (1,5), (2,2), (2,3), (3,2), and (3,5). These red cells correspond to the 'Fila' and 'Columna' values in the student list.

Alumno	Fila	Columna
Fran Alonso Jaen	1	2
Diego Insa Lopez	1	1
Victor Mateo Torres	2	2
Omar Pastor Nomeacuerdo	2	3
Carlos Pastor Bayo	1	5
David Pastor Bayo	3	5
Lucia Revert Loquesea	3	2

Figura 26. Ver asistencia

CAPÍTULO 7

7 Implementación e Integración

En los apartados anteriores de esta memoria se ha analizado, diseñado y detallado la aplicación Web a ser desarrollada en el presente proyecto. Con los resultados obtenidos en las etapas anteriores se pasa a implementar la aplicación Web que será la solución al proyecto.

En esta sección se detallará cómo se ha desarrollado la aplicación Web y qué programas han sido utilizados.

Para la realización de las distintas partes de la aplicación Web se han utilizado diferentes herramientas tales como Adobe Dreamweaver CS4 y Microsoft Visio. Para la gestión de la base de datos se ha utilizado PHPMyAdmin y MySQL Query Browser para la realización de procedimientos y consultas MySQL.

7.1 Tecnologías

7.1.1 HTTP

Se trata del protocolo más usado en Internet y el encargado de realizar las transacciones archivos de la Web. Se basa en el método de petición/respuesta, por lo que el cliente (navegador Web) realiza una solicitud y el servidor le responde con el resultado correspondiente. Estas respuestas son, habitualmente, documentos de texto en lenguaje HTML que el cliente se encarga de interpretar y mostrar al usuario final.

7.1.2 CSS

Es un lenguaje que establece la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). Este lenguaje pretende separar la estructura de un documento de su presentación, para ello puede incluirse el código en el propio documento o en un documento externo enlazado por el documento principal.

En el caso de AWAD, las hojas de estilo se encuentran en documentos externos, por lo que se gana en eficiencia y comprensión del código para futuras modificaciones del mismo.

Desde las CSS se ha diseñado todo el apartado visual de AWAD, tratando de buscar en todo momento una interfaz intuitiva, fácil y con una estética agradable para el usuario.

7.1.3 PHP

Se trata un lenguaje de programación interpretado, esto quiere decir que no se compila, sino que existe un intérprete que se encarga de ejecutar el código y sus instrucciones. PHP está pensado para la creación de páginas Web dinámicas, paginas del lado del servidor, donde se encuentra el intérprete. El código en PHP puede ser incrustado directamente dentro de código HTML (o XHTML). PHP es el esqueleto de AWAD. Todas las páginas son generadas dinámicamente a partir del código PHP. Son muchas las razones por las que elegimos PHP para la elaboración de AWAD, pero las más importantes son:

- Se trata de un lenguaje diseñado principalmente para la realización de aplicaciones Web, por lo que las tareas más habituales en el desarrollo de estas aplicaciones, pueden hacerse también en AWAD de manera más rápida, fácil y efectiva.
- Es multiplataforma y no propietario. Se puede ejecutar su código, sin ningún problema, en cualquier servidor intérprete de PHP, algo que hace que aumente la portabilidad de AWAD.
- Se encuentra en el entorno open source, por lo que se puede encontrar infinidad de recursos como manuales, tutoriales y scripts de ejemplos.
- A parte de todo esto, PHP hace las cosas manera rápida y efectiva.

En AWAD se puede comprobar el uso de PHP en todas partes, desde el registro de un nuevo usuario hasta la búsqueda y listado con accesos a la base de datos.



Figura 27. Esquema de funcionamiento de PHP.

7.1.4 MySQL

MySQL es un sistema de administración de bases de datos para bases de datos relacionales, por lo que hace uso de varias tablas para almacenar la información. Destaca por la facilidad de adaptación a los distintos entornos de desarrollo (como PHP en el caso de AWAD) y su integración en los diferentes sistemas operativos (Windows durante el desarrollo de AWAD). Su condición de open source permite su uso de forma gratuita y la modificación de su código fuente con total libertad (no resultado necesario para el desarrollo de AWAD).

A estas alturas las razones por las que elegimos MySQL para la gestión de la base de datos de AWAD ya deberían estar claras pues se integra a la perfección con PHP y Windows, trabaja con bases de datos relacionales como la de AWAD y además es fácil, rápido, eficiente y muy robusto.

7.1.5 AJAX

AJAX es una técnica de desarrollo Web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente (en el navegador del usuario) mientras se mantiene la comunicación de manera asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas completamente, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

AJAX no es una tecnología en sí, si no que se trata de una combinación de otras cuatro tecnologías:

- XHTML y CSS para mostrar la información y su diseño.
- Document Object Model (DOM), al que se accede a través de JavaScript para mostrar e interactuar, de forma dinámica, con la información presentada.
- El objeto XMLHttpRequest para realizar el intercambio de datos con el servidor de forma asíncrona.
- XML como formato para la transferencia de datos solicitados al servidor.

AJAX se encuentra presente en muchos apartados de AWAD en los que, con motivo de agilizar la interacción entre el sistema y el usuario, se muestra información sin que el usuario detecte ningún cambio en la fluidez de la página.

Aquí se ve un ejemplo de una función AJAX que borra de la base de datos un aula.

```
function borrar($cod,$id)
{
    $objResponse = new xajaxResponse();

    conectar();
    $deleteSQL = sprintf("DELETE FROM classroom WHERE cod_class=%s",$cod);
    $Result1 = mysql_query($deleteSQL) or die(mysql_error());
    desconectar();

    if(!$Result1)
    {
        $_SESSION['e_bbdd'] = "Error, no se ha podido borrar.";
        header("Location: ../../aplic/error_bbdd.php");
        exit();
    }

    //si no se produce ningun error
    $_SESSION['aviso'] = "<div class='success'>El Aula ha sido borrada correctamente, las preguntas relacionadas han sido borras y las preguntas tambien.</div>";

    $objResponse->call("removeltem",$id);
    return $objResponse;
}
```

7.1.6 Javascript

El lenguaje JavaScript es el complemento ideal del HTML, al permitir a la página ejecutar comportamientos dinámicos por sí misma, sin necesidad de estar sobrecargando el servidor del cual depende.

JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

JavaScript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, y a veces con ligereza.

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.

JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Toda esta potencia de JavaScript se pone a disposición del programador, que se convierte en el verdadero dueño y controlador de cada cosa que ocurre en la página.

Ejemplo de función javascript:

```
function selectUser(id)
{
    selec_user=id;
    if (style_ant!="")
    {
        ant=document.getElementById(style_ant);
        ant.style.background="";
    }
    obj=document.getElementById(id);
    obj.style.background='#CCCCCC';
    style_ant=id;
}
```

7.2 Herramientas

7.2.1 Adobe Dreamweaver CS4

Durante la implementación del proyecto, Dreamweaver CS4 ha sido la herramienta más utilizada, ya que con ella se ha llevado a cabo toda la programación de la aplicación Web con los lenguajes PHP, JavaScript y HTML.

Dreamweaver proporciona una plataforma muy completa para el diseño, desarrollo, mantenimiento y administración de sitios Web ya que soporta gran cantidad de tecnologías como:

- HTML, XHTML
- CSS
- JavaScript
- PHP
- ASP
- ASP.NET
- JSP
- XML
- WML
- Elementos Multimedia
- Etc.

Posee una interfaz gráfica de fácil manejo que nos permite generar código automáticamente (en nuestro caso PHP y HTML) y visualizar los cambios que efectuamos al mismo tiempo que los realizamos.

Este programa también ofrece muchas posibilidades en lo que se refiere a JavaScript y CSS (hojas de estilo).

Pese a las grandes ventajas que ofrece Dreamweaver, es obvio que en numerosas ocasiones ha sido necesario escribir código para obtener resultados deseados. En la siguiente figura se puede observar la herramienta en uso.

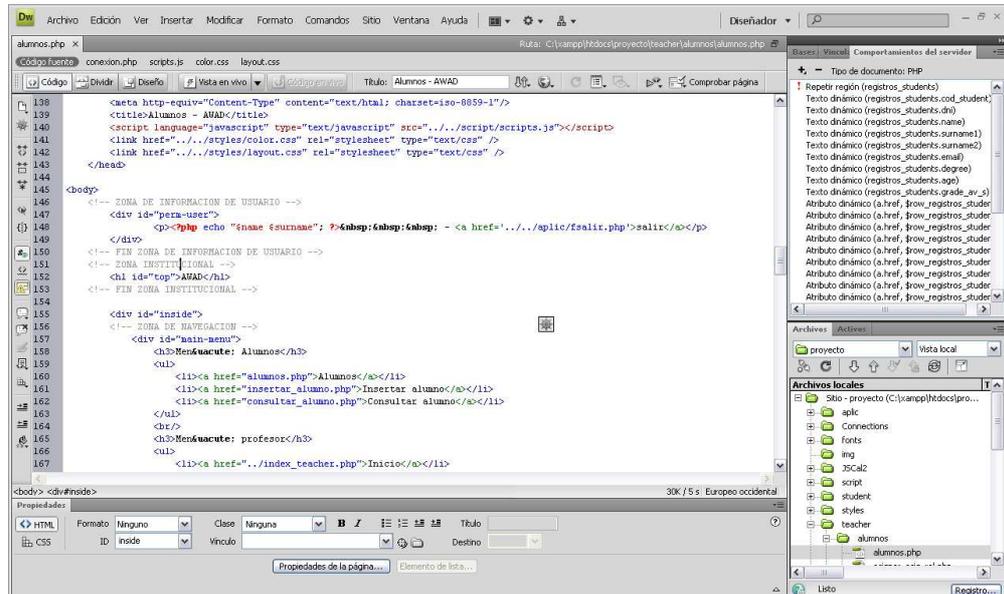


Figura 28. Dreamweaver CS4.

7.2.2 Adobe Fireworks

Para la parte de edición gráfica se ha usado Adobe Fireworks (FW), en concreto para la modificación de muchos de los iconos de la Web para hacerlos más intuitivos y amoldarlos lo más posible a la interfaz de la aplicación, para la creación de fondos degradados y para el logo del sitio Web.

Adobe Fireworks (FW), es programa que sirve para la edición de gráficos en mapa de bits o vectoriales. Originalmente fue desarrollado por Macromedia, compañía que fue comprada en 2005 por Adobe Systems. Fireworks está enfocado en la creación y edición de gráficos para Internet. Está diseñado para integrarse con otros productos de Macromedia, como Dreamweaver y Flash. Está disponible de forma individual o integrado en Adobe CS3.



7.2.2.1 MySQL WorkBench

MySQL Workbench es un software creado por la empresa informática Sun Microsystems, esta herramienta permite modelar diagramas de entidad-relación para bases de datos MySQL. Puede utilizarse para diseñar el esquema

de una base de datos nueva, documentar una ya existente o realizar una migración compleja.

La aplicación elabora una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real, ingeniería inversa para importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrator.

MySQL Workbench también puede generar el guión necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5. [WIKI MySQL_Workbench]

Las principales características son:

- Representación visual de tablas, vistas, procedimientos, funciones y claves ajenas.
- Soporta ingeniería inversa.
- Sincronización del modelo con la base de datos.
- Generación de scripts SQL.
- Arquitectura extensible a través de plugins y scripts.
- Exportar como SQL CREATE script.
- Importar modelos de DBDesigner 4.
- Soporta características de MySQL 5.
- Soporta diferentes notaciones para el diagrama entidad relación.
- Generar documentación.

MySQL Workbench se distribuye en dos versiones distintas:

- *MySQL Workbench Community Edition*. Bajo licencia GPL.
- *MySQL Workbench Standard Edition*. Disponible con suscripción de pagos. Incluye características adicionales.

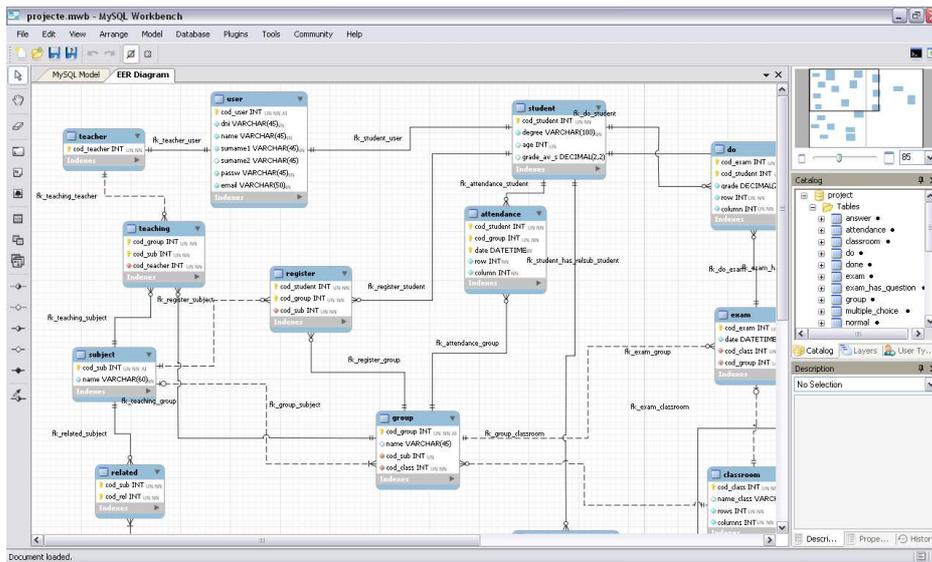


Figura 29. MySQL Workbench.

7.2.3 Microsoft Visio

Para la realización de los diseños del método OOWS se ha utilizado la herramienta de Microsoft Visio, en concreto la versión 2003.

Visio es una herramienta de dibujo vectorial. Aunque originalmente estaba pensada para dibujo técnico en el campo de Ingeniería y Arquitectura; con añadidos para desarrollar diagramas de negocios, su adquisición por Microsoft implicó drásticos cambios de tal forma que a partir de la versión de Visio para Microsoft Office 2003 el desarrollo de diagramas para negocios pasó de añadido a ser el núcleo central de negocio, minimizando las funciones para desarrollo de planos de Ingeniería y Arquitectura que se habían mantenido como principales anteriormente. [WIKI-VISIO, 2009].

Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, etc., que permiten iniciar al usuario en los lenguajes de programación.

En la siguiente figura podemos observar la herramienta en uso para la creación de los mapas del modelo navegacional del método OOWS.

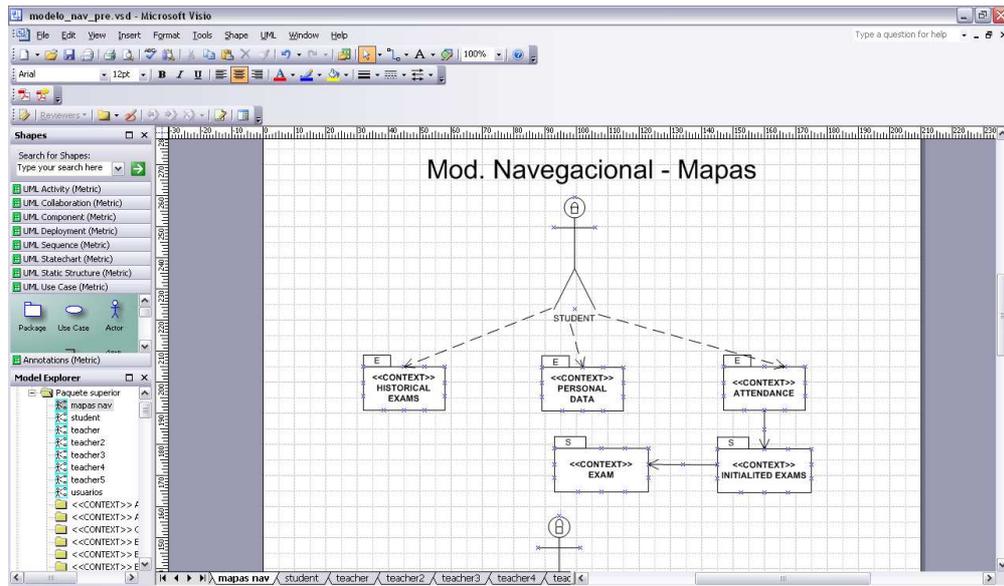


Figura 30. Microsoft Visio 2003

7.3 PhpMyAdmin

Para la creación y gestión de la base de datos MySQL se ha utilizado la herramienta PhpMyAdmin. Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas Web utilizando un navegador. Permite crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas bajo la licencia GPL. [PMA, 2009]. En la siguiente figura se observa la base de datos de la aplicación en PhpMyAdmin.

Tabla	Acción	Registros ¹	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> answer		20	InnoDB	utf8_spanish2_ci	40.0 KB	-
<input type="checkbox"/> attendance		1	InnoDB	utf8_spanish2_ci	64.0 KB	-
<input type="checkbox"/> classroom		0	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> do		2	InnoDB	utf8_spanish2_ci	48.0 KB	-
<input type="checkbox"/> done		2	InnoDB	utf8_spanish2_ci	60.0 KB	-
<input type="checkbox"/> exam		5	InnoDB	utf8_spanish2_ci	48.0 KB	-
<input type="checkbox"/> exam_has_question		5	InnoDB	utf8_spanish2_ci	48.0 KB	-
<input type="checkbox"/> group		10	InnoDB	utf8_spanish2_ci	64.0 KB	-
<input type="checkbox"/> multiple_choice		9	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> normal		0	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> question		9	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> register		4	InnoDB	utf8_spanish2_ci	64.0 KB	-
<input type="checkbox"/> related		11	InnoDB	utf8_spanish2_ci	60.0 KB	-
<input type="checkbox"/> related_subject		0	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> student		4	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> student_has_relob		2	InnoDB	utf8_spanish2_ci	48.0 KB	-
<input type="checkbox"/> subject		0	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> teacher		2	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> teaching		4	InnoDB	utf8_spanish2_ci	64.0 KB	-
<input type="checkbox"/> topic		22	InnoDB	utf8_spanish2_ci	72.0 KB	-
<input type="checkbox"/> user		7	InnoDB	utf8_spanish2_ci	48.0 KB	-

21 tablas(s) Número de filas: 153 MyISAM utf8_spanish2_ci 960.0 KB 0 Bytes

Figura 31. PhpMyAdmin.

La herramienta nos ha servido de gran ayuda a la hora de crear la base de datos de la Web ya que consta con una interfaz muy intuitiva y amigable. Además fue de gran ayuda a la hora de ver si el funcionamiento de la Web con la base de datos era correcta, ya que permite observar la información correspondiente a cada tabla. Por otro lado nos facilitó el poder crear archivos en los cuales se encontraba toda la base de datos de la aplicación.

7.4 Librería PHP para crear PDFs

PHP tiene entre sus características primarias la capacidad de generar archivos PDF. Con funciones específicas para personalizar y crear archivos PDF pero son muy difíciles de utilizar, por esta razón se ha buscado y encontrado una alternativa. R&OS pdf class es una clase la cual provee métodos muy potentes y simplificados para la creación de archivos PDF en PHP y además con las ventajas de que no es un modulo, esta formado simplemente por unos archivos PHP y es software libre. [R&OS, 2009].

7.5 Highcharts

Highcharts es una biblioteca de gráficos escrito en puro JavaScript, que ofrece una forma fácil de añadir gráficos interactivos a su sitio web o aplicación web. Highcharts actualmente soporta graficas de línea, área, areaspline, columna, barra, circular y los tipos de dispersión de éxitos.



Una de las características que me hizo decantarme por esta librería es por compatibilidad y licencia GNU.

Funciona en todos los navegadores modernos como el iPhone / IPAD e Internet Explorer desde la versión 6. Navegadores estándar SVG uso para el renderizado de gráficos. En Internet Explorer gráficos se dibujan usando VML.



7.6 Calendario en JavaScript

Durante la implementación era necesario un calendario en JavaScript para que el usuario de la aplicación Web pueda introducir fechas fácilmente y en el formato oportuno.

Crear un calendario en JavaScript puede ser una tarea bastante complicada y engorrosa, por eso igual que para crear PDFs se ha pensado en utilizar alguna opción externa que nos permita tener fácilmente un calendario en JavaScript.

Finalmente se ha utilizado JSCalendar, un calendario JavaScript de la compañía Dynarch que lo ofrece gratis si es para uso no comercial. Consta de una clase JavaScript para crear un calendario mensual. Es fácil de usar y sumamente customizable, apoya múltiples lenguas, y puedes escoger si las semanas comienzan con el sábado, el domingo, el lunes, o cualquier otro día. [JS-C, 2009].

En la siguiente figura vemos un ejemplo de uso de JSCalendar.

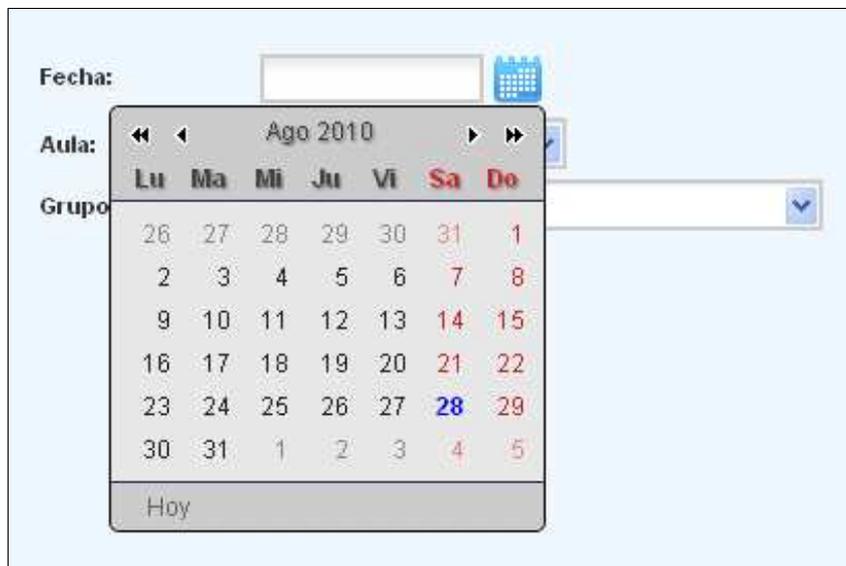


Figura 32. Ejemplo de uso de JSCalendar.

7.7 Framework jQuery

jQuery es una biblioteca o framework de javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la licencia MIT y de la GNU General Public License, Versión 2. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.



Ejemplo de código usando el Framework JQUERY:

```
$(document).ready(function()
{
    $("#add_respuestas").click( function()
    {
        n_res=n_res+1;
        $("#contenedor_respuestas").append('<div id="div_' + n_res + "'
class="insertRespuesta"><table
class="pregunta"><tr><th><label>Respuesta:</label></th><td><textarea id="text_'
+ n_res + "' name="text_' + n_res + "' cols="50"
rows="1"></textarea></td><td></td></tr><tr><th><label>Es solucion:
</label></th><td><input id="checkbox_' + n_res + "' name="checkbox_' + n_res + "'
type="checkbox"></td><td></td></th></table><a id="a_' + n_res + "'
class="eliminarPregunta" onclick="removeItem(this); return
false;">Eliminar</a></div>').children(':last').hide().fadeIn("slow");
    });
});
```

CAPÍTULO 8

8 Conclusiones

Eliminado: <sp>

Las sensaciones a la hora de finalizar este Proyecto Fin de Carrera son muy positivas. Quedé muy satisfecho con el desarrollo realizado y los resultados obtenidos, ya que habiendo sido, en algunos momentos, costoso y desquiciante fui capaz de solucionar la gran mayoría de los problemas que fueron apareciendo o que nosotros mismos nos propusimos. La solución a estos problemas fue hallada, en parte, en documentaciones sobre el tema e información al respecto en otras páginas web especializadas, pero las tareas que me resultaron más complejas se resolvieron por mucho esfuerzo, siendo paciente y muy insistente.

El comienzo, tras realizar un primer análisis y estructurar en rasgos generales la aplicación, resultó un poco complicado y largo pues el aprendizaje sobre el código de AWAD me llevó tiempo hasta que empecé a entenderlo y realizar pruebas. Es por ello que recuerdo las primeras pruebas como el auténtico inicio de este proyecto.

A pesar de que el resultado final se asemeja bastante con la idea de aplicación web que queríamos desarrollar, muchas ideas no pasaron a ser desarrolladas, principalmente, por falta de tiempo. Considero que dejamos un proyecto bien terminado pero abierto, con posibilidad de seguir incorporándole nuevos módulos o funcionalidades y otras mejoras y opciones. No hay más que pensar en formas de interactuar con otros usuarios. Las posibilidades son incontables y, seguramente, no resultarían muy complejas de incluir, ya que las bases están creadas.

Como se expuso en la introducción de este documento, en los últimos años, se ha notado un incremento considerable del uso de nuevas tecnologías en la educación. Mediante estas tecnologías, es posible dar un mayor apoyo al personal docente con respecto a los medios tradicionales.

Dado que el proyecto ya estaba empezado por Francisco Bosch Cerverá y permitía operaciones básicas como alta/baja/modificación de las tablas de la base de datos, quiero darle las gracias por su trabajo y dedicación, así como su buena documentación del código.

Como conclusión podemos decir que nuestro proyecto, es una herramienta más específica y no tan general como las soluciones comerciales existentes, de ahí que su uso conjuntamente con otras plataformas libres podría ofrecer funciones y características tan completas como las soluciones comerciales.

Personalmente, el presente proyecto significa mucho para mí y ha resultado muy interesante llevarlo a cabo. He podido aplicar los conocimientos adquiridos durante todos estos años de estudio, sobretodo en el ámbito de la programación y bases de datos. También he adquirido conocimientos en los lenguajes PHP, HTML y un poco de CSS y JavaScript ya que apenas tenía experiencia con ellos.

Con todo esto, ahora sí, concluimos con el Proyecto Fin de Carrera, el cual nos ha llevado también a concluir con la Ingeniería Técnica en Informática de Gestión, agradeciendo al director del mismo así como a todos aquellos compañeros que me han ayudado con consejos y posibles funcionalidades para la aplicación AWAD.

CAPÍTULO 9

9 Ampliaciones y mejoras futuras

Las ampliaciones o mejoras pensadas para un futuro se pueden separar en dos partes, las referentes a la aplicación Web y las referentes a la base de datos.

9.1 Mejoras a nivel de persistencia

En las referentes a la base de datos, se puede realizar un procedimiento para cada acción de insertar, modificar o borrar de cada tabla de la base de datos. Actualmente hay algunos procedimientos MySQL creados pero sólo en los casos en los que era necesario ya que se necesitaba un poco de programación para comprobar campos, dar formato o por que una acción en una tabla encadenaba otra acción en otra tabla. Estaría bien en un futuro que la comunicación entre la aplicación Web y la base de datos sólo fuera a través de llamadas a estos procedimientos quedando así totalmente separadas, esto supondría que al tener que realizar algún cambio en la base de datos sólo se tuviera que realizar a nivel de persistencia (en la base de datos) y no en la aplicación Web.

También se podría pensar en añadir algún *trigger* para llevar un mantenimiento en la base de datos en concreto para controlar los siguientes casos:

- Al borrar de SUBJECT en cascada se borra de RELATED, al borrar de RELATED comprobar si la asignatura sólo tenía una asignatura-relacionada y si es así borrarla también (la asignatura-relacionada en RELATED-SUBJECT).
- Al borrar de STUDENT en cascada se borra de DO, al borrar de DO comprobar si es el último alumno que esta relacionado con ese examen, si es así borrarlo también (el examen en EXAM).
- Al borrar de QUESTION en cascada se borra de HAS, al borrar de HAS comprobar si es la última pregunta relacionada con el examen al que pertenece, si es así borrarlo también (el examen en EXAM).

9.2 Mejoras a nivel de funcionalidad

A pesar de que el resultado final se asemeja bastante con la idea principal que queríamos desarrollar, muchas ideas no pasaron a ser desarrolladas, principalmente, por falta de tiempo.

Consideramos que dejamos un proyecto bien terminado pero abierto, con posibilidad de seguir incorporándole nuevos módulos o funcionalidades y otras mejoras y opciones. Las posibilidades son incontables y, seguramente, no resultarían muy complejas de incluir, ya que las bases están creadas.

Alguna de las mejoras pensadas son:

- **Creación de exámenes en función de su dificultad:** Para esto el sistema asignaría un valor de dificultad en función de los fallos y aciertos que tuvieran los alumnos para dicha pregunta, si en un examen el porcentaje de fallos en esa pregunta es bajo, la dificultad aumentaría. De esta forma el profesor a la hora de asignar las preguntas al examen, podría indicar el nivel de dificultad del examen.
- **Exámenes con tiempo finito:** Incorporar un reloj a modo de cuenta atrás en los exámenes, de esta forma el alumno sabría el tiempo restante que le queda, por otro lado el profesor no tendría que preocuparse de tener que parar el examen. Al finalizar el tiempo el examen se enviaría al servidor.
- **Exámenes con direcciones IP:** Otra posibilidad en cuanto a la evaluación se refiere sería asignar un conjunto de direcciones IP a la hora de empezar el examen, de esta forma sólo los ordenadores con dichas direcciones IP podrían iniciar el examen. Esto permitiría evitar que un alumno hiciese el examen desde otra aula. A pesar de no incorporar esta funcionalidad, hemos dotado al sistema de mecanismos para evitar este problema.

Bibliografía

[DBD, 2004]. Material de la asignatura “Diseño de base de datos” de la Facultad de informática de Valencia.

[IEEE830]. IEEE std 830 Recommended Practice for Software Requirements Specifications.

[IDR, 2006]. Material de la asignatura “Ingeniería de requerimientos” de la Facultad de informática de Valencia.

[WIKI-MER, 2009]. Wikipedia la enciclopedia libre, artículo Modelo Entidad-Relación, http://es.wikipedia.org/wiki/Modelo_entidad-relaci3n, 2009.

[DBDESIGNER, 2009]. Documentación online de la página Web del producto, <http://www.fabforce.net/dbdesigner4/doc/index.html>.

[MWB, 2009]. Información y documentación disponible en la página Web del producto, <http://www.mysql.com/products/workbench/>.

[IDR, 2006]. Material de la asignatura Ingeniería de requerimientos de la Facultad de informática de Valencia.

[WIKI-PHP, 2009]. Wikipedia la enciclopedia libre, artículo sobre el lenguaje de programación PHP, <http://es.wikipedia.org/wiki/.php>.

[WIKI-MY, 2009]. Wikipedia la enciclopedia libre, artículo sobre el sistema de gestión de base de datos MySQL, <http://es.wikipedia.org/wiki/Mysql>.

[WIKI-HTML, 2009]. Wikipedia la enciclopedia libre, artículo sobre el lenguaje de marcas HTML, <http://es.wikipedia.org/wiki/Html>.

[WIKI-HTTP, 2009]. Wikipedia la enciclopedia libre, artículo sobre el protocolo HTTP, <http://es.wikipedia.org/wiki/Http>.

[WIKI-IDE, 2009]. Wikipedia la enciclopedia libre, artículo sobre entornos de desarrollo integrados IDE, <http://es.wikipedia.org/wiki/>.

[WIKI-FRA, 2009]. Wikipedia la enciclopedia libre, artículo sobre Frameworks, <http://es.wikipedia.org/wiki/Framework>.

[WIKI-CAS, 2009]. Wikipedia la enciclopedia libre, artículo sobre el modelo en cascada http://es.wikipedia.org/wiki/Modelo_en_cascada.

[WIKI-SOFT, 2009]. Wikipedia la enciclopedia libre, artículo sobre software, http://es.wikipedia.org/wiki/Computer_software.

[BLACKB, 2009]. Página Web oficial de la compañía Blackboard. <http://www.blackboard.com/>.

[WIKI-BLACKB, 2009]. Wikipedia la enciclopedia libre, artículo sobre la compañía Blackboard Inc. <http://es.wikipedia.org/wiki/Blackboard>.

[SAKAI, 2009]. Página Web oficial del proyecto Sakai.
<http://sakaiproject.org/portal>.

[WIKI-DW, 2009]. Wikipedia la enciclopedia libre, artículo sobre diseño Web,
http://es.wikipedia.org/wiki/Diseño_web.

[NIELSEN-LORANGER, 2006]. Jakob Nielsen, Hoa Loranger; Libro *“Prioritizing Web Usability”*, Editorial New Riders, 2006.

[NIELSEN, 2000]. Jakob Nielsen; Libro *“Usabilidad. Diseño de sitios Web”*, Editorial Prentice may, 2000.

[CUEVA, 2004]. Juan Manuel Cueva Loelle, Artículo “Usabilidad en la web”;
<http://di002.edv.uniovi.es/~cueva/asignaturas/doctorado/2004/IngWeb.html>.

[W3S, 2009]. Sección de estadísticas en el portal w3schools
<http://www.w3schools.com/browsers/default.asp>.

[OOWS, 2004]. Material de la asignatura Desarrollo de aplicaciones para entornos web de la Facultad de informática de Valencia.

[OOWS2, 2002]. Joan Fons, Oscar Pastor, Pedro Valderas y Marta Ruiz. Artículo “OOWS: Un Método de Producción de Software en Ambientes Web”. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.

[DW, 2009]. Página Web oficial de la herramienta Dreamweaver CS4.
<http://www.adobe.com/es/products/dreamweaver>.

[WIKI-VISIO, 2009]. Wikipedia la enciclopedia libre, artículo sobre Microsoft Visio, http://es.wikipedia.org/wiki/Microsoft_Visio.

[R&OS, 2009]. Página Web oficial de la compañía R&OS, <http://ros.co.nz/pdf>.

[JS-C, 2009]. Página Web oficial de la compañía dynarch,
<http://www.dynarch.com/projects/calendar/>.

[MQB, 2009]. Manual oficial de la herramienta MySQL Query Browser,
<http://dev.mysql.com/doc/query-browser/es/index.html>.

[PMA, 2009]. Página Web oficial de la herramienta PhpMyAdmin,
http://www.phpmyadmin.net/home_page/index.php.

[Highcharts, 2010]. Página Web oficial del Framework Highcharts.
<http://www.highcharts.com/>.

[Srbyte, 2010]. Página Web oficial del SrByte.
<http://www.srbyte.com/2009/04/la-importancia-de-los-procedimientos.html>.

10 Anexo

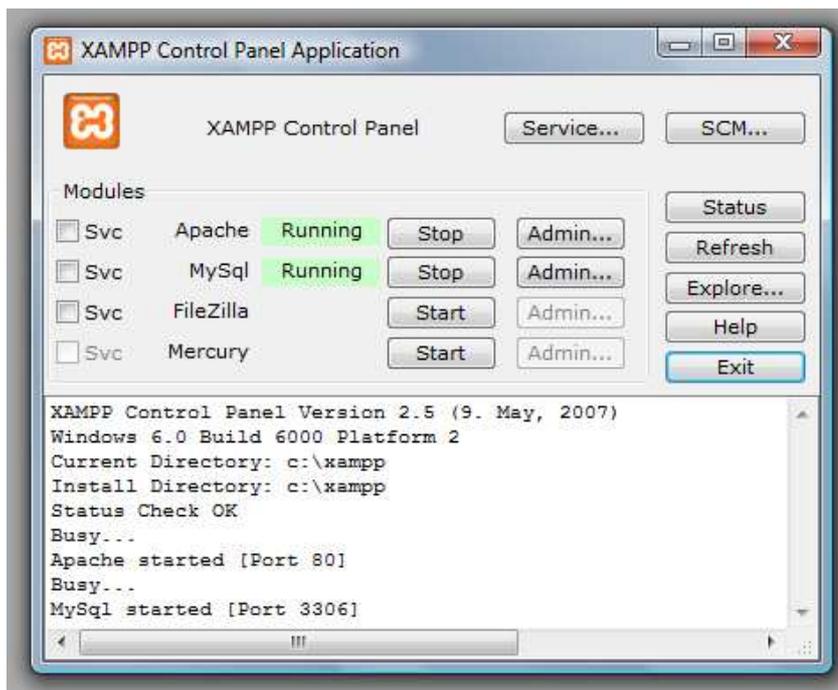
10.1 Instalación de AWAD

En esta sección vamos a explicar todo lo necesario para instalar AWAD en un servidor Web.

Lo primero que haremos es instalar XAMPP, gracias a esta aplicación se puede instalar Apache, PHP5 y MySQL de forma fácil y muy rápida. Sólo debes dirigirte a la sección de descargas de XAMPP y elegir la opción XAMPP Windows “installer” más reciente para comenzar a bajarla.

Dónde guardarlo: Al correr la instalación, lo primero que vas a tener que decidir es dónde guardarlo (recomiendo dejarlo en la carpeta que nos aparece por defecto c:\xampp). La ubicación es importante porque ahí es donde se ubicarán físicamente los sitios que quieras mostrar en el servidor.

Panel de Control: Luego de un par de siguientes y hacer clic en finalizar, el instalador te va a preguntar si deseas abrir el panel de control. Desde esa nueva ventana es que vas a poder prender y apagar el servidor Apache y MySQL (PHP se inicia cuando inicias Apache) cada vez que lo necesites.



Una vez instalado, tendremos que guardar la carpeta *awad*, que es donde tenemos alojado el proyecto, en el directorio `c:\xampp\htdocs\` para que puedan ser vistos.

El último para poder disfrutar de AWAD es importar la base de datos al nuevo servidor, para ello abriremos el navegador y escribiremos <http://127.0.0.1/phpmyadmin/>. Una vez dentro, crearemos una nueva base de datos y le pondremos como nombre *projet*. Accediendo Project desde el menu de la izquierda ya podremos importar la base de datos, basta con seleccionar la opción importar y seleccionar el archivo `bbdd.sql` que se adjunta en la memoria.

Ya tenemos instalado AWAD en el servidor, escribiendo <http://127.0.0.1/awad> en el navegador y si todo ha ido bien, accederemos a la pantalla de inicio de la aplicación.

A.W.A.D
APLICACIÓN WEB DE APOYO A LA DOCENCIA

Estás en: Página de registro

Regístrate

Todavía no usas AWAD? Regístrate aquí

Regístrate

Identificate

Email:
(* Correo UPV)

Password:

Identificate

Recordar sesión?
No marques esta casilla si estás utilizando un ordenador público o compartido

Universidad Politécnica de Valencia

Figura 33. Pantalla de login de la aplicación Web.