



**Universitat Autònoma
de Barcelona**

**Web para la gestión
electrónica de una Carpeta
de Aprendizaje**

Memoria del proyecto
De Ingeniería Técnica en
Informática de Sistemas

realizado por

David Forte Romera

y dirigido por

Asier Ibeas Hernández

Escuela Universitaria de Informática

Sabadell, Septiembre de 2009

El abajo firmante, **Asier Ibeas Hernández**, profesor de la Escuela Universitaria de Informática de la UAB,

CERTIFICA:

Que el trabajo al que corresponde la presente memoria ha estado realizado bajo su dirección por **David Forte Romera**.

Y para que conste firma la presente.

Sabadell, Septiembre del 2009

Firmado: Asier Ibeas Hernández

RESUMEN DEL PROYECTO

La rápida evolución de la tecnología ha permitido que la gran mayoría de personas tengan acceso tanto a una conexión a Internet como a un ordenador de forma habitual, haciendo posible un cambio de mentalidad a la hora de realizar las diferentes tareas de su trabajo. Con esa idea aplicada al campo de la docencia, se ha decidido implementar una aplicación web con el objetivo de aprovechar esta nueva realidad a la hora de hacer las diferentes actividades del curso para la asignatura de Robótica y Automatización Industrial.

El proyecto está dividido en dos partes bien diferenciadas. La primera es la parte del alumno, que como usuario de esta aplicación podrá añadir y modificar las respuestas a los ejercicios que se propongan durante el curso, además de obtener la información referente a la asignatura.

La segunda parte se trata de la sección del profesor, que como administrador tendrá la posibilidad de añadir o eliminar actividades e información de la asignatura además de tener acceso a todas las páginas de los alumnos para hacer el seguimiento.

El objetivo es conseguir una comunicación fluida entre profesor y alumno, así como un control más exhaustivo del trabajo realizado por cada estudiante y un seguimiento más continuado de la asignatura por parte de éste, además de los objetivos mencionados también se pretende agilizar el trabajo de corrección por parte del profesor.

Entre las prioridades de esta aplicación están las de minimizar lo máximo posible los inconvenientes por el cambio de metodología de trabajo, tanto de los alumnos como de los profesores, utilizando un diseño intuitivo, además de buscar un diseño robusto y fiable que permita garantizar el acceso seguro a la aplicación, y un control adecuado de la información contenida en la base de datos.

ÍNDICE

1	Introducción	7
1.1	Presentación	7
1.2	Objetivos	7
1.3	Estado del arte	7
1.4	Estructura de la memoria	8
2	Estudio de viabilidad	9
2.1	Introducción	9
2.2	Objeto	9
2.2.1	<i>Descripción de la situación actual</i>	9
2.2.2	<i>Perfil del usuario</i>	9
2.2.3	<i>Objetivos</i>	10
2.2.4	<i>Fuentes de información</i>	10
2.3	Sistema a realizar	10
2.3.1	<i>Descripción</i>	10
2.3.2	<i>Modelo de desarrollo</i>	11
2.3.3	<i>Recursos de desarrollo</i>	11
2.3.4	<i>Recursos de implementación</i>	12
2.4	Análisis de coste-beneficio	13
2.5	Evaluación de riesgos	14
2.6	Planificación	14
2.7	Viabilidad técnica	15
2.7.1	<i>Alternativas</i>	15
2.7.2	<i>Análisis de tecnologías</i>	16
2.7.3	<i>Tabla comparativa de tecnologías web</i>	19
2.7.4	<i>Tecnología web para el proyecto</i>	19

2.8 Conclusiones	20
3 Análisis	21
3.1 Requisitos funcionales	21
3.2 Requisitos no funcionales	21
3.3 Diagrama de casos de uso	22
3.3.1 Alumno	22
3.3.2 Profesor	23
3.4 Descripción de los casos de uso.....	24
3.4.1 Alumno	24
3.4.2 Profesor	28
4 Herramientas utilizadas	35
4.1 Struts 2	35
4.2 JavaScript	35
4.2.1 JSMath	36
4.2.2 Prototype	36
4.2.3 Tiny_mce	36
4.3 CSS	36
4.4 Jasypt	37
4.5 MySQL	37
4.5.1 MySQL Administrator	37
4.6 Hibernate	37
4.7 Tomcat	38
4.8 Eclipse	38
4.9 OpenSSL	38
5 Diseño de la aplicación	39
5.1 Base de datos	39
5.1.1 Diagrama	39

5.1.2 Estructura	40
5.2 Interfaz gráfica	44
5.2.1 Inicio	44
5.2.2 Alumno	46
5.2.2.1 Diario	48
5.2.2.2 Notas	50
5.2.3 Profesor	50
5.2.3.1 Lista de alumnos	52
5.2.3.2 Notas	52
5.2.3.3 Introducción de ejercicios	53
5.2.4 Subir archivos	54
5.2.5 Editor de textos	54
5.2.6 Botonera	55
6 Implementación	56
6.1 Capa del modelo	56
6.2 Capa de la vista	58
6.3 Capa del controlador	59
6.4 Internacionalización i18n	60
7 Pruebas	61
8 Conclusiones	67
9 Bibliografía	68
9.1 Fuentes bibliográficas	68
9.2 Fuentes electrónicas	68
9.3 Fuentes tecnológicas	69
Anexo	70

1. INTRODUCCIÓN

1.1 Presentación

El proyecto se basa en el desarrollo de una aplicación web que permita actualizar la metodología de trabajo utilizada hasta el momento en la asignatura de Robótica y Automatización Industrial que es una asignatura optativa de cuarto de Ingeniería informática en la Escuela de Bellaterra.

El desarrollo de dicha aplicación permitirá mayor agilidad en la corrección y seguimiento de los alumnos por parte del profesor, y en el caso de los alumnos se obtendrá más dinamismo en el seguimiento de la asignatura.

1.2 Objetivos

El principal objetivo de la aplicación es que pueda ser utilizada como herramienta de trabajo en la asignatura anteriormente mencionada haciendo que la adaptación por parte de los profesores sea rápida.

La aplicación tiene que ser dinámica para poder mostrar o modificar diferentes actividades, y segura para evitar intrusismo entre las cuentas de los distintos alumnos.

La parte de la administración también debe ser privada, y permitir ver y modificar el resto de cuentas de usuario.

1.3 Estado del arte

En la actualidad, con la irrupción de la web 2.0, se ha puesto más de moda que nunca el uso de aplicaciones web donde el papel que tienen los usuarios es cada vez más activo, las wikis son un claro ejemplo de esa tendencia, donde se favorece principalmente el trabajo en grupo utilizando una aplicación fácil de usar y de bajo coste. Otra alternativa es Moodle que es un sistema de gestión de recursos con el que se puede crear comunidades de aprendizaje en línea y que tiene disponibles diferentes módulos para la comunicación profesor-estudiante. Por último está la opción del Campus Virtual de la UAB mediante el cual los alumnos y profesores de una asignatura pueden comunicarse e intercambiarse diferente material pedagógico de forma rápida y sencilla.

En este proyecto se ha buscado una variación de esta idea con el objetivo de conseguir trabajos más individualizados y añadir ciertas comodidades al usuario final como un

INTRODUCCIÓN

editor WYSIWYG, o evitar que se tenga que crear una página nueva antes de poder empezar a editar un texto.

1.4 Estructura de la memoria

La memoria se ha dividido en nueve capítulos a través de los cuales se explicarán los objetivos del proyecto así como las tecnologías usadas para su desarrollo, además de cómo se ha diseñado y codificado la aplicación.

En el primer capítulo se verá una introducción al proyecto con los objetivos principales que se persiguen con su implementación además de las diferentes alternativas que se pueden encontrar para un uso similar.

El segundo capítulo está dedicado al estudio de viabilidad. En él se estudiará la situación actual, el coste de realizar este proyecto, los recursos necesarios durante su implementación y posterior puesta en marcha o la planificación para su desarrollo.

En el tercer capítulo se hará un estudio más en profundidad de los requisitos funcionales y no funcionales del proyecto. También se verán los diferentes casos de uso para cada uno de los futuros usuarios del proyecto.

El cuarto capítulo servirá para describir las tecnologías utilizadas para el desarrollo del proyecto además del software utilizado para llevarlo a cabo.

El quinto capítulo mostrara el diseño de la aplicación dando una especial atención tanto al diseño de la base de datos como al diseño de la interfaz gráfica.

El sexto capítulo se utilizará para explicar los detalles de la implementación, como el patrón de desarrollo que utiliza Struts 2 o la internacionalización del proyecto.

El séptimo capítulo estará dedicado al apartado de pruebas, en donde se comprobará el buen funcionamiento del proyecto utilizando los dos roles disponibles de la aplicación.

En el capítulo octavo, se encontrará la bibliografía utilizada durante el desarrollo del proyecto.

Finalmente, en el anexo se explicará la manera de crear un usuario administrador para poder comenzar a utilizar la aplicación.

2. ESTUDIO DE VIABILIDAD

2.1 Introducción

En este apartado se analizará la situación actual y los requisitos del proyecto. También se verán los objetivos principales así como el tipo de usuario al que se destinará la aplicación.

A continuación se mostrará el modelo de desarrollo que se utilizará durante la realización del proyecto además de su planificación.

Para finalizar, se evaluará la viabilidad del proyecto teniendo en cuenta todos estos factores.

2.2 Objeto

2.2.1 Descripción de la situación actual

Actualmente en la asignatura se utiliza una carpeta física donde los alumnos guardan la solución de las diferentes actividades propuestas durante el curso, y que en ciertas fechas límite tienen que entregar en mano al profesor para que este haga un seguimiento y una evaluación del alumno.

Las funcionalidades principales que se pretenden conseguir son un seguimiento más continuado de la asignatura por parte del alumno, además de aumentar la interacción con el profesor. En cuanto al profesor, el objetivo que se persigue es tener un mayor control del trabajo realizado por parte del alumno.

2.2.2 Perfil del usuario

Esta aplicación será utilizada tanto por los alumnos como por los profesores de la asignatura de Robótica y Automatización Industrial, los cuales tienen un nivel medio-alto en el manejo de aplicaciones informáticas.

2.2.3 Objetivos

Los objetivos que se persiguen con este proyecto son la fácil adaptabilidad de los profesores de la asignatura al nuevo formato de trabajo y asegurar un acceso seguro tanto a las cuentas de los alumnos como a la del administrador. En cuanto a las actividades que permitirá la aplicación, los alumnos tendrán la posibilidad de modificar diferentes apartados entre los que se encuentran la redacción de un diario, donde cada alumno deberá anotar el trabajo realizado durante esa jornada, la edición de los ejercicios de los apartados de Teoría, Problemas y Prácticas, además de la redacción de un trabajo relacionado con algún tópico de la asignatura.

Desde la cuenta del profesor, y como administrador de la aplicación, tendrá la posibilidad de editar y añadir los enunciados de los ejercicios relacionados con los apartados de Teoría, Problemas y Prácticas. También podrá incluir la información necesaria para el desarrollo de la asignatura, y por último, para conseguir un mayor control sobre las actividades realizadas por los alumnos, podrá acceder a todas las cuentas de los alumnos registrados desde las cuales podrá añadir los comentarios oportunos de las actividades realizadas por cada estudiante.

2.2.4 Fuentes de información

La recogida de información para el proyecto comenzará en primer lugar con diferentes entrevistas con el profesor responsable de la asignatura para tener una idea general de la forma de trabajo y de los objetivos que se pretenden conseguir.

En una segunda fase se recogerá información más enfocada a la implementación de la aplicación utilizando diferentes páginas Web y libros de programación de lenguajes orientados a la programación Web y bases de datos.

2.3 Sistema a realizar

2.3.1 Descripción

Diseño e implementación de una aplicación Web que permita introducir a los alumnos las respuestas de las actividades propuestas por el profesor, guardándolas en una base de datos para una posterior visualización o modificación según las indicaciones del profesor, el cual, tendrá acceso a todos los archivos de los alumnos para poder añadir una corrección o un comentario en caso de ser necesario. La interfaz de usuario tiene que ser agradable e intuitiva para facilitar el uso de la aplicación y evitar posibles

errores de los usuarios, además de tener un buen nivel de seguridad para el acceso de todos los usuarios. Como se puede observar en la Figura 1 el sistema a realizar se compone de páginas exclusivas para el profesor y páginas para los alumnos desde las cuales se tendrá acceso a la base de datos para poder recuperar toda la información utilizada en la aplicación.

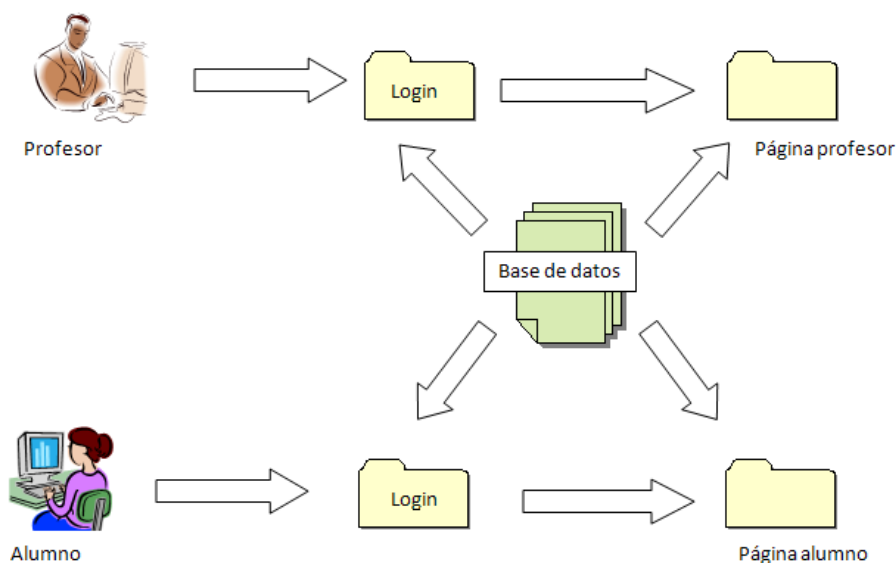


Figura 1: Esquema de la descripción

2.3.2 Modelo de desarrollo

El modelo de desarrollo elegido para realizar este proyecto es el modelo lineal ya que el trabajo a realizar se puede dividir en diferentes etapas consecutivas que tendrán que ir realizándose antes de comenzar la siguiente.

2.3.3 Recursos de desarrollo

- *Hardware:*
 - Procesador: Athlon 64 X2 4400+
 - Memoria RAM: 2GB
 - Disco Duro: 250 GB
 - Conexión a Internet

- *Software:*
 - Servidor Web: Apache Tomcat 5.5
 - Programación Web: Struts 2, JavaScript, CSS

 - Base de datos: MySQL
 - Editor de base de datos: MySQL Administrator
 - Entorno de desarrollo: Eclipse 3.2
 - Sistema operativo: Windows XP
 - Navegador: Firefox, Internet Explorer, Opera, Chrome, Safari
 - Documentación del proyecto: MS Office y MS Project

2.3.4 Recursos de implementación

- *Hardware del servidor:*
 - Procesador: Pentium 4 o superior.
 - Memoria RAM: 1GB
 - Espacio Disco Duro: 20GB
 - Conexión a Internet

- *Software del servidor:*
 - Servidor Web: compatible con java y SSL
 - Programación Web: java
 - Base de datos: MySQL

- *Software profesor/alumno:*
 - Navegador Web.

- *Hardware profesor/alumno:*
 - PC con conexión a Internet.

2.4 Análisis de coste-beneficio

Todo proyecto conlleva unos costes que han de tenerse en cuenta antes de comenzar implementarlo. En este caso se han previsto los siguientes:

Coste de material:

<i>Recurso</i>	<i>Coste</i>
Frameworks de Java (Struts 2, Hibernate) + Eclipse	0 €
Apache-Tomcat	0 €
MySQL	0 €
Amortización del PC del alumno	100 €
Amortización MS Project, MS Office	20,83 €
Total	120, 83 €

Coste de personal:

Estudiante: 20 €/hora

<i>Tarea</i>	<i>Horas</i>	<i>Coste</i>
Creación de una carpeta de estudiante	388	7760 €
Diagnóstico	5	100 €
Entrevista con el profesor	2	40 €
Realización de un boceto general del proyecto	3	60 €
Estudio de Viabilidad	31	620 €
Especificación del lenguaje de programación	1	20 €
Especificación de la parte del alumno	15	300 €
Especificación de la capacidad de almacenamiento	6	120 €
Especificación de la forma de introducir los datos	8	160 €
Especificación de la parte del profesor	15	300 €
Especificación de la forma de modificar el contenido	8	160 €
Especificación de la forma de evaluación	6	120 €
Realización del proyecto	300	6000 €
Diseño de la base de datos	30	600 €
Programación web	150	3000 €
Diseño y programación de la interfaz	100	2000 €
Pruebas	20	400 €
Fase final	52	1040 €
Documentación	50	1000 €
Exposición	2	40 €

El coste previsto para el proyecto es de 7760 €. Teniendo en cuenta que se trata de un proyecto final de carrera, los costes se consideran nulos ya que todos los materiales son gratuitos o pueden utilizarse de forma gratuita en la universidad. El coste de personal es asumido por el alumno ya que es un trabajo que forma parte de la carrera en curso.

2.5 Evaluación de riesgos

Incompatibilidad entre navegadores: Pueden surgir problemas en el momento de visionar la página en los distintos navegadores, de manera que habrá que diseñar el proyecto de forma que funciones correctamente en todos.

Problemas de conexión: Un fallo con la conexión a Internet puede suponer un retraso en el calendario previsto para las diferentes actividades.

Problemas en el servidor: Cualquier fallo en el servidor haría que fuera imposible acceder a la página tanto a los profesores como a los alumnos.

Seguridad: El acceso a la aplicación debe ser mediante usuario y contraseña para evitar que un alumno pueda acceder a la solución de los ejercicios propuesta por otro alumno, o que se pueda acceder sin consentimiento al área de administración.

2.6 Planificación

Fases del proyecto:

<i>Nº</i>	<i>Descripción de la actividad</i>	<i>Duración</i>
1	Entrevistas con el profesor	2 h
2	Realización de un boceto general de la aplicación	3 h
3	Instalación y configuración del servidor	2 h
4	Diseño de la base de datos	30 h
5	Programación web	150 h
6	Diseño y programación de la interfaz gráfica	100 h
7	Pruebas	20 h
8	Elaboración de la documentación	50 h

ESTUDIO DE VIABILIDAD

En la figura 1 se puede observar el diagrama de Gantt del proyecto, en el se puede ver una estimación de la duración de cada una de las tareas en que se divide el proyecto una vez asignados los recursos disponibles para cada una.

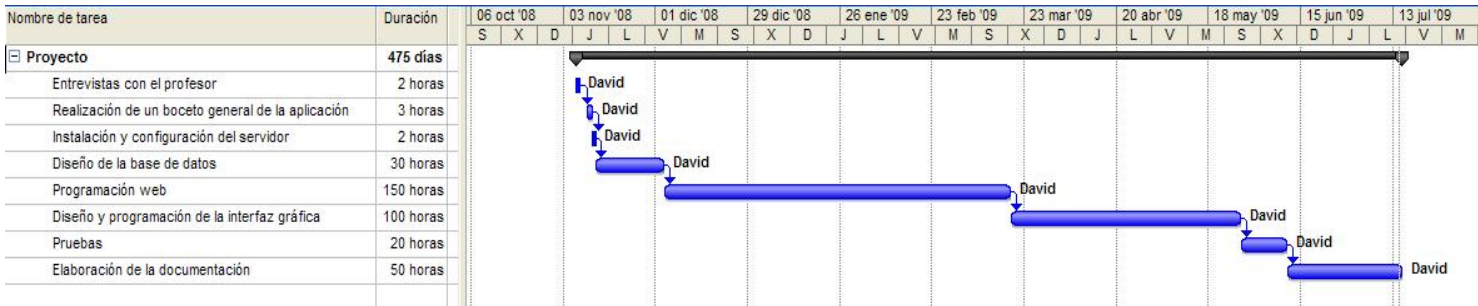


Figura 2: Diagrama de Gantt

2.7 Viabilidad técnica

2.7.1 Alternativas

Moodle

Como se ha comentado en el estado del arte, Moodle es una aplicación software para la creación de cursos a través de internet, con lo que se pretende facilitar la comunicación a distancia entre el profesor y sus alumnos.

Entre sus principales características se encuentra que es un software libre e intuitivo, además de ser flexible y personalizable al ser de código abierto.

Moodle permite utilizar diferentes sistemas de comunicación como foros, chats y mensajería interna, también permite incluir tanto tareas individuales, entre las que se incluyen las diferentes actividades del curso o enlaces a diferentes webs, como tareas en grupo, entre las que se incluye las wikis y los foros. Por último Moodle tiene la posibilidad de añadir diferentes actividades de evaluación entre las que se encuentran los cuestionarios o los exámenes autoevaluables.

El principal uso que tiene Moodle es la creación de cursos on-line o semipresenciales ya que toda la información del curso puede estar disponible en el sistema además de disponer de un amplio repertorio de herramientas de comunicación como ya se ha comentado, aunque también es posible su uso para salas virtuales de profesores, grupos de trabajo o para web de empresa con varios usos posibles como son las imagen corporativa en Internet o como intranet corporativa.

Campus Virtual

El Campus Virtual es una aplicación web utilizada por la gran mayoría de asignaturas impartidas en la UAB y con la que se consigue una comunicación bastante fluida entre los profesores de una asignatura y los alumnos matriculados en ella.

La principal baza de esta opción es la facilidad de uso y las diversas formas de comunicación entre los usuarios, donde destaca el uso de foros y el servicio de mensajería interna.

El Campus Virtual, aunque también presenta la posibilidad de utilizar una wiki para cada una de las asignaturas, se usa principalmente para hacer llegar el material docente a los alumnos y para que estos puedan realizar las diferentes entregas en forma de archivos adjuntos en las fechas establecidas.

Wiki

Una wiki es una aplicación que permite a los usuarios editar su contenido a través de un navegador web.

El principal uso que tienen las wikis es la edición colectiva de ciertos documentos, y a su favor tiene que es una forma sencilla y rápida de modificar los contenidos de una página, aunque tiene el inconveniente de tener que hacer un control más exhaustivo del contenido para evitar modificaciones no deseadas por parte de otros usuarios.

Conclusiones

Después de ver las diferentes alternativas que hay disponibles se ha decidido realizar una aplicación específica debido a que la forma de trabajar de estas aplicaciones no se ajusta a la metodología de trabajo deseada ya que están enfocadas a un uso más colaborativo entre los diferentes usuarios, en el caso de las wikis y de Moodle, o están enfocados al intercambio de archivos con los diferentes contenidos, en el caso del Campus Virtual.

2.7.2 Análisis de tecnologías:

Microsoft .NET

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones, que facilita el proceso de construcción de aplicaciones multipropósito, tanto en entornos cliente como en entornos web. La plataforma .NET incorpora una serie de servicios

ESTUDIO DE VIABILIDAD

universales como son la gestión de memoria y recolección de basura, seguridad, control de código, etc.... Por otro lado está la .NET framework que es una biblioteca de clases independiente del lenguaje, que facilita al desarrollador todos los elementos necesarios para incorporar interfaces de usuario nativas y web, acceso a bases de datos, comunicación a través de red, etc.

.NET se basa en estándares ya establecidos como XML, SOAP, WSDL y en general busca la creación de aplicaciones fáciles de integrar, de modificar y agilizar la implementación de cualquier tipo de soluciones, para cualquier dispositivo sin importar su hardware, sistema operativo o lenguaje con el que estén hechas las aplicaciones que ejecuta, intentando aprovechar para ello la infraestructura de internet, redes y protocolos.

Una de las partes más importantes de esta tecnología es el CLR (Common Language Runtime), que es una aplicación similar a una máquina virtual que se encarga de gestionar la ejecución de las aplicaciones. El CLR ofrece a las aplicaciones numerosos servicios que facilitan su desarrollo y mantenimiento y favorecen su fiabilidad y seguridad.

Los objetivos de .NET son:

- Reducir la complejidad en el desarrollo de aplicaciones.
- Proporcionar un entorno de ejecución robusto y seguro.
- Soporte de múltiples lenguaje de programación.
- Intercambio de datos entre distintas aplicaciones.
- Crear una librería de clases común para todos los lenguajes.

JAVA (J2EE)

Es una plataforma que habilita soluciones para desarrollo, uso efectivo y manejo de multicapas en aplicaciones centralizadas en el servidor. J2EE utiliza la plataforma Java 2 SE para tender una completa, estable, segura y rápida plataforma Java en el ámbito empresarial. Permite ahorrar a la compañía porque habilita una plataforma que reduce de manera significativa los costos y la complejidad de desarrollo de soluciones multicapa, resultando en servicios que pueden ser desarrollados rápidamente y ampliados fácilmente.

Java es un lenguaje de desarrollo de propósito general orientado a objetos y está basado en la arquitectura del lado del servidor. Usa una máquina virtual para la ejecución de los programas, lo que permite que las aplicaciones escritas en Java se puedan ejecutar en cualquier máquina, independientemente del sistema operativo y la configuración de hardware. Java también incluye un gran conjunto de componentes reutilizables para ayudar al programador (Struts2, Tapestry, Spring, Hibernate,...). Esta

ESTUDIO DE VIABILIDAD

arquitectura concentra la mayoría de los procesos de la aplicación en el servidor o en una sección de este. Este tipo de arquitectura tiene algunas ventajas como:

Múltiples clientes: Una arquitectura basada en el servidor requiere una clara separación entre la capa del cliente (interfaz) y la capa del servidor, en la cual se realizan los procesos de la aplicación. Con esto se consigue que una misma aplicación soporte múltiples clientes simultáneamente, todos ellos con distintas interfaces.

Operaciones robustas: Una arquitectura basada en el servidor soporta escalabilidad, confiabilidad, disponibilidad y recuperabilidad.

Los objetivos de Java son:

- Sencillez.
- Lenguaje orientado a objetos.
- Robusto porque simplifica la gestión de memoria y elimina las complejidades de la gestión de punteros.

- Seguro.
- Independiente de la arquitectura, pudiendo ser usado con hardware y sistemas operativos diversos.
- Multihilo.

PHP

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

El fácil uso y la similitud con los lenguajes de programación más comunes permiten que su aprendizaje y uso en aplicaciones complejas sea rápida.

PHP se interpreta y ejecuta en el servidor y el cliente solo recibe el resultado de la ejecución. También permite la conexión a múltiples servidores de base de datos tales como MySQL, Postgres, Oracle,... con lo que se consigue aplicaciones web muy robustas.

Los objetivos de PHP son:

- Lenguaje multiplataforma
- Capacidad de conexión con la mayoría de controladores de base de datos
- Amplia documentación.
- Es libre, por lo tanto de fácil acceso.
- Programación web dinámica.

2.7.3 Tabla comparativa de tecnologías web

Características	.NET	Java (J2EE)	PHP
Código compilado (Pre-compilado es más rápido)	Si, tanto pre-compilado como compilado dinámicamente cuando se hace una petición.	Si, tanto pre-compilado como compilado dinámicamente cuando se hace una petición.	No, un programa externo puede mejorar el funcionamiento, pero no viene instalado en la mayoría de servidores.
Lenguaje de scripts	No	No	Si
Orientado a objetos	Si	Si	No
Lenguajes de desarrollo soportados	C++, C#, VisualBasic .NET, JScript .NET, Python, Perl, J#, etc...	Java	PHP
Navegador para interpretar HTML específico	Si (Internet Explorer)	No	No
Código abierto	No	Si	Si
Multiplataforma	No	Si	Si
Coste	Elevado	Nulo	Nulo

2.7.4 Tecnología utilizada para el proyecto

Aunque PHP podría ser una opción perfectamente válida para desarrollar el proyecto debido a que es un lenguaje fácil de aprender y del que se poseen algunos conocimientos previos, además de ser de código abierto y muy utilizado para aplicaciones web, se ha descartado porque presenta algunas limitaciones como son el rendimiento, debido a que es un lenguaje script, y problemas a la hora de gestionar distintos usuarios con diferentes interfaces

Por lo tanto sin contar con PHP, están las opciones de Java y .NET y aunque con ambas tecnologías se puede conseguir los mismos resultados, se ha tomado la decisión de utilizar Java junto a alguno de sus frameworks dedicados al desarrollo web debido a que es una tecnología libre y que permite programar la aplicación con independencia de la plataforma en que será utilizada posteriormente además de ser una de las tecnologías más utilizadas actualmente para este tipo de aplicaciones.

2.8 Conclusiones

Con esta aplicación se pretende cubrir ciertas necesidades existentes en la asignatura de Robótica y Automatización Industrial transportando la metodología de trabajo actual a un entorno virtual con lo que se conseguirá una mayor eficiencia en el trabajo diario de la asignatura, más control del trabajo realizado por los alumnos y aumentar la comunicación profesor-alumno.

Por lo tanto una vez comprobados los inconvenientes y las ventajas que podría ocasionar el desarrollo de esta aplicación, se puede afirmar que es un proyecto completamente viable.

3. ANÁLISIS

En este capítulo se podrá ver el comportamiento de la aplicación así como los requisitos funcionales, que son los procesos que se pueden realizar, como crear un nuevo usuario o modificar un ejercicio, y los requisitos no funcionales, que son las condiciones que debe cumplir el proyecto, como que se cumplan los estándares, que la aplicación sea segura y que no contenga errores.

3.1 Requisitos funcionales

Los requisitos funcionales que debe cumplir el proyecto son los siguientes:

- *Eficiencia y facilidad de uso:* La aplicación debe ser intuitiva y que la navegación sea clara haciendo que el usuario se sienta cómodo al trabajar con ella.
- *Control de acceso:* Es muy importante mantener un buen nivel de seguridad para restringir el acceso a cuentas ajenas, por lo tanto es necesario un protocolo de seguridad basada en usuario y contraseña.
- *Gestión de contenidos:* Modificación e introducción de nuevo material para los alumnos, así como correcciones de los ejercicios realizados.
- *Gestión de usuarios:* El alumno podrá añadir la respuesta a todas las actividades propuestas para la asignatura.

3.2 Requisitos no funcionales

Los requisitos no funcionales que debe cumplir el proyecto son los siguientes:

- *Seguridad:* El proyecto tiene que garantizar cierta privacidad para los datos de identificación de los usuarios, por lo tanto la base de datos estará protegida por contraseña y todos los passwords introducidos por los alumnos se cifrarán utilizando el algoritmo SHA-1.
- *Control de errores:* todos los posibles errores que comentan los usuarios estarán contemplados y controlados.
- *Cumplimiento de estándares:* El proyecto deberá ajustarse al máximo a los estándares para evitar malas visualizaciones en los diferentes navegadores en que pueda ser usado.

3.3 Diagramas de casos de uso

Los diagramas de caso de uso muestran las diferentes acciones que puede realizar cada usuario sea profesor o alumno.

3.3.1 Alumno

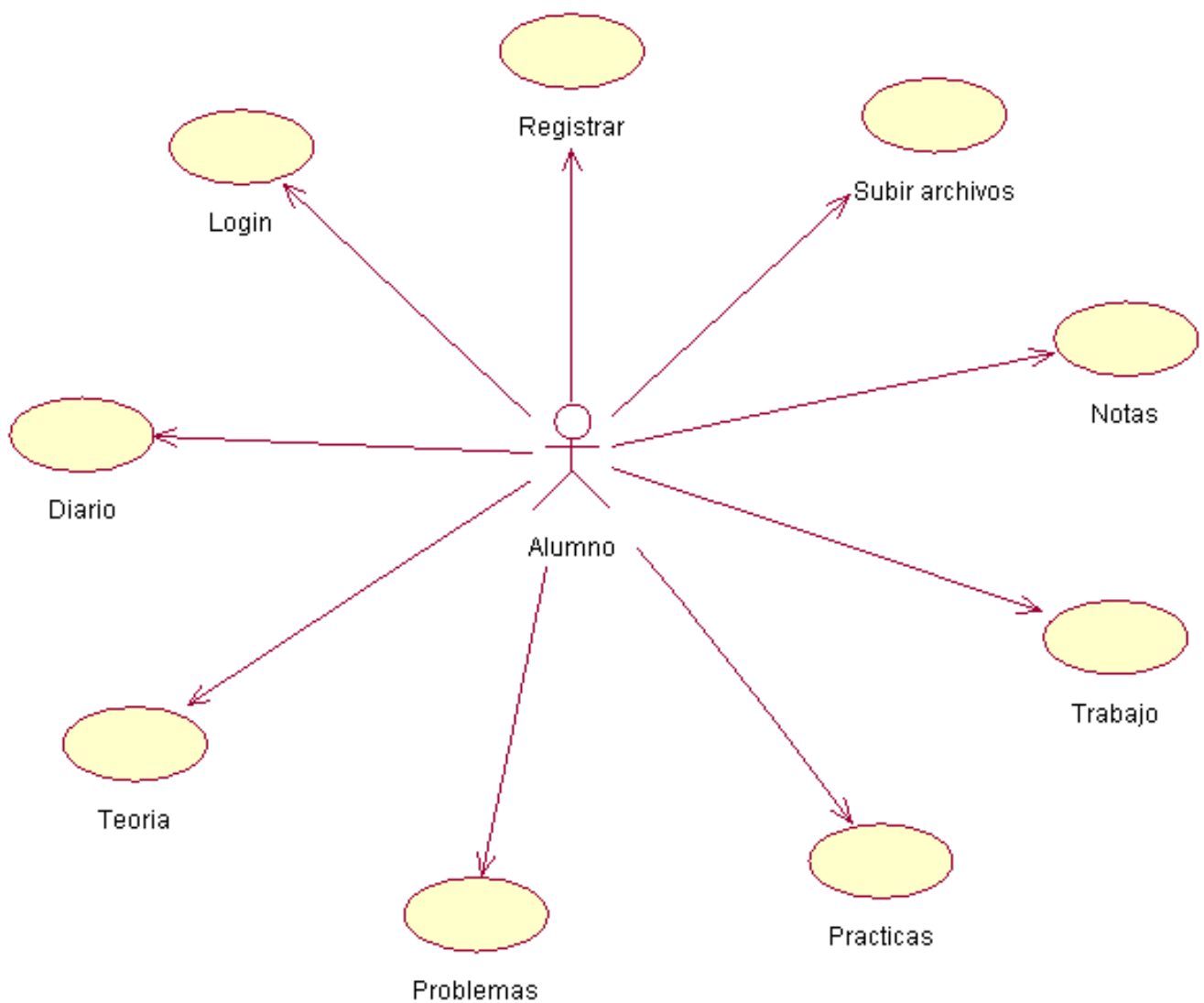


Figura 3: Diagrama de casos de uso del alumno

3.3.2 Profesor

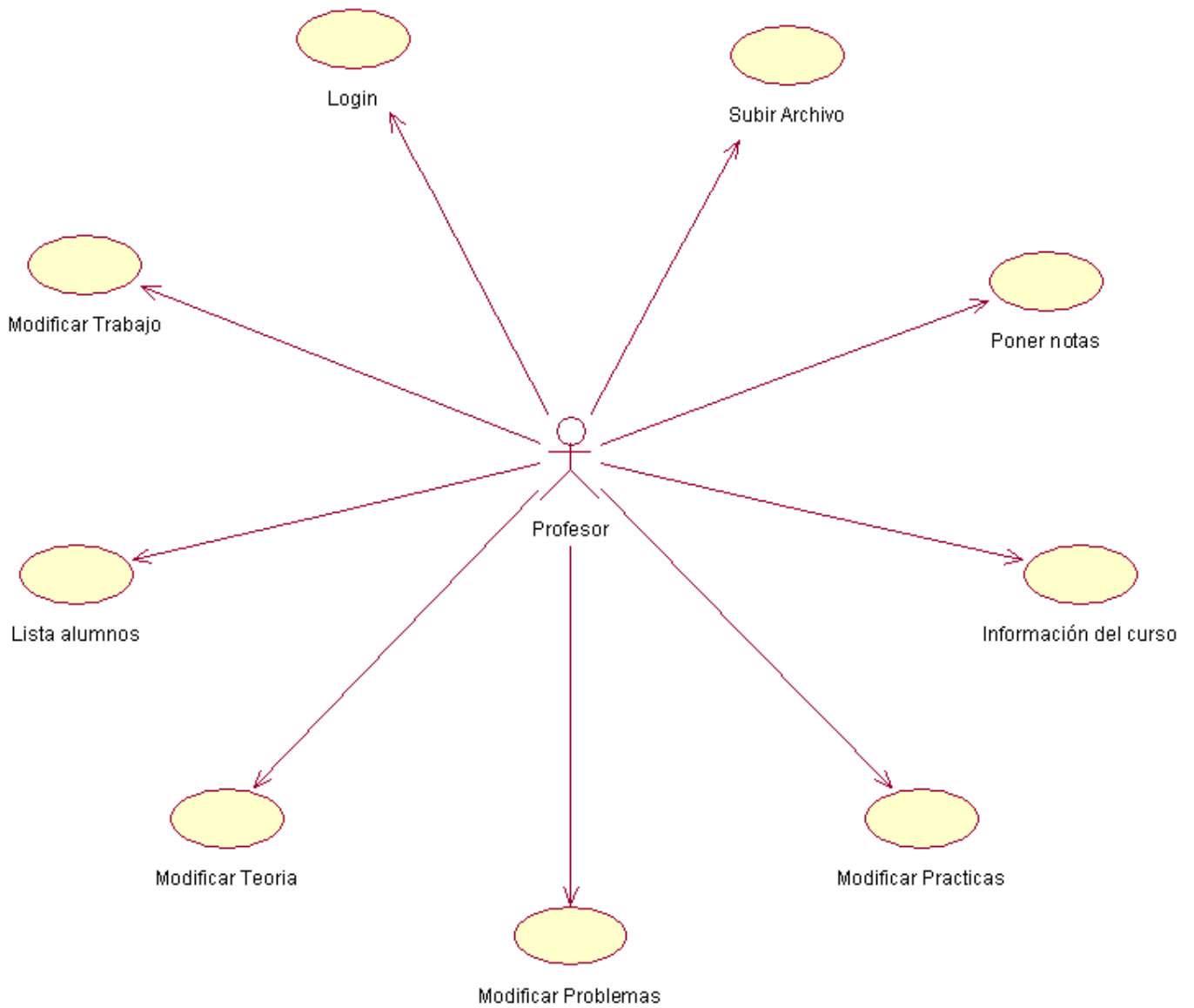


Figura 4: Diagrama de casos de uso del profesor

3.4 Descripción de los casos de uso

En este punto se verán de forma más detallada cada uno de los casos de uso que se muestran en el punto anterior.

3.4.1 Alumno

CASO DE USO	Registrar
Descripción	Todos los alumnos matriculados a la asignatura deberán registrarse para poder realizar todas las actividades relacionadas con la asignatura. Para ello deberán acceder a la página de registro haciendo clic en el botón <i>Registrar</i> de la página de <i>Login</i> .
Actores	Alumno
Flujo Principal	El navegador mostrará un pequeño formulario donde el usuario deberá introducir su NIU, una contraseña y su confirmación. Al pulsar el botón <i>Registrar</i> se guardarán los datos con los que el nuevo usuario podrá acceder a las actividades de la asignatura.
Flujos Alternativos	<u>Datos incorrectos:</u> Si hay algún dato mal introducido se muestra un mensaje de error indicando el campo incorrectamente rellenado. <u>Usuario ya existe:</u> Si el usuario introduce el NIU de otro alumno que ya existe en la base de datos se muestra el mensaje de error y no se completa el registro.

CASO DE USO	Login
Descripción	Los usuarios que ya se hayan registrado pueden loguearse para así acceder a las diferentes actividades de la asignatura. Para ello han de introducir su NIU y contraseña en el menú que aparece en la página de <i>Login</i> y pulsar el botón <i>Enviar</i> .
Actores	Alumno
Flujo Principal	Después de introducir los datos en el formulario y pulsar el botón <i>Enviar</i> en el navegador se mostrará la página principal del alumno desde la cual se podrá acceder a todos los apartados disponibles.
Flujos Alternativos	<u>Datos incorrectos:</u> Si alguno de los datos no coincide con los de la base de datos, el navegador mostrará un mensaje de error indicando al usuario que se ha equivocado introduciendo parte de la información.

ANÁLISIS

CASO DE USO	Diario
Descripción	Los usuarios podrán ver los comentarios que han ido introduciendo en días anteriores, además de poder insertar un nuevo comentario con la fecha actual.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	El usuario pulsa sobre el botón <i>Diario</i> en el menú de actividades. Aparece en el cuadro principal de la página el contenido de la pestaña <i>Visualizar</i> .
Sub-flujos	<p><u>Pestaña visualizar</u></p> <p>Se mostrará un calendario donde el usuario podrá ver las diferentes anotaciones que ha ido añadiendo a lo largo del curso, para ello tendrá que pulsar sobre el día que desea ver y que tendrá diferente color dependiendo de si contiene un comentario o no.</p> <p><u>Pestaña nueva Entrada</u></p> <p>En el navegador se mostrará un editor de texto con el que el usuario podrá introducir un nuevo comentario que tras pulsar el botón <i>Guardar</i> se añadirá a la base de datos con la fecha actual del servidor.</p>

CASO DE USO	Teoría
Descripción	Los usuarios podrán seleccionar entre los temas disponibles el ejercicio que desea solucionar de los propuestos en el apartado de teoría.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	El usuario pulsará sobre el botón <i>Preparación de Teoría</i> en el menú de actividades. Aparecerá en el cuadro principal de la página el contenido de la pestaña <i>Visualizar</i> . Se mostrará en el cuadro de actividades dos selectores para elegir el tema y el ejercicio a visualizar.
Sub-flujos	<p><u>Pestaña visualizar:</u></p> <p>El navegador mostrará el enunciado del ejercicio seleccionado y la respuesta que el usuario ha añadido.</p> <p><u>Pestaña editar:</u></p> <p>El navegador mostrará un editor de texto con el que el usuario podrá escribir la respuesta al ejercicio seleccionado. Para guardar los cambios en la base de datos se deberá pulsar el botón <i>Guardar</i>.</p>

ANÁLISIS

CASO DE USO	Problemas
Descripción	Los usuarios podrán seleccionar entre los temas disponibles el ejercicio que desean solucionar de los propuestos en el apartado de problemas.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	El usuario pulsará sobre el botón <i>Problemas</i> en el menú de actividades. Aparecerá en el cuadro principal de la página el contenido de la pestaña <i>Visualizar</i> . Se mostrará en el cuadro de actividades dos selectores para elegir el tema y el ejercicio a visualizar.
Sub-flujos	<u>Pestaña visualizar:</u> El navegador mostrará el enunciado del ejercicio seleccionado y la respuesta que el usuario ha añadido. <u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que el usuario podrá escribir la respuesta al ejercicio seleccionado. Para guardar los cambios en la base de datos se deberá pulsar el botón <i>Guardar</i> .

CASO DE USO	Prácticas
Descripción	Los usuarios podrán seleccionar entre los temas disponibles el ejercicio que desean solucionar de los propuestos en el apartado de prácticas.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	El usuario pulsará sobre el botón <i>Prácticas</i> en el menú de actividades. Aparecerá en el cuadro principal de la página el contenido de la pestaña <i>Visualizar</i> . Se mostrará en el cuadro de actividades dos selectores para elegir el tema y el ejercicio a visualizar.
Sub-flujos	<u>Pestaña visualizar:</u> El navegador mostrará el enunciado del ejercicio seleccionado y la respuesta que el usuario ha añadido. <u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que el usuario podrá escribir la respuesta al ejercicio seleccionado. Para guardar los cambios en la base de datos se deberá pulsar el botón <i>Guardar</i> .

ANÁLISIS

CASO DE USO	Trabajo
Descripción	Los usuarios podrán añadir un trabajo sobre la asignatura así como la valoración personal sobre sus compañeros en este trabajo. También podrán ver la información referente a este apartado.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	El usuario pulsará sobre el botón <i>Trabajo</i> en el menú de actividades. Aparecerá en el cuadro principal de la página la pestaña <i>Trabajo</i> con la solución del alumno a este apartado.
Sub-flujos	<p><u>Pestaña trabajo:</u> El navegador mostrará el trabajo introducido por el usuario, así como la valoración que ha hecho sobre sus compañeros.</p> <p><u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que el usuario podrá escribir el trabajo que ha realizado. Para guardar los cambios en la base de datos deberá pulsar el botón <i>Guardar</i>.</p> <p><u>Pestaña información:</u> El navegador mostrará la información necesaria sobre este apartado.</p> <p><u>Pestaña valoración de los compañeros:</u> El navegador mostrará un editor de texto con el que el usuario podrá añadir la valoración que hace sobre sus compañeros de trabajo. Para guardar los cambios en la base de datos deberá pulsar el botón <i>Guardar</i>.</p>

CASO DE USO	Notas
Descripción	Los usuarios podrán consultar las notas obtenidas en cada apartado de la asignatura, así como los criterios de puntuación que han seguido los profesores.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	El usuario pulsa sobre el botón <i>Notas</i> en el menú de <i>Actividades</i> . Aparece en el cuadro principal de la página la pestaña <i>Visualizar</i> .
Sub-flujos	<p><u>Pestaña visualizar:</u> El navegador muestra una tabla con las diferentes actividades de la asignatura y la puntuación que ha obtenido el alumno en cada una de ellas.</p> <p><u>Pestaña Información:</u> El navegador muestra los criterios de valoración y toda la información relacionada con la evaluación de la asignatura.</p>

CASO DE USO	Subir archivo
Descripción	El alumno podrá subir una fotografía a su cuenta de usuario para posteriormente poderla añadir en alguno de los ejercicios que resuelva.
Actores	Alumno
Precondición	Identificarse
Flujo Principal	Tras pulsar el icono <i>Subir archivo</i> el alumno accederá a una página donde podrá buscar la fotografía que desea subir al servidor, para que se añada a su cuenta debe pulsar el botón <i>Subir Archivo</i> y posteriormente se verá el archivo subido en la tabla inferior.
Sub-flujos	<u>Eliminar archivo:</u> El alumno podrá eliminar un archivo seleccionándolo en la tabla y pulsando el botón <i>Eliminar Archivo</i> . <u>Renombrar:</u> El alumno podrá renombrar un archivo, para ello tendrá que seleccionarlo en la tabla y escribir en cuadro de texto inferior el nombre nuevo del archivo y su extensión, a continuación deberá pulsar el botón <i>Renombrar</i> para que el archivo cambie de nombre.

3.4.2 Profesor

CASO DE USO	Login
Descripción	El profesor debe loguearse para poder acceder a cualquier apartado de la administración así como a la cuenta de cualquiera de los alumnos registrados.
Actores	Profesor
Flujo Principal	El profesor debe introducir su usuario y contraseña en el formulario de login y pulsar el botón <i>Enviar</i> a continuación, en el navegador, se mostrará la pagina principal del profesor desde la cual podrá acceder a todas las secciones de la aplicación.
Flujos Alternativos	<u>Datos incorrectos:</u> Si alguno de los datos introducidos no coincide con los de la base de datos, en el navegador se mostrará un mensaje de error indicando al usuario que se ha equivocado introduciendo parte de la información.

ANÁLISIS

CASO DE USO	Lista Alumnos
Descripción	El profesor podrá ver una lista con todos los alumnos registrados en la aplicación a través de la cual podrá acceder a cualquiera de esas cuentas.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsa sobre el botón <i>Lista de Alumnos</i> . En la parte principal de la página aparece una tabla con los enlaces a las cuentas de los alumnos y a través de los cuales el profesor puede tener total acceso a cada una de ellas y añadir las correcciones necesarias.

CASO DE USO	Información del curso
Descripción	El profesor podrá insertar la información general sobre la asignatura que el alumno deba conocer y que aparecerá en la página principal de todos los alumnos.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsa sobre el botón <i>Modificar Contenido</i> para que se abra un submenú donde se encuentra el botón <i>Información del curso</i> . En el cuadro principal de la página aparecerá la pestaña <i>Visualizar</i> .
Sub-flujos	<u>Pestaña visualizar:</u> El navegador mostrará la información ya introducida anteriormente en este apartado. <u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que el profesor podrá escribir la información a mostrar. Para guardar los cambios en la base de datos se deberá pulsar el botón <i>Guardar</i> .

CASO DE USO	Modificar Teoría
Descripción	El profesor podrá añadir, modificar o eliminar ejercicios relacionados con el apartado de teoría de la asignatura.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsará sobre el botón <i>Modificar Contenido</i> para que se abra el submenú donde se encuentra el botón <i>Teoría</i> , al pulsar ese botón se abrirá en el cuadro de actividades dos selectores para poder escoger el tema y el ejercicio a visualizar de los que se hayan introducido anteriormente.
Sub-flujos	<p><u>Pestaña visualizar:</u> El navegador mostrará el enunciado del ejercicio seleccionado en los selectores del cuadro de actividades.</p> <p><u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que se podrá escribir el enunciado del ejercicio o modificar el ya seleccionado. También habrá dos pequeñas áreas de texto donde se pondrá el número de tema y ejercicio, además de un botón que hará que el ejercicio sea visible o no para los alumnos.</p> <p><u>Pestaña Eliminar:</u> El navegador mostrará una tabla con todos los ejercicios introducidos separados por tema, pudiendo seleccionar cada uno de ellos para ser eliminado de la base de datos tras pulsar el botón <i>Eliminar</i>.</p>
Flujos Alternativos	<p><u>Número de ejercicio ya existe:</u> Si se introduce un número de ejercicio que ya existe para ese tema, se mostrará por pantalla un error indicando que el número de ejercicio está equivocado.</p> <p><u>No se pone número de ejercicio/tema:</u> Si no se modifica el número de ejercicio o el número de tema, se mostrará por pantalla un error indicando el problema.</p>

CASO DE USO	Modificar Problemas
Descripción	El profesor podrá añadir, modificar o eliminar ejercicios relacionados con el apartado de problemas de la asignatura.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsará sobre el botón <i>Modificar Contenido</i> para que se abra el submenú donde se encuentra el botón <i>Problemas</i> , al pulsar ese botón se abrirá en el cuadro de actividades dos selectores para poder escoger el tema y el ejercicio a visualizar de los que se hayan introducido anteriormente.
Sub-flujos	<p><u>Pestaña visualizar:</u> El navegador mostrará el enunciado del ejercicio seleccionado en los selectores del cuadro de actividades.</p> <p><u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que se podrá escribir el enunciado del ejercicio o modificar el ya seleccionado. También habrá dos pequeñas áreas de texto donde se pondrá el número de tema y ejercicio, además de un botón que hará que el ejercicio sea visible o no para los alumnos.</p> <p><u>Pestaña Eliminar:</u> El navegador mostrará una tabla con todos los ejercicios introducidos separados por tema, pudiendo seleccionar cada uno de ellos para ser eliminado de la base de datos tras pulsar el botón <i>Eliminar</i>.</p>
Flujos Alternativos	<p><u>Número de ejercicio ya existe:</u> Si se introduce un número de ejercicio que ya existe para ese tema, se mostrará por pantalla un error indicando que el número de ejercicio está equivocado.</p> <p><u>No se pone número de ejercicio/tema:</u> Si no se modifica el número de ejercicio o el número de tema, se mostrará por pantalla un error indicando el problema.</p>

CASO DE USO	Modificar Prácticas
Descripción	El profesor podrá añadir, modificar o eliminar ejercicios relacionados con el apartado de prácticas de la asignatura.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsará sobre el botón <i>Modificar Contenido</i> para que se abra el submenú donde se encuentra el botón <i>Prácticas</i> , al pulsar ese botón se abrirá en el cuadro de actividades dos selectores para poder escoger el tema y el ejercicio a visualizar de los que se hayan introducido anteriormente.
Sub-flujos	<p><u>Pestaña visualizar:</u> El navegador mostrará el enunciado del ejercicio seleccionado en los selectores del cuadro de actividades.</p> <p><u>Pestaña editar:</u> El navegador mostrará un editor de texto con el que se podrá escribir el enunciado del ejercicio o modificar el ya seleccionado. También habrá dos pequeñas áreas de texto donde se pondrá el número de tema y ejercicio, además de un botón que hará que el ejercicio sea visible o no para los alumnos.</p> <p><u>Pestaña Eliminar:</u> El navegador mostrará una tabla con todos los ejercicios introducidos separados por tema, pudiendo seleccionar cada uno de ellos para ser eliminado de la base de datos tras pulsar el botón <i>Eliminar</i>.</p>
Flujos Alternativos	<p><u>Número de ejercicio ya existe:</u> Si se introduce un número de ejercicio que ya existe para ese tema, se mostrará por pantalla un error indicando que el número de ejercicio está equivocado.</p> <p><u>No se pone número de ejercicio/tema:</u> Si no se modifica el número de ejercicio o el número de tema, se mostrará por pantalla un error indicando el problema.</p>

ANÁLISIS

CASO DE USO	Modificar Trabajo
Descripción	El profesor podrá añadir la información necesaria para que el alumno realice el trabajo sobre la asignatura.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsará sobre el botón <i>Modificar Contenido</i> para que se abra un submenú donde se encuentra el botón <i>Trabajo</i> . En el cuadro principal de la página aparecerá la pestaña <i>Visualizar</i> .
Sub-flujos	<p><u>Pestaña visualizar:</u> El navegador mostrará la información ya introducida anteriormente en este apartado.</p> <p><u>Pestaña editar información:</u> El navegador mostrará un editor de texto con el que el profesor podrá escribir la información a mostrar. Para guardar los cambios en la base de datos se deberá pulsar el botón <i>Guardar</i>.</p>

CASO DE USO	Poner Notas
Descripción	El profesor podrá poner una nota en cada uno de los apartados de la asignatura a cada uno de los alumnos. También podrá añadir información acerca de la forma de evaluación de la asignatura.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	El profesor pulsará sobre el botón <i>Notas</i> en el menú de actividades y el navegador mostrará una lista con todos los alumnos registrados en la aplicación, a los cuales se podrá puntuar según su trabajo realizado en la asignatura.
Sub-flujos	<p><u>Pestaña visualizar:</u> Se mostrará una lista con todos los alumnos registrados y los diferentes apartados en que se divide la asignatura para que sean puntuados. Los cambios se guardarán en la base de datos después de pulsar el botón <i>Guardar</i>.</p> <p><u>Pestaña Información:</u> Se mostrará en el navegador la información referente al criterio de evaluación seguido en la asignatura.</p> <p><u>Pestaña Editar Información:</u> Se mostrará un editor de textos con el que se podrá insertar o modificar la información que hace referencia a la evaluación. Los cambios se guardarán en la base de datos cuando se pulse el botón <i>Guardar Información</i>.</p>

CASO DE USO	Subir archivo
Descripción	El profesor podrá subir una fotografía a su cuenta de usuario para posteriormente poderla añadir en alguno de los enunciados que inserte.
Actores	Profesor
Precondición	Identificarse
Flujo Principal	Tras pulsar el icono <i>Subir archivo</i> el profesor accederá a una página donde podrá buscar la fotografía que desea subir al servidor, para que se añada a su cuenta debe pulsar el botón <i>Subir Archivo</i> y posteriormente se verá el archivo subido en la tabla inferior.
Sub-flujos	<p><u>Eliminar archivo:</u> El profesor podrá eliminar un archivo seleccionándolo en la tabla y pulsando el botón <i>Eliminar Archivo</i>.</p> <p><u>Renombrar:</u> El profesor podrá renombrar un archivo, para ello tendrá que seleccionarlo en la tabla y escribir en cuadro de texto inferior el nombre nuevo del archivo y su extensión, a continuación deberá pulsar el botón <i>Renombrar</i> para que el archivo cambie de nombre.</p>

4. HERRAMIENTAS UTILIZADAS

En este apartado se comentarán las diferentes herramientas utilizadas para el desarrollo del proyecto, además de los lenguajes de programación usados para su realización.

4.1 Struts 2

Para el desarrollo principal del proyecto se ha optado por Struts 2 (*Struts 2 in action*) que es un framework de Java dedicado al desarrollo web. Struts 2 está basado en el patrón MVC (Modelo-Vista-Controlador) que divide la aplicación en tres capas claramente diferenciadas:

- *El modelo:* Es el encargado de procesar las acciones solicitadas por el controlador. Es decir, dentro del modelo se encuentran las clases encargadas de realizar las operaciones necesarias.
- *La vista:* Es la parte encargada de generar la interfaz con la que la aplicación interacciona con el usuario, es decir, son el conjunto de páginas JSP que puede ver el usuario desde su navegador.
- *El Controlador:* Es el encargado de comunicar la vista y el modelo respondiendo a los eventos generados por el usuario en la vista, haciendo que se produzcan los cambios en el modelo y devolviendo los resultados de nuevo a la vista.

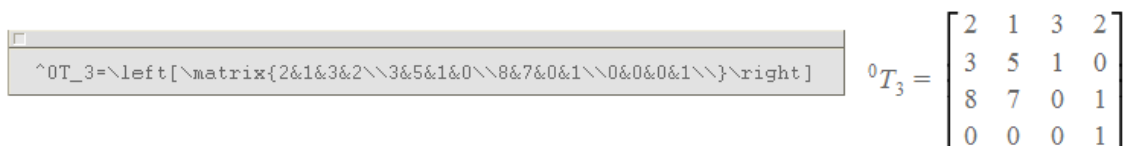
4.2 JavaScript

Pese a que el proyecto gira alrededor de Struts 2 no hay que olvidar que se pueden utilizar otras tecnologías que enriquezcan el contenido web y mejoren la experiencia del usuario. Con esta idea se utilizará JavaScript (*La biblia de JavaScript*), que es un lenguaje interpretado, que no necesita compilación, y que es ampliamente utilizado en aplicaciones web.

Dentro del proyecto se ha utilizado para integrar el editor de textos así como para mejorar la presentación dando diferentes efectos en los menús.

4.2.1 JSMath

Para poder introducir las fórmulas matemáticas necesarias para la asignatura, se ha decidido utilizar JSMath que es una implementación de Tex en JavaScript. JSMath tiene la particularidad de que es capaz de generar las fórmulas usando directamente tipografía Tex o con sus propias fuentes nativas, en vez de usar imágenes como es típico en HTML. La siguiente imagen muestra un ejemplo con JSMath:



The image shows a code editor window on the left containing the LaTeX code: `^0T_3=\left[\matrix{2&1&3&2\\3&5&1&0\\8&7&0&1\\0&0&0&1}\right]`. To the right of the code editor, the rendered output is displayed as a 4x4 matrix: ${}^0T_3 = \begin{bmatrix} 2 & 1 & 3 & 2 \\ 3 & 5 & 1 & 0 \\ 8 & 7 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

Figura 5: Ejemplo de JSMath

4.2.2 Prototype

Prototype es un framework escrito en JavaScript y que se utiliza para implementar aplicaciones AJAX.

En el proyecto únicamente se ha utilizado la función $\$F()$ para traspasar la información entre el editor de textos y las pestañas de visualización.

4.2.3 Tiny_MCE

Para hacer más cómoda la entrada de información en la aplicación se ha optado por utilizar un editor de textos WYSIWYG. Tiny_MCE, que está escrito en JavaScript, cumple ese requisito ya que convierte los textareas de la aplicación en ese tipo de editor. El aspecto que presenta es similar a la barra de herramientas de Microsoft Word y permite dar formato al texto viendo directamente el resultado final.

4.3 CSS

El lenguaje utilizado para definir la presentación de la aplicación es el CSS (*Manual avanzado de CSS*). A partir de un único archivo permite definir la forma y la posición que ocupará cada uno de los elementos mostrados en la página facilitando modificaciones en el apartado visual de la aplicación de forma rápida y sencilla.

4.4 Jasypt

Como se ha comentado en capítulos anteriores, la seguridad es un tema importante en este tipo de aplicaciones, por ese motivo se ha decidido cifrar las contraseñas introducidas por los alumnos en la base de datos. Jasypt (<http://www.jasypt.org/getting-started-easy.html>) es una librería java que sirve para ese propósito.

En el proyecto se ha decidido utilizar para el cifrado el algoritmo SHA-1 que ofrece un buen nivel de seguridad.

4.5 MySQL

MySQL (<http://www.mysql.com/>) es un gestor de base de datos relacional que como el resto de opciones para este proyecto es de código abierto. Se ha escogido este gestor porque es además rápido, fiable y robusto además de que trabaja bien tanto con grandes como con pequeños volúmenes de datos. MySQL es una opción ampliamente usada en aplicaciones web y puede ser usado en diferentes plataformas.

4.5.1 MySQL Administrator

Es un software multiplataforma y libre que permite administrar una base de datos MySQL de forma visual.

4.6 Hibernate

Es un motor de persistencia de código abierto que facilita el mapeo de atributos de una base de datos relacional y los objetos de una aplicación utilizando archivos XML para establecer las relaciones, en otras palabras, se utiliza para hacer persistente los objetos de un programa en una base de datos relacional.

Se ha optado por esta opción para evitar el uso de JDBC ya que es más complejo y dependiente de la estructura de los datos, de manera que utilizando Hibernate (<http://www.hibernate.org>) se utiliza una forma de trabajar más natural y sencilla utilizando directamente objetos.

4.7 Tomcat

Tomcat (<http://tomcat.apache.org/>) es un servidor web con soporte para JSP y Servlets. Este servidor es libre, gratuito y multiplataforma y será el encargado de interpretar el código de la aplicación para ser entregado al navegador del usuario.

En el proyecto tendrá la función de contenedor de páginas JSP con sus respectivos CSS, también se encargará de sostener el sistema Struts 2, gestionar la seguridad mediante el protocolo SSL e interactuar con la base de datos.

4.8 Eclipse

Es el entorno de desarrollo elegido para implementar el proyecto, al igual que el resto de herramientas utilizadas es de código abierto y multiplataforma. Desde Eclipse se puede programar en múltiples lenguajes entre los que se encuentran JSP, JavaScript, Java y XML.

4.9 OpenSSL

Es una herramienta que sirve para crear certificados para usarlos con aplicaciones web. Esos certificados aseguran que las credenciales de la aplicación no son fraudulentas.

DISEÑO DE LA APLICACIÓN

5. DISEÑO DE LA APLICACIÓN

En este capítulo se verá detalladamente tanto la base de datos usada para el proyecto, así como la interfaz grafica que presenta la aplicación tanto en el apartado de administrador como en el de usuario.

5.1 Base de datos

La base de datos es una de las partes más importantes del proyecto porque sobre ella se mantiene todo el peso de la aplicación, ya que tanto profesores como alumnos tendrán que acceder para recuperar o insertar información, y un fallo podría desestabilizar toda la aplicación.

5.1.1 Diagrama

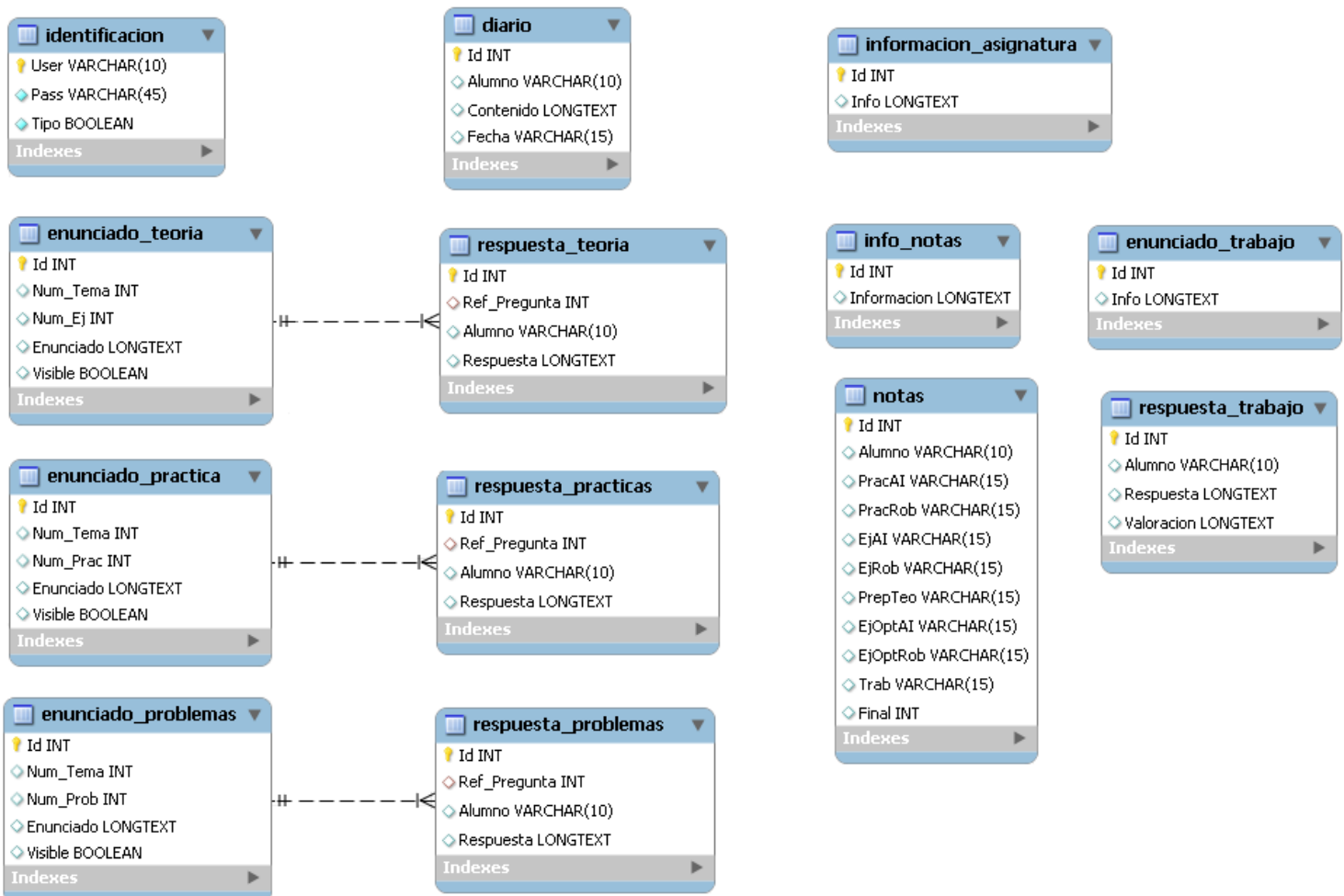


Figura 6: Diagrama de la base de datos

DISEÑO DE LA APLICACIÓN

5.1.2 Estructura

En este apartado se mostrarán con más detalle las diferentes tablas que forman parte de la base de datos.

Tabla identificación: Esta tabla contiene toda la información necesaria para identificarse como profesor o como alumno.

Campo	Tipo	Descripción
<i>User</i>	Varchar	Clave primaria. Es el campo que contiene el nombre de usuario de la cuenta tanto de profesor como de alumnos. Para los alumnos se usará el NIU
<i>Pass</i>	Varchar	Contraseña de las cuentas de profesor y alumnos
<i>Tipo</i>	Booleano	Indica el tipo de usuario de la cuenta, 1 si es profesor y 0 si es alumno.

Tabla diario: Contiene las entradas que han introducido los alumnos en el diario.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas.
<i>Alumno</i>	Varchar	Guarda el usuario al que pertenece la entrada
<i>Contenido</i>	Longtext	Contiene el comentario introducido por el alumno.
<i>Fecha</i>	Varchar	Este campo contiene la fecha en que se añadió el comentario.

Tabla respuesta_teoría: Esta tabla contiene la respuesta de cada alumno a los ejercicios propuestos en el apartado de teoría.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Ref_Pregunta</i>	Int	Este campo indica el identificador que tiene la pregunta a la que pertenece la respuesta.
<i>Alumno</i>	Varchar	Guarda el usuario al que pertenece la entrada.
<i>Respuesta</i>	Longtext	Este campo contiene la respuesta que ha propuesto el alumno para el ejercicio.
<i>Modificación</i>	Varchar	Este campo contiene la fecha de la última modificación del alumno a este ejercicio.

DISEÑO DE LA APLICACIÓN

Tabla respuesta_practicas: Esta tabla contiene la respuesta de cada alumno a las tareas propuestas en el apartado de prácticas.

Campo	Tipo	Descripción
Id	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
Ref_Pregunta	Int	Este campo indica el identificador que tiene la pregunta a la que pertenece la respuesta.
Alumno	Varchar	Guarda el usuario al que pertenece la entrada.
Respuesta	Longtext	Este campo contiene la respuesta que ha propuesto el alumno para el ejercicio.
Modificación	Varchar	Este campo contiene la fecha de la última modificación del alumno a este ejercicio.

Tabla respuesta_problemas: Esta tabla contiene la respuesta a los ejercicios propuestos en el apartado de problemas de la asignatura.

Campo	Tipo	Descripción
Id	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
Ref_Pregunta	Int	Este campo indica el identificador que tiene la pregunta a la que pertenece la respuesta.
Alumno	Varchar	Guarda el usuario al que pertenece la entrada.
Respuesta	Longtext	Este campo contiene la respuesta que ha propuesto el alumno para el ejercicio.
Modificación	Varchar	Este campo contiene la fecha de la última modificación del alumno a este ejercicio.

Tabla respuesta_trabajo: Contiene el trabajo sobre la asignatura que ha hecho cada alumno además de la valoración de sus compañeros.

Campo	Tipo	Descripción
Id	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
Alumno	Varchar	Guarda el usuario que ha realizado la entrada.
Valoración	Longtext	Contiene la valoración que ha hecho el alumno de sus compañeros de trabajo.
Respuesta	Longtext	Este campo contiene el trabajo realizado por el alumno.
Modificación	Varchar	Este campo contiene la fecha de la última modificación de esta entrada.

DISEÑO DE LA APLICACIÓN

Tabla notas: Contiene la puntuación del alumno en los diferentes apartados de la asignatura.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Alumno</i>	Varchar	Guarda el usuario al que hacen referencia las notas de esta entrada.
<i>PracAI</i>	Varchar	Contiene la nota de las prácticas de Automatización Industrial.
<i>PracRob</i>	Varchar	Contiene la nota de las prácticas de Robótica.
<i>EjAI</i>	Varchar	Contiene la nota de los problemas obligatorios de la parte de Automatización Industrial.
<i>EjRob</i>	Varchar	Contiene la nota de los problemas obligatorios de la parte de Robótica.
<i>PrepTeo</i>	Varchar	Contiene la nota de los ejercicios de teoría.
<i>EjOptAI</i>	Varchar	Contiene la nota de los ejercicios optativos de la parte de Automatización Industrial.
<i>EjOptRob</i>	Varchar	Contiene la nota de los ejercicios optativos de la parte de Robótica.
<i>Trab</i>	Varchar	Contiene la nota del trabajo sobre la asignatura.
<i>Final</i>	Int	Contiene la nota final obtenida por el alumno en la asignatura.

Tabla info_notas: Esta tabla contiene la información referente al método de evaluación de la asignatura.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Información</i>	Longtext	Contiene los datos relacionados con el método de evaluación de la asignatura.

Tabla información_asignatura: Contiene la información general referente a la asignatura.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Info</i>	Longtext	Contiene la información general de la asignatura.

DISEÑO DE LA APLICACIÓN

Tabla enunciado_teoría: Contiene los ejercicios del apartado de teoría que tendrán que solucionar los alumnos.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Num_Tema</i>	Int	Indica el tema al que pertenece el ejercicio.
<i>Num_Ej</i>	Int	Guarda el número de ejercicio.
<i>Enunciado</i>	Longtext	Guarda el enunciado del ejercicio.
<i>Visible</i>	Booleano	Indica si el ejercicio estará disponible para los alumnos.

Tabla enunciado_problemas: Contiene los ejercicios del apartado de problemas.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Num_Tema</i>	Int	Indica el tema al que pertenece el problema.
<i>Num_Prob</i>	Int	Guarda el número de problema.
<i>Enunciado</i>	Longtext	Guarda el enunciado del problema.
<i>Visible</i>	Booleano	Indica si el problema estará disponible para los alumnos.

Tabla enunciado_practica: Contiene los ejercicios del apartado de prácticas.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Num_Tema</i>	Int	Indica el tema al que pertenece el ejercicio.
<i>Num_Prac</i>	Int	Guarda el número de ejercicio.
<i>Enunciado</i>	Longtext	Guarda el enunciado del ejercicio.
<i>Visible</i>	Booleano	Indica si el ejercicio estará disponible para los alumnos.

Tabla enunciado_trabajo: Contiene la información referente al trabajo sobre la asignatura que deberán hacer los alumnos.

Campo	Tipo	Descripción
<i>Id</i>	Int	Clave primaria. Campo que identifica cada una de las entradas de la tabla.
<i>Info</i>	Longtext	Contiene los datos relacionados con el trabajo a realizar.

DISEÑO DE LA APLICACIÓN

5.2 Interfaz gráfica

Para implementar la interfaz gráfica se ha diseñado un archivo CSS desde cero junto con JavaScript para conseguir los efectos gráficos deseados en algunos botones de la aplicación.

5.2.1 Inicio

En la figura 7 se puede contemplar la página de entrada a la aplicación, donde profesores y alumnos podrán acceder a sus respectivas cuentas.

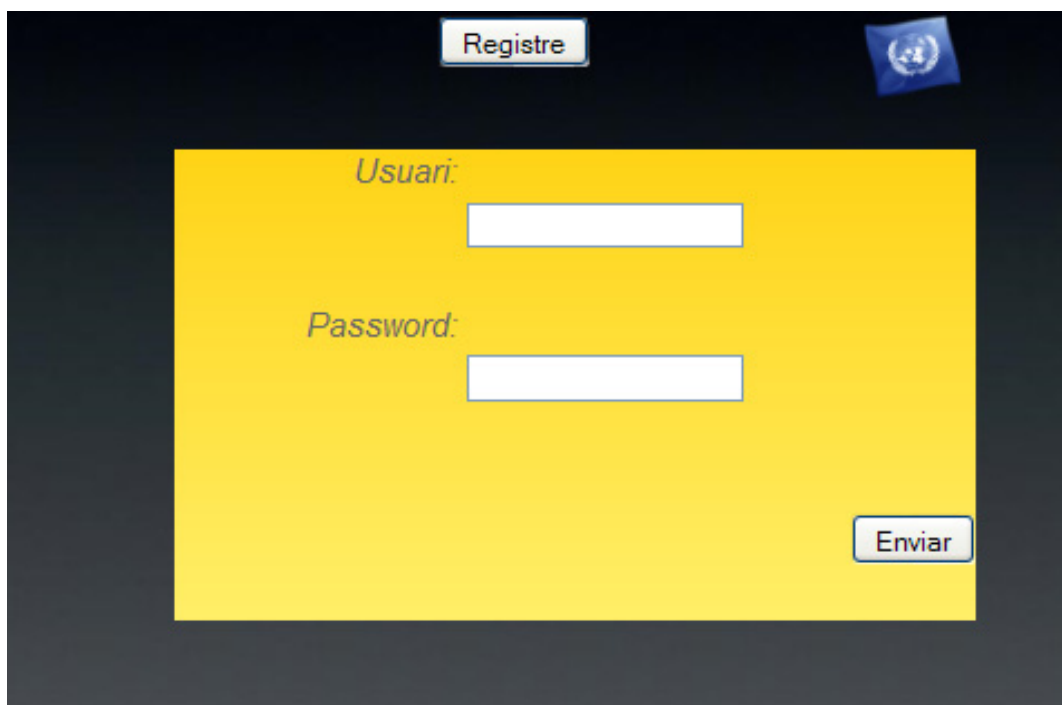


Figura 7: Página principal de Login

DISEÑO DE LA APLICACIÓN

Desde esta página principal el usuario podrá escoger el idioma con el que desea trabajar en la aplicación.

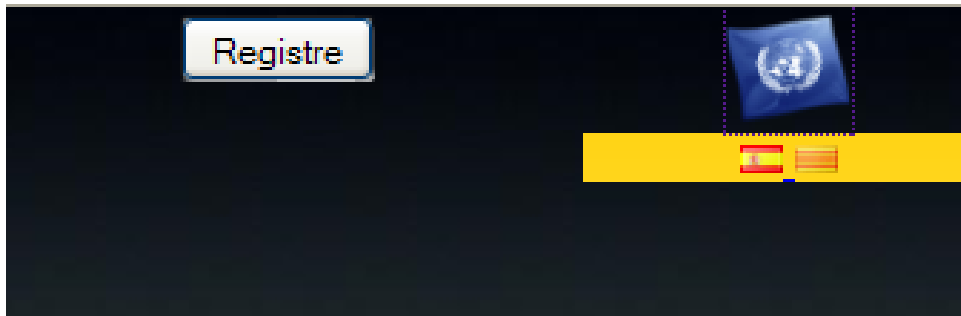


Figura 8: Selección de idiomas

En la esquina superior se encuentra también un botón que abre la página de registro. Desde dicha página todos los alumnos deberán registrarse antes de poder usar la aplicación.

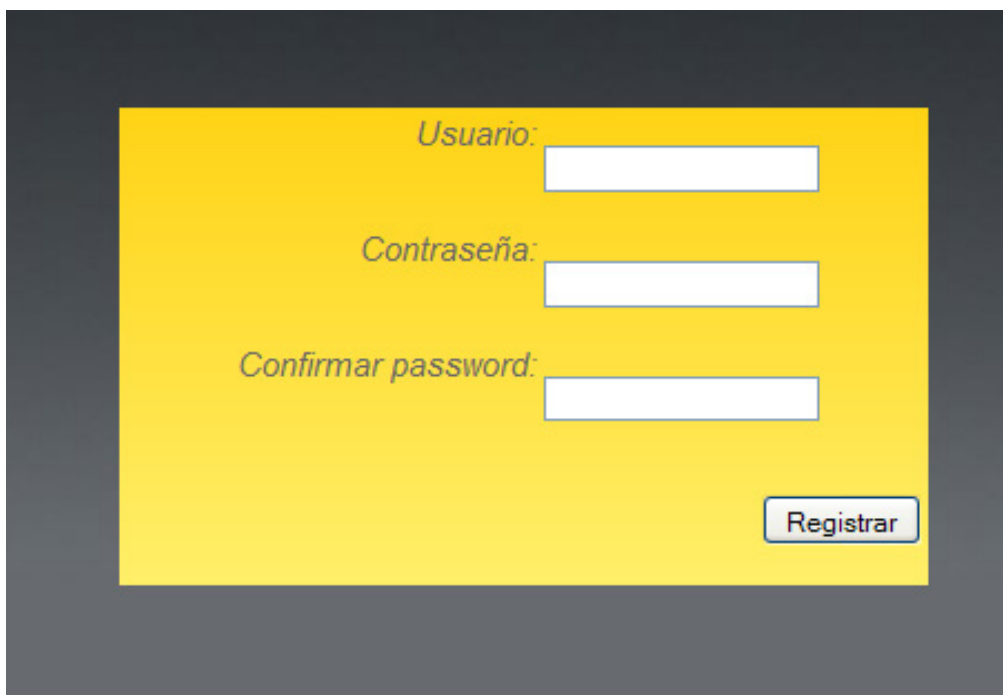
A screenshot of a registration form. The form is set against a yellow background. It contains three input fields: 'Usuario:', 'Contraseña:', and 'Confirmar password:'. Each field is followed by a white rectangular input box. In the bottom right corner of the form, there is a button labeled 'Registrar'.

Figura 9: Página de registro

DISEÑO DE LA APLICACIÓN

5.2.2 Alumno

Después de registrarse, el alumno accederá a la página principal de su cuenta.

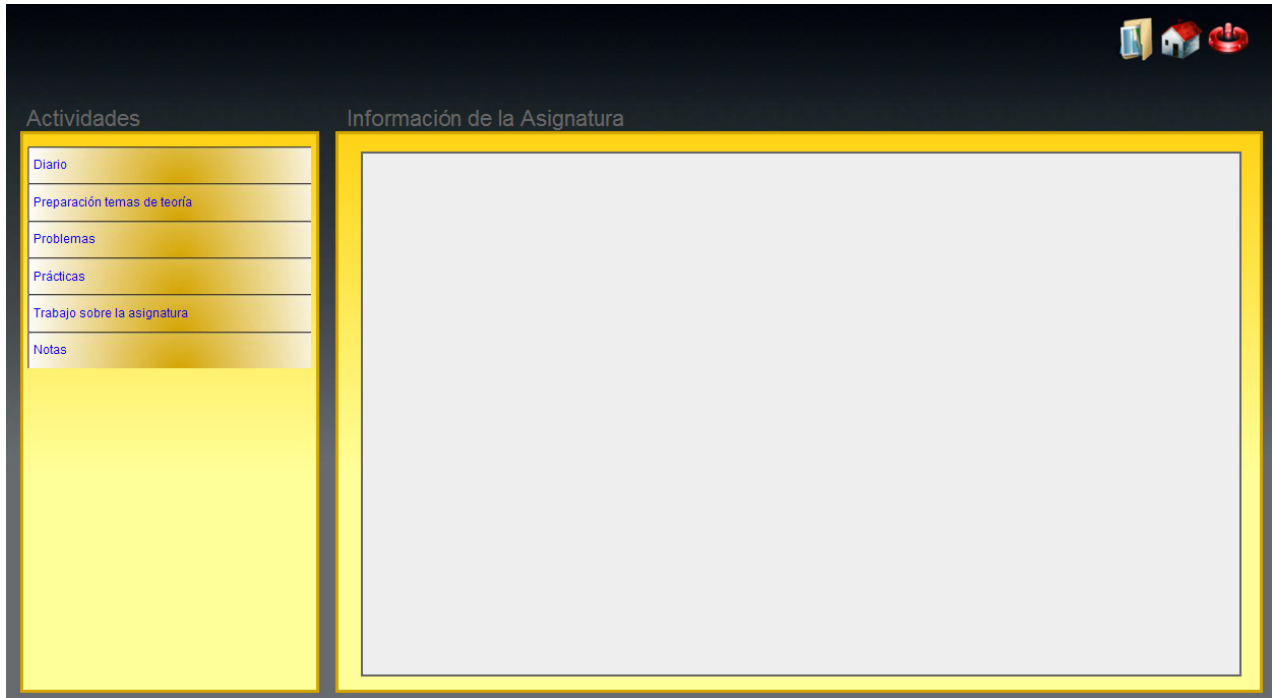


Figura 10: Página principal del alumno

La aplicación consta de un menú principal donde el alumno podrá seleccionar las actividades que desea realizar.

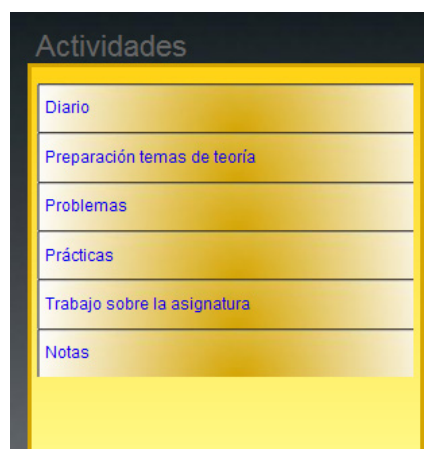


Figura 11: Menú de actividades

DISEÑO DE LA APLICACIÓN

Al pulsar sobre alguno de los apartados donde se tienen que realizar los ejercicios, como son la sección de *problemas*, *prácticas* o *preparación de temas de teoría*, se desplegará un submenú donde el alumno podrá seleccionar el tema y ejercicio que desea realizar de los que haya disponibles.

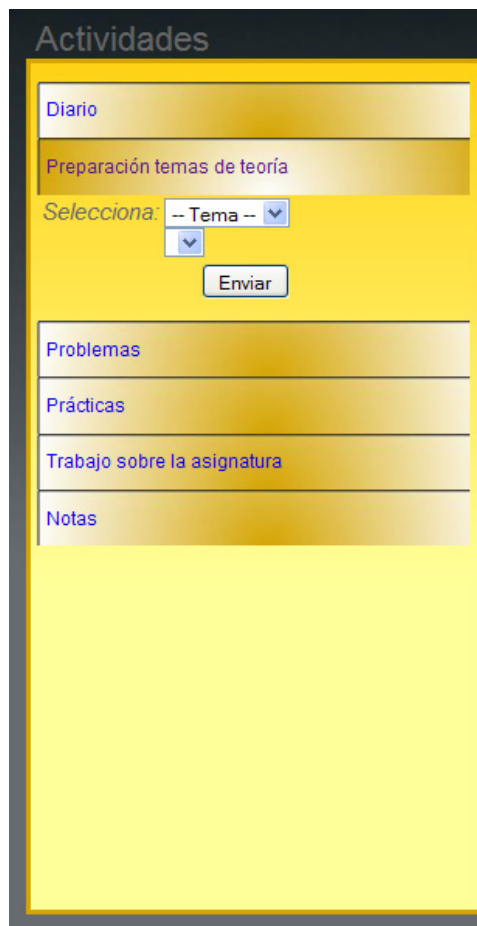


Figura 12: Menú de actividades desplegado

En la zona central de la página, se encuentra el cuadro principal donde el alumno trabajará con la aplicación, dependiendo de la actividad seleccionada en el menú de *Actividades*. Las pestañas de esta zona central variarán dependiendo de la opción seleccionada, como se puede ver en las siguientes imágenes:

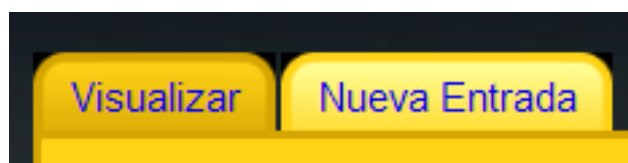


Figura 13: Pestañas del apartado de diario

DISEÑO DE LA APLICACIÓN

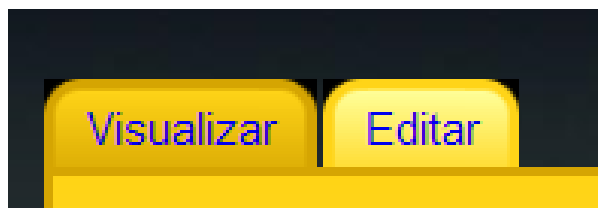


Figura 14: Pestañas del apartado de Teoría

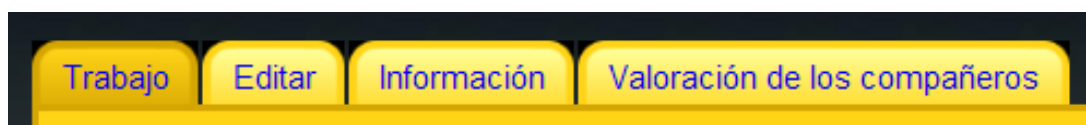


Figura 15: Pestañas del apartado de Trabajo

5.2.2.1 Diario

Cuando el alumno selecciona en el menú de *actividades* la opción *Diario*, se muestra en el panel central una tabla con los días del mes seleccionado en el selector superior y el año indicado en la zona inferior.



Figura 16: Apartado del diario

DISEÑO DE LA APLICACIÓN

Los días del diario presentarán diferente color de fondo en función de si se ha añadido un comentario o no en ese día.

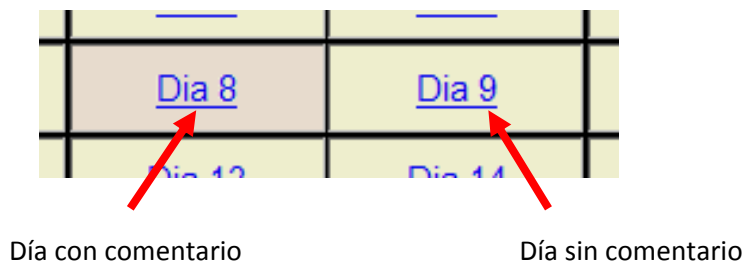


Figura 17: Detalle de los días en el diario

Si se pulsa sobre uno de los días, se desplegará una pantalla donde se podrá ver el comentario introducido.

Finalmente, para añadir o modificar un comentario con la fecha actual, desde la pestaña *Nueva Entrada* se accederá al editor de texto desde el que se podrá introducir la información.

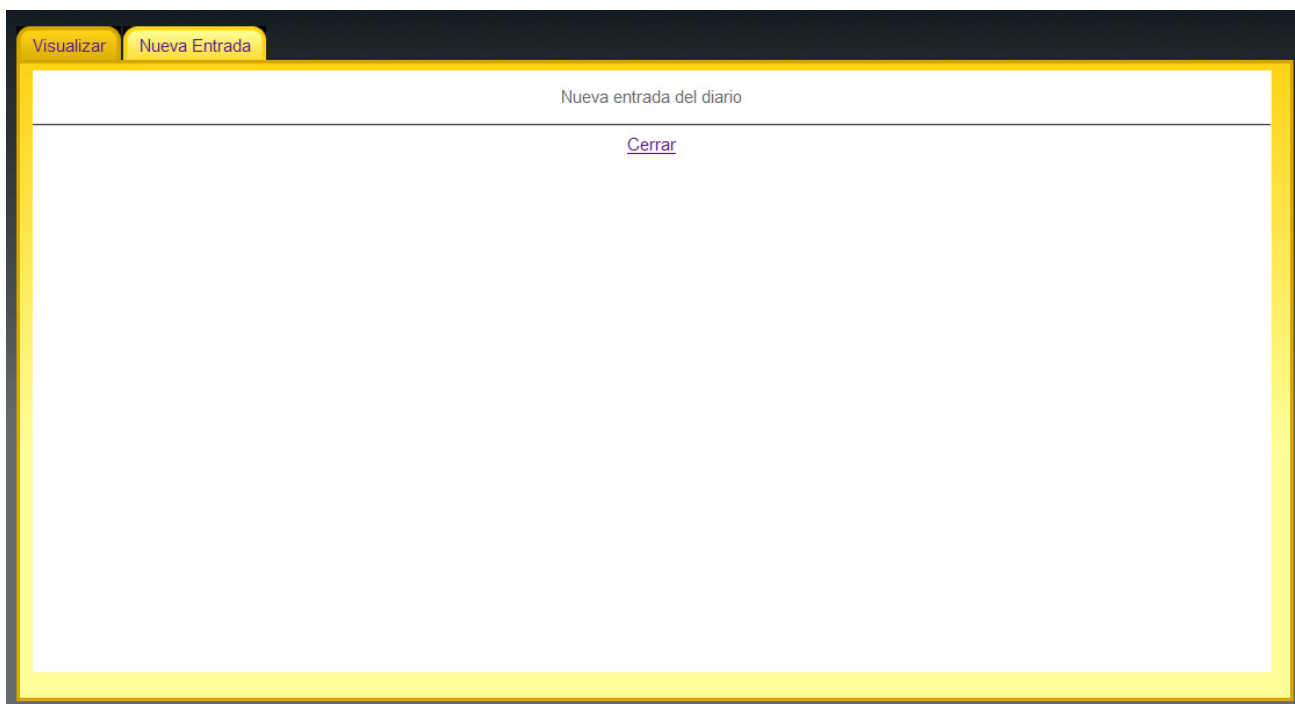


Figura 18: Comentario en un día del diario

DISEÑO DE LA APLICACIÓN

5.2.2.2 Notas

Este sub-apartado mostrará una tabla con las notas del alumno en cada una de las actividades del curso.

Prácticas de Laboratorio de Automatización Industrial	No Presentado
Ejercicios Obligatorios de Automatización Industrial	No Presentado
Prácticas de Robótica	No Presentado
Ejercicios Obligatorios de Robótica	No Presentado
Preparación de Temas de Teoría	No Presentado
Ejercicios Optativos de Automatización Industrial	No Presentado
Ejercicios Optativos de Robótica	No Presentado
Desarrollo y Presentación de un Trabajo de Libre Elección	No Presentado
Nota Final	No Presentado

Figura 19: Puntuación del alumno

5.2.3 Profesor

La página del profesor cuenta con el mismo apartado visual que la sección del alumno. En este caso el menú principal contiene otras opciones, como se puede ver en la siguiente imagen:

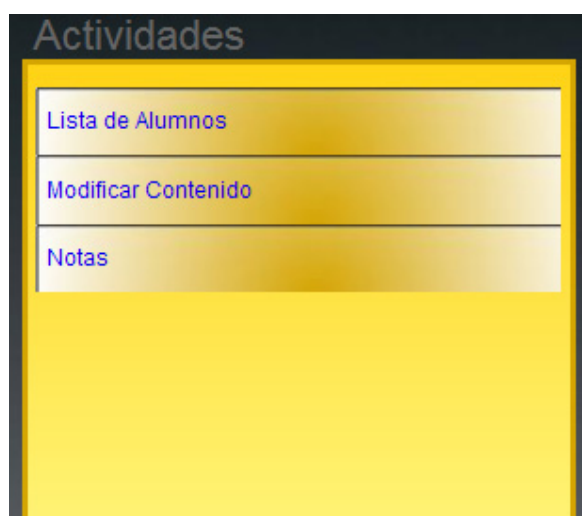


Figura 20: Menú de actividades

DISEÑO DE LA APLICACIÓN

Este menú contiene diferentes niveles que se abren dependiendo de la opción escogida. Al pulsar sobre el botón *Modificar Contenido* se despliega un submenú con los diferentes apartados que el profesor puede modificar.



Figura 21: Menú de actividades desplegado.

Igual que en la sección del alumno, los botones de *Teoría*, *Problemas* y *Prácticas* desplegarán unos selectores que permitirán escoger un tema y un ejercicio de los que ya se hayan guardado anteriormente en la base de datos con el fin de modificarlos.



Figura 22: Menú profesor con sección abierta

DISEÑO DE LA APLICACIÓN

5.2.3.1 Lista de alumnos

Desde este apartado el profesor podrá seleccionar la cuenta del alumno que quiere ver.

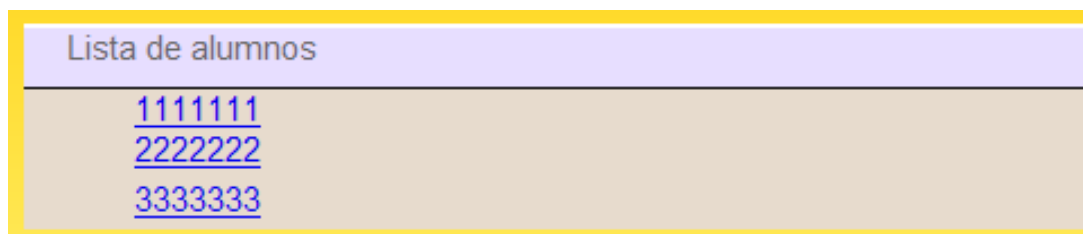


Figura 23: Lista de alumnos registrados

5.2.3.2 Notas

Desde este apartado el profesor podrá introducir las notas de los alumnos registrados en la aplicación. También desde esta sección, en la pestaña *Editar Información*, podrá añadir la información referente al método de evaluación seguido.

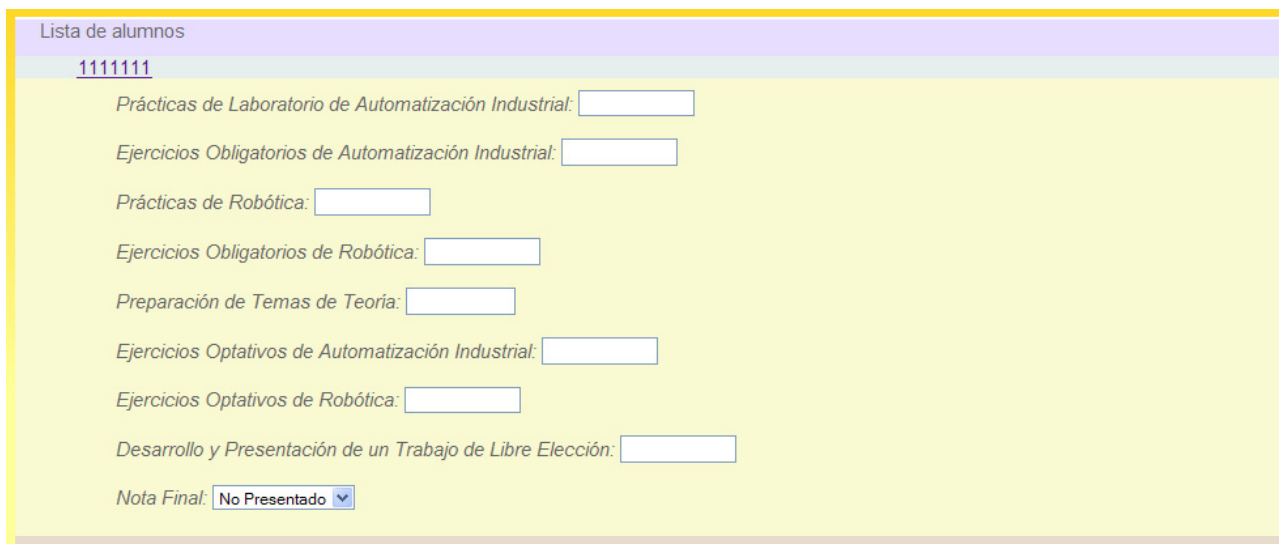
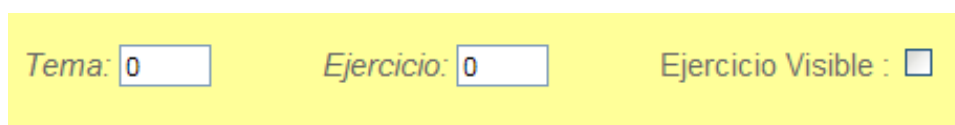
A screenshot of a web application interface showing the "Editar Información" form for a specific student. The header bar is purple and contains the text "Lista de alumnos". Below it, the student ID "1111111" is displayed. The form area has a light yellow background and contains several input fields for different assessment components: "Prácticas de Laboratorio de Automatización Industrial:", "Ejercicios Obligatorios de Automatización Industrial:", "Prácticas de Robótica:", "Ejercicios Obligatorios de Robótica:", "Preparación de Temas de Teoría:", "Ejercicios Optativos de Automatización Industrial:", "Ejercicios Optativos de Robótica:", and "Desarrollo y Presentación de un Trabajo de Libre Elección:". At the bottom of the form, there is a dropdown menu labeled "Nota Final:" with the selected option being "No Presentado".

Figura 24: Apartado de introducción de notas

DISEÑO DE LA APLICACIÓN

5.2.3.3 Introducción de ejercicios

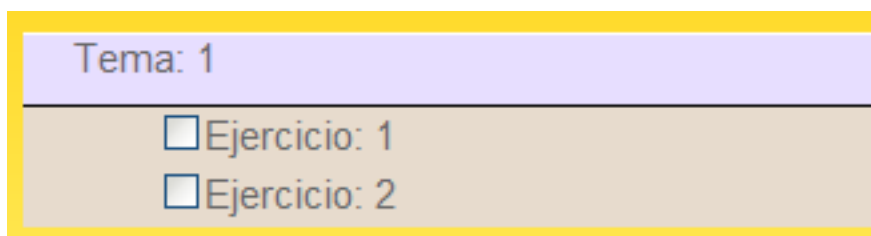
Tanto en la sección de *Teoría*, como en la de *Problemas y Prácticas* de la página del profesor se podrá ver la pestaña *Visualizar*, que mostrará el ejercicio seleccionado, la pestaña *Editar*, donde además del editor de textos se encontrarán dos áreas de texto para insertar el número de tema y de ejercicio además de poder seleccionar si el ejercicio es visible para el alumno.



Formulario de introducción de ejercicios con tres campos: 'Tema: 0', 'Ejercicio: 0' y 'Ejercicio Visible: '. El fondo del formulario es amarillo.

Figura 25: Insertar número de tema y ejercicio

La tercera pestaña que mostrarán estos apartados es la de *eliminar*, donde se verán una lista con los ejercicios introducidos en la base de datos para el apartado seleccionado, y que podrán ser eliminados desde ahí.



Pestaña de eliminar con un encabezado 'Tema: 1' y una lista de ejercicios: 'Ejercicio: 1' y 'Ejercicio: 2', cada uno con un checkbox. El fondo de la pestaña es beige.

Figura 26: Pestaña de eliminar

DISEÑO DE LA APLICACIÓN

5.2.4 Subir archivos

A este apartado se podrá acceder desde el icono superior de la página y mostrará una lista con los archivos subidos por el usuario así como la opción de eliminar o renombrar cada uno de esos archivos.

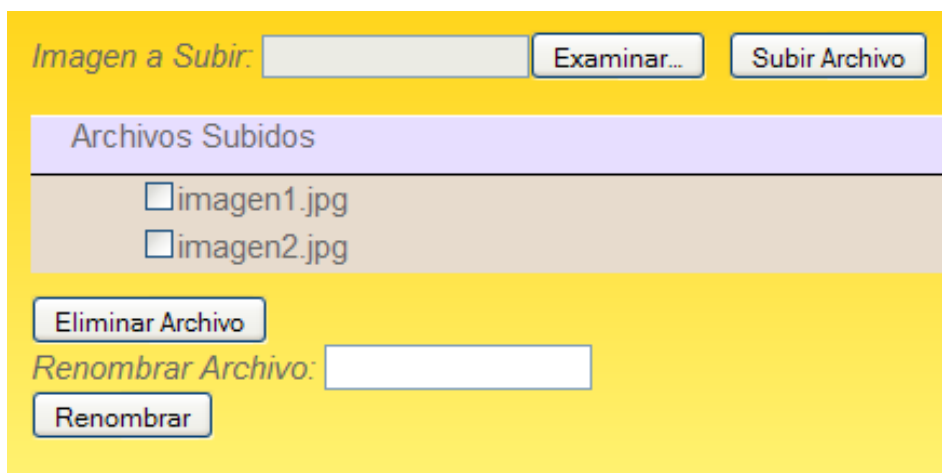


Figura 27: Página de subir archivos

5.2.5 Editor de textos

Como se ha comentado en puntos anteriores, en diferentes apartados se podrá introducir texto, y para facilitar la tarea podrá encontrarse en las diferentes pestañas de *Editar* un editor de texto como el de la siguiente figura:

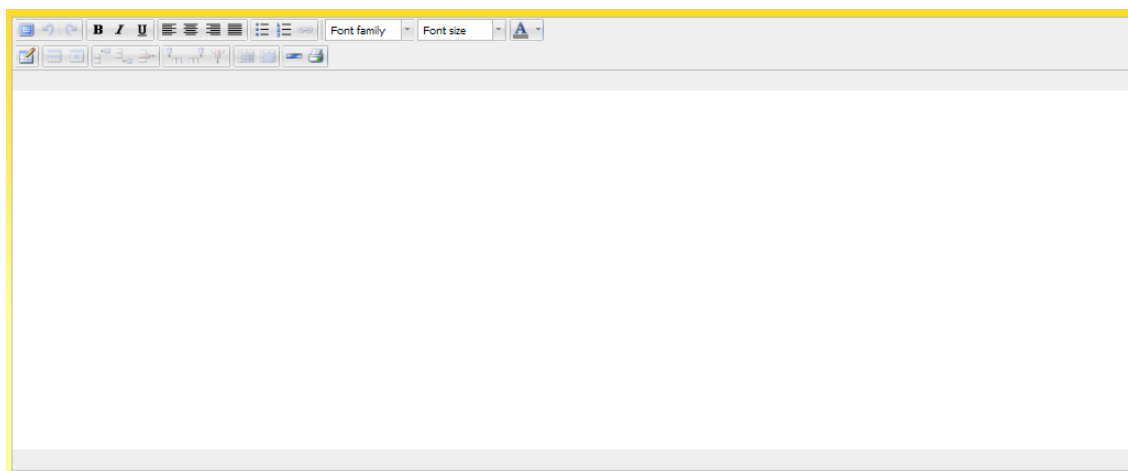


Figura 28: Editor de textos

DISEÑO DE LA APLICACIÓN

5.2.6 Botonera

En la zona superior de la página se podrá encontrar un conjunto de iconos mediante los cuales el usuario podrá salir de la aplicación (icono 1), volver a la página inicial de su cuenta (icono 2), obtener ayuda sobre el apartado en el que se encuentra (icono 3), acceder a la página de subir archivos (icono 4), y por último, en esta zona, está el icono para volver a la página del profesor (icono 5), que solo será visible cuando el profesor acceda a la cuenta de un alumno desde la sección dedicada a ello de la cuenta de administrador.



Figura 29: Botonera superior

6. IMPLEMENTACIÓN

Como se ha comentado en capítulos anteriores el proyecto se realizará utilizando el Framework Struts 2 que utiliza el patrón MVC. Por lo tanto en este apartado se mostrará cada una de las partes que forman este patrón así como el tema de la persistencia con Hibernate.

6.1 Capa del modelo

Esta capa es la parte donde se procesan las acciones y se devuelven los resultados. También es la parte encargada de acceder a la base de datos. Todo esto está implementado en clases de Java.

A continuación se mostrará una de las clases utilizadas en el proyecto donde se mostrará cómo se hace una lectura de la base de datos utilizando Hibernate y se traspa esa información a la capa de la vista.

Se puede ver en la línea número siete como se guarda toda la información que contiene la tabla de la base de datos relacionada con la clase ModTrabajo en la lista modifTrabajo. A continuación en las líneas diez y once se pasa la información de la clase java a la página jsp donde Contenido es el nombre que tiene el textarea de la capa de la vista.

```
1. public void lecturaBBDD(){
2.     Session sessionBBDD = HibernateUtil.getSessionFactory().getCurrentSession();
3.     sessionBBDD.beginTransaction();
4.
5.     try{
6.         List<ModTrabajo> modifTrabajo = new ArrayList<ModTrabajo>();
7.         modifTrabajo = (List<ModTrabajo>)sessionBBDD.createQuery("from ModTrabajo").list();
8.
9.         if(!modifTrabajo.isEmpty()){
10.             setContenido(modifTrabajo.get(0).getInfo());
11.             setId(modifTrabajo.get(0).getId());
12.         }
13.         sessionBBDD.flush();
14.         sessionBBDD.clear();
15.         sessionBBDD.getTransaction().commit();
16.     }
17.     catch(HibernateException e){
18.         e.printStackTrace();
19.     }
20.     catch(RuntimeException re){
21.         System.out.println("Error "+ re.toString());
22.         throw re;
23.     }
24. }
```

Figura 30: Lectura de la base de datos

IMPLEMENTACIÓN

Para guardar información en la base de datos también se realizará desde la clase de Java, y al igual que la lectura, se utilizará Hibernate para pasar la información hasta la base de datos.

En la línea número siete y ocho se puede ver como se recoge la información procedente de la capa de la vista y se guarda en una variable del tipo de la clase, en este caso ModTrabajo, a continuación dependiendo de si se inserta por primera vez la información o de si se hace una actualización, Hibernate realiza un save en la línea número diecisiete o hace un update en la línea once respectivamente donde se guarda la información recuperada de la página jsp en la base de datos.

```
1. public void escrituraBBDD(){
2.     try{
3.         Session sessionBBDD = HibernateUtil.getSessionFactory().getCurrentSession();
4.         sessionBBDD.beginTransaction();
5.
6.         ModTrabajo theEvent = new ModTrabajo();
7.         theEvent.setId(getId());
8.         theEvent.setInfo(getContenido());
9.
10.        if(getId() !=0){
11.            sessionBBDD.update(theEvent);
12.            sessionBBDD.flush();
13.            sessionBBDD.clear();
14.            sessionBBDD.getTransaction().commit();
15.        }
16.        else{
17.            sessionBBDD.save(theEvent);
18.            sessionBBDD.flush();
19.            sessionBBDD.clear();
20.            sessionBBDD.getTransaction().commit();
21.        }
22.    }
23.    catch(RuntimeException re){
24.        System.out.println("error al escribir el fichero en ModTrabajo"+re);
25.        throw re;
26.    }
27. }
```

Figura 31: Escritura en la base de datos

IMPLEMENTACIÓN

Para relacionar Hibernate con cada tabla de la base de datos se utilizará un archivo XML con el mismo nombre que la clase de Java con el que se relaciona. A continuación se muestra uno de estos archivos utilizados en el proyecto.

Se puede ver como Hibernate mapea cada una de las columnas de la tabla, con el campo id de la línea siete se hace referencia a la clave primaria de la tabla, mientras que con el campo property de la línea once se hace referencia a la otra columna de la tabla.

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
3. "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
4.
5. <hibernate-mapping>
6.     <class name="acciones.profesor.ModTrabajo" table="enunciado_trabajo" >
7.         <id name="Id" column="Id" type="java.lang.Integer">
8.             <generator class="native" />
9.         </id>
10.
11.         <property name="Info" type="java.lang.String">
12.             <column name="Info"/>
13.         </property>
14.
15.     </class>
16. </hibernate-mapping>
```

Figura 32: Archivo XML para conexión con la base de datos

6.2 Capa de la vista

Esta capa del modelo está formada por el conjunto de las páginas que el usuario puede ver desde su navegador. Para su diseño se utilizarán paginas JSP que permiten el uso combinado de código HTML y de etiquetas Struts 2. Para mejorar su aspecto, también se usará JavaScript y CSS que mejorarán mucho la experiencia final del usuario.

A continuación se muestra parte del código JSP de una de las páginas, donde se puede apreciar el uso de las etiquetas Struts 2. Donde se puede ver la forma que tiene una etiqueta div, en la línea seis, o como se inserta la etiqueta textarea mediante Struts 2 en la línea siete.

IMPLEMENTACIÓN

```
1. <s:div id="div_editar" cssClass="CuadroInfo">
2.     <s:form id="form_editar" name="form_editar" method="post"
3.         action="modtrabajo" cssClass="FormularioEditar" theme="css_xhtml">
4.         <s:submit name="boton" value="%${getText('boton.guardar')}%"
5.             cssClass="BotonGuardar" theme="css_xhtml" />
6.         <s:div cssClass="TextareaContenido">
7.             <s:textarea name="Contenido" id="Contenido" cols="20" rows="5">
8.             </s:textarea>
9.             <s:hidden name="Id" id="Id" />
10.        </s:div>
11.    </s:form>
12.</s:div>
```

Figura 33: Código de la página JSP

6.3 Capa del controlador

Struts 2 facilita un controlador interno llamado *FilterDispatcher* con lo cual lo único que hay que hacer es mapear las acciones que se necesiten.

Cuando se acciona un evento en la página JSP, una clase de tipo action recoge los datos del formulario asociado para después llamar a la clase correspondiente de la capa del modelo. El mapeo de la página JSP con la clase Java se realiza mediante un fichero XML llamado *struts.xml*, que es el utilizado por el *FilterDispatcher* para saber que clase utilizar en cada momento.

En el código se puede ver como todas las acciones con nombre modtrabajo utilizarán la clase ModTrabajo.java y el resultado será enviado a la página jsp modificartrabajo.

```
<action name="modtrabajo" class="acciones.profesor.ModTrabajo">
    <result>/paginas/profesor/modificartrabajo.jsp</result>
</action>
```

Figura 34: Código XML de la capa del controlador

6.4 Internacionalización i18n

La internacionalización consiste en diseñar la página con soporte para múltiples idiomas. Este proceso con Struts 2 se realiza de forma sencilla, simplemente se tiene que crear un fichero “.properties” para cada uno de los idiomas que se quieran añadir, en esos ficheros se guardarán las variables de texto junto con su traducción al idioma deseado.

```
# Login de l'usuari
login.user=Usuari
login.pass>Password
login.error=Error al login
login.errorPass= Error al password
login.registrar=Registre
login.confirmar=Confirmar password

# Menu de la part de l'alumne
alumno.diario=Diari
alumno.teoria=Preparació temes de teoria
alumno.problemas=Problemes
alumno.practicas=Pràctiques
alumno.trabajo=Treball sobre l''assignatura
alumno.notas=Notes
alumno.archivo=Pujar arxiu
```

Figura 35: Ejemplo archivo de internacionalización

7. PRUEBAS

Una vez terminada la codificación de la aplicación y las pequeñas pruebas que se han ido realizando en paralelo llega el momento de realizar un conjunto de pruebas más exhaustivo para asegurar el correcto funcionamiento de la aplicación.

Para realizar las pruebas, se ha decidido utilizar los dos posibles roles que permite la aplicación, en primer lugar se asumirá el rol de profesor, previamente registrado, para introducir la diversa información y ejercicios en la base de datos. A continuación se adoptará el rol de alumno que tras registrarse, tendrá la posibilidad de acceder a su cuenta y realizar los ejercicios disponibles.

Tras la identificación como profesor, el primer paso es insertar la información sobre la asignatura que verán todos los alumnos, para ello desde el menú se accede al apartado de *Información del curso*.



Figura 36: Información de la asignatura

PRUEBAS

A continuación, para seguir introduciendo este tipo de información, se accede al apartado de notas y desde *Editar información* se escribe la forma de evaluación que se seguirá en la asignatura.

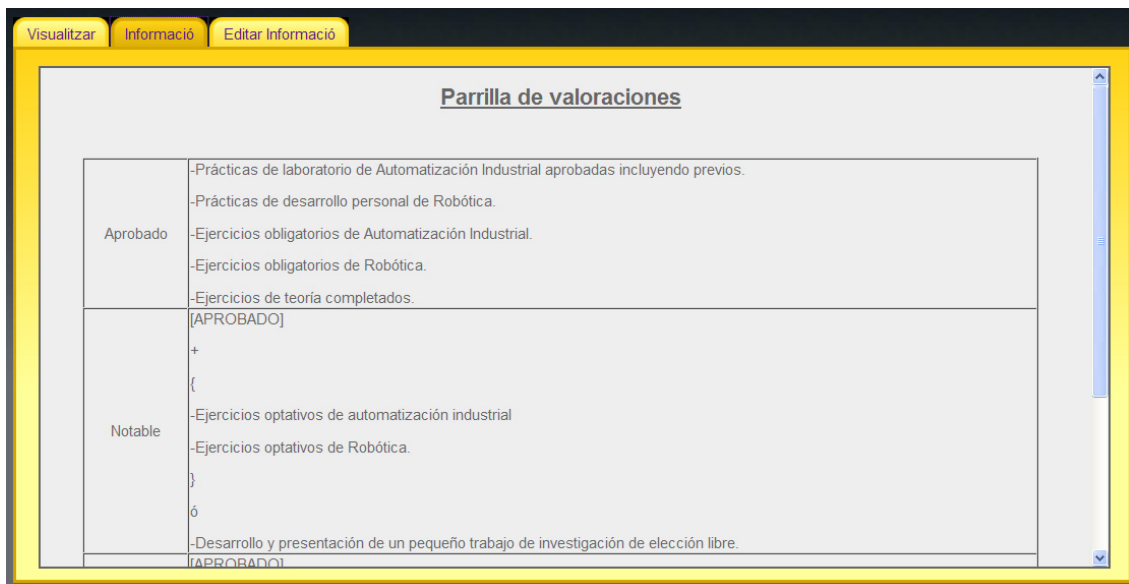


Figura 37: Información de las notas

Para finalizar la información referente a la asignatura, se ha añadido en el apartado de trabajo los datos necesarios para su realización.



Figura 38: Información del trabajo

PRUEBAS

Para terminar de comprobar el resto de apartados se introducirá un ejercicio representativo en cada una de las secciones. Antes de continuar, se subirán las fotografías necesarias para los enunciados de los ejercicios introducidos.

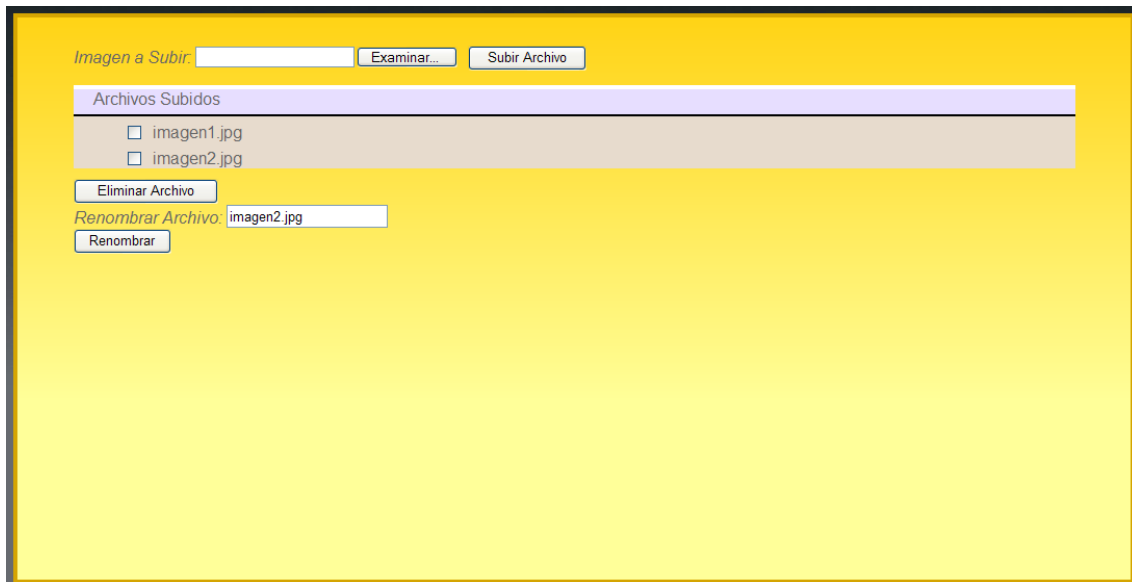


Figura 39: Página para subir imágenes

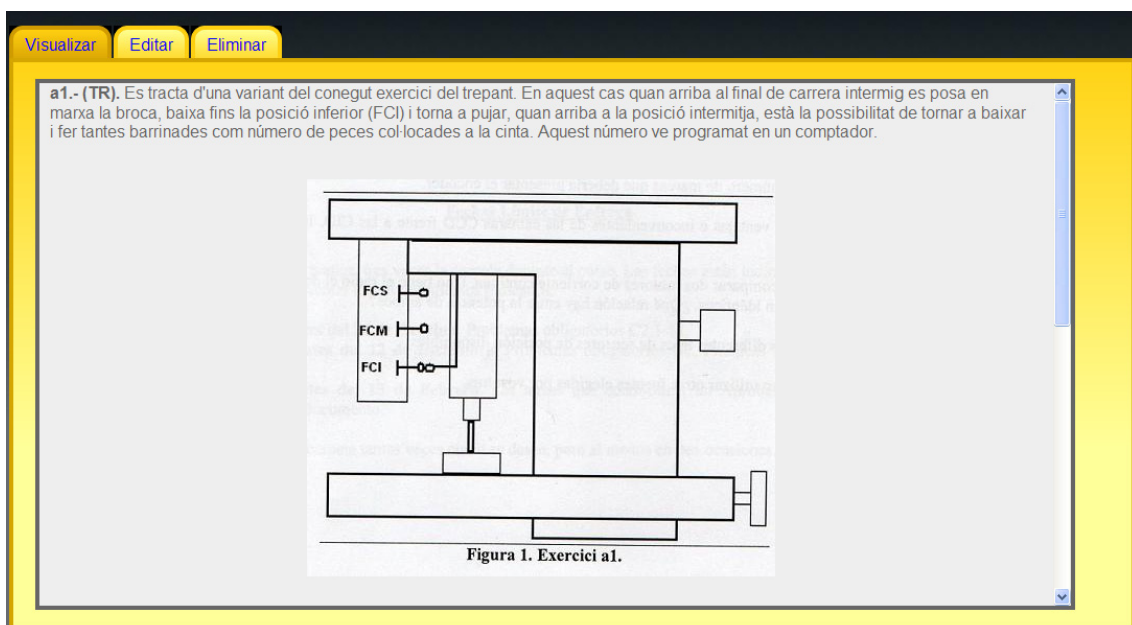


Figura 40: Problemas añadido a la aplicación

PRUEBAS

Una vez comprobado que toda la parte del profesor funciona correctamente, pasamos a utilizar la aplicación como alumno, en primer lugar, antes de poder acceder a la cuenta, deberemos registrarnos y a continuación, tras identificarnos, se accederá a la página principal donde ya se podrá ver toda la información de la asignatura introducida previamente con la cuenta del profesor.



Figura 41: Página de inicio en la cuenta del alumno

PRUEBAS

Ahora es el momento de resolver alguno de los ejercicios propuestos por el profesor, para ello se ha seleccionado el apartado de Problemas. En la siguiente figura se puede ver una parte del ejercicio resuelto.

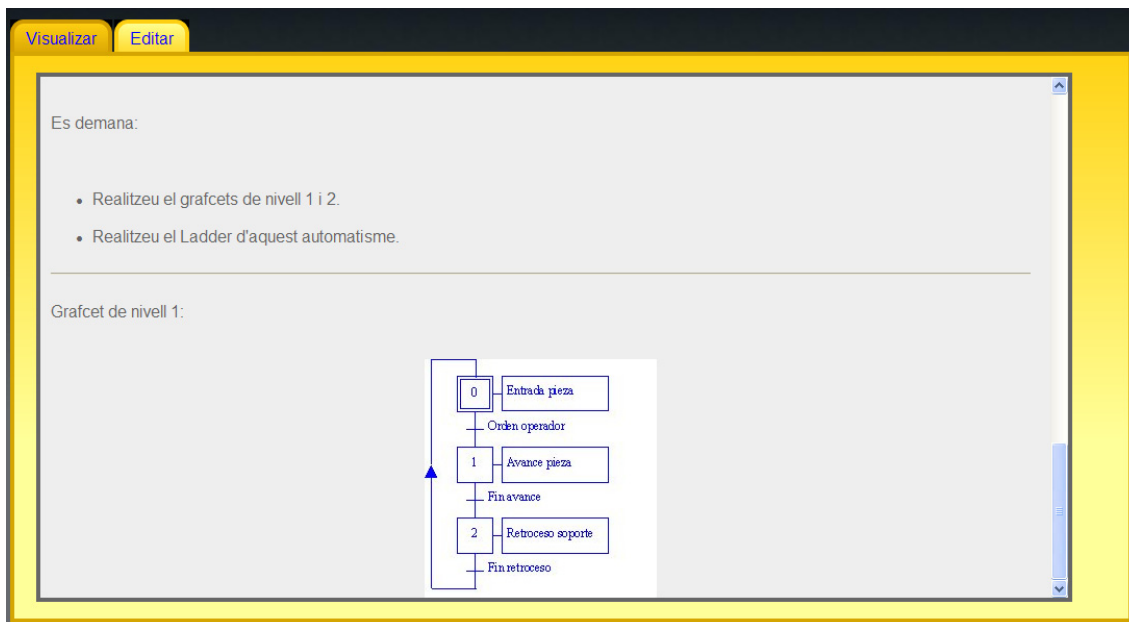


Figura 42: Resolución de un problema

Para finalizar, dando un salto al final del curso, nuevamente desde la cuenta del profesor introducimos la puntuación del alumno para corroborar que realiza bien esta última operación.

Lista de alumnos

1111111

Prácticas de Laboratorio de Automatización Industrial: 7.5

Ejercicios Obligatorios de Automatización Industrial: 7

Prácticas de Robótica: 6.5

Ejercicios Obligatorios de Robótica: 7

Preparación de Temas de Teoría: 8

Ejercicios Optativos de Automatización Industrial: []

Ejercicios Optativos de Robótica: []

Desarrollo y Presentación de un Trabajo de Libre Elección: 8

Nota Final: Notable

Figura 43: Página de insertar notas

PRUEBAS

Desde la cuenta del alumno accedemos al apartado de notas para comprobar la puntuación final de las tareas realizadas.



Actividad	Puntuación
Prácticas de Laboratorio de Automatización Industrial	7.5
Ejercicios Obligatorios de Automatización Industrial	7
Prácticas de Robótica	6.5
Ejercicios Obligatorios de Robótica	7
Preparación de Temas de Teoría	8
Ejercicios Optativos de Automatización Industrial	
Ejercicios Optativos de Robótica	
Desarrollo y Presentación de un Trabajo de Libre Elección	8
Nota Final	Notable

Figura 44: Apartado de notas del alumno

Una vez finalizado este paquete de pruebas, se puede concluir con que el funcionamiento de la aplicación es correcto para cada una de sus secciones.

8. CONCLUSIONES

Llegado a este punto de la memoria, es el momento de valorar si con la nueva aplicación que se ha implementado se ha conseguido cubrir todos los objetivos que se habían planteado antes de comenzar su construcción.

Con este proyecto se ha querido diseñar una solución tecnología para satisfacer las necesidades de una asignatura concreta, pero siempre con la mente puesta en la posibilidad de exportar la aplicación a otras asignaturas con una metodología de trabajo similar. Los objetivos eran conseguir una aplicación fácil de usar e intuitiva para favorecer la rápida adaptación de los futuros usuarios. Otro de los grandes objetivos que se habían planteado desde el principio era favorecer su uso en múltiples plataformas para evitar limitaciones en el uso de servidores, además de permitir su utilización en la mayor cantidad de navegadores y resoluciones de pantalla posible para que los usuarios pudieran seguir utilizando sus configuraciones preferidas.

Teniendo en cuenta los objetivos perseguidos y el resultado obtenido se puede concluir con que se han logrado alcanzar todas las metas propuestas al iniciar este proyecto.

Desde el punto de vista personal, el proyecto me ha aportado sobre todo dos cosas, la primera es que he aprendido a utilizar tecnologías que desconocía totalmente como Struts 2, Hibernate o JSP y me ha dado la oportunidad de profundizar más en los conocimientos previos que tenía sobre JavaScript y CSS. La segunda cosa que me ha aportado este proyecto ha sido la experiencia de crear un sitio web completo desde cero, con la posibilidad de comprobar las consecuencias de tomar una decisión u otra en cada momento del desarrollo.

9. BIBLIOGRAFÍA

9.1 Fuentes bibliográficas

- Donald Brown, Chad Michael Davis, Scott Stanlick. *Struts 2 in action*, 2008 Ed. Manning.
- Francisco Charte Ojeda. *La biblia de HTML*, 2005 Ed. Anaya Multimedia.
- Andy Budd, Cameron Moll, Simon Collison. *Manual avanzado de CSS*, 2007 Ed. Anaya Multimedia.
- Lázaro Issi Camy. *La biblia de JavaScript*, 2002 Ed. Anaya Multimedia.

9.2 Fuentes electrónicas

- Struts 2 : <http://struts.apache.org/2.0.14/>
- Tiny_MCE: <http://tinymce.moxiecode.com/>
- Hibernate: <http://www.hibernate.org>
- jsMath: <http://www.math.union.edu/~dpvc/jsMath/authors/2.x/basics.html>
- Jasypt: <http://www.jasypt.org/getting-started-easy.html>
- Tomcat: <http://tomcat.apache.org/>
- Mysql: <http://www.mysql.com/>

9.3 Bibliografía tecnológica

- Francisco Charte Ojeda, Jorge Serrano Pérez. *Programación con Microsoft Visual Studio .NET*, 2002 ed. Anaya Multimedia
- Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett. *Professional AJAX 2nd Edition*, 2007 ed. Wiley Publishing
- Comparativa de tecnologías:
<http://www.promoteware.com/Module/Article/ArticleView.aspx?id=10>
- <http://es.wikipedia.org/wiki/.NET>
- <http://es.wikipedia.org/wiki/PHP>
- <http://es.wikipedia.org/wiki/J2EE>

ANEXO

Generar una contraseña de administrador

Este anexo pretende ser una ayuda para la creación de una cuenta de administrador, ya que la única forma de poder crear este tipo de usuario es generando la encriptación del password de forma externa a la aplicación y añadiéndola posteriormente a la base de datos.

El proceso se realizará desde una línea de comandos. Después de acceder al directorio donde se ha descomprimido el archivo adjunto (Jasypt) se deberá ir al directorio *bin* y se insertará la siguiente línea:

```
digest.bat input="contraseña" saltSizeBytes=0 iterations=1 algorithm=SHA-1
```

A continuación se muestra un ejemplo para la contraseña "123456".

```
----ENVIRONMENT-----  
Runtime: Sun Microsystems Inc. Java HotSpot(TM) Client VM 11.3-b02  
  
----ARGUMENTS-----  
algorithm: SHA-1  
saltSizeBytes: 0  
input: 123456  
iterations: 1  
  
----OUTPUT-----  
fEqNCco3Yq9h5ZUg1D3CZJT41Bs =
```

Figura 45: Cifrado de una contraseña

Una vez obtenido el cifrado de la contraseña se debe añadir una nueva entrada a la base de datos en la tabla identificación, donde en el campo *User* se añadirá el número del usuario, en el campo *Pass* se añadirá esta contraseña y en el campo *Tipo* se introducirá un 1 para indicar que es un profesor.

El abajo firmante, **David Forte Romera**, alumno de la Escuela Universitaria de Informática de la UAB,

CERTIFICA:

Que ha realizado la presente memoria y el proyecto “Web para la gestión electrónica de una Carpeta de Aprendizaje”.

Y para que conste firma la presente.

Sabadell, Septiembre del 2009

Firmado: David Forte Romera

