



**Universitat Autònoma  
de Barcelona**

# Aplicación web con tecnología Java para la gestión de competencias en los estudios de grado.

Memoria del proyecto  
de Ingeniería Técnica en  
Informática de Sistemas  
realizado por

*Jose Alberto Fuentes Torrubia*

y dirigido por

*Jordi Pons Aróztegui*

**Escuela de Ingeniería**

Sabadell, *Septiembre* de 2011

El sotasignat, ***Jordi Pons Aróztegui***,  
professor de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present  
memòria  
ha estat realitzat sota la seva direcció per  
***Jose Alberto Fuentes Torrubia***

I per a que consti firma la present.  
Sabadell, ***setembre*** de ***2011***

-----  
Signat: ***Jordi Pons Aróztegui***

## FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

<b>Títol del projecte:</b> Aplicación web con tecnología Java para la gestión de competencias en los estudios de grado.	
<b>Autor[a]:</b> Jose Alberto Fuentes Torrubia	<b>Data:</b> <i>Septiembre 2011</i>
<b>Tutor[a]/s[es]:</b> Jordi Pons Aróztegui	
<b>Titulació:</b> Ingeniería Técnica Informática de Sistemas	
<b>Paraules clau</b> (mínim 3) <ul style="list-style-type: none"><li>• Català: Aplicació Web, Base de dades, Competències</li><li>• Castellà: Aplicación Web, Base de datos, Competencias</li><li>• Anglès: Web Application, Database, Competitions</li></ul>	
<b>Resum del projecte</b> (extensió màxima 100 paraules) <ul style="list-style-type: none"><li>• Català: L'objectiu d'aquest projecte és desenvolupar una aplicació web amb tecnologia Java com a alternativa a PHP. La web consisteix a facilitar la gestió de competències dels estudis dels graus universitaris. Dins de l'aplicació tant coordinadors com a administradors podran realitzar assignacions i manteniment d'aquestes competències. Els usuaris no registrats podran obtenir informació de les competències a través dels estudis seleccionats. Conclou amb una comparació de la tecnologia usada amb PHP sobre l'aplicació, detallant els avantatges i inconvenients d'una enfront de l'altra.</li><li>• Castellà: El objetivo de este proyecto es desarrollar una aplicación web con tecnología Java como alternativa a PHP. La web consiste en facilitar la gestión de competencias de los estudios de los grados universitarios. Dentro de la aplicación tanto coordinadores como administradores podrán realizar asignaciones y mantenimiento de dichas competencias. Los usuarios no registrados podrán obtener información de las competencias a través de los estudios seleccionados. Concluye con una comparación de la tecnología usada con PHP sobre la aplicación, detallando las ventajas e inconvenientes de una frente a la otra.</li><li>• Anglès: The objective of this project is to develop a web application using Java technologic and alternative to PHP. The web application is to facilitate the management competences of studies of university degrees. Into the implementation of both coordinators and administrators can make assignments and maintenance of these competences. Unregistered users can obtain information competences getting the selected studies. It concludes with a comparison of the technology used with PHP on the application, detailing the advantages and disadvantages of one against the another one.</li></ul>	

## INDICE

---

1. INTRODUCCIÓN .....	1
1.1 OBJETIVOS.....	2
1.2 ESTRUCTURA DE LA MEMORIA.....	2
2. ESTUDIO DE VIABILIDAD .....	5
2.1 OBJETIVOS.....	5
2.2 PARTES INTERESADAS.....	6
2.3 PLANIFICACION .....	7
2.4 DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES.....	9
2.5 REFERENCIAS .....	9
2.6 PRODUCTO Y DOCUMENTACIÓN DEL PROYECTO .....	10
2.7 DIAGNOSTICO DEL SISTEMA ACTUAL .....	10
2.8 ALTERNATIVAS TECNOLOGICAS.....	10
2.9 EVALUACION DE RIESGOS.....	16
2.9.1 PLAN DE CONTINGENCIAS.....	17
2.10 PRESUPUESTO.....	18
2.11 CONCLUSIONES.....	19
3. ANÁLISIS.....	21
3.1 INTRODUCCIÓN .....	21
3.2 DEFINICION DE REQUISITOS FUNCIONALES .....	21
3.3 DEFINICION DE REQUISITOS NO FUNCIONALES .....	26
3.4 CASOS DE USOS .....	27
4. DISEÑO.....	29
4.1 INTRODUCCION .....	29
4.2 SELECCIÓN DE TECNOLOGIA.....	29
4.3 DEFINICION DE LA BD .....	32
4.3.1 TABLA DE ESTUDIOS .....	32
4.3.2 TABLA DE MÓDULOS .....	32
4.3.3 TABLA DE MATERIAS .....	32
4.3.4 TABLA DE ASIGNATURAS .....	32
4.3.5 TABLA DE TIPO DE COMPETENCIA .....	33
4.3.6 TABLA DE NIVEL DE COMPETENCIA.....	33
4.3.7 TABLA DE COMPETENCIAS .....	33
4.3.8 TABLA DE COMPETENCIAS-MATERIAS .....	33
4.3.9 TABLA DE COMPETENCIAS-ASIGNATURAS.....	33
4.3.10 ESQUEMA BD.....	34
4.4 INTERFAZ DE USUARIO .....	36
4.4.1 PLANTILLA DE LA APLICACIÓN.....	36
4.4.2 MAPA DE LA APLICACIÓN WEB .....	39
5. IMPLEMENTACION.....	45
5.1 DISEÑO DE CLASES.....	47
5.2 ORGANIZACIÓN DE ARCHIVOS .....	49
5.3 CONEXIÓN Y CONSULTA DE BASE DE DATOS .....	51
5.4 VALIDACIÓN DE FORMULARIOS .....	52
6. PRUEBAS .....	53
6.1 PRUEBAS UNITARIAS.....	53
6.2 PRUEBAS DE INTEGRACIÓN .....	54
7. COMPARACION DE TECNOLOGIAS.....	55

7.1 ANALISIS.....	56
7.1.1 MODULARIZACIÓN .....	56
7.1.2 MANTENIBILIDAD .....	57
7.1.3 CRECIMIENTO DEL SISTEMA.....	57
7.1.4 COSTE DE DESARROLLO .....	57
7.1.5 FORMACIÓN .....	58
7.1.6 INTEGRACIÓN EXTERNA .....	58
7.1.7 SEGURIDAD.....	58
7.1.8 RENDIMIENTO .....	58
7.1.9 ESCALABILIDAD.....	59
7.1.10 CONCLUSIONES DEL ESTUDIO .....	59
8. CONCLUSIONES.....	61
8.1 OBJETIVOS CONSEGUIDOS Y NO CONSEGUIDOS.....	61
8.2 AMPLIACIONES Y MEJORAS .....	61
8.3 DESVIACIÓN TEMPORAL RESPECTO A LA PLANIFICACION .....	62
8.4 VALORACION.....	64
9. REFERENCIA WEB.....	65

## INDICE DE TABLAS

---

TABLA 1: Clasificación de los objetivos .....	6
TABLA 2: Evaluación de riesgos.....	17
TABLA 3: Plan de contingencias .....	17
TABLA 4: Presupuesto por roles.....	18
TABLA 5: Presupuesto basado en la planificación .....	18
TABLA 6: Gastos de recursos necesarios_ .....	18
TABLA 7: Control de acceso .....	21
TABLA 8: Tabla Estudios .....	22
TABLA 9: Tabla Módulos .....	22
TABLA 10: Tabla Materias .....	22
TABLA 11: Tabla Asignaturas .....	23
TABLA 12: Tabla Tipo de competencia .....	23
TABLA 13: Tabla Nivel de competencia.....	23
TABLA 14: Tabla Competencias.....	24
TABLA 15: Tabla Competencias.....	24
TABLA 16: Tabla Estudios_Módulos .....	24
TABLA 17: Tabla Módulos_Materias .....	25
TABLA 18: Tabla Materias_Competencias.....	25
TABLA 19: Tabla Materias_Asignaturas .....	25
TABLA 20: Tabla Competencias_Asignaturas.....	25
TABLA 21: Relaciones entre tablas base de datos.....	35

## INDICE DE FIGURAS

---

FIGURA 1: Planificación .....	7
FIGURA 2: Diagrama de Gantt .....	8
FIGURA 3: Logo Java .....	10
FIGURA 4: Logo Hibernate .....	11
FIGURA 5: Logo Ibatis .....	11
FIGURA 6: Logo Apache Derby .....	11
FIGURA 7: Logo MySQL .....	12
FIGURA 8: Logo PostgreSQL .....	12
FIGURA 9: Logo Microsoft Access .....	12
FIGURA 10: Logo Microsoft SQL Server .....	13
FIGURA 11: Logo Oracle .....	13
FIGURA 12: Logo Struts .....	14
FIGURA 13: Logo EJB 3.0 .....	14
FIGURA 14: Logo Ruby on Rails .....	15
FIGURA 15: Logo Spring Framework MVC .....	15
FIGURA 16: Logo Hibernate .....	29
FIGURA 17: Logo MySQL .....	30
FIGURA 18: Logo Log4j .....	30
FIGURA 19: Logo Spring Framework MVC .....	31
FIGURA 20: Esquema de base de datos .....	34
FIGURA 21: Plantilla de la aplicación .....	36
FIGURA 22: Captura Aplicación Web .....	38
FIGURA 23: Mapa aplicación Web usuario .....	39
FIGURA 24: Captura de Contacto .....	40
FIGURA 25: Mapa aplicación Web coordinador .....	40
FIGURA 26: Captura Competencias manager .....	42
FIGURA 27: Captura Estudios_Modulos .....	42
FIGURA 28: Captura relación Estudio/Módulos .....	42
FIGURA 29: Mapa aplicación Web administrador .....	43
FIGURA 30: Menú Coordinador .....	45
FIGURA 31: Menú Administrador .....	45
FIGURA 32: Menú Usuario .....	46
FIGURA 33: Flujo de permisos .....	46
FIGURA 34: Diagrama de clases .....	48
FIGURA 35: Organización de archivos .....	50
FIGURA 36: Código de acceso a base de datos .....	51
FIGURA 37: Código propiedad sessionFactory .....	51
FIGURA 38: Módulo MVC .....	56
FIGURA 39: Grafico comparación de tecnologías .....	59
FIGURA 40: Comparación de planificaciones .....	63

## 1. INTRODUCCIÓN

---

El proyecto que a continuación se va a desarrollar consiste en realizar una aplicación web para gestionar las competencias de los estudios de grado universitarios.

Aunque actualmente para el desarrollo de aplicaciones de esta similitud se utiliza PHP como tecnología de programación, nosotros lo vamos a implementar utilizando JAVA como tecnología de desarrollo.

El objetivo es aprender y profundizar en dicha tecnología y poder comentar tanto ventajas como inconvenientes de su uso para elaborar aplicaciones de este tipo, comparándola finalmente con PHP para obtener las diferencias más destacables entre ambas tecnologías para realizar el mismo problema.

La aplicación web que se va a desarrollar está enfocada a resolver en un entorno universitario el problema de adjudicación, modificación y consulta de competencias y niveles de cumplimiento según asignaturas, materias y estudios para cada uno de los posibles tipos de usuarios registrados, habiendo alumnos, administradores y coordinadores.

La presente documentación junto con la aplicación web se entrega como proyecto de final de carrera de Ingeniería Técnica en Informática de Sistemas.

La aplicación web debe ser capaz de alcanzar todos los propósitos y objetivos que más adelante detallaremos. Para la creación de estos propósitos y objetivos hemos mantenido, a lo largo de la creación del proyecto, varias entrevistas o reuniones con el coordinador de estudios de la escuela y hemos ido perfilando la aplicación web con diferentes propósitos y variantes. Entre el cliente (escuela) y el proyectista hemos llegado a unos objetivos comunes.

La funcionalidad de esta aplicación web ha de favorecer la usabilidad a la hora de adjudicar, modificar y consultar competencias asignadas a las distintas asignaturas de los distintos grados.

## 1.1 OBJETIVOS

---

Los objetivos del proyecto son los siguientes:

- Crear la aplicación web para administrar mejor las tareas de adjudicación y modificación de competencias y niveles de cumplimientos en estudios universitarios oficiales.
- Facilitar al personal docente una herramienta simple para poder gestionar las tareas mencionadas anteriormente.
- Utilizar Eclipse como entorno de desarrollo integrado de código abierto multiplataforma para desarrollar la aplicación.
- Aprendizaje de nuevas tecnologías de desarrollo web.
- Realizar comparativa de frameworks.

## 1.2 ESTRUCTURA DE LA MEMORIA

---

En la memoria intentaremos explicar el funcionamiento y la estructura que tendrá nuestra aplicación web una vez se concluya.

La importancia de este proyecto fin de carrera no es la aplicación en sí, sino el estudio de la tecnología JAVA con el fin de realizar los objetivos, obteniendo las ventajas e inconvenientes de implementarla con JAVA.

Esta memoria complementa la aplicación web aunque, la mayoría de tiempo empleado en el proyecto, haya sido con la aplicación web, parte más funcional y útil.

Tanto la memoria como la aplicación a entregar en la escuela, dependen directamente una de la otra ya que hasta que no se ha acabado la aplicación web no se ha podido finalizar la memoria.

A continuación se explicarán los diferentes capítulos de que consta esta memoria:

- En el bloque llamado Introducción, explicaremos de qué trata el proyecto que vamos a desarrollar y los objetivos que nos ha llevado a realizarlo y la tecnología con la que lo vamos a desarrollar, haciendo mención a la tecnología con la que vamos a realizar la comparación.
- En el siguiente bloque se explica la planificación que se ha llevado a cabo para la realización del proyecto, con el estudio de viabilidad, los objetivos que esta planificación quiere alcanzar y las ventajas e inconvenientes de la tecnología utilizada frente a PHP.

- Encontraremos también el análisis de requerimientos que hemos tenido que tener en cuenta para cumplir todos los objetivos propuestos.
- En el capítulo de diseño se explicará cómo está estructurada la aplicación web, haciendo referencia al diseño de base de datos, la tecnología con la que diseñaremos la aplicación y las consideraciones gráficas que se han tenido en cuenta.
- En la parte de implementación se explicará el modelos con el que se ha realizado la aplicación, el diseño de clases, la organización de los archivos del proyecto, explicando el tipo de validación realizada a los formularios y como accedemos a base de datos capturas. En resumen, exponemos detalladamente la implementación seguida para la elaboración de la aplicación.
- La parte de pruebas realizadas para que el entorno sea de calidad, también ha tenido una sección para asegurarnos de que no se produce ningún error en la hora del tráfico de datos de la aplicación, o intentar acceder a puestos restringidos.
- Para finalizar la memoria, explicaremos las conclusiones que este proyecto ha comportado y las posibles líneas de futuro a desarrollar basándonos en las posibles modificaciones que podría adquirir en un futuro la aplicación, ofreciendo ventajas e inconvenientes de la tecnología usada frente a la tecnología con la que la vamos a comparar.



## 2. ESTUDIO DE VIABILIDAD

---

El estudio de viabilidad se estudiará a continuación dando a conocer la planificación de dicho proyecto consiguiendo con ello una orientación aproximada de lo que se requiere.

Los valores que vamos a señalar a continuación tienen un peso especial en este estudio y son los siguientes:

- Objetivos
- Partes interesadas
- Planificación
- Definición, acrónimos y abreviaciones
- Referencias
- Producto y documentación del proyecto
- Diagnóstico del sistema actual
- Alternativas y soluciones a las tecnologías utilizadas
- Evaluación de riesgos
- Presupuesto
- Conclusiones

### 2.1 OBJETIVOS

---

Los objetivos del proyecto en este estudio de viabilidad que han sido de apoyo para la elaboración del mismo han sido los siguientes:

1. Mejorar la gestión de competencias (niveles, tipos, asignaturas...) del alumnado dependiendo de sus estudios y materias.
2. Buena interfaz gráfica y fácil usabilidad de la aplicación web para el usuario.
3. Creación de una pequeña intranet distinguida entre alumno, coordinador y administrador.
4. Base de datos estable cumpliendo los requisitos fundamentales de seguridad y funcionalidad.
5. Estudio de Java como tecnología usada para implementar la aplicación.
6. Comparación con PHP como tecnología más implantada para la elaboración de aplicaciones similares.

Según la prioridad que estableceremos en la tabla 1 clasificaremos los objetivos según prioridades:

	Alta	Media	Baja
1	X		
2	X		
3		X	
4	X		
5	X		
6		X	

TABLA 1: Clasificación de los objetivos.

## 2.2 PARTES INTERESADAS

Podemos clasificar las partes interesadas claramente en tres grupos:

- **Usuarios**
- **Equipo de trabajo** (el equipo de trabajo es una única persona ya que el proyecto es individual).
- **Tutor**

En cuanto al grupo de usuarios podemos diferenciar tres grupos:

- **Alumnado:** Consulta la web para poder ver según sus estudios y materias sus correspondientes competencias.
- **Administrador:** Asegurarse de que el entorno funciona correctamente y actualizar base de datos periódicamente.
- **Coordinador:** Asigna competencias u niveles a los distintas materias y asignaturas.

Con respecto al equipo de trabajo lo organizaremos por ámbito de trabajo:

- **Jefe del proyecto:** Coordinación, dirección, control y responsabilización en último término de la ejecución del proyecto.
- **Analista funcional:** Estudio de la situación actual, de los objetivos, de la viabilidad y del impacto hardware/software.
- **Analista:** Encargado del diseño del proyecto (diseño de tablas de BD, evaluación y control de pruebas unitarias, puesta en producción...)
- **Programador:** Encargado de realizar la estructura lógica, codificación y pruebas unitarias de la aplicación.

## 2.3 PLANIFICACION

Para la creación de la planificación del proyecto vamos a utilizar el software Microsoft Project.

Id		Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesor
1			Inicio del proyecto	1 día	vie 03/06/11	lun 06/06/11	
2			<b>Planificacion</b>	<b>7 días</b>	<b>lun 06/06/11</b>	<b>mié 15/06/11</b>	
3			Estudio de viabilidad	4 días	lun 06/06/11	vie 10/06/11	
4			Primera reunion con l	1 día	vie 10/06/11	lun 13/06/11	3
5			Aprobación de estudio	2 días	lun 13/06/11	mié 15/06/11	4
6			<b>Analisis de la aplicación</b>	<b>11,13 días</b>	<b>jue 16/06/11</b>	<b>vie 01/07/11</b>	
7			Análisis de requisitos	3 días	jue 16/06/11	mar 21/06/11	
8			Análisis de datos (Base	2 días	mar 21/06/11	jue 23/06/11	7
9			Análisis de seguridad	2 días	jue 23/06/11	lun 27/06/11	8
10			Documentacion del an	2 días	lun 27/06/11	mié 29/06/11	9
11			Segunda reunion con	1 día	mié 29/06/11	jue 30/06/11	10
12			Aprobación de análisis	1 día	jue 30/06/11	vie 01/07/11	11
13			<b>Diseño de la aplicación</b>	<b>11 días</b>	<b>lun 04/07/11</b>	<b>mar 19/07/11</b>	
14			Diseño Técnico (Base	4 días	lun 04/07/11	vie 08/07/11	
15			Diseño de la interfaz	4 días	vie 08/07/11	jue 14/07/11	14
16			Documentacion del di	2 días	jue 14/07/11	lun 18/07/11	15
17			Aprobación del diseño	1 día	lun 18/07/11	mar 19/07/11	16
18			<b>Desarrollo de la aplicación</b>	<b>27 días</b>	<b>mié 20/07/11</b>	<b>vie 26/08/11</b>	
19			Preparación del entor	1 día	mié 20/07/11	jue 21/07/11	
20			Desarrollo aplicación	20 días	jue 21/07/11	jue 18/08/11	19
21			Desarrollo de la interf	5 días	jue 18/08/11	jue 25/08/11	20
22			Documentación de de	1 día	jue 25/08/11	vie 26/08/11	21
23			<b>Test y pruebas</b>	<b>7,75 días</b>	<b>lun 29/08/11</b>	<b>mié 07/09/11</b>	
24			Pruebas unitarias	2 días	lun 29/08/11	mié 31/08/11	
25			Pruebas de integración	3 días	mié 31/08/11	lun 05/09/11	24
26			Documentacion de pruebas	1 día	lun 05/09/11	mar 06/09/11	25
27							
28			Presentación PFC	1 día	jue 15/09/11	vie 16/09/11	
29			Cierre del proyecto	1 día	vie 16/09/11	lun 19/09/11	28

FIGURA 1: Planificación

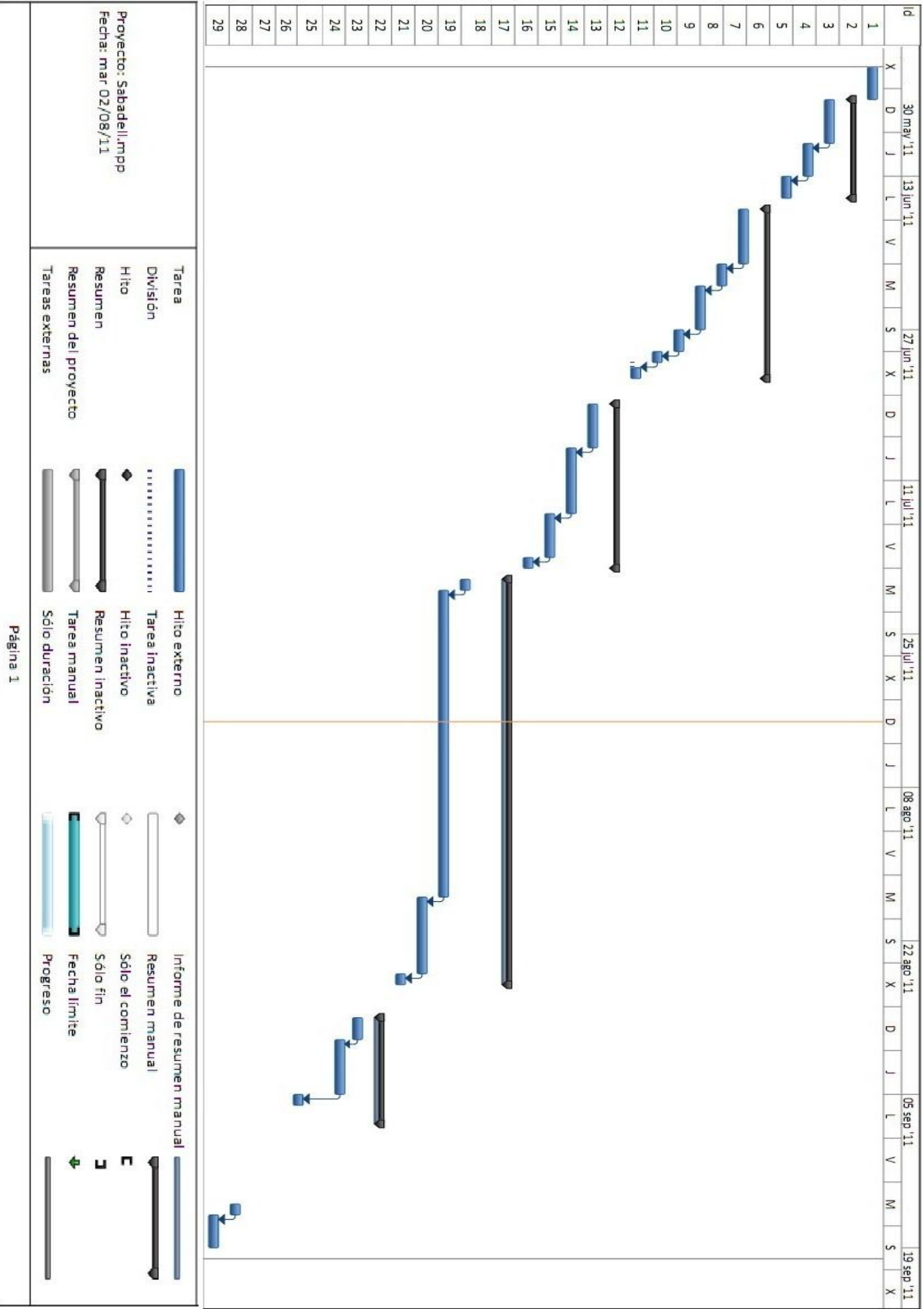


FIGURA 2: Diagrama de Gantt

## 2.4 DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES

---

A continuación definiremos los acrónimos y abreviaciones que utilizaremos en el resto de la documentación:

- BD: Base de datos.
- SGBD: Sistema gestor de base de datos.
- Licencia BSD: Licencia de software otorgada principalmente para los sistemas operativos derivador de Unix. Es una licencia de software libre permisiva. Esta licencia tiene menos restricciones en comparación con otras. La licencia BSD permite el uso del código fuente en software no libre.
- Open Source: Término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.
- ORM: Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.
- JavaScript: Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

## 2.5 REFERENCIAS

---

Referenciaremos las direcciones que explican las normativas vigentes en cuanto protección de datos, normativa de proyecto y documentación extraída como ayuda de internet:

- Normativa de proyecto de ingeniería técnica informática:

## 2.6 PRODUCTO Y DOCUMENTACIÓN DEL PROYECTO

---

A la escuela se le entregará los mencionados materiales:

- Aplicación informática que cumpla unos requisitos establecidos.
- Una memoria del proyecto.

## 2.7 DIAGNOSTICO DEL SISTEMA ACTUAL

---

No había ninguna aplicación web realizada porque la puesta en marcha de los grados empezó en el curso 2009/2010 y se realizaba toda la gestión de forma manual.

Actualmente para el curso 2010/2011 se ha puesto en marcha una aplicación de Guías Docentes, en las que el coordinador de estudios puede asignar competencias a las asignaturas.

No hemos tenido acceso a dicha aplicación por lo que nuestra elaboración parte desde cero.

## 2.8 ALTERNATIVAS TECNOLOGICAS

---

La decisión de utilizar Java como lenguaje de programación ha sido por la tendencia empresarial a utilizar dicha tecnología para el desarrollo de aplicaciones empresariales.

1. En cuanto a la herramienta de Mapeo Objeto-relaciona las alternativas con más peso para dicha herramienta son las siguientes:

- **JPA:** Java Persistence API, es la API de persistencia desarrollada para la plataforma Java EE. Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).



FIGURA 3: Logo Java

- **HIBERNATE:** Herramienta de Mapeo Objeto-relacional (ORM) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.



FIGURA 4: Logo Hibernate

- **iBATIS:** Es un framework (método de trabajo) de código abierto basado en capas desarrollado por Apache Software Foundation, que se ocupa de la capa de Persistencia (se sitúa entre la lógica de Negocio y la capa de la Base de Datos).



FIGURA 5: Logo Ibatis

2. Encontramos muchas más alternativas para el SGBD entre las que destacaremos las más influyentes en el mercado:

Podemos diferenciar entre SGBD libres:

- **Apache Derby:** Sistema gestor de base de datos relacional escrito en Java que puede ser empotrado en aplicaciones Java y utilizado para procesos de transacciones online. Tiene un tamaño de 2 MB de espacio en disco. Inicialmente distribuido como IBM Cloudscape, Apache Derby es un proyecto open source licenciado bajo la Apache 2.0 License. Actualmente se distribuye como Sun Java DB.



FIGURA 6: Logo Apache Derby

- **MySQL:** Sistema de gestión de bases de datos relacional, multihilo y multiusuario.



FIGURA 7: Logo MySQL

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

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales.



FIGURA 8: Logo PostgreSQL

Y SGBD no libres:

- **Microsoft Access:** Programa utilizado en los sistemas operativos Microsoft Windows, para la gestión de bases de datos, creado y modificado por Microsoft y orientado a ser usado en entornos personales o en pequeñas organizaciones. Permite manipular los datos en forma de tablas (formadas por filas y columnas), crear relaciones entre tablas, consultas, formularios para introducir datos e informes para presentar la información.



FIGURA 9: Logo Microsoft Access

- **Microsoft SQL Server:** Sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.



FIGURA 10: Logo Microsoft SQL Server

- **Oracle:** Sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos. Su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL o MySQL. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.



FIGURA 11: Logo Oracle

3. Frameworks de desarrollo web existen muchos, destacaremos los más utilizados:

- **Struts:** Esta es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE (Java 2, Enterprise Edition). Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en que Java Enterprise esté disponible. Lo convierte en una herramienta altamente disponible.



FIGURA 12: Logo Struts

- **EJB 3:** Los Enterprise Java Beans (EJB) son una plataforma para construir aplicaciones de negocio portables, reusables y escalables haciendo uso de Java. El desarrollador ve a un EJB como una porción de código que se ejecuta en un contenedor EJB, que no es más que un ambiente especializado (runtime) que posee determinados componentes de servicio.



FIGURA 13: Logo EJB 3.0

- **Ruby on Rails:** Framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración.



FIGURA 14: Ruby on Rails

- **Spring Framework MVC:** El Spring Framework MVC (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java.



FIGURA 15: Spring Framework MVC

A la hora de realizar el proyecto de viabilidad debemos tener en cuenta diferentes factores o problemas que pueden hacer que el proyecto se demore con respecto a la estimación realizada. En la siguiente lista indicaremos los riesgos que podemos encontrar a la hora de realizar dicha aplicación:

**R1. Incompatibilidad entre los navegadores de Internet:**

Problemas que pueden surgir al visualizar la página según los navegadores existentes.

**R2. Seguridad:**

Si la seguridad no es apropiada los usuarios podrían ver datos privados.

**R3. Planificación temporal optimista:**

Estudio de viabilidad. No se acaba en la fecha prevista aumentando los recursos.

**R4. Presupuesto poco ajustado:**

Estudio de viabilidad. Menos calidad, pérdidas económicas.

**R5. Herramientas de desarrollo inadecuadas:**

Implementación. Retraso en la finalización del proyecto, menos calidad...

**R6. Dificultad para comunicarse con el cliente:**

Estudio de viabilidad, análisis, pruebas, formación. Falta de requisitos o son inadecuados, retrasos etc.

**R7. Cambio de requisitos:**

Retraso en el desarrollo y resultado.

**R8. Abandono del proyecto antes de la finalización:**

En cualquier fase. Pérdidas económicas, frustración.

Podremos ver la probabilidad que se dé el riesgo y el posible impacto que tendrá sobre el proyecto con la tabla 2:

- Escala Probabilidad: Alto-Medio-Bajo
- Escala Impacto: Catastrófico-Crítico-Medio-Indiferente

	Probabilidad	Impacto
R1	Media	Medio
R2	Alta	Critico
R3	Alta	Critico
R4	Alta	Medio
R5	Media	Medio
R6	Bajo	Critico
R7	Alto	Critico
R8	Bajo	Catastrófico

TABLA 2: Evaluación de riesgos

### 2.9.1 PLAN DE CONTINGENCIAS

Para poder afrontar la anterior lista de riesgos, vamos a desarrollar en la tabla 3, un plan de contingencias para solucionar dichos riesgos si se produjesen:

	Solución que se adoptará
R1	Diseño de página de manera que funcione correctamente en cualquier navegador, a tener en cuenta en la fase de pruebas y depuración de errores.
R2	Para el acceso a BD se requerirá Usuario y Contraseña que recogerá el servidor mediante la ayuda de Hibernate con la seguridad que ello conlleva para que cualquiera no pueda acceder a datos confidenciales.
R3	Afrontar posibles pérdidas y desechar funcionalidades secundarias.
R4	Se tendrán en cuenta las leyes LOPD y LSSI.
R5	Afrontar posibles pérdidas y volver a negociar con el cliente la nueva situación.
R6	Proposición de nuevas alternativas.
R7	Acordar una fecha en la que estén de acuerdo las dos partes.
R8	Renegociar con el cliente, aplazar funcionalidad, modificar planificación y presupuesto.
R9	Sin solución.

TABLA 3: Plan de contingencias

## 2.10 PRESUPUESTO

En este apartado vamos a estudiar los costes asociados del proyecto y el coste total.

- Primero vamos a valorar en la tabla 4 los ingresos medios de los trabajadores, orientativos para los diferentes roles dentro de la organización empresarial:

Recursos Humanos	Ingresos
Director del proyecto	150€/h
Jefe de proyecto	100€/h
Analista	50€/h
Programador	30€/h

TABLA 4: Presupuesto por roles

- Basándonos en la tabla 3 de ingresos medios de los empleados, lo aplicaremos a las horas totales estimadas del proyecto (visto en el punto Planificación) obteniendo la tabla 5:

Recursos Humanos	Horas	Total * horas
Director del proyecto	10h	1.500€
Jefe de proyecto	24h	2.400€
Analista	33h	1.650€
Programador	180€	5.400€
<b>TOTAL</b>		<b>10.950€</b>

TABLA 5: Presupuesto basado en la planificación

- Para desarrollar el proyecto debemos de añadir los gastos de recursos necesarios para su creación obteniendo la tabla 6 con los costes totales:

	Coste de amortización	Coste unitario	Periodo utilización
PC Programador	100€	1.200€	3meses
MSOffice	20€	250€	3meses
MSProject	30€	360€	3meses
<b>TOTAL</b>	<b>150€</b>	<b>30€</b>	

TABLA 6: Gastos de recursos necesarios

Resumen de los costes totales:

- Coste de desarrollo del proyecto → **10.950€**
- Coste de amortización del material → **150€**

**TOTAL: 11.100€**

## 2.11 CONCLUSIONES

---

Tras la realización del estudio de viabilidad hemos resuelto que este proyecto se realizará en un periodo de 3 meses con un total de 294 horas de trabajo aproximadamente.

Podemos decir que el proyecto es viable ya que conlleva un gasto de realización soportable, además de resolver los objetivos principales de dicha tarea.





### 3. ANÁLISIS

---

#### 3.1 INTRODUCCIÓN

---

La aplicación tiene la función principal de asignar competencias a materias en los nuevos estudios de grado por parte de un administrador y el coordinador de estudios correspondiente.

Además de esta función principal, cada alumno, logándose, podrá consultar la información completa sobre el grado al que pertenece.

Todo alumno registrado puede ponerse en contacto con el administrador para realizarle consultas o modificación de sus datos académicos.

#### 3.2 DEFINICION DE REQUISITOS FUNCIONALES

---

1. **CONTROL DE ACCESO:** El usuario no tendrá la posibilidad de registrarse, ya que solo podrá consultar la jerarquía tras elegir un estudio.

En cambio tanto administrador como coordinador deben registrarse para poder acceder a algunas partes de la aplicación.

Para acceder a dichas partes deben logarse. Un usuario (sea administrador o coordinador) logado se considera cuando introduce tanto “usuario” como “password” correctamente, como se muestra en la tabla 7.

Este “usuario” y “password” será suministrado por el administrador.

Desde cualquier punto de la aplicación el usuario puede logarse, saltándole el registro automáticamente si accede a un área con necesidad de identificación por parte del usuario.

En cualquier momento el usuario puede efectuar el logout.

Campos	Descripción	Categoría
Username	Nombre de usuario	Obligatorio
Password	Contraseña de usuario	Obligatorio

TABLA 7: Control de acceso

2. MANTENIMIENTO (Alta, baja y modificación) DE TABLAS: El mantenimiento es trabajo únicamente del administrador y del coordinador.

Veremos a continuación los mantenimientos de los que están encargados cada uno:

- ADMINISTRADOR:

- ESTUDIOS: El administrador debe registrar los estudios disponibles añadiendo los siguientes campo declarados en la tabla 8:

Campo	Descripción	Categoría
IdEstudio	Identificador de estudio	Obligatorio
Nombre	Nombre de estudio	Obligatorio
Acrónimo	Acrónimo de estudio	Optativo

TABLA 8: Tabla Estudios

- MODULOS: Debe registrar los módulos añadiendo los siguientes campo indicados en la tabla 9:

Campo	Descripción	Categoría
IdModulo	Identificador de modulo	Obligatorio
Nombre	Nombre de modulo	Obligatorio

TABLA 9: Tabla Módulos

- MATERIAS: El administrador debe registrar las materias con los campos indicados en la tabla 10:

Campo	Descripción	Categoría
IdMateria	Identificador de materia	Obligatorio
Nombre	Nombre de la materia	Obligatorio
IdTipo	Identificador de tipo	Obligatorio
Carácter	Identificador de carácter	Obligatorio
Créditos	Créditos de la materia	Obligatorio

TABLA 10: Tabla Materias

- ASIGNATURAS: Debe registrar las asignaturas con los campos indicados en la tabla 11:

Campo	Descripción	Categoría
IdAsignatura	Identificador de asignatura	Obligatorio
Nombre	Nombre de asignatura	Obligatorio
IdTipo	Identificador de tipo	Obligatorio
IdCaracter	Identificador de carácter	Obligatorio

TABLA 11: Tabla Asignaturas

- TIPO DE COMPETENCIAS: En la tabla 12 se muestran los campos que el administrador debe ingresar para el registro de los tipos de competencias:

Campo	Descripción	Categoría
IdTipoCompetencia	Identificador de tipo de competencia	Obligatorio
Descripcion	Descripción del tipo de competencia	Obligatorio

TABLA 12: Tabla Tipo de competencia

- NIVEL DE COMPETENCIA: Debe registrar los siguientes campos indicados en la tabla 13, para el registro de los niveles de competencias:

-

Campo	Descripción	Categoría
IdNivelCompetencia	Identificador del nivel de competencia	Obligatorio
Descripción	Descripción del tipo de competencia	Obligatorio

TABLA 13: Tabla Nivel de competencia

- COMPETENCIAS (GLOBALES): En la tabla 14 se muestran los campos que el administrador debe ingresar para el registro de competencias globales para toda la universidad:

Campo	Descripción	Categoría
IdCompetencia	Identificador de competencia	Obligatorio
IdTipoCompetencia	Identificador del tipo de competencia	Obligatorio
Descripcion	Descripción de la competencia	Obligatorio

TABLA 14: Tabla Competencias

- COORDINADOR:

- COMPETENCIAS (ESPECIFICAS): Debe registrar los siguientes campos visibles en la tabla 15, para el registro de competencias específicas del coordinador:

Campo	Descripción	Categoría
IdCompetencia	Identificador de competencia	Obligatorio
IdTipoCompetencia	Identificador del tipo de competencia	Obligatorio
Descripcion	Descripción de la competencia	Obligatorio

TABLA 15: Tabla Competencias

3. ASIGNACIONES: Las asignaciones es trabajo únicamente del coordinador.

- COORDINADOR:

- ENTRE ESTUDIOS-MODULOS: En la tabla 16 se muestran los campos que debe rellenar el coordinador para poder relacionar un estudio con un modulo :

Campo	Descripción	Categoría
IdEstudio	Identificador de Estudio	Obligatorio
IdModulo	Identificador de Modulo	Obligatorio

TABLA 16: Tabla Estudios\_Modulos

- ENTRE MODULOS-MATERIAS: El coordinador debe relacionar módulos con materias rellenando los campos que se muestran en la tabla 17:

Campo	Descripción	Categoría
IdModulo	Identificador de Modulo	Obligatorio
IdMateria	Identificador de Materia	Obligatorio

TABLA 17: Tabla Modulos\_Materias

- ENTRE MATERIAS-ASIGNATURAS: En la tabla 18 se muestran los campos que debe rellenar el coordinador para poder relacionar la materia seleccionada con una asignatura :

Campo	Descripción	Categoría
IdMateria	Identificador de Materia	Obligatorio
IdAsignatura	Identificador de Asignatura	Obligatorio

TABLA 18: Tabla Materias\_Asignaturas

- ENTRE MATERIAS-COMPETENCIAS: Como se puede observar en la tabla 19, el coordinador deberá relacionar materias con competencias a través de sus campos:

Campo	Descripción	Categoría
IdMateria	Identificador de Materia	Obligatorio
IdCompetencia	Identificador de Competencia	Obligatorio

TABLA 19: Tabla Materias\_Competencias

- ENTRE COMPETENCIAS-ASIGNATURAS: En la tabla 20 se muestran los campos que debe rellenar el coordinador para poder relacionar la competencia con las asignaturas :

Campo	Descripción	Categoría
IdAsignatura	Identificador de asignatura	Obligatorio
IdCompetencia	Identificador de competencia	Obligatorio
IdNivelCompetencia	Identificador del nivel de competencia	Obligatorio
Evaluacion	Evaluación de competencias en las asignaturas	Optativo

TABLA 20: Tabla Competencias\_Asignaturas

4. CONSULTAS: Las consultas las podrán realizar todos los usuarios sin necesidad de registrarse en la aplicación web.

Mostrara las competencias relacionadas con cada según el estudio elegido.

Esta consulta se visualizará en el campo Competencias del menú que veremos explicado más detalladamente a continuación.

---

### 3.3 DEFINICION DE REQUISITOS NO FUNCIONALES

---

#### 1. Cumplimiento de estándares:

Se debe intentar que la aplicación cumpla con los estándares para tener una correcta visualización en distintos navegadores (Internet Explorer, Firefox, Chrome).

La aplicación deberá poder utilizar tanto con los navegadores Internet Explorer de Microsoft, Firefox de Mozilla y Chrome de Google.

Podrán existir diferencias en la visualización pero en ningún caso esta diferencia representará un funcionamiento incorrecto de la aplicación.

#### 2. Control de todas las entradas de usuarios:

Para minimizar el uso indebido de la aplicación, o con el fin de extraer estadísticas de su uso como posible ampliación futura, habrá que tener controlada la entrada de los usuarios a la aplicación, así como las diferentes acciones que realicen de la misma.

#### 3. Seguridad:

La base de datos estará protegida con contraseña para garantizar que sólo pueda acceder a la base de datos del sistema el administrador de la aplicación, ya sea para realizar copias de seguridad, hacer algún tipo de reparación o modificación etc., ésta estará protegida por contraseña.

#### 4. Respuesta a los errores:

Los posibles errores que haga el usuario serán validados a nivel de cliente, evitando campos vacíos, mal rellenados, etc.

También orientado a la seguridad de la aplicación, a la hora de rellenar los formularios, se controlará el tipo de información introducida, y en caso de no coincidir con la requerida se informará al usuario del error correspondiente intentando minimizar la repetición de entrada de datos en los sucesivos intentos.

#### 5. Eficiencia y facilidad:

La aplicación web debe estar bien estructurada, que la navegación sea clara e intuitiva, ofreciendo al usuario lo que necesite.

La aplicación desde un principio ha de ofrecer al usuario lo que busca, como poder efectuar el login una vez accede a la aplicación.

Una vez dentro de la aplicación, todas las funciones que nos ofrece deben ser fáciles de utilizar, como la visibilidad de los nuevos grados.

A la vez todo debe estar bien estructurado y bien indicado para hacer la aplicación lo más amigable posible al usuario que la utilice.

6. Normalización del proyecto según la normativa de proyectos de ingeniería técnica:

Con el fin de que la aplicación sea reconocida como proyecto de final de carrera, ésta cumplirá con todos los requisitos que se indican en el documento de la normativa de proyectos de ingeniería técnica, como los plazos de presentación, el material a entregar, estructura y formato de la memoria etc.

Este documento lo podemos encontrar en la página:

[http://www.uab.cat/Document/541/595/Normativa\\_PFCNovembre2010.pdf](http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf).

---

### 3.4 CASOS DE USOS

Tenemos tres casos de usos de nuestra aplicación, la que corresponde con un usuario no registrado, la del administrador y la del coordinador.

Las explicaremos a continuación:

- USUARIO NO REGISTRADO:



- Información sobre Competencias.
- Información sobre la aplicación.
- Correo de contacto

- ADMINISTRADOR:



- Información sobre Competencias.
- Información sobre la aplicación.
- Logeo como usuario registrado.
- Correo de contacto



- Logout.
- Administración de usuarios.
- Administración de estudios.
- Administración de materias.
- Administración de asignaturas.
- Administración de niveles de competencia.
- Administración de tipo de competencia.
- Administración competencias globales.

- COORDINADOR:



- Información sobre Competencias.
- Información sobre la aplicación.
- Logeo como usuario registrado.
- Correo de contacto



- Logout.
- Administración competencias específicas.
- Asignación de competencias con materias y asignaturas indicando el nivel su evaluación.

## 4. DISEÑO

---

### 4.1 INTRODUCCION

---

En este capítulo se darán a conocer los principales puntos de diseño de la aplicación.

Primero indicaremos las tecnologías elegidas explicando los principales motivos de su selección y sus características más importantes.

A continuación se hará el diseño de la base de datos, y por último se presentará una descripción de la interfaz de usuario, el diseño de la página principal y el mapa de las diferentes páginas que forman la aplicación.

### 4.2 SELECCIÓN DE TECNOLOGIA

---

Existe multitud de framework para el desarrollo Java Web con los que poder abordar el problema como por ejemplo Struts, Spring MVC, JSF, .Net...

Nuestra aplicación la vamos a realizar utilizando diferentes framework cada uno dedicado a simplificar un ámbito del desarrollo, entre ellos destacar:

- **Hibernate:** Como hemos mencionado anteriormente es una herramienta de Mapeo Objeto-relacional (ORM) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.



FIGURA 16: Logo Hibernate

#### ¿Por qué utilizar Hibernate?

Hibernate es una ORM de libre distribución, que además, es de las más maduras y completas. Actualmente su uso está muy extendido y además está siendo desarrollada de forma muy activa. Una característica muy importante que distingue Hibernate de otras soluciones al problema de la persistencia, como los EJBs de entidad, es que la clase Hibernate persistente puede utilizarse en cualquier contexto de ejecución, es decir, no se necesita un contenedor especial para ello.

- **Tiles**: Framework que facilita la utilización de esquemas (layout) y plantillas (templates) para aplicaciones web. Este framework utiliza un archivo de configuración donde se declaran las redirecciones de las plantillas.

- **SpringSecurity**: Proporciona servicios de seguridad para aplicaciones de software empresariales basados en J2EE.

#### ¿Por qué utilizar SpringSecurity?

Porque está muy ligado al framework Spring MVC por lo que se trabaja de la misma manera en cuanto a dependencias y aparte ofrece el servicio integro sin tener que desarrollar uno mismo la seguridad completa de la aplicación.

- **MySQL**: Sistema de gestión de bases de datos relacional, multihilo y multiusuario. Su popularidad como aplicación web está muy ligada también a PHP.



FIGURA 17: Logo MySQL

#### ¿Por qué utilizar MySQL?

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. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

- **Log4j**: Biblioteca open source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o “logs”.



FIGURA 18: Logo Log4j

- **Spring Framework MVC:** Framework de Java que nos facilitará la creación de aplicaciones.



FIGURA 19: Logo Spring Framework MVC

### **¿Por qué utilizar Spring Framework MVC?**

Lo hemos utilizado porque, se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa y sustituto del modelo de Enterprise JavaBean. Por su diseño ofrece mucha libertad a los desarrolladores en Java y posee soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

### 4.3 DEFINICION DE LA BD

---

En este punto podremos ver con más detalle la base de datos, la cual nos servirá para guardar los datos necesarios para conseguir un buen funcionamiento de la aplicación.

Pudiendo acceder de forma fácil a la misma, ya sea para hacer consultas, modificarla, futuras ampliaciones.

En las tablas principales se guarda información sobre los usuarios que utilizan la aplicación. También veremos que existen otras tablas que nos sirven como complemento de las principales. A continuación se hará un breve resumen sobre cada una de las tablas que forman parte de la base de datos.

#### 4.3.1 TABLA DE ESTUDIOS

---

Se recoge las titulaciones existentes a través del identificador nombre, acrónimo del estudio, además de señalar dicho estudio a que modulo pertenece. Con lo cual relacionar la tabla de usuarios con la de estudios y esta a la vez con la de módulos que describiremos a continuación.

#### 4.3.2 TABLA DE MÓDULOS

---

En esta tabla se recoge toda la información de los grupos de materias de los que se dispone. Está organizado por identificador de módulo y el nombre que lo describe.

#### 4.3.3 TABLA DE MATERIAS

---

En esta tabla representamos los grupos de asignaturas que tienen competencias asignadas. Las identificamos a través de su código identificativo, nombre, tipo, carácter, créditos y al grupo al que pertenece (identificador de módulo).

#### 4.3.4 TABLA DE ASIGNATURAS

---

Se recogen las asignaturas existentes a través del identificador, el nombre, tipo, carácter y créditos de cada una de las asignaturas, indicando además a que materia pertenece.

#### 4.3.5 TABLA DE TIPO DE COMPETENCIA

---

Recogemos los tipos de competencias que definiremos en esta tabla, a través de su identificador y su descripción.

#### 4.3.6 TABLA DE NIVEL DE COMPETENCIA

---

Recogemos los niveles que poseerán las competencias que los definiremos con el atributo identificador de nivel y su descripción.

#### 4.3.7 TABLA DE COMPETENCIAS

---

Recogemos las competencias comunes en los grados. Las competencias son los conocimientos generales que debe adquirir el usuario dependiendo de sus estudios.

Estas competencias tienen tipos, que serán recogidas de la tabla Tipo de Competencia antes descrita. Además posee una descripción donde se redacta la competencia.

#### 4.3.8 TABLA DE COMPETENCIAS-MATERIAS

---

Tabla en la que relacionamos un identificador de competencias con un identificador de materias.

#### 4.3.9 TABLA DE COMPETENCIAS-ASIGNATURAS

---

Tabla donde relacionamos Asignatura con Competencia y con Nivel de Competencia, todos a través del identificador de cada uno de ellos que elegiremos correspondientemente de las tablas Asignaturas, Competencias y NivelCompetencia.

A esta tabla le añadiremos el campo Evaluación, donde se describirá el tipo de evaluación que obtendrá.

#### 4.3.10 ESQUEMA BD

A continuación podremos ver el esquema de la BD de nuestra aplicación en la figura 20.

En él se podrán ver las claves principales y foráneas de cada tabla y la relación entre tablas.

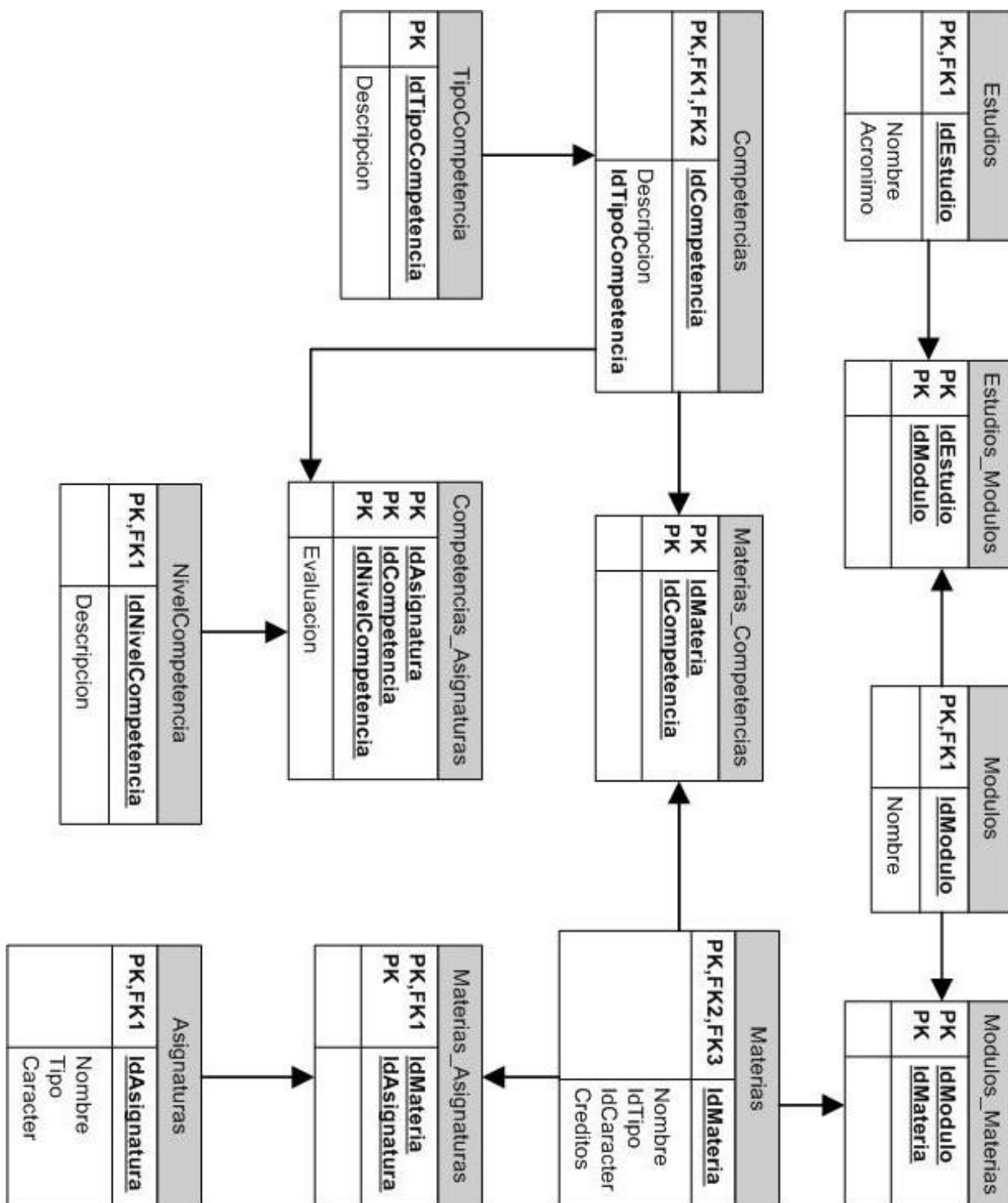


FIGURA 20: Esquema de base de datos

Describiremos en la tabla 21 las relaciones entre tablas mostrando sus claves primarias y foráneas con las que se relacionan con otras tablas:

Relaciones	Pk	Fk
Estudios ( IdEstudio, Nombre, Acronimo )	IdEstudio	
Modulos ( IdModulo, Nombre )	IdModulo	
Materias ( IdMateria, Nombre, IdTipo, IdCarácter, Creditos )	IdMateria	IdTipo/ <b>Tipo</b> IdCaracter/ <b>Caracter</b>
Asignaturas ( IdAsignatura, Nombre, IdTipo, Id Carácter )	IdAsignatura	IdAsignatura/ <b>Materias _Asignaturas</b> IdTipo/ <b>Tipo</b> IdCaracter/ <b>Caracter</b>
Competencias (IdCompetencia, IdTipoCompetencia, Descripcion )	IdCompetencia	IdTipoCompetencia/ <b>TipoCompetencia</b>
TipoCompetencia ( IdTipoCompetencia, Descripcion )	IdTipoCompetencia	
NivelCompetencia ( IdNivelCompetencia, Descripcion )	IdNivelCompetencia	
Estudios_Modulos ( IdEstudio, IdModulo )	IdEstudio, IdModulo	IdEstudio/ <b>Estudios</b> IdModulo/ <b>Modulos</b>
Modulos_Materias ( IdModulo, IdMateria )	IdModulo, IdMateria	IdMateria/ <b>Materias</b> IdModulo/ <b>Modulos</b>
Materias _Asignatuas ( IdMateria, IdAsignatura )	IdMateria, IdAsignatura	IdMateria/ <b>Materias</b> IdAsignatura/ <b>Asignaturas</b>
Materias _Competencias ( IdMateria, IdCompetencia )	IdMateria, IdCompetencia	IdMateria/ <b>Materias</b> IdCompetencia/ <b>Competencias</b>
Competencias _Asignaturas ( IdCompetencia, IdAsignatura, IdNivelCompetencia, Evaluacion )	IdCompetencia, IdAsignatura, IdNivelCompetencia	IdAsignatura/ <b>Asignaturas</b> IdCompetencia/ <b>Competencias</b> IdNivelCompetencia/ <b>NivelCompetencia</b>

TABLA 21: Relaciones entre tablas base de datos

## 4.4 INTERFAZ DE USUARIO

---

El diseño de una aplicación Web es una de las etapas más importantes durante su desarrollo. Un buen diseño, que resulte intuitivo, sencillo y claro, permitirá al usuario final navegar por la aplicación con comodidad.

Esta parte tiene como objetivo la definición de la interfaz de usuario de la aplicación, en la que se detallará el diseño de la plantilla principal que se utilizará y se describirá el mapa de la aplicación Web donde se podrán ver las diferentes páginas con las que esta está formada y la relación que hay entre ellas.

Este diseño determinará en detalle la funcionalidad ofrecida por la aplicación Web.

### 4.4.1 PLANTILLA DE LA APLICACIÓN

---

Toda aplicación web posee como base una plantilla donde su contenido se modificará en la parte principal de la misma.

En la figura 21 podemos ver las diferentes secciones de las que se compone la plantilla principal de nuestra aplicación web y una breve descripción de cada una de sus partes.

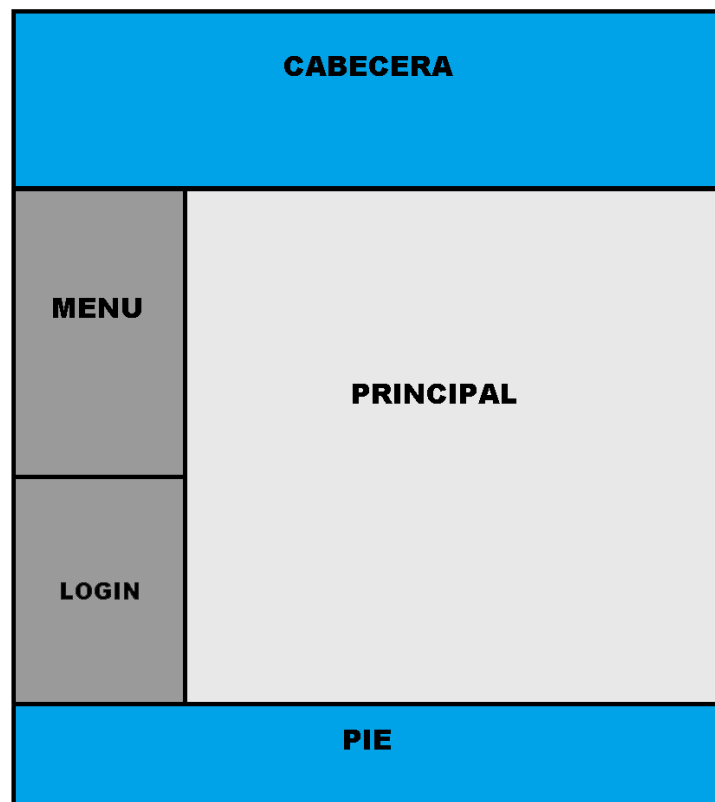


FIGURA 21: Plantilla de la aplicación

- CABECERA: Sección de la página donde estableceremos el logotipo de la aplicación además de servir como acceso directo a la página inicial.

- LOGIN: En esta parte de la página es donde el usuario podrá efectuar el logeo a la aplicación. Tendrá que introducir tanto “usuario” como “password” obligatoriamente.

En la sección LOGIN, una vez logado correctamente se mostrara la hora de inicio y la opción de Logout además de un mensaje bienvenida en la sección PRINCIPAL.

- MENU: Se mostrará las diferentes opciones de menú por donde navegar en la aplicación.

Dependiendo de los privilegios del usuario registrado tendrá acceso a diferentes campos en el menú.

Un **usuario no logado** en la aplicación podrá consultar información dentro del menú *Inicio* donde se dará la bienvenida a la aplicación. La sección de *Aplicación* donde se explicará el funcionamiento de la aplicación y el menú *Grados* obteniendo información sobre los grados disponibles en la escuela además de sus correspondientes competencias. Por último tendrá acceso al campo *Contacto* donde podrá ponerse en contacto a través de un correo electrónico con el administrador.

Un **usuario logado Administrador** tendrá acceso al menú de administración donde podrá administrar todas las tablas de la BD anteriormente explicadas.

Un **usuario logado Coordinador** tendrá acceso al menú de asignaciones anteriormente explicadas.

- PRINCIPAL: Sección que ofrecerá el contenido principal de cada página de la aplicación.

- PIE: Esta sección contiene accesos directos fijos independientes del punto de la aplicación en la que nos encontremos.

Los accesos directos que encontraremos en esta sección serán: Universidad Autónoma de Barcelona (Sabadell), quienes somos y contacto.

A continuación mostraremos una captura de la aplicación web completa en la figura 22:



FIGURA 22: Captura aplicación Web

#### 4.4.2 MAPA DE LA APLICACIÓN WEB

En este apartado vamos a mostrar el mapa web de cada uno de los perfiles que posee nuestra aplicación con sus diferentes páginas de las que está compuesta y la relación que hay entre ellas.

Vamos a diferenciar el mapa de la aplicación web en sus tres partes diferenciadas:

- Usuario
- Coordinador
- Administrador

• **Usuario:** Como hemos mencionado ya, los campos a los que puede acceder un usuario no logado son comunes tanto para coordinador como administrador. En la figura 23 se muestra una imagen de las páginas a las que accede.

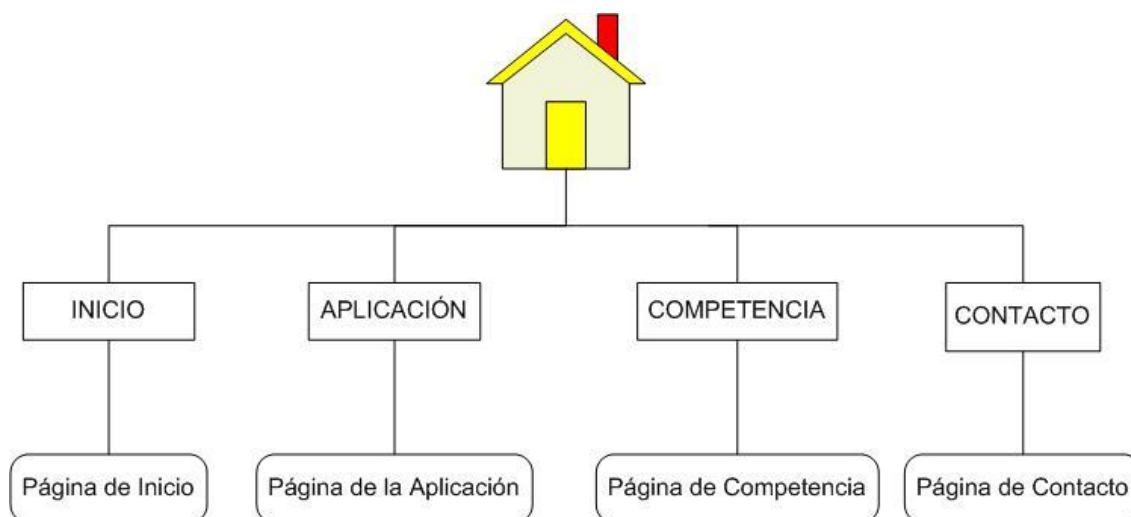


FIGURA 23: Mapa aplicación Web usuario

Como podemos observar, el usuario puede acceder a los campos de Inicio, Aplicación, Competencia y Contacto.

El campo Inicio se da la bienvenida a la aplicación y se explica el funcionamiento de la misma.

En el campo Aplicación detallaremos como interactuar con la aplicación y la descripción de cada una de las partes que la componen.

La pagina de Competencia ofrece la posibilidad de tras elegir un Estudio de los existentes, poder ver según las competencias, con que materias está relacionada y según las materias, que competencias y niveles posee.

Por último en el campo Contacto, el usuario tiene la posibilidad de enviar un correo electrónico al administrador, solamente añadiendo su correo electrónico para poder ser contestado, el asunto y el cuerpo del mensaje. En la figura 24 podemos ver una pantalla del campo Contacto:

**Contacto**

Tu email:	<input style="width: 95%;" type="text"/>
Asunto:	<input style="width: 95%;" type="text"/>
Mensaje:	<input style="width: 95%;" type="text"/>
<input type="button" value="Enviar correo"/>	

FIGURA 24: Captura de Contacto

- **Coordinador**: El coordinador al logarse en la aplicación le aparece un campo nuevo llamado Menú Coordinador, en el cual como vemos en la figura 25 contiene otro submenú que detallaremos a continuación:

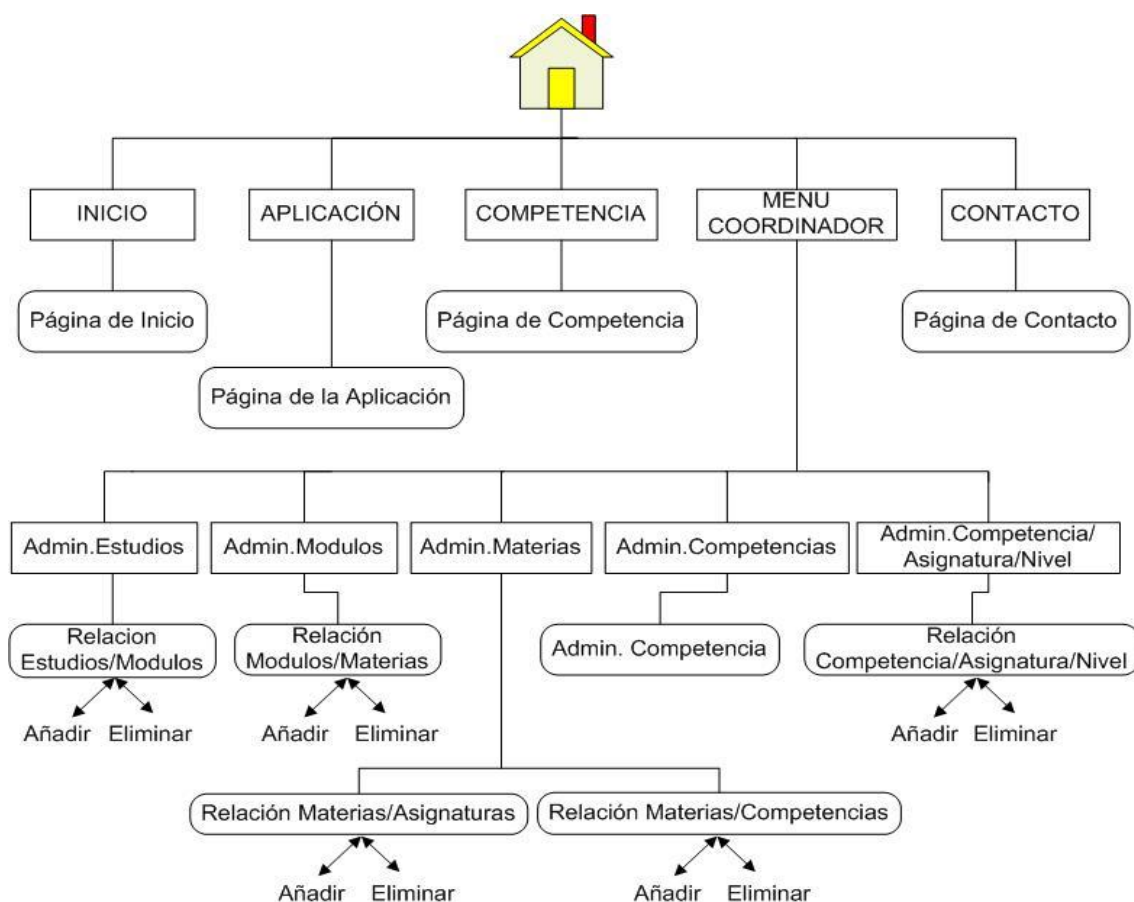


FIGURA 25: Mapa aplicación Web coordinador

El menú coordinador contiene un submenú que está compuesto por varios campos. El campo Admin.Competencias se dedica, a diferencia de los demás, en que el coordinador puede realizar el mantenimiento de las competencias, añadiendo nuevas y eliminando las no útiles.

Como podemos observar en la figura 26 se realiza todo mediante una interfaz intuitiva y sencilla. A la vez que el coordinador va realizando el mantenimiento se muestran todas las competencias existentes con toda la información que posee.

**Competencias manager**

Descripcion:	<input type="text"/>
Tipo:	TipoB ▼
<a href="#">Añadir Competencia</a>	

**Listado de Competencias**

Descripcion	Tipo	
Antonio	TipoB	<a href="#">Eliminar</a>
ce	TipoB	<a href="#">Eliminar</a>

FIGURA 26: Captura competencias manager

En cuanto a los demás submenús, se encargan de relacionar un beans con otro.

Explicaremos como ejemplo el campo de Admin.Estudios, ya que todos los demás siguen el mismo patrón de diseño y ejecución.

Al acceder vemos una tabla, como se puede observar en la figura 27, donde se relaciona un estudio con un modulo además de campos con más información. Tenemos la posibilidad a través de dos botones de añadir una nueva relación o de eliminar dicha relación. Las relaciones son uno a muchos, ya que un estudio puede poseer varios módulos, por lo que con el botón de añadir relación podemos agregar tantos módulos a un estudio como sea necesario.

### Estudios-Modulos

Nombre Estudio	Acronimo	Modulo		
EstudioA	Ea	ModuloA	+	×
EstudioA	Ea	ModuloB	+	×

FIGURA 27: Captura Estudios\_Módulos

Al pulsar en el botón añadir relación observamos otra pantalla donde se relaciona el estudio escogido con el modulo que deseemos, como se puede observa en la figura 28.

### Añadir relación Estudio/Modulos

Estudio:	<input type="text" value="EstudioA"/>
Modulo:	<input type="text" value="ModuloB"/>
<input type="button" value="Guardar"/>	

FIGURA 28: Captura relación Estudio/Módulos

- **Administrador:** Como el coordinador, el administrador al registrarse correctamente se le presenta el Menú Administrador en donde realiza además de todo lo que puede realizar el coordinador, el mantenimiento de los demás beans (Estudios, Módulos, Materias, Asignaturas, Competencias, Tipo de competencia, Carácter, Tipo y Nivel de Competencia) utilizando la misma interfaz que la explicada anteriormente en el campo Admin.Competencias.

En la figura 29 mostraremos el mapa web del administrador.

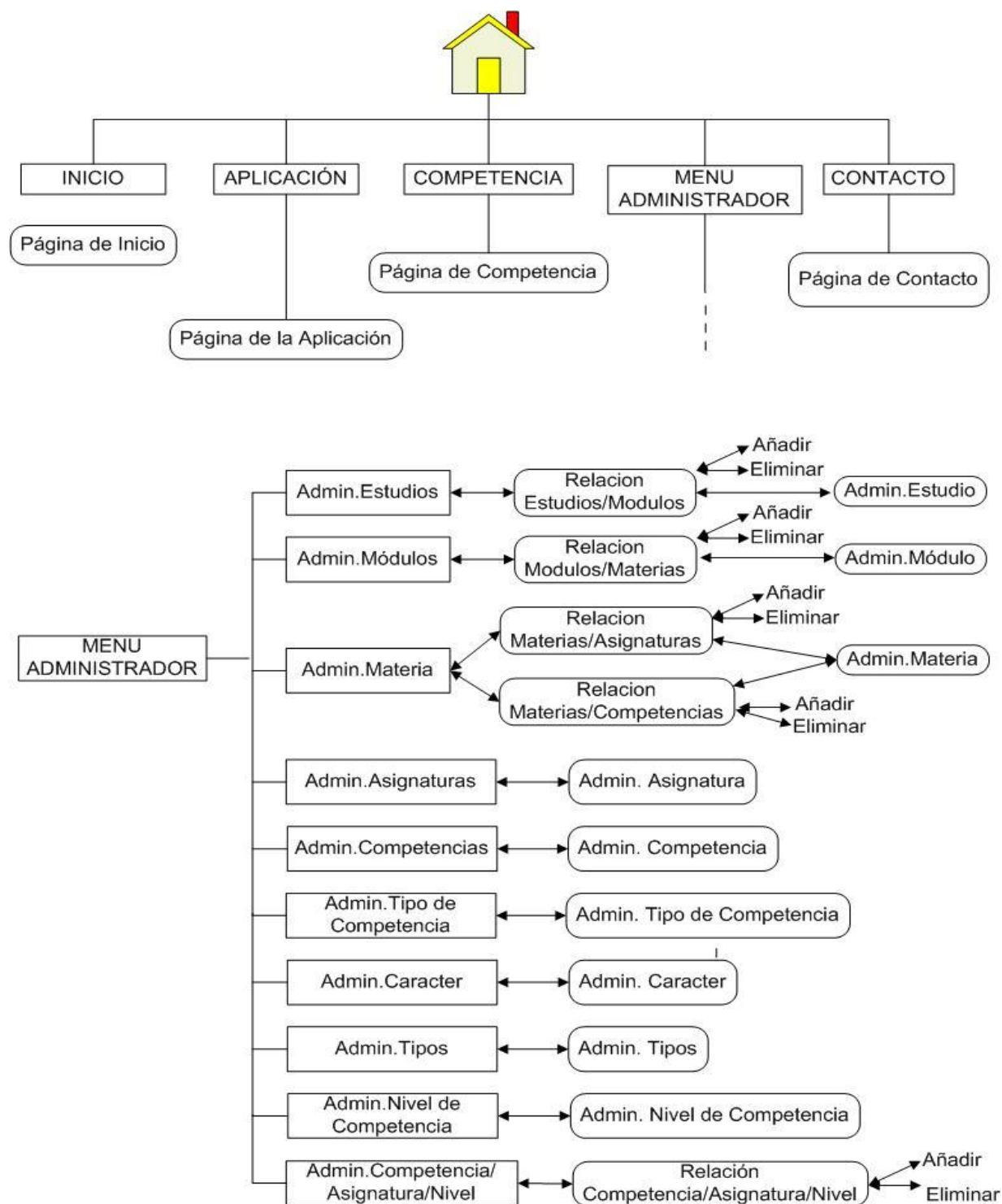


FIGURA 29: Mapa aplicación Web administrador



## 5. IMPLEMENTACION

---

La aplicación web ha sido elaborada de tal forma que tanto, los usuarios no registrados como coordinadores y administrador, tengan acceso de forma fácil e intuitiva a cualquier información que quieran consultar.

Principalmente la aplicación está dividida en dos bloques, el bloque de acceso restringido y el bloque de acceso público.

Como su nombre indica, el bloque de acceso restringido está destinado principalmente a los coordinadores dados de alta. Éstos tienen una clave de acceso y un usuario para acceder a estas zonas restringidas donde podrán ser capaces de asignar y consultar competencias sin necesidad de entender código HTML o JAVA.

Esta gestión de seguridad de sesión se lleva a cabo con Spring Security.

Como sabemos la seguridad comprende dos operaciones: La primera operación es conocida como "autenticación", por el cual se establece si un usuario (que quiere realizar una acción en nuestra aplicación) es quien dice ser, y la segunda operación es llamada "autorización" que se refiere al proceso de decidir si a un usuario le es permitido realizar una determinada acción en nuestra aplicación.

**Menú Coordinador**

MENU
Inicio
Aplicacion
Grados
Menu Coordinador
Contacto
INICIO DE SESIÓN
Gracias por logearse en la aplicacion web.
Puede disfrutar de sus servicios.
<a href="#">Logout</a>

FIGURA 30: Menú Coordinador

**Menú Administrador**

MENU
Inicio
Aplicacion
Grados
Menu Administrador
Contacto
INICIO DE SESIÓN
Gracias por logearse en la aplicacion web.
Puede disfrutar de sus servicios.
<a href="#">Logout</a>

FIGURA 31: Menú Administrador

El bloque de acceso público es donde el usuario podrá acceder a *Inicio*, donde se dará la bienvenida a la aplicación. El siguiente punto es *Aplicación*, donde se detalla el funcionamiento de la aplicación y como navegar a través de ella. El campo más importante para el usuario no registrado es el de *Competencia* donde se mostrarán las competencias existentes a través de materias o las materias

relacionadas con las competencias, todo esto a través de la selección de un estudio existente. El cual al seleccionar uno de ellos, se detallarán los módulos con los que se relaciona y así sucesivamente tanto con materias, asignaturas, competencias, tipos de competencias y niveles, creando una estructura de mayor a menor.

### **Menú Usuario**

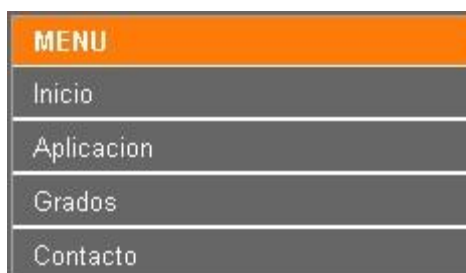


FIGURA 32: Menú Usuario

A continuación explicaremos el flujo de permisos según si estas registrado o no.

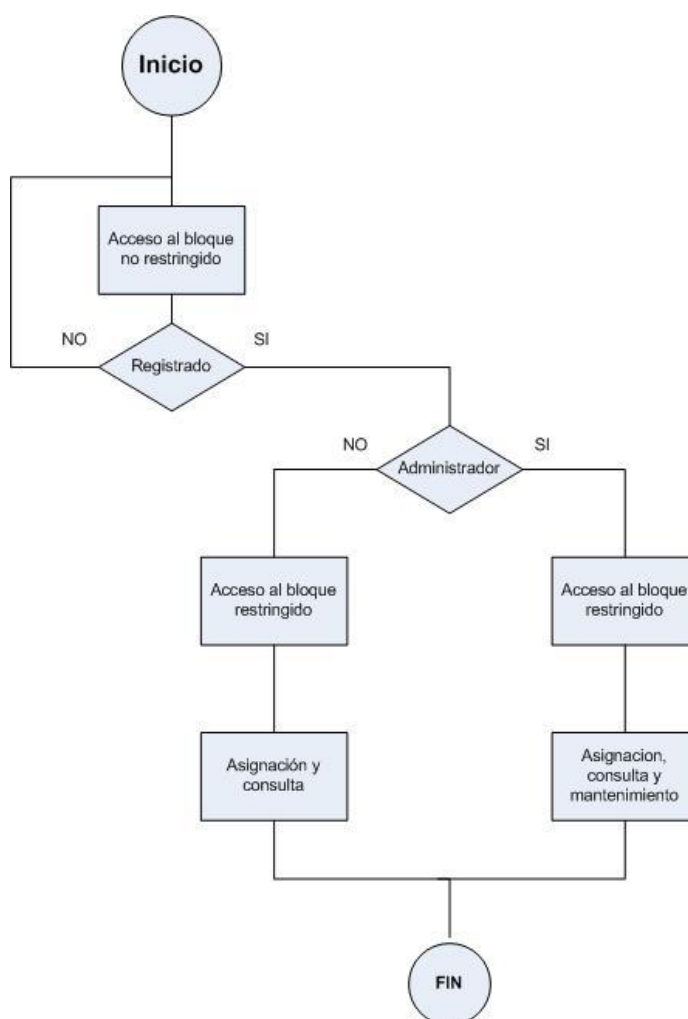


FIGURA 33: Flujo de permisos

La aplicación ha sido elaborada como hemos mencionado anteriormente con Spring como framework de implementación. Este framework está basado en el modelo MVC (Modelo-Vista-Controlador) que es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos que explicamos a continuación:

**Modelo:** Es el objeto que representa los datos, manejándolos y controlando su transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

**Vista:** Es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo generando la interfaz de usuario.

**Controlador:** Es el objeto que proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Aunque se pueden encontrar diferentes implementaciones de MVC, nosotros vamos a describir el patrón seguido en el desarrollo de nuestra aplicación:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.), a través de esa acción se le envía la notificación al controlador.
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario y gestiona el evento que le llega.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega a Tiles los objetos de la vista, que se encarga de redireccionar a la vista correspondiente. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Este patrón MVC, se ve claramente en la aplicación web.

En la figura 34 vemos el diagrama de clases de nuestro proyecto web en el entorno de desarrollo Eclipse, distinguiendo el modelo MVC.

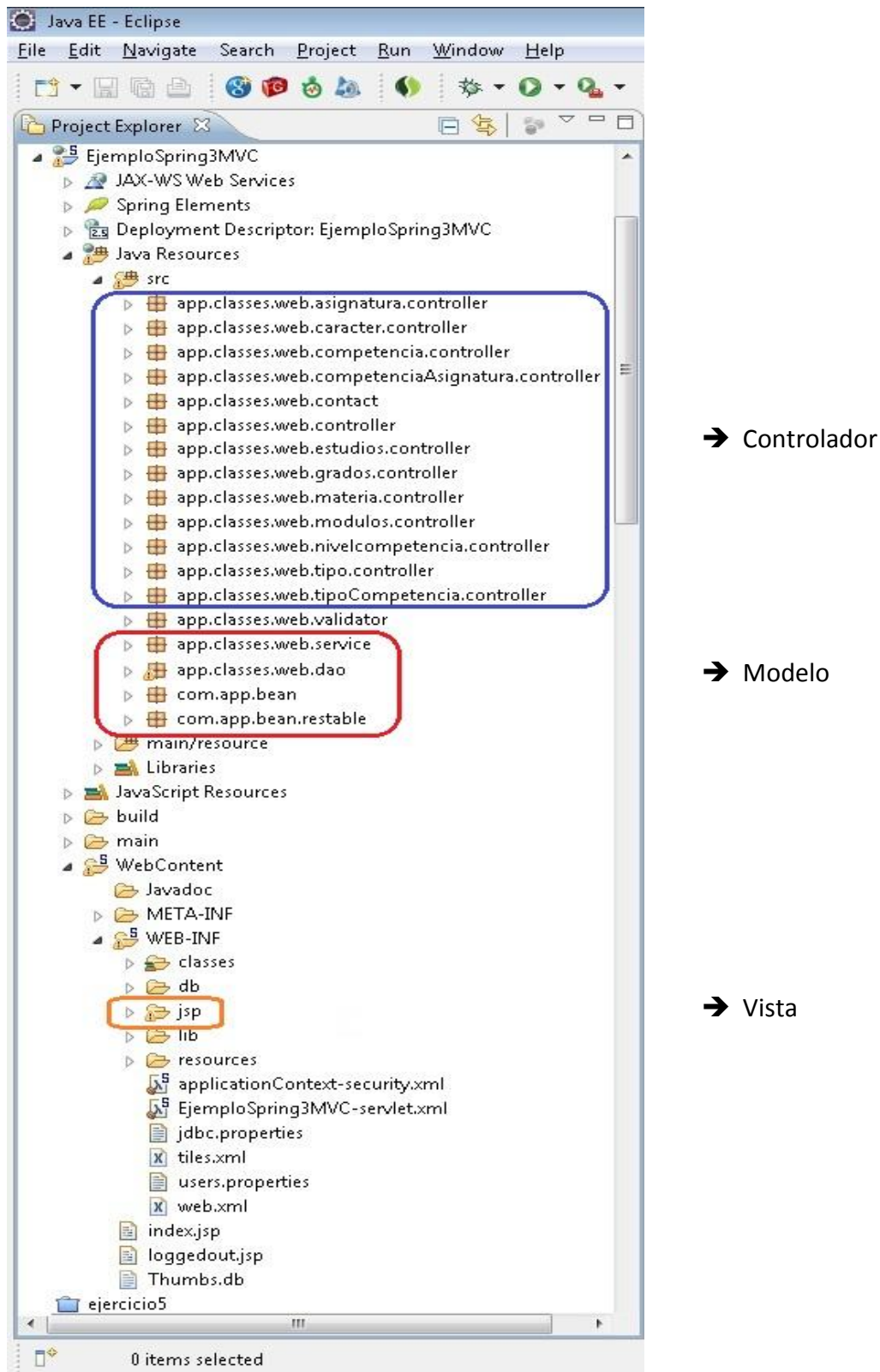


FIGURA 34: Diagrama de clases

## 5.2 ORGANIZACIÓN DE ARCHIVOS

---

Lo primero que hacemos al empezar a desarrollar la aplicación es crearnos un nuevo proyecto Java que automáticamente Eclipse lo estructura de la siguiente manera:

En la carpeta Java Resources/src, se crean las clases Java.

Esta incluye desde la declaración de los Beans (Estudio, Modulo, Materia...) hasta los controladores, la lógica de negocio y las validaciones de formulario (de los cuales hablaremos más adelante), todo estructurado en carpetas agrupadas según su funcionalidad.

La siguiente importante es WEB-INF que contiene varias carpetas:

- db: Base de datos completa exportada formato .sql.
- jsp: Como hemos mencionado anteriormente, se encuentran todos las páginas .jsp que forman la interfaz de usuario.
- lib: Contiene todas las librerías necesarias para el proyecto.
- resources/img: Contiene las imágenes que están incluidas en la interfaz de usuario.

A parte de estas carpetas tenemos archivos sueltos que son de configuración de la aplicación, los más destacables son:

- applicationContext-security.xml: Se declara la seguridad de la aplicación indicando que paginas necesitas autenticación y que permisos se dan a cada uno de los usuarios de la aplicación.
- EjemploSpring3MVC-servlet.xml: Se establece todos los parámetros de configuración del proyecto.  
Se indica la ruta de los controladores donde debe buscar nuestra aplicación cuando un usuario envía una notificación a través de la interfaz de usuario.  
Se establece la ruta de los archivos de configuración tanto de tiles.xml, log4j y configuración del acceso de base de datos (lo veremos detalladamente más adelante).
- web.xml: Se determina el archivo de inicio del proyecto y los archivos de configuración que debe manejar el sistema para su funcionamiento (los dos archivos indicados anteriormente).

En la figura 35 se observa la organización de archivos descrita.

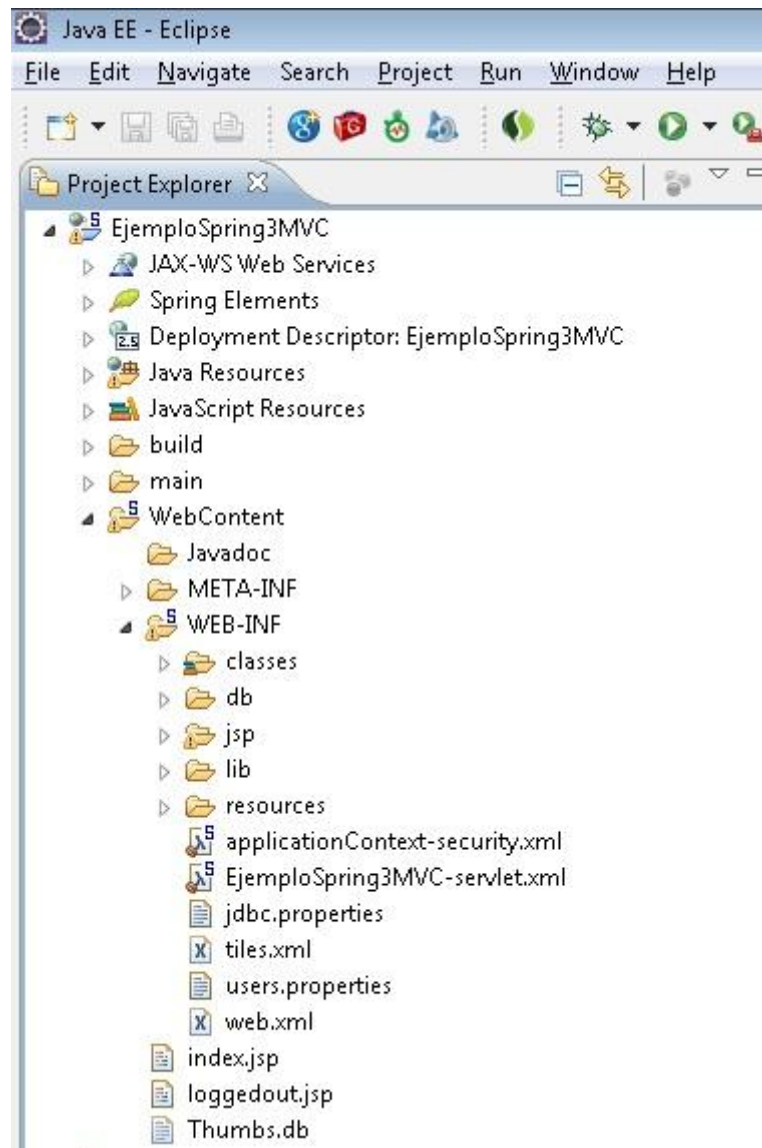


FIGURA 35: Organización de archivos

## 5.3 CONEXIÓN Y CONSULTA DE BASE DE DATOS

---

Para realizar la conexión a base de datos, se establece en el archivo de configuración de la aplicación (EjemploSpringMVC-servlet.xml) la declaración de la base de datos, estableciendo con anterioridad un archivo llamado jdbc.properties donde se establecen la url de la base de datos, el usuario y la contraseña de la misma, el driver que vamos a utilizar y el gestor de base de datos (FIGURA 36).

```
<!-- Propiedades BD -->

<bean id="userDataSource"
class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close"
p:driverClassName="${jdbc.driverClassName}"
p:url="${jdbc.databaseurl}"
p:username="${jdbc.username}"
p:password="${jdbc.password}"
/>
```

FIGURA 36: Código de acceso a base de datos

Por tanto cada vez que queramos acceder a base de datos nos creamos un nuevo objeto llamado SessionFactory, que lo suministra Hibernate para tener que abstraernos de abrir sesión y cerrarla además de las capturas de excepciones, y en su declaración le inyectamos la propiedad de la base de datos antes expuesta, como se observa en la figura 37:

```
<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
<property name="dataSource">
<ref local="userDataSource" />
</property>
...

```

FIGURA 37: Código propiedad SessionFactory

Otro aspecto importante es la forma en que se han realizado las validaciones a diferentes formularios que hay para la aplicación Web, como: controlar que los campos obligatorios estén llenos, que se cumpla la longitud mínima y máxima que se indica en algunos campos, etc.

Lo hemos ubicado en una carpeta donde se organizaran todas las validaciones de cada uno de los beans.

La validación la hemos realizado en el servidor y no en cliente, ya que siempre se debe validar desde servidor, debe ser la prioridad máxima de seguridad. La validación del lado del cliente es opcional y depende de si deseas enriquecer la interfaz, pero no aporta seguridad alguna individualmente.

Un usuario puede desactivar JavaScript del navegador o un “malintencionado” duplique el formulario y lo envíe sin validación, pudiendo alterar el correcto funcionamiento de la base de datos.

## 6. PRUEBAS

---

La finalidad de este apartado es conseguir un correcto funcionamiento de la aplicación. Y para garantizar este correcto funcionamiento, se ha sometido a diferentes tipos de pruebas para detectar todos aquellos errores que durante la codificación y programación de las diferentes páginas no han podido detectarse. Este punto es fundamental en el desarrollo de una aplicación de estas características, ya que hemos de conseguir una correcta visualización y una correcta respuesta a los posibles errores ante el usuario que la use.

Para detectar estos errores, se han llevado a cabo tanto pruebas unitarias como de integración.

### 6.1 PRUEBAS UNITARIAS

---

Las pruebas unitarias son aquellas que permiten analizar los diferentes módulos del sistema de forma individual. En el desarrollo de esta aplicación, cada vez que se acababa un módulo, lo hemos sometido a una serie de pruebas con la peor intención posible para encontrar los posibles errores. Así como pruebas siguiendo las indicaciones de la aplicación para ver que realiza lo que tiene que realizar.

Las diferentes pruebas que se han realizado dentro de esta categoría son las siguientes:

- Validación de campos vacíos: Se ha comprobado que para continuar realizando alguna acción determinada, deben estar todos los campos obligatorios rellenados. Y que un mensaje de error indicando qué campos son los que faltan por rellenar.

- Validación del formato de los datos de entrada: Se ha comprobado que aparte de estar todos los campos obligados rellenados, tal y como indica el punto anterior, también que estos campos tengan el formato correcto pedido para que no haya problemas a la hora de insertar los datos en la base de datos. Y que un mensaje de error indicando qué datos son los que tienen un formato incorrecto.

- Validación de la información en la base de datos: Se ha comprobado que al insertar, eliminar o modificar cualquier dato en la base de datos, se haga correctamente y que queríamos que se efectuara.

- Correcta visualización por parte del módulo: Se ha comprobado que el módulo en cuestión se visualice bien tanto en Mozilla Firefox como Internet Explorer.

## 6.2 PRUEBAS DE INTEGRACIÓN

---

Estas pruebas son aquellas que se realizan de forma conjunta una vez tenemos todos los módulos desarrollados.

Lo que se intenta encontrar son aquellos errores derivados de acciones como el paso de parámetros entre diferentes páginas de la aplicación, el envío y recuperación de los datos con formularios o las variables de sesión de las diferentes áreas privadas.

A continuación, se detallan las pruebas realizadas:

- Creación y eliminación de beans: Se ha comprobado que una vez el usuario ha iniciado sesión y todos los campos obligatorios son rellenados correctamente se efectúa correctamente insertando los datos introducidos en la base de datos.

- Crear relación entre beans: Se ha comprobado que una vez el usuario ha iniciado sesión y todos los campos obligatorios son rellenados correctamente se realiza la relación entre beans correctamente introduciendo los datos en la base de datos.

- Opciones de los diferentes usuarios: Se han creado diferentes perfiles de usuario, para poder comprobar que dependiendo del perfil de usuario que se trate tenga acceso a unas determinadas opciones u otras.

## 7. COMPARACION DE TECNOLOGIAS

---

La comparación la vamos a realizar con la tecnología PHP.

Tras haber elegido Java como lenguaje de desarrollo para nuestra aplicación web vamos a detallar las diferencias principales entre estas tecnologías.

Empezaremos hablando un poco de cada una de ellas.

Un poco de JAVA:

Java como lenguaje de programación, al contrario que PHP no puede ser tratado de una manera tan a la ligera y superficial, ya que si estamos hablando del desarrollo web, debemos centrarnos en un sector de todo el mundo que rodea a Java, concretamente en el de JSP, Servlets y demás.

De manera genérica, se trata de un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Todo el desarrollo del lenguaje fue controlado por el Java Community Process por parte de Sun hasta que finalmente entre Noviembre de 2006 y Mayo de 2007, estos liberaron la mayor parte de sus tecnologías bajo licencia GNU GPL, por tanto podemos considerar el lenguaje Java como se Software Libre.

Si echamos un vistazo a la actualidad, nos damos cuenta de que java se ha convertido en uno de los lenguajes más usados y más demandados por los desarrolladores.

La comunidad Java es actualmente uno de los grupos más extendidos en el universo de Internet y diversos sitios web dedicados al apoyo, información y soporte de esta tecnología. Según avancemos en su análisis iremos nombrando algunas de ellas.

Un poco de PHP:

PHP se trata de un lenguaje de programación interpretado en el servidor. Creado originalmente por Rasmus Lerdorf en 1994, en la actualidad está publicado bajo la licencia PHP.

En la actualidad es ampliamente usado en entornos de desarrollo web por su facilidad de uso, su integración perfecta con HTML y su versatilidad de uso en diferentes Sistemas Operativos.

Se han introducido importantes mejoras tales como mejoras de rendimiento, mejora en el soporte de Programación orientada a Objetos, soporte mejorado de conexiones a Base de datos, etc con la nueva versión publicada.

Con el crecimiento de PHP surgieron proyectos asociados, tales como Frameworks, IDE's (Entorno de desarrollo integrado) que le han dado al lenguaje una robustez y consistencia aún mayor y que trataremos posteriormente.

Vamos a realizar un análisis a los aspectos más importantes en la elección de una tecnología para la creación de una aplicación web:

## 7.1 ANALISIS

### 7.1.1 MODULARIZACIÓN

Al tratarse de un análisis entre dos tecnologías de desarrollo web, consideramos el término Modularización como la separación en capas definidas en un modelo MVC (Modelo Vista-Controlador). La modularidad de un sistema tiene vital importancia en el aspecto de la consistencia, robustez, mantenibilidad y demás aspectos.

Fijándonos en la estructura de PHP y Java podemos decir que existe una gran diferencia entre ambos en este ámbito.

La tecnología Java usada en cualquier portal web posee una estructura claramente diferenciada, pudiendo diferenciar con facilidad el modelo MVC con sus diferentes módulos:



FIGURA 38: Modelo MVC

En cuanto a PHP, podemos decir que carece de modelo MVC puesto que todas las capas lógicas son implementadas en un mismo archivo .php.

### 7.1.2 MANTENIBILIDAD

---

La mantenibilidad del sistema es una parte fundamental en el ciclo de vida de cualquier proyecto que estemos tratando, y está estrechamente relacionada con la tecnología que hayamos elegido en la etapa de diseño.

Para realizar el análisis de las técnicas que estamos tratando debemos remontarnos de nuevo al punto anterior para conseguir sacar una conclusión firme. Un sistema en el que exista una estructura clara de sus componentes será más fácilmente mantenible en un futuro ya que será necesario el seguimiento de una metodología ya definida y evitará un empobrecimiento de su código y por tanto de su rendimiento.

### 7.1.3 CRECIMIENTO DEL SISTEMA

---

Cuando se realiza el diseño de un proyecto y se elige una tecnología a usar, durante la etapa de desarrollo hay que pensar en que una vez implementado y puesto en funcionamiento, serán necesarias nuevas mejoras que no supongan un coste demasiado elevado ni que mucho menos produzca un empobrecimiento del sistema actual.

Por experiencia sabemos que una vez finalizado el desarrollo, puede ocurrir que las mejoras del sistema no sean implementadas por su programador original y sí por otra persona o empresa externa. El hecho de que PHP sea una técnica poco estructurada y que el desarrollador no sea pleno conocedor de la estrategia seguida en la programación original, puede dar origen a un empobrecimiento del código, repercutiendo normalmente a su rendimiento.

### 7.1.4 COSTE DE DESARROLLO

---

El coste de un proyecto PHP siempre será menor que en un proyecto Java. Mientras que la programación de un sistema PHP es mucho más directa con resultados inmediatos, el uso de Java supone el montaje de la estructura mencionada en los puntos anteriores, lo que alarga el tiempo de desarrollo y con esto, su coste.

Otro punto a tener en cuenta en la estimación de costes es el hecho de que, para la programación de un sistema java es necesaria mayor preparación y experiencia, lo que puede aumentar el coste total.

El sistema Java está más orientado a grandes proyectos ya que, como hablaremos más adelante proporciona una mayor escalabilidad que PHP.

### 7.1.5 FORMACIÓN

---

Como llevamos viendo desde el inicio del documento, está demostrado que Java es una tecnología mucho más amplia y desarrollada que PHP, lo que nos llevará a un coste de formación mayor.

### 7.1.6 INTEGRACIÓN EXTERNA

---

El mundo Java es muy amplio y variado. Esto supone una ventaja tanto en el ámbito del desarrollo como en su repercusión final. Por un lado existen diferentes Frameworks que facilitarán la tarea de los programadores, pudiendo obtener los mismos resultados (o incluso mejores) que sin ellos en un tiempo más breve. Por otro lado, existe lo que podemos llamar “módulos” ya desarrollados y de libre distribución que podemos usar en nuestro proyecto.

Si realizamos la comparación con PHP, existe una gran diferencia, ya que este último como vimos al principio es mucho menos estructurado y es mucho más complicado encontrar un módulo complejo completo integrable con facilidad.

### 7.1.7 SEGURIDAD

---

El aspecto de la seguridad siempre ha sido un punto a tener muy en cuenta en cualquier sistema informático y un portal web es especialmente vulnerable por estar expuesto a todo el público en internet.

Como sistemas de seguridad usados en proyectos Java cabe destacar los que se implementan a nivel de Servidor de aplicaciones y los que están incluidos en Frameworks externos (como por ejemplo “Spring Security”), ambos eficaces y transparentes a usuarios y programadores. En el desarrollo de un portal web con PHP debemos controlar la seguridad de acceso a nuestro sistema de una forma mucho más manual, realizando comprobaciones minuciosas de los diferentes ataques que podemos recibir.

### 7.1.8 RENDIMIENTO

---

Quizás una de las ventajas de PHP frente a Java sea en cuestión de rendimiento ya que el primero es mucho menos pesado, lo que produce una sensación al usuario de rapidez y mayor usabilidad.

### 7.1.9 ESCALABILIDAD

Un tema importante a mencionar es la escalabilidad del sistema, es decir, por la cual un sistema no empeora su rendimiento y funcionalidad ante un número creciente de usuarios.

Desde hace mucho tiempo siempre se ha dicho que la tecnología Java es mucho más escalable que PHP, demostrándose la pérdida de rendimiento de este último ante un aumento considerable de usuarios concurrentes.

### 7.1.10 CONCLUSIONES DEL ESTUDIO

Según el tipo de proyecto que estamos realizando, creemos que lo más aconsejable hubiera sido haberlo realizado con tecnología PHP ya que la curva de aprendizaje de la tecnología hubiera sido menos horizontal, lo que conlleva mayor rapidez a la hora del desarrollo de la misma y por consiguiente menos presupuesto.

Por otro lado cabe destacar el haberlo implementado con Java por el tema de mantenibilidad y crecimiento del proyecto, ya que el framework al estar enfocado con un modelo MVC sigue una regla de programación lineal por lo que si otro programador se propusiera realizar mejoras o mantenimiento no tendría que perder mucho tiempo en revisar el código al estar todo perfectamente estructurado por partes.

En la figura 39 vemos como para cada uno de los campos descritos anteriormente para nuestra aplicación que tecnología es superior:

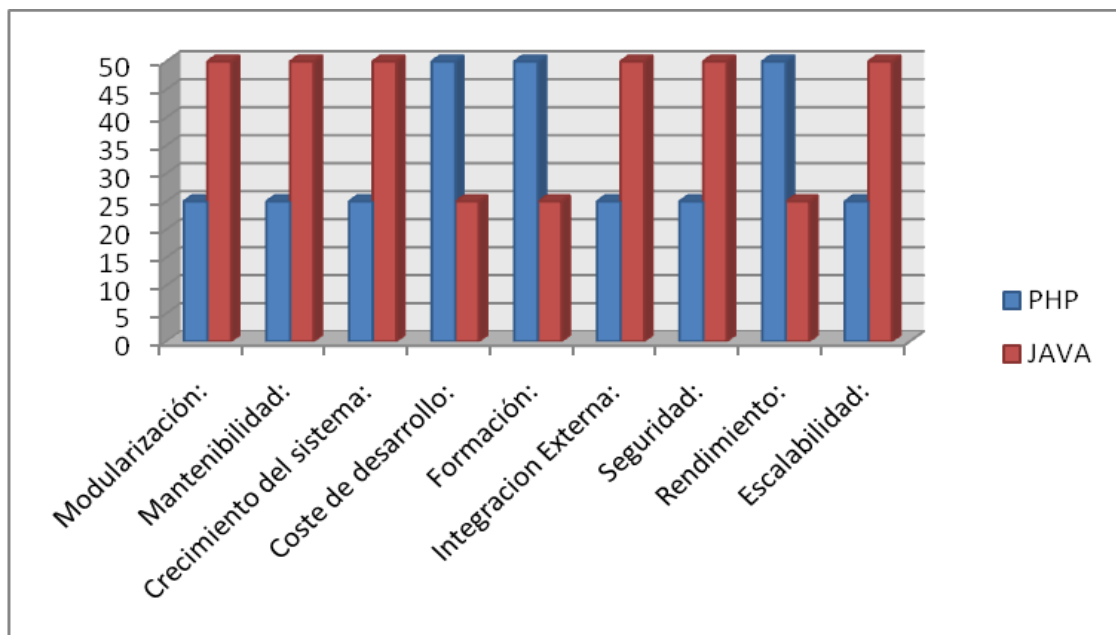


FIGURA 39: Gráfico comparación de tecnologías

En resumen, si nos enfocamos en el proyecto en sí, hubiera sido aconsejable por el tema de coste de desarrollo, formación y rendimiento haberlo implementado en PHP, pero si pensamos en una posible futura ampliación de la aplicación es más recomendable desarrollarlo con la tecnología empleada por su fácil modularización, mantenibilidad, integración externa y seguridad.

## 8. CONCLUSIONES

---

### 8.1 OBJETIVOS CONSEGUIDOS Y NO CONSEGUIDOS

---

En este punto, se analiza si se han cubierto los objetivos que se fijaron inicialmente.

Una vez terminada la aplicación, esta nos permite la creación, eliminación y asignación de competencias, consiguiendo cumplir el objetivo, a través de una intranet en donde al logarse nos permite realizar dichas tareas. Esto hace que hayamos cumplido los objetivos 1 y 3 establecidos en el punto 2.1.

La interfaz de usuario es simple e intuitiva ya que se ha simplificado al máximo los pasos necesarios para realizar las operaciones pertinentes por lo que se ha cumplido el objetivo número 2.

La base de datos cumple todos los requerimientos de seguridad, estabilidad y normalización cumpliendo el objetivo número 4.

Al completar la aplicación con todos los objetivos se ha obligado a estudiar con profundidad Java como tecnología desarrolladora de la aplicación, profundizando bastante en aspectos antes desconocidos, esto hace cumplir el objetivo número 5.

En el punto 6 de la documentación se realiza la comparativa con PHP obteniendo ventajas e inconvenientes con respecto a la tecnología utilizada y exponiendo las conclusiones obtenidas. Esto cumple el último punto de los objetivos (número 6).

### 8.2 AMPLIACIONES Y MEJORAS

---

La aplicación sirve de base para posibles ampliaciones, por lo que pueden agregarse numerosas mejoras y ampliaciones. Las mejoras que hemos visto destacables son las siguientes:

Con respecto a la visualización se podría mejorar la experiencia de usuario, introduciendo Ajax en la aplicación, esto haría que se ejecutara en el cliente (navegador) y se realizarían cambios sobre las páginas sin necesidad de recargarlas constantemente, lo que significa aumentar la interactividad, velocidad y usabilidad.

Se podría implementar mediante JavaScript las validaciones de los formularios como complemento de las validaciones desarrolladas en servidor.

Otra posible mejora sería implementar la exportación a documentos pdf, Excel o Word, además de obtener informes de la información obtenida en el campo Competencias del menú, pudiendo así obtener físicamente la información.

### 8.3 DESVIACIÓN TEMPORAL RESPECTO A LA PLANIFICACION

---

Después de haber realizado el proyecto, se ha constatado que la planificación inicial del proyecto fue un poco optimista.

De la cual podemos decir que el total de horas dedicado ha variado aunque no considerablemente ya que de 294h previstas se han dedicado 340h aproximadamente, pero lo que no se ha cumplido estrictamente ha sido la planificación de horas semanales que se había indicado (5 horas diarias de lunes a viernes), ya que no todos los días se disponía de ese tiempo. Lo que se ha hecho pues, ha sido dedicar diariamente todas aquellas horas que se ha podido, así como algunos fines de semana que en un principio no se habían tenido en cuenta.

La variación de horas dedicadas ha surgido por problemas en el desarrollo de la aplicación, porque sea realizado una planificación temporal de la elaboración de la documentación algo reducida y a causa de un retraso en la fecha establecida de entrega del proyecto fin de carrera por la posibilidad de presentarlo en otra convocatoria posterior, por lo que ese tiempo intermedio se ha utilizado para proseguir con el desarrollo de la aplicación y mejora de la documentación.

Estos cambios se pueden ver en la siguiente figura, donde se comparó la planificación inicial (diagrama de Gantt de color verde) y la planificación real (diagrama de Gantt de color azul) del proyecto. Se puede observar la diferencia de fecha de entrega y por consiguiente el aumento en horas en trabajo de desarrollo y documentación.

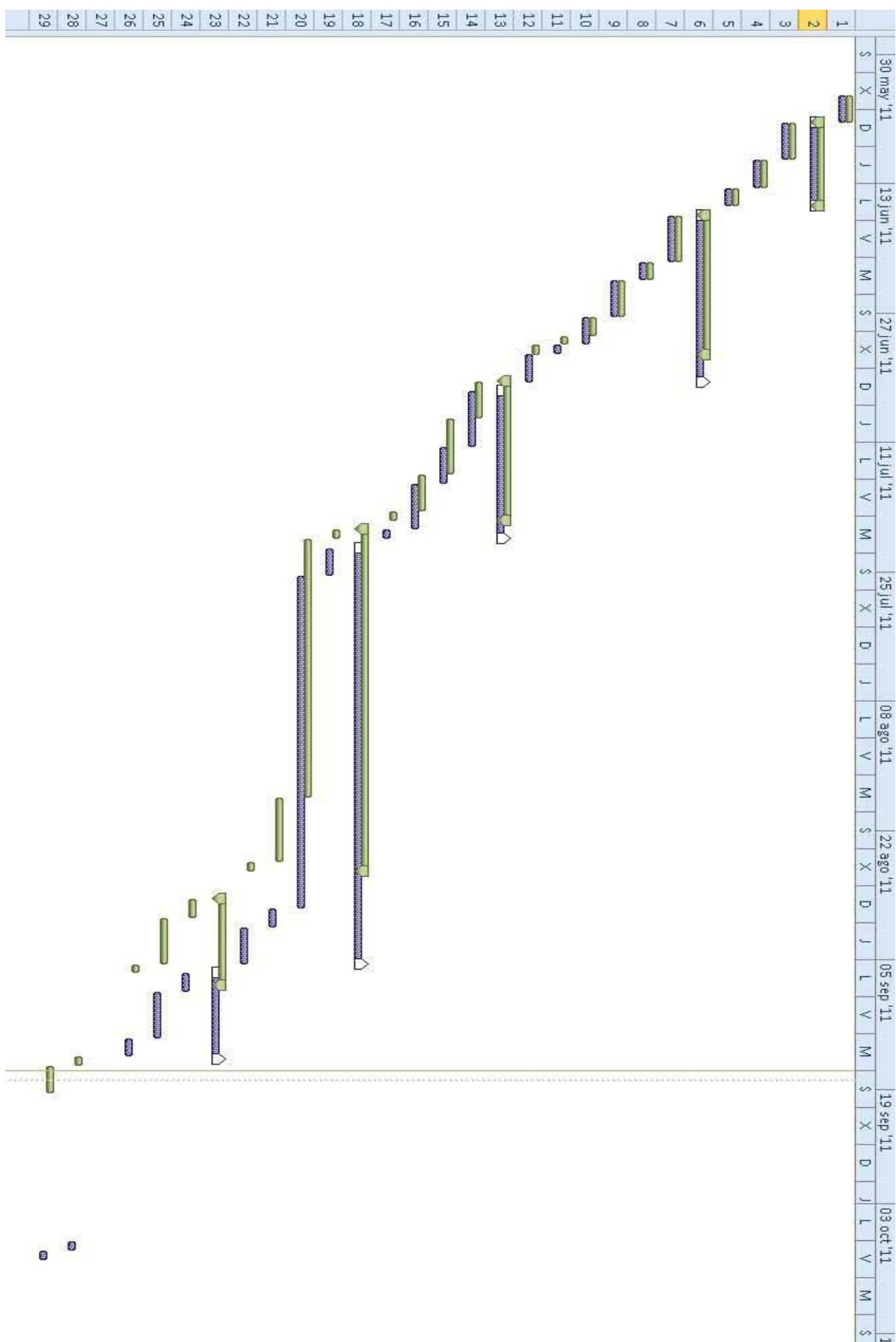


FIGURA 40: Comparación de planificaciones

## 8.4 VALORACION

---

La elaboración de este proyecto me ha supuesto grandes ventajas tanto a nivel profesional como a nivel personal, ya que su realización me ha permitido profundizar con este lenguaje que actualmente está muy implantado en el ámbito empresarial. Espero que todos los conocimientos que he aprendido sean de gran ayuda para cualquier tipo de trabajo que pueda llegar a realizar en un futuro.

Por otra parte durante la etapa de desarrollo ha habido momentos donde las cosas salían a la primera y eso animaba a seguir adelante, y momentos donde la programación se hacía un poco pesada y pasaba todo lo contrario.

Por último, la valoración general que hago de esta experiencia es muy positiva, ya que se trata de la primera vez que realizo una aplicación partiendo de cero con Java, lo que me ha llevado a una profunda investigación sobre la tecnología y su desarrollo para su correcta implementación.

## 9. REFERENCIA WEB

---

Enlaces Spring Framework:

- <http://www.springhispano.org/>
- <http://www.springsource.org/>

Enlaces Spring Security:

- <http://www.adictosaltrabajo.com/>
- <http://carloscacique.blogspot.com>

Enlaces Hibernate:

- <http://www.vaannila.com>
- <http://krams915.blogspot.com>

Enlaces Tiles:

- <http://www.yaxche-soft.com>
- <http://tiles.apache.org/>

Enlaces MySQL:

- <http://www.mysql.com/>

Enlaces Log4j:

- <http://logging.apache.org/log4j/1.2/>

Enlace Wikipedia:

- <http://es.wikipedia.org/wiki/Wikipedia:Portada>

Jose Alberto Fuentes Torrubia

Sabadell, Septiembre de 2011