



**Universitat Autònoma  
de Barcelona**

**Portal WEB para la gestión de  
tutorías para proyectos de fin  
de carrera.**

Memoria del proyecto

De Ingeniería Técnica en  
Informática de Gestión

Realizado por

*Laura Polo Cruz*

y dirigido por

*Rafael Cortés Fité*

**Escuela de ingeniería**

Sabadell, Setiembre de 2010

El que firma, **Rafael Cortés Fité**  
profesor de la Escuela de Ingeniería de la UAB,

**CERTIFICA:**

Que el trabajo al que corresponde la presente  
memoria, ha sido realizado bajo su dirección  
por **Laura Polo Cruz**

I para que conste firma el presente.  
Sabadell, Setiembre de 2010

-----

Firmado: Rafael Cortés Fité

### ***Resumen del proyecto:***

Este proyecto consiste en la realización de una aplicación web que recoge las funcionalidades de un gestor de tutorías para los proyectos final de carrera. Tienen acceso 3 tipos de usuario: alumno, profesor y gestor. El alumno puede visualizar las tutorías asignadas por el tutor, participar en un foro público entre los usuarios del gestor, compartir los archivos del proyecto con el tutor, reservar una serie de recursos disponibles para la realización del proyecto y comunicarse con el profesor mediante una mensajería interna. El profesor tiene las mismas funcionalidades que el alumno, pero se le añaden las siguientes: introducir tutorías, asignar proyectos, crear proyectos, e introducir una planificación de tareas para el desarrollo del proyecto. El gestor se encarga de la gestión de enlaces de interés publicados por los profesores, de los usuarios y los recursos.

## Contenido

Capítulo 1. INTRODUCCIÓN .....	8
2.1 Resumen de contenido.....	8
2.2 Motivaciones personales:.....	9
2.3 Estado del arte:.....	9
2.4 Objetivos y alcance previstos:.....	10
Capítulo 2. ESTUDIO DE VIABILIDAD .....	12
2.1 Introducción: .....	12
2.2 Objeto .....	13
2.2.1 Situación a resolver .....	13
2.2.2 Perfil del usuario .....	13
2.2.3 Objetivos .....	14
2.3 Descripción del sistema .....	17
2.3.1 Descripción.....	17
2.4 Recursos .....	18
2.4.1 Recursos humanos .....	18
2.4.2 Recursos hardware.....	18
2.4.3 Recursos software.....	19
2.5 Planificación del proyecto .....	20
2.5.1 Planificación inicial .....	20
2.5.2 Modelo de desarrollo.....	22
2.6 Análisis de costes .....	23
2.6.1 Costes de recursos humanos:.....	23

2.6.2	Costes de recursos hardware:.....	23
2.6.3	Costes de recursos software.....	23
2.6.4	Evaluación de riesgos: .....	26
2.7	Conclusiones .....	27
Capítulo 3. ANALISIS DEL PROYECTO.....		29
3.1	Requerimientos funcionales.....	29
3.1.1	Visión general .....	29
3.1.2	Requisitos funcionales de usuario .....	29
3.1.2.1	Usuario no registrado.....	29
3.1.2.2	Profesor o tutor.....	30
3.1.2.3	Alumno.....	30
3.1.2.4	Gestor base de datos.....	31
3.1.3	Diagrama de casos de uso.....	31
3.1.3.1	Usuario no registrado.....	31
3.1.3.2	Alumno.....	32
3.1.3.3	Profesor.....	33
3.1.3.4	Gestor de contenidos .....	34
3.1.4	Menú de navegación.....	35
3.1.4.1	Alumno.....	35
3.1.4.2	Profesor.....	36
3.1.4.3	Gestor de la base de datos .....	37
3.2	Restricciones de diseño.....	38
Capítulo 4. Diseño .....		40

4.1	Introducción .....	40
4.2	Capas de la aplicación.....	41
4.2.1	Entorno de usuario.....	41
4.2.2	Motor de aplicación .....	41
4.2.3	Capa de datos.....	42
4.3	Estructura de la base de datos .....	42
4.4	Interfaz .....	45
4.4.1	Definición de la interfaz para distintos perfiles de usuario:.....	45
4.4.1.1	Interfaz Usuario no registrado.....	45
4.4.2	Interfaz Usuario registrado.....	46
4.5	Estructura de archivos .....	49
Capítulo 5. Implementación.....		51
5.1	Capa de presentación:.....	51
5.2	Capa de negociación:.....	54
5.3	Capa de datos:.....	59
Capítulo 6. CONCLUSIONES.....		60
6.1	Desviación de la planificación .....	60
6.2	Línea de mejoras y ampliación .....	60
Capítulo 7. BIBLIOGRAFIA .....		63
7.1	Referencias electrónicas:.....	63



# Capítulo 1. INTRODUCCIÓN

---

## 2.1 Resumen de contenido

El proyecto consiste en la creación de una aplicación Web que gestione las tutorías de los proyectos final de carrera de la UAB. Este proyecto pretende dar un servicio actualmente inexistente y con ello hacer frente a los posibles problemas surgidos por una mala organización y gestión de las tutorías, así como de los recursos disponibles para el estudiante, tales como: asignar una misma hora a distintos alumnos, no tener un control sobre el proyecto que se va a tratar en la tutoría así como del estudiante correspondiente, asignar recursos o horas de laboratorio no disponibles, etc.

Por tanto, con la nueva aplicación web, conseguiremos mejorar la comunicación entre las diferentes partes implicadas mediante un entorno de colaboración más eficiente que permita un tiempo de respuesta más rápido y haga posible un acercamiento más dinámico entre tutor y alumno. Una mejor gestión y uso de los recursos mediante una serie de herramientas y servicios proporcionados por el aplicativo, caracterizado por una interfaz amigable, sencilla y clara.

El proyecto ofrecerá un servicio deslocalizado con el que se podrá acceder desde cualquier sitio y en cualquier momento. Contendrá un sistema de gestión documental compartido entre profesores y alumnos para agilizar el intercambio de información entre ambos. Se proporcionará el servicio de un foro para incluir un lugar de colaboración entre todos los alumnos y profesores y se presentará una serie de links de interés, artículos y publicaciones que puedan resultar de ayuda a



las tareas del alumno. Y sobre todo, se gestionará de forma segura los diferentes perfiles, contenidos y permisos en función de los usuarios.

Para crear la aplicación se ha utilizado XAMPP que integra la distribución apache y contiene la tecnología php y base de datos mysql, controlada por la interfaz phpmyadmin. Se ha escogido este tipo de herramientas por su fácil acceso y disponibilidad así como por su sencillez de instalación.

## **2.2 Motivaciones personales:**

Las motivaciones personales a la hora de realizar el proyecto residen en que se trata de un aplicativo que una vez finalizado puede ser utilizado por la universidad. Tiene una aplicación real y resulta más interesante en el momento de buscar los requerimientos y requisitos que la aplicación ha de cumplir.

## **2.3 Estado del arte:**

No existen precedentes de aplicaciones específicas que pretendan trabajar los aspectos que el proyecto intenta proveer, pues se tratan de unas funcionalidades determinadas para la universidad. Aun así, existen gestores de contenido que permiten una estructura de soporte para la administración de contenidos generales que permiten crear y mantener una Web con una cierta facilidad.

Tenemos por ejemplo gestores como SharePoint de Microsoft Office, Joomla e incluso software libre como Mambo. Sin embargo, aun disponiendo de este tipo de software, tenemos los siguientes inconvenientes: un coste de compra para aquellos que no son "open source", así como el coste de aprendizaje del funcionamiento del

gestor por parte del usuario, una implantación compleja y la siguiente adaptación del gestor a las necesidades propias, una vinculación al desarrollador, y por tanto, inflexibilidad ante cambios o ampliaciones imprescindibles, y limitaciones de sistema operativo y plataforma de base de datos que muchas veces imposibilitan su implantación. Como resultado de esta situación, obtenemos que aunque se dispone de aplicativos que pueden ayudar, para nuestro caso, pueden resultar no necesariamente útiles y eficientes.

## **2.4 Objetivos y alcance previstos:**

Los objetivos del proyecto van enfocados a facilitar y agilizar la gestión de las tutorías tanto a alumnos como a profesores, así como a gestionar los recursos proporcionados por la universidad para la creación de proyectos. Una de las funcionalidades principales será el control de acceso de diferentes usuarios, cada uno con un perfil: profesor, alumno y administrador. Cada uno con un aplicativo adaptado a sus características. Se pretende alcanzar, así mismo, la implementación del acceso a los servicios del portal mediante una interfaz amigable para simplificar la adaptación y aprendizaje de la aplicación. Para que esta interface sea realmente útil se debe dotar a cada usuario de un fácil acceso a las funcionalidades asociadas a su perfil y a sus datos completamente disponibles y actualizados. También se plantea como una clara mejora la posibilidad de acceder a un foro donde los usuarios puedan poner en común sus inquietudes y compartir conocimiento. Se quiere aprovechar la velocidad y flexibilidad de internet para agilizar el intercambio de información entre alumnos y profesores, potenciando la capacidad de evaluación continuada sobre los contenidos compartidos.

Una forma de ampliar y completar el proyecto podría ser mediante entrevistas o consulta a profesores encargados de llevar proyectos que puedan aportar información sobre cómo les gustaría que fuese la aplicación y que acciones o aspectos debería de tratar para mejorar su utilidad.

# Capítulo 2. ESTUDIO DE VIABILIDAD

---

## 2.1 Introducción:

El proyecto consiste en realizar una aplicación Web que permita tanto a alumnos como a profesorado gestionar las tutorías de los proyectos finales de carrera de la UAB, así como de los recursos dispuestos a tales efectos. Pretende dar un servicio inexistente hasta la actualidad y evitar los problemas típicos surgidos por carencias o inadecuada organización.

Esta aplicación facilitará la comunicación entre las dos partes implicadas mediante una interfaz sencilla y clara, accesible desde cualquier ordenador con acceso a Internet.

Dispondrá de una serie de recursos que tendrán como objetivo simplificar el intercambio de documentos, el acceso a links de interés para alumnos y tutores, y un foro para que tanto estudiantes como profesores se relacionen y comuniquen entre sí de una manera rápida y eficiente.

Al ser accesible desde Internet, será necesario tener un sistema de identificación de usuarios, que controle la entrada al aplicativo con seguridad. De esta manera se pretende evitar que usuarios mal intencionados o no autorizados tengan acceso a la aplicación.

Actualmente no existe ninguna aplicación que tenga como objetivo abarcar todas estas características específicas para la UAB. Sin embargo, sí que existen gestores

de contenido generales que son adaptables a las necesidades del usuario. Aún así, resulta inconveniente por una serie de desventajas que suponen un coste extra importante.

## **2.2 Objeto**

### ***2.2.1 Situación a resolver***

Actualmente la universidad no dispone de ningún método que facilite la gestión de las tutorías y recursos disponibles para la realización del proyecto final de carrera. Con este proyecto lo que intentamos es dar solución a esta situación de forma eficiente.

Con el portal Web, accesible desde Internet, se pretende que los usuarios puedan organizar las horas de tutorías, reservar los recursos que se necesiten, conocer las horas libres de laboratorio, y tener siempre a mano una serie de artículos, links y diferente contenido específico, que sean de interés para los proyectos.

La principal intención es mantener informados y actualizados a los diferentes usuarios de la aplicación con respecto a su proyecto final de carrera de forma rápida y fácil.

### ***2.2.2 Perfil del usuario***

La aplicación está orientada a usuarios de la UAB. Concretamente a tutores y alumnos que están en pleno proceso de realización de proyecto final de carrera. Para el uso de la herramienta, no se necesita un conocimiento extenso de informática y de las tecnologías utilizadas pues se pretende facilitar su uso mediante una interfaz sencilla y amigable.

### 2.2.3 *Objetivos*

Los objetivos básicos del proyecto son los que proporcionan una aplicación Web, junto con las características específicas que tratan este proyecto. Por tanto, las funcionalidades que se van a introducir son las siguientes:

1. Proporcionar una interfaz simple y asequible a todo tipo de usuarios. De manera que la adaptación y manejo de la aplicación no comporte ninguna complicación extra para el usuario.
2. Llevar a cabo un control de usuarios y acceso a la Web. Para cada usuario se habilitará un menú distinto, con funcionalidades personalizadas, según su perfil. Para poder acceder al aplicativo será necesario efectuar un registro inicial por parte del usuario. Tras este trámite inicial el usuario podrá acceder a los servicios disponibles siempre que lo desee mediante una posterior identificación.
3. Acceso a un foro tanto para profesores como para alumnos, con el principal objetivo de solucionar problemas y exponer cuestiones que tengan interés en los proyectos final de carrera.
4. Todo alumno dispondrá de un gestor documental, compartido con su tutor, para el intercambio de archivos del proyecto. De esta manera se persigue agilizar la comunicación entre las dos partes y asegurar una correcta gestión del versionado de trabajos.
5. Consulta de datos a una base de datos gestionada por el administrador. Por tanto, cada perfil tendrá a su disposición una serie de consultas:
  - a. *Profesor:*
    - i. Consulta de horas de tutorías asignadas para una fecha, proyecto o estudiante determinado.

- ii. Consulta de proyectos asignados y descripción del mismo.
- iii. Consulta de estudiantes del cual es tutor.
- iv. Consulta de horas de laboratorio y recursos disponibles.

b. *Alumno:*

- i. Consulta de tutorías.
- ii. Consulta de horario de laboratorio y recursos disponibles por fecha, hora y laboratorio.
- iii. Consulta de tareas asignadas durante el desarrollo del aplicativo.

6. Actualización e introducción de datos vía Web:

a. *Administrador*

- i. Gestionar los recursos de la plataforma, constituidos en una base de datos de configuración, para establecer la disponibilidad de estos.
  - 1. Recursos a gestionar:
  - 2. Profesores
  - 3. Laboratorio
  - 4. Sistemas
  - 5. Material

b. *Profesor:*

- i. Introducción de nuevos proyectos y alumnos.
- ii. Introducción y modificación de horas de tutorías.
- iii. Introducción y modificación de comentarios en tutorías.

c. *Alumno:*

- i. Validación del usuario en el entorno de proyectos.
- ii. Petición de tutorías.
- iii. Petición de laboratorio y recursos.

iv. Introducción y modificación de comentarios en tutorías.

7. Realización y visualización de informes de seguimiento, tales como horas total por proyecto, calendario de tutorías, tareas establecidas por proyecto, cumplimiento de las entregas, asistencia, etc.
8. Aviso vía correo electrónico de modificaciones realizadas en horas de tutoría, horas de laboratorio libres u ocupadas, recursos disponibles y tareas establecidas.
9. Cálculo de estadísticas.

Número objetivo	Nombre	Tipo	Usuario	Realizado
1	Interfaz simple	Crítico	General	Sí
2	Control de usuario	Crítico	General	Sí
3	Acceso a un foro	Prioritario	General	Sí
4	Gestor documental	Secundario	General	Sí
5	Consulta de tutorías	Crítico	Profesor	Sí
5a <sup>ii</sup>	Consulta de proyectos	Crítico	Profesor	Sí
5a <sup>iv</sup>	Consulta disponibilidad de recursos	Prioritario	Profesor	Sí
5b <sup>i</sup>	Consulta de tutorías	Crítico	Alumno	Sí
5b <sup>ii</sup>	Consulta disponibilidad de recursos	Prioritario	Alumno	Sí
5b <sup>iii</sup>	Consulta planificación	Prioritario	Alumno	Sí
6a	Gestionar recursos	Crítico	Gestor	Sí
6b <sup>i</sup>	Introducción nuevos proyectos	Crítico	Profesor	Sí
6b <sup>ii</sup>	Introducción y modificación de tutorías	Crítico	Profesor	Sí
6b <sup>iii</sup>	Introducción y modificación comentarios en tutorías	Secundario	Profesor	No
6c <sup>ii</sup>	Petición de tutorías	Crítico	Alumno	Sí
6c <sup>iii</sup>	Petición de recursos	Prioritario	Alumno	Sí
6c <sup>iv</sup>	Introducción y modificación de comentarios en tutorías	Secundario	Alumno	No
7	Visualización de informes varios	Secundario	General	No
8	Aviso por correo electrónico	Secundario	General	No



9	Cálculo de estadísticas	Secundario	General	No
---	-------------------------	------------	---------	----

## 2.3 Descripción del sistema

### 2.3.1 Descripción

Este proyecto constituye una solución inexistente actualmente en la organización y gestión de las tutorías de los proyectos final de carrera. Consistirá en una aplicación Web que ofrecerá una serie de servicios a los usuarios implicados en los proyectos de la UAB: alumnos y profesores. Pretende crear un portal desde el cual, el usuario podrá consultar aquellos datos que le interese saber y realizar peticiones sobre recursos disponibles.

Concretamente ofrecerá un servicio de gestión de tutorías y recursos ofertados para los proyectos final de carrera. También tendrán acceso a una serie de links y artículos relacionados con los proyectos que puedan resultar de ayuda para su realización. Se dispondrá de un foro, solo para usuarios registrados, desde el cual podrán comunicarse las diferentes partes, de manera que se les permitirá resolver dudas y formular cuestiones, así como aconsejar o informar sobre distintos asuntos.

Existirá un gestor documental con el que alumno y tutor podrán compartir archivos, una forma de agilizar los intercambios y actualizaciones de archivos.

Para poder acceder a la aplicación, inicialmente será necesario un registro, a partir del cual se definirá un *login* y un *password*. Estos datos serán necesarios para poder entrar definitivamente al menú que interesa a cada uno de los usuarios.

Una vez un usuario se ha registrado en el entorno de trabajo del sistema, solo tendrá acceso completo a los servicios del aplicativo si sus credenciales pueden ser

contrastadas con éxito sobre una lista, previamente actualizada por los servicios administrativos del centro, de alumnos matriculados de proyecto.

Para esta aplicación se declararán 3 tipos de perfiles: profesor o tutor, alumno, y administrador. Cada uno de ellos tendrá un rol distinto, de manera que las funcionalidades serán diferentes según el tipo de usuario que se identifique.

## **2.4 Recursos**

Para la realización de este proyecto será necesario el uso de determinados recursos, los cuales se pueden dividir según su naturaleza en los siguientes: físicos (hardware), lógicos (software) y humanos.

### ***2.4.1 Recursos humanos***

Como recurso humano básico se precisa de la figura del desarrollador de la aplicación. Este debe conocer los diferentes lenguajes de programación, así como de los aplicativos utilizados para la creación del proyecto. En este caso, el desarrollador es el propio alumno, y será el encargado de realizar todas las tareas específicas para la creación del proyecto: programador, analista, testeo, y corrector de errores.

### ***2.4.2 Recursos hardware***

Para la realización del proyecto será necesario un PC que soporte los requisitos mínimos del sistema operativo. Se han escogido unas especificaciones estándares para un ordenador de gama baja:

- ✓ Procesador Pentium 3 o superior
- ✓ Memoria RAM: 256 MB

- ✓ Disco Duro: 20 GB
- ✓ Grabador DVD-ROM
- ✓ Monitor SVGA
- ✓ Impresora Inyección a tinta
- ✓ Accesorios: ratón, teclado.

### ***2.4.3 Recursos software***

En este subgrupo se incluyen todas las aplicaciones que se van a usar a lo largo del proceso de desarrollo del proyecto, así como los lenguajes necesarios para tal efecto.

- ✓ Sistema operativo Windows XP Home Edition o superior.
- ✓ Navegador web: Mozilla Firefox
- ✓ Cliente windows FTP: WinSCP
- ✓ Microsoft Office y Microsoft Project
- ✓ Editor de texto Notepad++
- ✓ Editor web orientado a php: Zend Studio 7.0.2
- ✓ Complemento Firebug
- ✓ Xampp 1.7.3 :
  - Servidor Apache 2.2.14
  - Base de datos Mysql 5.1.41
  - Gestor base de datos phpMyAdmin 3.2.4
  - PHP 5.3.1
- ✓ Html
- ✓ Javascript
- ✓ CSS

## 2.5 Planificación del proyecto

### 2.5.1 Modelo de desarrollo

El modelo de desarrollo escogido para la planificación del proyecto es el secuencial.

Para facilitar el desarrollo y seguir siempre una estructura lineal se ha creído conveniente seguir un modelo en que los módulos sean dependientes del anterior.

De esta manera se sigue un orden correcto en la creación del proyecto, en que se evitan posibles problemas surgidos por la realización de una tarea sin haber acabado la que la precede.

### 2.5.2 Planificación inicial

Descripción de las etapas que se van a seguir en el transcurso del desarrollo del proyecto:

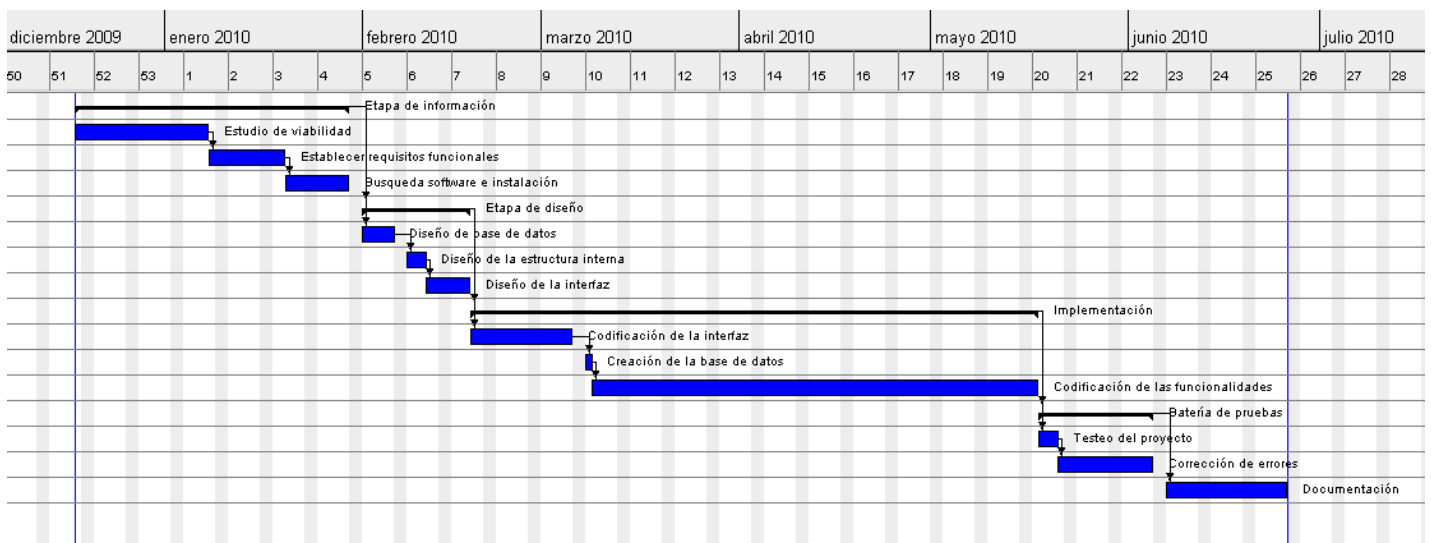
- ✓ Etapa de *información*: durante esta etapa se realizan las tareas propias de búsqueda de información: entrevistas a futuros usuarios de la aplicación para aumentar la lista de objetivos y búsqueda de proyectos parecidos para conocer similitudes y divergencias en cuanto a funcionalidades y servicios. A partir de los datos obtenidos, se establecen los requisitos funcionales del programa y se obtienen conclusiones importantes que afectan a la realización del proyecto.
- ✓ Etapa de *diseño*: en esta etapa se define el diseño visual y estructural del proyecto, así como de las correspondientes revisiones y correcciones para la

obtención de mejoras. Aquí toma importancia la interfaz del aplicativo y la disposición de la base de datos.

- ✓ *Implementación*: codificación de los requisitos y las funcionalidades exigidas en el proyecto, así como la implementación del diseño realizado en la etapa anterior.
- ✓ *Testeo* del proyecto: revisión del proyecto mediante una secuencia de pruebas pensadas de forma muy específica para el descubrimiento de posibles errores y conflictos. Una vez hallados se efectúan las correcciones pertinentes, o, únicamente se modifica el software para la introducción de mejoras.
- ✓ *Documentación*: plasmar en papel todo el proceso elaborado para la creación del proyecto, así como la definición y explicación de las distintas etapas en el desarrollo del aplicativo.

Tareas	Duración (horas)
<b>Información</b>	<b>29</b>
<b>Análisis de requerimientos</b>	<b>15</b>
<b>Búsqueda de software</b>	<b>6</b>
<b>Estudio de viabilidad</b>	<b>18</b>
<b>Diseño visual</b>	<b>10</b>
<b>Interfaz principal del aplicativo</b>	<b>10</b>
<b>Diseño estructura de datos</b>	<b>15</b>
<b>Diseño de la base de datos</b>	<b>10</b>
<b>Diseño de módulos</b>	<b>5</b>

<b>Implementación</b>	<b>130</b>
<b>Codificación de funcionalidades</b>	<b>96</b>
<b>Codificación de la interfaz</b>	<b>23</b>
<b>Creación de la base de datos</b>	<b>2</b>
<b>Batería de pruebas</b>	<b>34</b>
<b>Testeo de proyecto</b>	<b>9</b>
<b>Corrección de errores</b>	<b>25</b>
<b>Documentación</b>	<b>37</b>



El diagrama de Gantt se ha realizado teniendo en cuenta que se realizan 2 horas diarias aproximadamente para el desarrollo del proyecto. Observando el gráfico, vemos como el total de días es de 195 días.

## 2.6 Análisis de costes

### 2.6.1 Costes de recursos humanos:

Como coste se valora el tiempo invertido en la formación del desarrollador para poder adquirir los conocimientos necesarios que permiten la realización de proyectos como el que estamos tratando. Así como las horas empleadas en búsqueda de información, software y los desplazamientos requeridos a tal efecto. Por todo ello, se tendrá en cuenta en los siguientes cálculos de coste, un precio hipotético por hora del analista programador/desarrollador del aplicativo. Este precio asciende aproximadamente a 25€/hora.

### 2.6.2 Costes de recursos hardware:

Costes de los elementos físicos necesarios para llevar a cabo el proyecto.

Descripción	Coste (€)
PC	600€
Impresora	40€
Papel / Encuadernación	20€

### 2.6.3 Costes de recursos software

Costes relacionados con la etapa de información:

Recogida de información necesaria para iniciar el proyecto.

Descripción	Horas dedicadas	Coste Total (€)
Análisis requerimientos	15	375€
Búsqueda de software	6	150€
Estudio de viabilidad	18	450€

Costes relacionados con el diseño:

Estudio del diseño visual que se utilizará para conseguir una interfaz amigable, así como el diseño de funcionamiento interno de la aplicación.

Descripción	Horas dedicadas	Coste Total (€)
Diseño gráfico	10	250€
Diseño base de datos	5	225€

Costes relacionados con la programación

Horas dedicadas a la implementación de la aplicación en el lenguaje escogido, junto con las pruebas de validación de la aplicación y corrección de errores.

Descripción	Horas dedicadas	Coste Total(€)
Programación	121	3025€
Test de pruebas	9	225€
Corrección de errores	25	625€



### Costes relacionados con la documentación

Elaboración de la documentación del proyecto.

Descripción	Horas dedicadas	Coste Total (€)
Borradores	5	125€
Revisiones	9	225€
Memoria	23	575€

### Costes relacionados con el software

Coste de los programas que se han necesitado para la creación del proyecto.

Descripción	Coste Total (€)
Microsoft Windows XP Internet Explorer	150 €
Microsoft Office 2007	80 €
Xampp	0€
Microsoft Project 2007	0 €
Zend Studio	190 €
Notepad++	0 €
Mozilla Firefox y firebug	0 €

Hay que incluir los gastos de la conexión ADSL: 30€/mes aproximadamente.

Costes totales del proyecto:

Descripción	Coste (€)
Análisis	975€
Diseño	475€
Implementación	3875€
Documentación	925€
Total Desarrollo	6250€
Total HW/SW	420€
<b>Total Proyecto</b>	<b>6670€</b>

#### ***2.6.4 Evaluación de riesgos:***

Hay una serie de riesgos que acompañan a la creación de un proyecto que pueden afectar al resultado final, a los costes totales, y a la cantidad de horas que se invierten en su desarrollo.

Por la parte del programador, pueden venir dados por una falta de conocimiento en las herramientas utilizadas para la creación del proyecto, que pueden dificultar su realización y consecuentemente obtener resultados menos idóneos. Del mismo modo, las horas que implican la elaboración del proyecto pueden aumentar de forma considerable. Por lo tanto corremos el riesgo de sufrir un aumento de tiempo poco rentabilizable, y sobre todo, cometer errores importantes que puedan conllevar un cambio significativo en la realización del proyecto y un retroceso en su creación.

Por otro lado, tenemos factores externos que pueden complicar el proyecto de forma considerable. En nuestro caso, al tratarse de una aplicación web, la seguridad de los datos en red es un aspecto al que hay que otorgarle gran consideración. Una mala gestión de estos junto con un control erróneo, pueden provocar intrusiones de usuarios externos a la aplicación. Los usuarios aportan información personal que debe de estar vigilada en todo momento para ofrecer y garantizar siempre la perfecta privacidad de estos.

De igual forma, se debe tener un sistema seguro en cuanto a perfiles de usuarios, pues se ha de evitar que distintos roles se mezclen y se tenga acceso a una serie de operaciones y herramientas no correspondientes al tipo de usuario logado.

Un aspecto a tener en cuenta en proyectos web son las incompatibilidades entre navegadores. Para este proyecto se decantará por el uso del Mozilla Firefox, dado que actualmente es el más utilizado por los usuarios de internet.

## **2.7 Conclusiones**

Este proyecto está orientado para el uso universitario. Como ya se comentó en puntos anteriores, puede suponer una gran ventaja y una evolución en cuanto a la gestión de las tutorías y seguimiento de los proyectos final de carrera. Se pueden obtener beneficios reduciendo el número de horas dedicadas a estas tareas, a disminuir el número de errores producidos por una carencia de control en la gestión de los proyectos, así como en facilitar el acceso a unos recursos de forma rápida y fácil.

Su creación no supone un coste excesivo, pues se intenta utilizar, dentro de lo posible, software opensource y recursos disponibles de manera gratuita.

No hay competencia directa de productos similares que puedan hacer de su elaboración un proyecto inviable en cuanto a precio y servicios, en consecuencia, hacen del proyecto un aplicativo interesante.

Por todo ello, aun cuando sea necesario un proceso de aprendizaje e implantación por parte del usuario final, resulta un proyecto útil, eficiente, económico y consecuentemente viable.

# Capítulo 3. ANALISIS DEL PROYECTO

---

## 3.1 Requerimientos funcionales

### 3.1.1 *Visión general*

En este apartado analizaremos todos los requisitos del proyecto para cada uno de los distintos usuarios que pueden acceder a la aplicación. Como ya se dijo en apartados anteriores, cada usuario podrá realizar una serie de operaciones adecuadas a su perfil. Y por tanto, llegados a este punto, detallaremos cuales son, estos requisitos funcionales.

Recordemos que este aplicativo puede ser usado por los siguientes usuarios:

- Alumno
- Profesor
- Gestor de la base de datos

### 3.1.2 *Requisitos funcionales de usuario*

#### 3.1.2.1 Usuario no registrado

Se considerará usuario no registrado toda persona que acceda al entorno de gestión de tutorías sin un identificador valido. Los requisitos funcionales para este perfil de usuario son:

- Registrarse
- Visualización de contenido estático

### **3.1.2.2 Profesor o tutor**

Se considerará usuario registrado con acreditación de tutor a toda persona que acceda al entorno de gestión de tutorías y se le hayan otorgado las credenciales oportunas. Los requisitos funcionales para este perfil de usuario son:

- Identificarse
- Visualización foro
- Realizar entrada foro
- Añadir tareas proyecto
- Visualización proyectos adquiridos e información relacionada
- Añadir proyecto
- Modificar datos proyecto
- Acceso archivos compartidos con alumno
- Añadir/modificar tutoría proyecto
- Eliminar tutoría proyecto
- Enviar comentario alumno
- Acceso contenido con enlaces de interés

### **3.1.2.3 Alumno**

Se considerará usuario registrado con acreditación de tutor a toda persona que acceda al entorno de gestión de tutorías y se le hayan otorgado las credenciales oportunas. Los requisitos funcionales para este perfil de usuario son:

- Identificarse
- Visualización recursos disponibles
- Reserva recursos disponibles
- Visualización información proyecto
- Visualización tareas asignadas al proyecto
- Visualización de tutorías

- Enviar comentario tutor
- Visualización foro
- Participación al foro
- Acceso archivos compartidos con tutor
- Visualización enlaces de interés

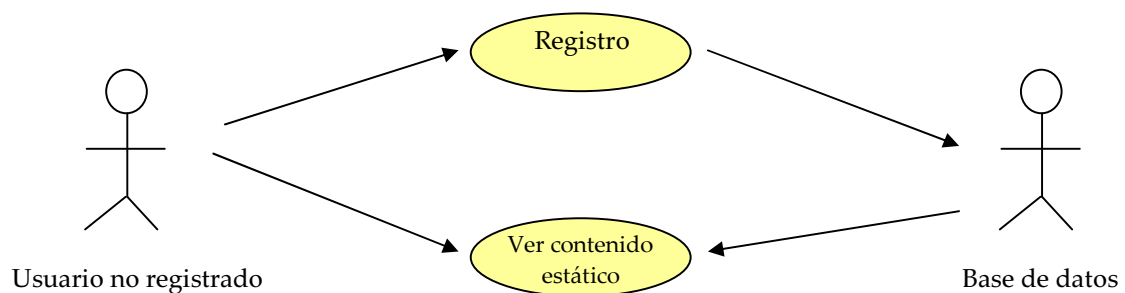
### 3.1.2.4 Gestor base de datos

Los requisitos funcionales para este perfil de usuario son:

- Identificarse
- Introducir recursos
- Eliminar recursos
- Modificar recursos
- Listar usuarios
- Eliminar usuarios
- Introducir nuevo usuario
- Modificar enlaces de interés
- Eliminar enlaces de interés
- Introducir enlace de interés

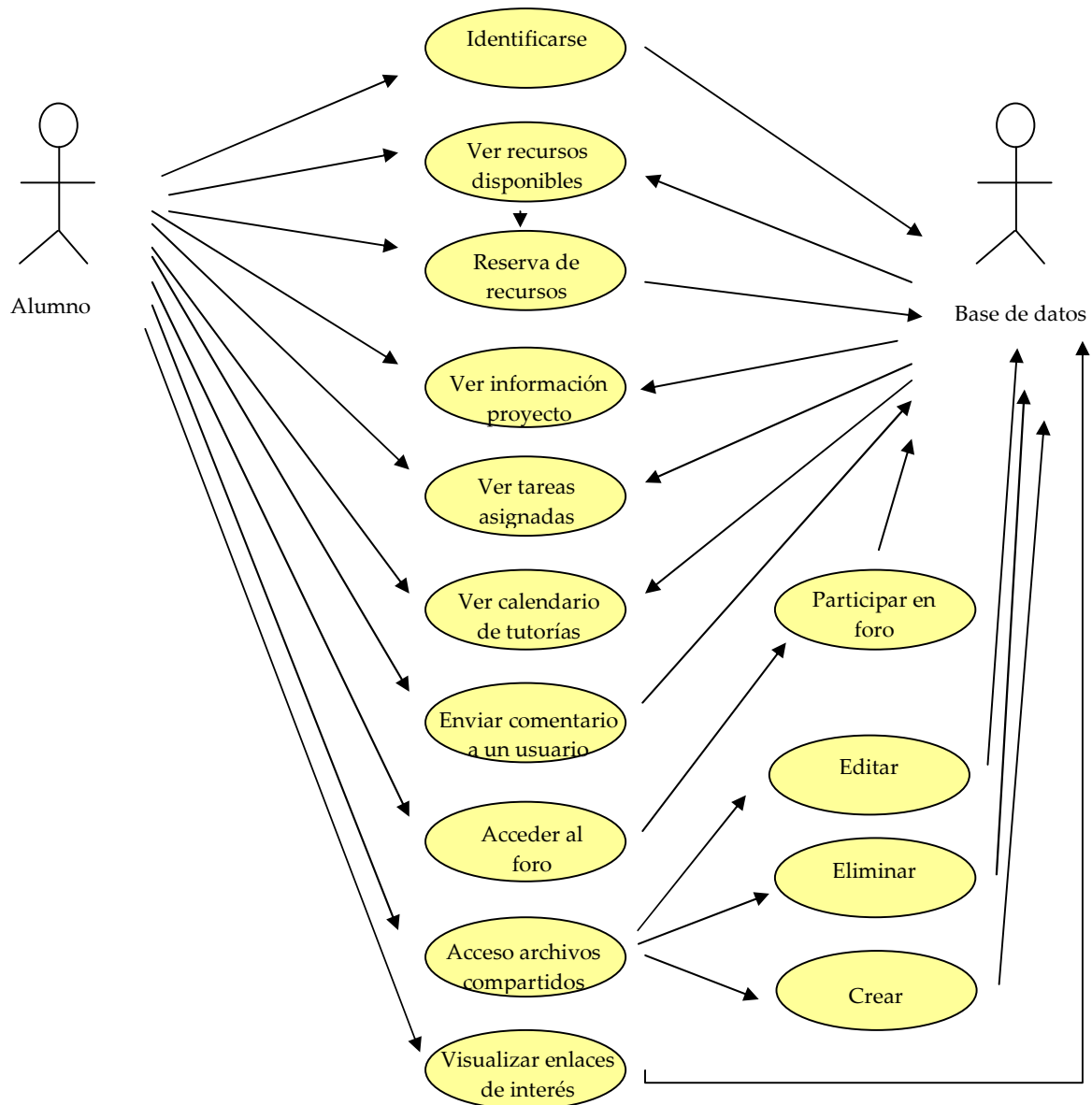
### *3.1.3 Diagrama de casos de uso*

#### 3.1.3.1 Usuario no registrado



### 3.1.3.2 Alumno

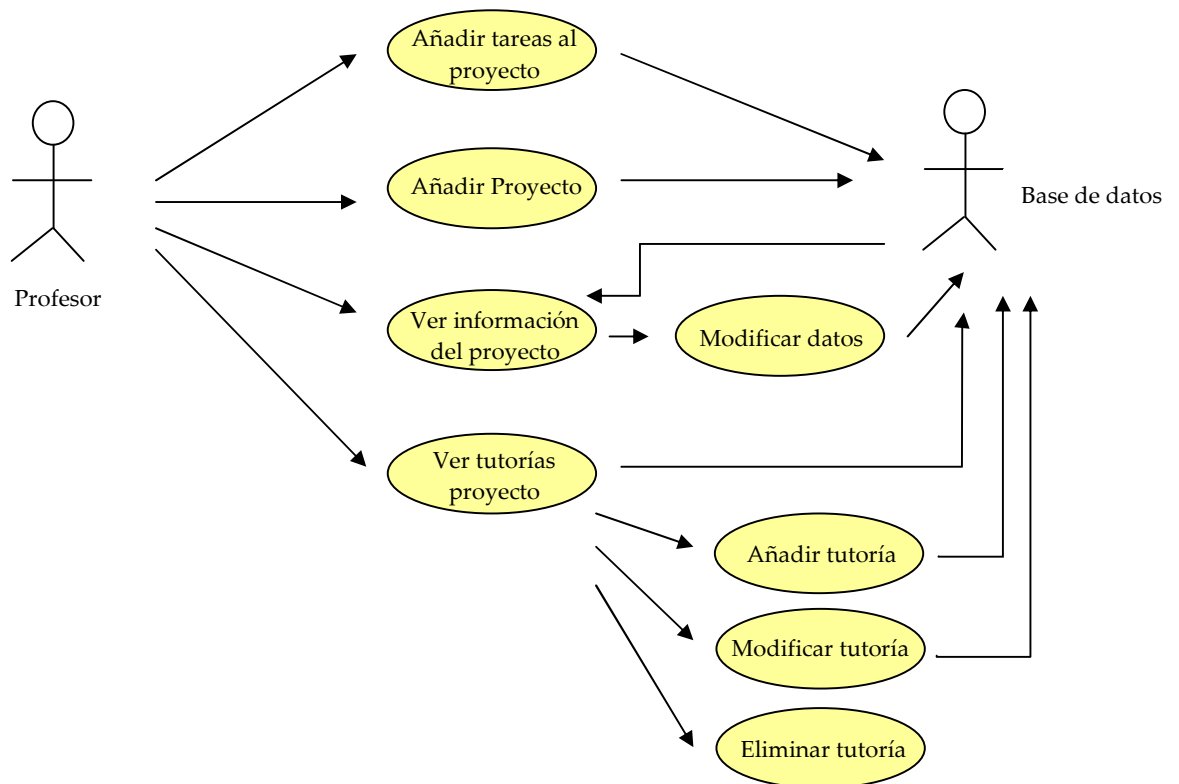
El perfil de usuario “Alumno” heredará las capacidades básicas del usuario no registrado y dispondrá, adicionalmente, de las siguientes posibilidades de interacción.



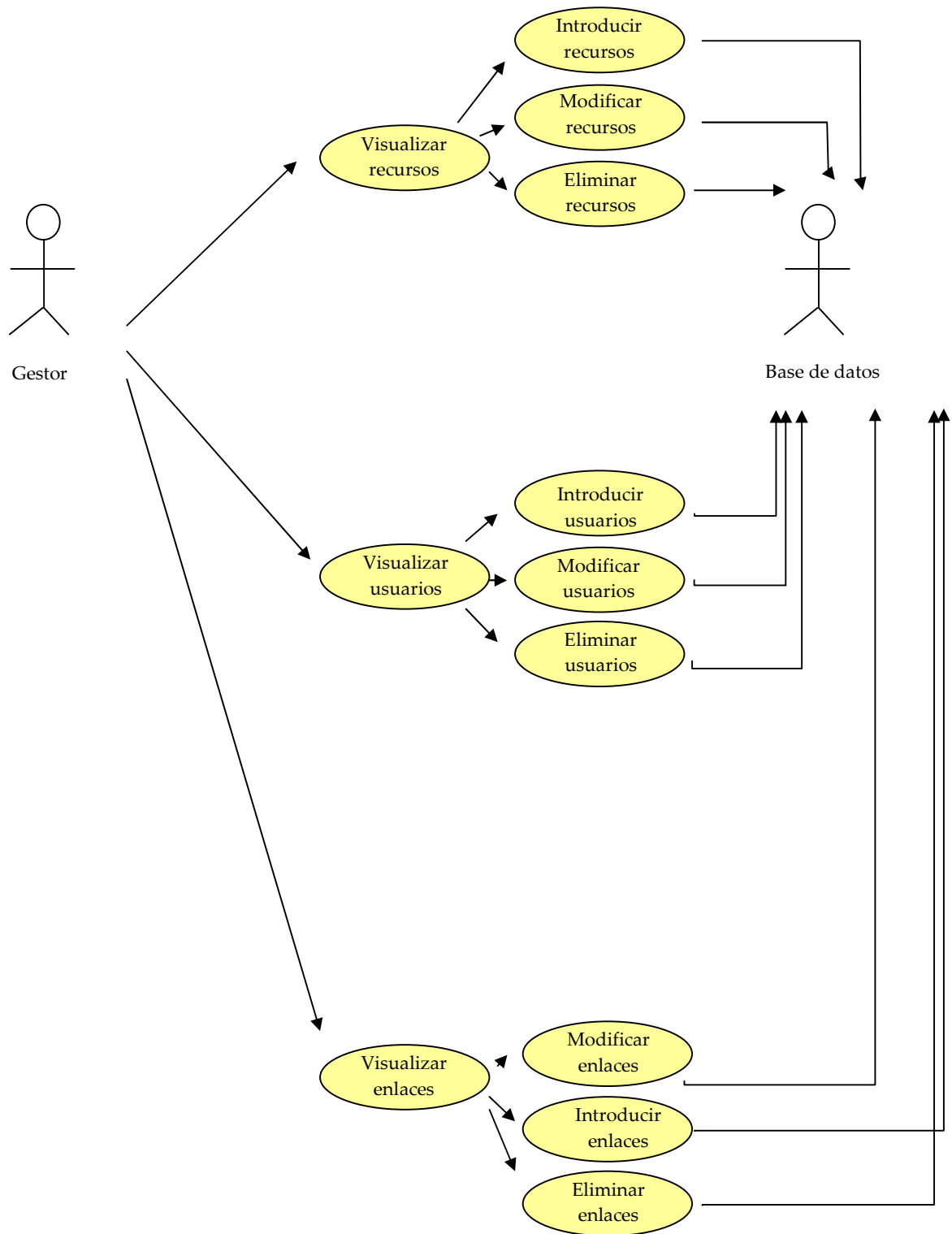


### 3.1.3.3 Profesor

Para hacer más comprensible el esquema de casos de uso, aquellos que son compartidos con el alumno, no se han vuelto a poner: identificarse, acceso a archivos compartidos, acceso al foro, ver enlaces de interés i enviar comentario a otro usuario.



### 3.1.3.4 Gestor de contenidos



### ***3.1.4 Menú de navegación***

Uno de los módulos importantes en la creación de este proyecto es el menú de navegación. Este menú facilitará el uso de la aplicación a cada uno de los usuarios. Como ya se comentó anteriormente, cada usuario tendrá un menú distinto según el tipo de perfil al que pertenezca. A continuación se describe cada una de las categorías que lo componen.

#### ***3.1.4.1 Alumno***

##### ***Recursos disponibles***

En esta categoría el usuario podrá visualizar los distintos recursos que hay disponibles para la creación del proyecto, y reservar su uso para unas fechas determinadas. Primero se seleccionará el recurso, se mostrará las fechas disponibles, y a partir de aquí, se seleccionará la fecha determinada que interese.

##### ***Proyecto***

En esta sección aparecerá toda la información relacionada con el proyecto que se está creando: nombre del proyecto, descripción, información del tutor (nombre, despacho, mail y horas de tutoría), fecha de inicio, información académica del alumno (nombre, NIU, mail) y las tareas que se han ido asignando en el transcurso del desarrollo del proyecto por el tutor. Se podrá introducir comentarios a cada una de las tareas, y asignarles un estado determinado: finalizada, en proceso, y sin iniciar.

##### ***Calendario de tutorías***

El alumno podrá visualizar un calendario con las tutorías estipuladas por el tutor. El alumno podrá enviar un mail al profesor en el caso de que alguna fecha u horario no se adapte a la situación del estudiante. Así como en el caso de que desee reservar una tutoría extra por algún motivo concreto.

### ***3.1.4.2 Profesor***

#### ***Proyectos asignados***

Lista de todos los proyectos del que se ha nombrado tutor. De cada proyecto podrá obtener toda la información detallada, la misma que visualiza el alumno. Además, podrá llevar un control de notas de seguimiento por proyecto, así como visualizar el estado de las tareas, y la introducción de nuevas.

#### ***Calendario de tutorías***

El tutor escogerá un proyecto en concreto y a partir de aquí se mostrara el calendario de tutorías para este. El profesor podrá eliminar, modificar e introducir una nueva tutoría. El usuario será avisado de forma automática vía mail de la gestión de la tutoría. De igual forma, existirá la opción integrada en el entorno de trabajo del usuario para comunicarse con el alumno mediante el envío de un mail para cualquier asunto en concreto.

Categorías comunes a profesor y alumno:

#### ***Inicio***

Da la opción de volver a la página inicial de la aplicación desde cualquier sección.

### *Foro*

Permite acceso al foro compartido por estudiantes y profesores. Tanto uno como otro, podrá visualizar los comentarios hasta ese momento existentes, y añadir nuevas entradas.

### *Archivos compartidos*

En esta sección se mostrará el directorio de archivos compartidos entre tutor y alumno por proyecto. Para el caso del profesor, primeramente se listará la totalidad de proyectos adscritos a él, para que una vez seleccionado uno, se muestren los archivos correspondientes. Una vez en el directorio deseado, tanto alumno como profesor podrán ver y editar archivos existentes, y crear y subir de nuevos.

### *Enlaces de interés*

Se mostrará una serie de links de interés para alumnos y profesores, con utilidad en la creación de proyectos final de carrera.

### **3.1.4.3 Gestor de la base de datos**

Como se explica en puntos anteriores, el gestor será el encargado de gestionar todos los datos de la base de datos y usuarios.

### *Gestionar Recursos*

En esta categoría el gestor de contenido podrá listar la totalidad de recursos, así como modificar o añadir de nuevos. Tendrá acceso total a la base de datos de estos y por tanto podrá editar cualquier atributo respecto a ellos.

### *Gestionar Usuarios*

El administrador se encargará de dar permisos a los usuarios según su perfil, así como controlar que los registros se realicen de forma correcta. Podrá visualizar todos los usuarios, y seleccionar aquellos para los que quiera realizar alguna acción concreta.

### *Gestionar enlaces de interés*

Podrá eliminar, modificar y crear los links de interés que se muestran al profesor y estudiante.

## **3.2 Restricciones de diseño**

Para garantizar una seguridad en la edición de los datos, en las acciones tales como modificar o eliminar, siempre se requerirá una confirmación de dicha operación para evitar que el usuario cometa errores involuntarios debido a un uso incorrecto de la aplicación.

En algunas secciones determinadas, el usuario interactuará directamente con la base de datos insertando, eliminando o creando entradas. Para mantener siempre una coherencia entre los datos y tablas, se comprobará cada una de estas consultas, y se gestionará cada uno de los posibles errores que puedan surgir por una mala utilización del aplicativo.

Por otro lado, en las consultas de tipo 'insert' o 'update', siempre se comprobaran los datos a introducir mediante una función de tipo "sanitize" consistente en sanear los datos del usuario. De esta manera se procura evitar que usuarios malintencionados puedan explotar la posibles vulnerabilidad de nuestra aplicación mediante ataques XSS (*'Cross Site Scripting'*), CSRF (*'Cross-site request forgery'*) y SQL Injection.

El aplicativo podrá ser utilizado por tres perfiles de usuarios distintos. Para mantener una seguridad en el acceso a los datos, se utilizará una variable de sesión que será comprobada en cada página para controlar en todo momento el tipo de usuario que está utilizando el aplicativo y por tanto, la información que contendrá dicha página, así como el menú de navegación correspondiente.

# Capítulo 4. Diseño

---

## 4.1 Introducción

En este apartado describiremos con detalle el diseño de la aplicación web que va a ser desarrollada. Esto consiste en recolectar los datos necesarios y escoger los métodos que trataran estos datos para dar consistencia al proyecto. Esta fase es de gran relevancia, pues un buen diseño puede significar una disminución de los posibles errores futuros, así como una fácil introducción de mejoras.

Para ello, dividiremos la labor en dos partes. En la primera se realizará el diseño gráfico de la aplicación web, intentando obtener un resultado eficaz en términos de usabilidad y claridad. Y por otro lado, para cualquier proyecto que necesite trabajar con datos almacenados, es indispensable una buena organización previa de la información, por tanto, un buen diseño de la estructura de la base de datos mediante un diagrama entidad-relación.

Para la codificación del diseño, es necesario el uso de un lenguaje de programación que se adapte a las necesidades del proyecto. Debido a la naturaleza web del mismo, un lenguaje de gran utilidad y fácil aprendizaje es PHP para el servidor, y html junto con CSS para la parte gráfica. También se hará uso en determinadas ocasiones de Javascript, y para la realización de las consultas a base de datos SQL.



## 4.2 Capas de la aplicación

La programación por capas consiste en diferenciar tres niveles a la hora de diseñar una aplicación: el entorno de usuario, el motor de la aplicación y la capa de datos. Para cada una de estas capas, se utilizará una serie de herramientas distintas adecuadas a sus características.

### 4.2.1 *Entorno de usuario*

Esta capa la forma la parte gráfica del aplicativo. Será la encargada de interactuar con el usuario, de manera que ésta le proporcione información y le muestre el contenido de la aplicación, y por otro lado, recogerá los datos que ingrese el usuario mediante distintos y variados formularios. Estos datos son enviados a la capa de motor, la cual se encargará de gestionarlos. El resultado final del entorno debe de ser simple y claro para el usuario, con el objetivo de facilitar su uso y comprensión.

Para realizar la codificación de esta capa, utilizaremos el lenguaje de marcado xhtml. Éste permite crear páginas web compatibles con la mayoría de los navegadores.

Para darle formato al contenido de las páginas, utilizaremos hojas de estilo CSS. De una forma sencilla y rápida conseguiremos obtener una interfaz amigable.

Finalmente para la validación de los formularios y comprobación de que los datos introducidos por el usuario son correctos, utilizaremos Javascript.

### 4.2.2 *Motor de aplicación*

En el motor de la aplicación se genera y selecciona la información que formará parte del contenido de la aplicación. Esta capa se comunica con el entorno de

usuario para recibir las peticiones y presentar los resultados finales después de la elección de aquella información que cumple con una serie de reglas establecidas en esta misma capa. Para ello, se comunica también con la capa de datos. Mediante una serie de operaciones puede recuperar y almacenar aquellos datos que necesite haciendo uso del gestor de la base de datos.

Para poder realizar todo este proceso, se utilizará el lenguaje de programación PHP. Este lenguaje se puede introducir en código html de forma fácil y permite la creación de páginas dinámicas.

### ***4.2.3 Capa de datos***

En la capa de datos es donde está almacenada la información que gestionará el proyecto. El gestor de la base de datos es el encargado de realizar aquellas operaciones y acciones que permiten la inserción, recuperación, modificación y eliminación de los datos.

Se almacena aquella información que es dinámica y que por tanto varía a lo largo del tiempo.

Para nuestro proyecto se ha escogido MySQL, un sistema de gestión de base de datos relacional. En el siguiente punto se detalla de forma específica cada una de las tablas que se incluyen y crean la base de datos de este proyecto, y las relaciones entre ellas mediante un diagrama entidad relación.

## **4.3 Estructura de la base de datos**

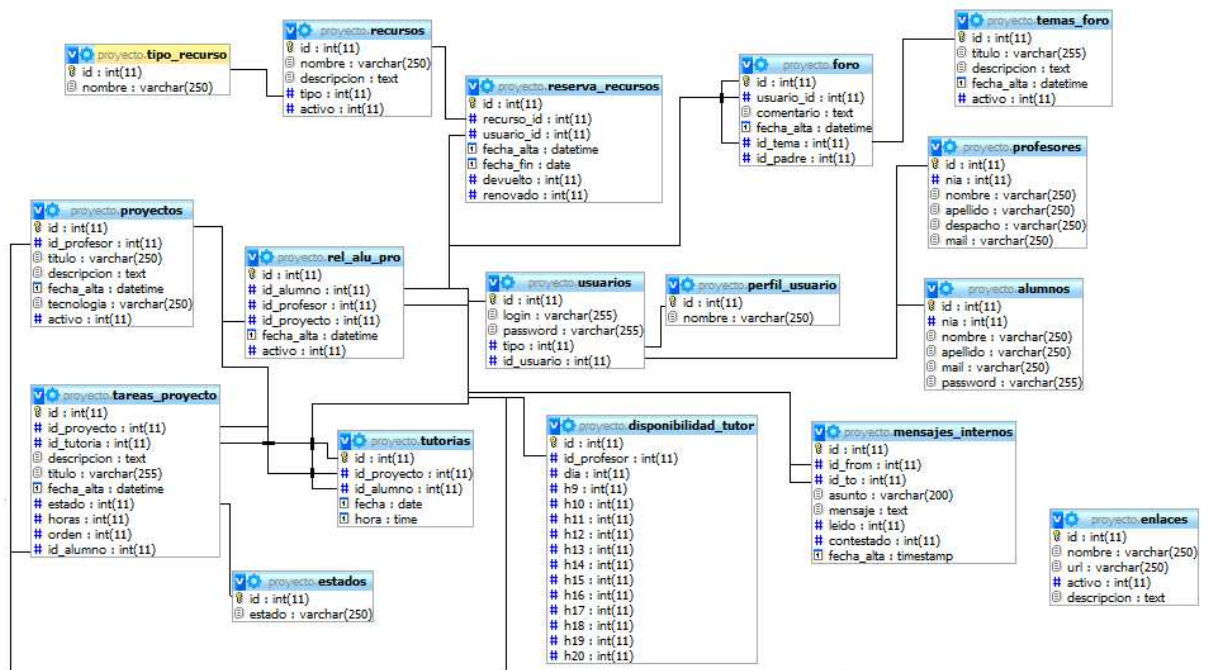
La estructura de la base de datos está compuesta por 17 tablas relacionadas entre ellas. Cada una tiene sus propios atributos, entre los cuales, la clave primaria identificada por el atributo: id. Este atributo es único para cada una de las entradas de la tabla correspondiente. A continuación, una descripción de las principales:

- ✓ **Usuarios:** esta tabla contiene información de los usuarios registrados de la aplicación. Se guarda el tipo de usuario al que pertenece, es decir, el identificador del perfil de usuario, un login y un password, y un identificador de usuario correspondiente al id de la tabla alumnos o profesores según se trate de uno u otro.
- ✓ **Perfil de usuarios y Permisos de usuario:** en la tabla de perfiles tenemos una entrada para cada uno de los tipos que se explicó anteriormente: alumno, profesor y gestor. Cada uno asociado a un permiso. Y la de permisos, describe los diferentes menús al que tienen acceso los perfiles.
- ✓ **Reserva de recursos:** en esta tabla se guarda toda la información relacionada con las reservas de los recursos realizado por los usuarios registrados. Contiene el id del usuario que ha reservado, el recurso, la fecha en que se hizo la reserva, y la fecha en la que el recurso debe ser devuelto.
- ✓ **Proyectos:** esta tabla encierra los atributos necesarios para mantener los datos de cada proyecto dado de alta por un profesor. Entre ellos tenemos, el identificador del profesor que lleva el proyecto, la fecha de alta de este, una descripción detallada del proyecto, las herramientas que se utilizarán y un título que lo identifique.
- ✓ **Rel\_alu\_pro:** debido a que puede darse el caso en que un mismo proyecto puede estar realizado por dos usuarios distintos, se creó esta tabla. Contiene las relaciones entre proyectos y alumnos. Es decir, cada entrada hace referencia a un proyecto y a un alumno, así, pueden haber diferentes entradas con el mismo proyecto, pero distinto alumno.
- ✓ **Tutorías y Tareas de proyecto:** la tabla de tutorías contiene la fecha y hora junto a un id de proyecto para cada tutoría concertada, y la de

tareas, una descripción de la misma, un identificador de proyecto asociado, y un estado.

- ✓ Las tablas de **Alumnos** y **Profesor** incluyen toda la información necesaria de cada uno de estos usuarios. Estos datos serán cedidos por la universidad.
- ✓ **Mensajes internos:** contienen los mensajes enviados tanto por parte del tutor como del alumno vía mensajería interna. Contiene un identificador de quien lo envía, quien lo recibe, un campo que guarda el mensaje, el asunto y unos campos que sirven de flag para saber si los mensajes han sido leídos y contestados.

A continuación se muestra un diagrama de la base de datos que muestra todas las tablas explicadas en los puntos anteriores. Cada tabla está representada por un rectángulo, el cual contiene cada uno de los atributos de las entidades. Las relaciones entre ellas están simbolizadas mediante una línea:



## 4.4 Interfaz

Se ha utilizado un modelo de interfaz sencillo y de fácil comprensión para que al usuario no le resulte complicado la utilización del mismo. Se han cogido aspectos similares al propio campus virtual de la UAB para seguir con la línea de estilo que en este se presenta y conseguir una homogeneidad entre los distintos aplicativos.

### 4.4.1 Definición de la interfaz para distintos perfiles de usuario:

#### 4.4.1.1 Interfaz Usuario no registrado

El usuario no registrado es aquel que no se ha identificado como usuario de la UAB y que no ha implementado el formulario inicial que le da acceso a la aplicación. Para este usuario tenemos dos tipos de interfaz distintos según en el paso en el que se encuentre:



En este caso, se nos presenta un formulario de identificación como usuario de la UAB. Se debe introducir el NIU y el password del campus para verificar la identificación. Si es así, se nos presentará en pantalla el siguiente formulario de registro:










#### ***4.4.2 Interfaz Usuario registrado***

Este usuario ha implementado el formulario de registro, se ha identificado con un login y un password y ha accedido a la aplicación. Hay 3 tipos de perfiles que acceden al aplicativo: el gestor de la aplicación, el profesor y el alumno. A cada uno de ellos se le mostrara un menú de navegación distinto que cumpla con los permisos asignados correspondientes. El estilo y modelo de interfaz es igual para todos.



1. En esta parte de la interfaz tenemos el menú principal. Cada uno de los iconos representa una de las secciones de las que dispone la aplicación. Empezando de izquierda a derecha, tenemos:

- a.  → hace referencia a aspectos relacionados con el proyecto. Información del proyecto, tareas, tutorías, asignaciones, etc.
- b.  → muestra la información relacionada con los recursos disponibles para la realización del proyecto.
- c.  → obtendremos un esquema tipo árbol con los archivos del proyecto que se han ido subiendo y modificando durante el proceso de creación del mismo.
- d.  → muestra un listado con enlaces interesantes para los estudiantes.
- e.  → direcciona a la parte del foro.
- f.  → muestra un formulario con datos de la cuenta del usuario. Estos pueden ser actualizados.

g.  → direcciona a mensajería interna. Un servicio de mensajes entre el tutor y el alumno.

2. En esta parte del diseño encontramos el submenú. Éste presenta un listado de opciones distinto según nos encontremos en una sección u otra. Esta parte del diseño, se ha realizado con la tecnología ajax, de manera que tan solo se carga el submenú, y no la página entera, cuando clicamos sobre una de las secciones.

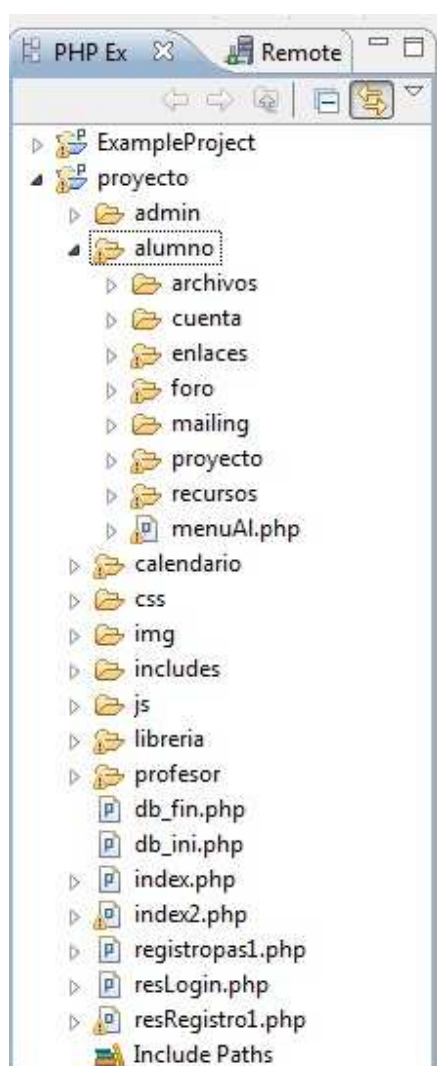
El submenú de una misma sección, puede ser distinto según el perfil del usuario logado. De forma general, para el usuario 'alumno' el submenú consta básicamente de operaciones que tan solo muestran información del proyecto, recursos y enlaces. Tan solo puede modificar datos de su propia cuenta, realizar reservas de recursos y participar en el foro. El usuario 'profesor' tiene, además de lo anterior, la capacidad de gestionar los proyectos (crear, modificar y eliminar proyectos y asignaciones a estos) e introducir nuevos enlaces en el apartado correspondiente. El usuario 'gestor', en su submenú tendrá permisos completos para gestionar proyectos, enlaces, cuentas de usuario, foro y recursos.

3. Este es el cuerpo principal de la aplicación. En este se mostrará todo el contenido de la aplicación: formularios, información, resultados de búsqueda, etc.



## 4.5 Estructura de archivos

Los archivos de la aplicación se han organizado por sección y por usuario para facilitar la comprensión y búsqueda de ficheros en caso de futuras modificaciones y ampliaciones.



De la raíz principal cuelgan los siguientes módulos y archivos:

- Modulo alumno
- Modulo profesor
- Modulo admin
- CSS
- Js
- Imágenes
- Includes
- Librería
- Archivos de configuración de base de datos y índice de la aplicación

Modulo alumno, profesor y admin: cada modulo contiene todos los archivos relacionados con el perfil correspondiente. La aplicación se basa en los submódulos de alumno, y en cada perfil se ha creado una carpeta para cada una de las secciones

del menú principal que difieren a las del perfil alumno. Dentro de cada una de estas carpetas tenemos, por un lado, un archivo que contiene todas las opciones

del submenú para esa categoría en concreto, y luego un archivo que implementa cada opción de este submenú.

CSS: contiene todos los archivos de hojas de estilo de toda la aplicación. Definen el diseño del aplicativo.

JS: contiene archivos con distintos scripts en javascript y jquery que se utilizan en algunas secciones de la aplicación para el tratamiento de formularios y la carga de la página mediante ajax.

Imágenes: en esta carpeta se guardan todas las imágenes que se incluyen en el gestor. Tanto las imágenes para el menú principal, como aquellas imágenes de determinadas secciones.

Includes: son archivos que contienen una parte del código que luego se incluyen en archivos de los módulos alumno, profesor y admin. Se ha realizado de esta manera para simplificar el código en aquellas partes que eran comunes y se repetían en distintas ocasiones.

Librería: contiene archivos con funciones específicas creadas para determinadas operaciones del aplicativo que facilitan la implementación del mismo.

Archivos de configuración e índices: los archivos de configuración contienen la información necesaria para el acceso a base de datos. Los índices constituyen las páginas iniciales de login y registro previas a la entrada al gestor.

# Capítulo 5. Implementación

---

## 5.1 Capa de presentación:

Como ya se introdujo en capítulos anteriores, la capa de presentación es aquella parte del proyecto encargada de comunicar al usuario con el sistema. La codificación de ésta se ha realizado con los siguientes lenguajes de programación: HTML y CSS.

HTML es el lenguaje de marcado por excelencia de las páginas web actuales. Un lenguaje de marcado codifica el documento con etiquetas que contienen información de la estructura y presentación del texto.

CSS (hojas de estilo) es un lenguaje utilizado para definir la presentación de un documento escrito en HTML. Esto es, especifica información sobre una serie de características para un elemento en concreto del código html: la fuente, el color, el tamaño, alineación del texto, etc.

Con la mezcla de estos dos lenguajes conseguimos separar la parte de estructuración del documento, con la de la presentación. Así, ante posibles cambios tanto en el estilo como en la estructura, resultan fáciles de realizar.

La información CSS puede introducirse dentro del mismo código HTML o en archivos externos. En este proyecto se ha optado por la separación del código CSS en ficheros distintos que posteriormente se incluyen mediante una referencia al archivo a partir de un path.

Se ha intentado seguir un patrón de diseño similar en toda la aplicación para que al usuario le resultara intuitivo su uso.

Así, todos los formularios tienen el mismo diseño, hay una página inicial en cada uno de los ítems del menú principal, que explica con brevedad que es cada una de las opciones del menú, y cada bloque del menú está compuesto por unos elementos comunes, como son el título y una pequeña descripción del apartado en el que se encuentra.

Para conseguir una codificación estructurada se han utilizado divisiones del código mediante la etiqueta `<div></div>`. Cada div contiene partes del documento que comparten la misma presentación, información y situación física en el diseño web. En este aspecto, señalamos la distinción de 4 tipos de divs importantes identificados por el atributo "id" con el siguiente nombre: principal, cabecera, submenú, contenido y contenido2. En el div con identificador "contenido2" se introduce toda la información que se va a presentar al usuario de una sección determinada. En el div con id igual a "submenú" contiene las opciones de menú de un ítem determinado del menú principal.

El div identificado por "contenido" incluye los dos divs anteriores: contenido2 y submenú.

El div con id "cabecera" contiene toda la parte superior del diseño del aplicativo. Esto es el menú principal, el logo y el nombre de usuario.

Finalmente el div principal incluye todos los otros divs. Aquí un esquema de la división del código:

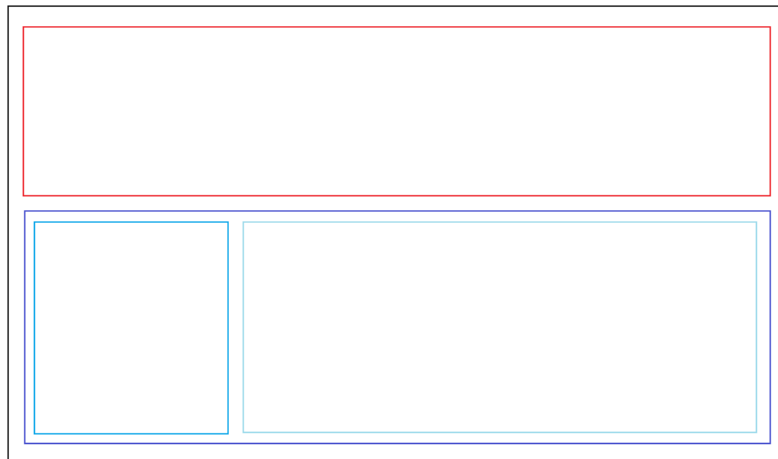


Figura1.División de la aplicación en distintos divs

Cada uno de estos rectángulos de la figura 1 representa uno de los divs anteriormente explicados: div principal (línea negra), div cabecera (línea roja), div contenido (línea azul oscuro), div submenú (línea azul), div contenido2 (línea azul claro).

Para la organización de la información contenida en el div contenido2 se han utilizado varios recursos. Por un lado, para el caso en que se haya necesitado mostrar un listado de elementos, en que el número de elementos podía variar según variables externas, se ha hecho uso de las listas, codificadas con la etiqueta: `<ul></ul>`. Cada ítem de la lista está identificado por los tags: `<li></li>`.

Estas listas siempre están incluidas dentro de un "div", al que se le ha dado un formato determinado para mostrar la información de una forma entendible y amigable para el usuario.

Existe también la opción del uso de la etiqueta `<table></table>` utilizada para organizar la información en tablas de forma ordenada. Dado que su uso no es 100% recomendable para la maquetación de webs por distintos factores (la legibilidad del código, las tablas son más lentas al cargar la página, para los

buscadores es más amigable un código sin tablas) su uso ha sido limitado a casos excepcionales. Entre ellos, los casos en que la información es estática y siempre hay el mismo tipo de información.

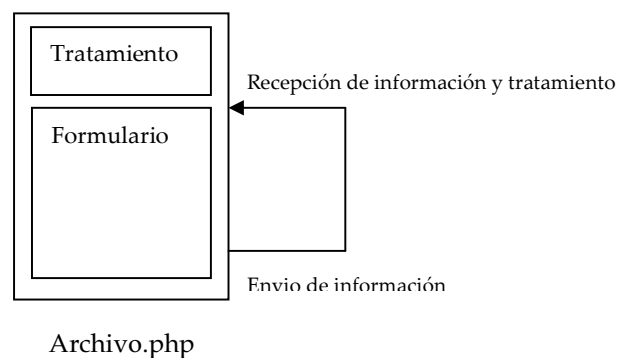
## 5.2 Capa de negociación:

Para la codificación de la capa de negociación se ha utilizado PHP. Este es un lenguaje de programación interpretado, lo cual quiere decir que debe de ser ejecutado por medio de un intérprete, en este caso, por el lado del servidor. El código PHP es cogido por el servidor como entrada, y devuelve como salida páginas web con contenido dinámico.

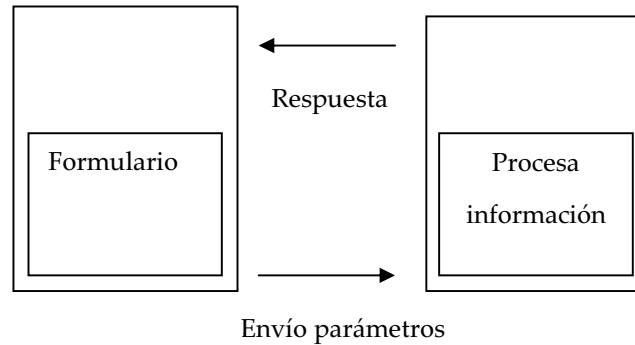
La ventaja de este lenguaje es que puede ser fácilmente incrustado en código HTML, y su aprendizaje es rápido y sencillo por su gran parecido con otros tipos de lenguajes.

Se ha utilizado PHP para poder trabajar con la base de datos y acceder a la información que en ella se almacena.

La forma de acceder a esa información se ha implementado de dos maneras distintas según la necesidad. En una primera, el mismo archivo que contiene la información de las secciones y los formularios correspondientes se encarga de tratar la información enviada por parámetros.



El segundo modo, mediante un archivo secundario que recibe los parámetros, trata la información y devuelve una respuesta.



En la codificación de la aplicación cabe destacar varios bloques importantes. Estos son: la distinción de perfiles, la implementación del foro, la mensajería interna, la asignación de tutorías y la reserva de recursos.

Para la diferenciación entre los distintos perfiles que pueden utilizar la aplicación, se han creado 3 archivos .php distintos. Uno para el profesor, otro para el alumno y finalmente otro para el gestor. De esta manera, cuando un usuario hace login, se identifica el tipo de perfil al que pertenece, y según sea uno u otro, se carga un archivo distinto. Así, cada archivo contiene el menú adaptado a cada uno de los usuarios.

El identificador de usuario y el tipo de usuario se guardan en variables de sesión, que se mantienen durante todo el tiempo en que el usuario está usando el aplicativo. Estas variables se utilizan para recuperar esos valores, saber en cada momento el identificador de usuario actual, y utilizarlos en diferentes consultas en las que esos datos sean necesarios.

Para la creación del foro no se ha utilizado ninguna librería externa, pues se ha creído interesante implementarlo personalmente.

La estructura de este resulta de la siguiente manera: existen una serie de temas, creados por los propios usuarios, donde se exponen comentarios relacionados con el tema que lo encabeza. Entonces, una vez se ha seleccionado un tema, se visualiza un listado de comentarios. Para cada uno de estos comentarios, se da la opción de ver las respuestas a dichos comentarios y responder también si se desea. Por otro lado, se puede comentar de forma global sin hacer referencia a ningún comentario en concreto.

La implementación del foro se ha hecho siguiendo una relación padre-hijo entre los comentarios. De manera que en la base de datos, para cada comentario hay un campo con el nombre 'id\_padre' y otro con nombre 'id\_tema'. El campo 'id\_padre' es el identificador del comentario al que se ha respondido. Si el campo id\_padre está a 0 indica que es un comentario global y que por tanto, no es la respuesta de ningún otro. El campo 'id\_tema' identifica el tema al que pertenece.

Para la implementación de los aspectos de las tutorías se ha dividido en distintos bloques.

En la parte del profesor, tenemos la asignación de tutorías, y la configuración de la disponibilidad.

En disponibilidad, el profesor selecciona para cada uno de los días de la semana (de lunes a viernes) las horas libres de las que dispone para consultas y tutorías con sus alumnos de forma general. Así, en base de datos se guarda el identificador de día y las horas asignadas. No hace referencia a ningún mes y año, solo a día de la semana.

En asignación de tutorías, el profesor selecciona un alumno, una fecha, y una hora de tutoría.

El alumno recibe toda esta información de la siguiente manera: visualiza las horas que tiene asignadas a tutorías en un calendario. Cuando el alumno clica sobre el



día en que tiene destacada una tutoría, se le redirige a una página con un detalle de horas.

Cada hora está marcada con un color distinto según esté programada para tutoría, como hora disponible para el profesor, o no haya programación alguna. Así, en el caso en que al alumno la hora asignada no le vaya bien, puede ver al momento que otras horas, para ese día, el tutor puede estar disponible.

Para la comunicación de este hecho, entre otros, se ha dispuesto el servicio de mensajería interna en el gestor.

La mensajería interna se ha codificado de tal manera que, un usuario con perfil de alumno, siempre enviará un mensaje dirigido a su tutor, sin opción de escoger a quien quiere enviar un mensaje. Esto es así porque la finalidad principal de la mensajería interna en el gestor es la comunicación única entre tutor-alumno de forma rápida y directa. Por otro lado, el usuario con perfil de tutor / profesor puede escoger al alumno al que quiere enviarle un mensaje. El profesor visualiza un listado de los últimos 10 mensajes más actuales, y para cada uno de ellos, se le indica el alumno que lo envía y da opción de responder.

Tanto para el alumno como para el tutor, los mensajes nuevos son destacados. La forma de detectar que un mensaje es nuevo o no se ha implementado de la siguiente manera: en base de datos, cada uno de los comentarios realizados tienen un campo con nombre 'leído' que indica si ese mensaje ya ha sido mostrado por pantalla o no. De manera que, cuando se recuperan los mensajes, se mira el flag 'leído', y si esta a '0' se le asigna la imagen que lo destaca como nuevo y se modifica el flag a '1' para que la próxima vez que se cargue, no lo destaque.

La reserva y renovación de recursos ha sido implementada de la siguiente manera. Se dispone una lista de los tipos de recursos que se pueden reservar. El usuario selecciona el tipo. Recibimos por parámetro el tipo de recurso y lo utilizamos para

conseguir el listado de recursos que cumplen con esa condición de tipo. Una vez se visualiza, se da la opción de ver la disponibilidad de ese recurso, o de todos los recursos de ese tipo seleccionado en un primer momento.

Para saber si un recurso está disponible o no, accedemos a la información de la base de datos que nos proporciona la tabla reserva\_recursos. A partir de aquí, un recurso está disponible si cumple con las siguientes condiciones:

- Que no aparezca en la tabla
- En el caso de que aparezca, que la fecha de devolución sea anterior a la actual y que el recurso haya sido devuelto.

Para saber si ha sido devuelto tenemos un campo con el nombre 'devuelto' que indica si se ha devuelto o no el recurso.

Una vez el usuario visiona la disponibilidad del recurso, puede seleccionarlo para reservarlo. Introduce la fecha en la que se devolverá, se controla que esa fecha no supere los 30 días desde el actual, y entonces se introduce entrada en base de datos.

Para la renovación de los recursos primeramente se buscan los recursos reservados por parte del usuario. Se buscan aquellos en que, el campo en base de datos esté igual a 0, que el campo 'renovado' esté a 0, y que la fecha de devolución sea superior a la actual. Una vez se han obtenido, se visualizan y el usuario selecciona el recurso que quiere renovar. La información del recurso se redirige a una página que calcula la nueva fecha de devolución sumando 15 días a la fecha de devolución actual existente en base de datos y cambia el campo 'renovado' a 1, para que no pueda volver a renovarlo.

### **5.3 Capa de datos:**

La capa de datos es la encargada de almacenar toda la información a la que será necesario acceder para cargar el contenido de la aplicación. Para ello utilizamos la base de datos MySQL, y el gestor phpMyAdmin que facilita el tratamiento y visualización de la información almacenada.

MySQL viene integrada en el paquete XAMPP que mencionamos en capítulos anteriores. Aparte de la base de datos, también incluye un servidor web Apache y los intérpretes de PHP y PERL.

# Capítulo 6. CONCLUSIONES

---

## 6.1 Desviación de la planificación

Como consecuencia de una serie de imprevistos surgidos durante las fechas planificadas para el desarrollo del proyecto, el plazo de entrega del mismo, se ha retrasado en dos meses más.

Un motivo de retraso es un aumento del tiempo de aprendizaje estimado, necesitado para la realización de diferentes tareas del proyecto. En la planificación del proyecto, se concentra en el conjunto de tareas que engloba la Implementación. Otro motivo de retraso, es la falta de tiempo del proyectista debido a problemas personales, que han impedido de forma general, mantener una implicación en horas que permita la realización del proyecto en las fechas estimadas inicialmente.

## 6.2 Línea de mejoras y ampliación

Debido a la amplitud del proyecto y a las posibles ramificaciones que puede tomar, se ha realizado una versión funcional básica que abarca una serie de herramientas muy concretas, y que pueden ser ampliadas y mejoradas para dar una mayor usabilidad al aplicativo.

Basándonos en la tabla de objetivos descrita anteriormente en el apartado X, podemos obtener las siguientes conclusiones: todos los objetivos críticos explicados en secciones anteriores han sido desarrollados al 100%. Los objetivos de carácter prioritario, también se han realizado al 100%. Respecto a los objetivos secundarios observamos que no se han llevado a cabo ninguno de ellos. El motivo

principal ha sido la falta de tiempo por la extensión de tiempo en la implementación de otros puntos más importantes.

Por tanto, a partir de estos datos, podemos tener en cuenta los objetivos no realizados para la realización de ampliaciones y mejoras en el aplicativo.

Un primer caso es la implementación de un módulo que se encargue del seguimiento del alumno por parte del profesor. Actualmente el gestor proporciona la herramienta para poder incluir una planificación de proyecto. Una posible ampliación consistiría en dar la opción al tutor de incluir notas a cada una de las tareas para poder llevar el seguimiento del alumno durante el desarrollo del proyecto. Otra opción, es no solo incluir notas en la planificación, sino, para cada tutoría asignada, poder llevar un registro de las cualificaciones que el alumno va obteniendo por parte del tutor después de la revisión de una tutoría.

De esta manera se obtendría un histórico del trabajo realizado por el alumno mucho más específico y detallado.

Otra futura línea de ampliación es la inclusión de informes con datos de distinta índole que puedan ofrecer al alumno y al profesor información útil de cara al proyecto final de carrera y su planificación.

Un aspecto no mencionado en los objetivos y que puede ser interesante de implementar, es el tratamiento de las horas en las tutorías. Tanto en la sección de disponibilidad del tutor como en la asignación de tutorías, las horas se seleccionan enteras. Podría mejorarse añadiendo precisión en su selección de forma que fuese posible escogerlas por fracciones. De esta manera, el alumno tendría una visualización de las tutorías mucho más detallada, y el profesor tendría opción de asignar tutorías con mayor flexibilidad y adaptada a sus necesidades.

Por lo que respecta a la parte técnica, actualmente, el gestor solo está maquetado para Mozilla Firefox porque se consideró que era uno de los más usados comúnmente por los usuarios. Una ampliación de esta visualización del gestor, hacia otros navegadores como puede ser Internet Explorer, puede suponer una mejora interesante a tener en cuenta en proyectos futuros.

# Capítulo 7. BIBLIOGRAFIA

---

## 7.1 Referencias electrónicas:

### Páginas web:

[<http://www.iconspedia.com>] Iconos Media,

[<http://www.desarrolloweb.com>] Desarrollo web

[<http://www.programacionweb.net/>] Programación web

[<http://www.webestilo.com>] Web Estilo

### Libros electrónicos:

[<http://www.librosweb.es/ajax/>] Libros web, Introducción a Ajax

[<http://www.librosweb.es/referencia/css/index.html/>] Libros web, Referencia de CSS.

[<http://www.librosweb.es/xhtml/index.html>] Libros web, Introducción a XHTML.

[<http://www.librosweb.es/css/index.html>] Libros web, Introducción CSS

[<http://www.librosweb.es/javascript/index.html>] Libros web, Introducción a Javascript.

### Tutoriales:

[<http://www.webestilo.com/php/>] Manual de PHP

[<http://php.net/manual/es/index.php>] Manual de PHP

### Foros:

[<http://www.forosdelweb.com/>] Foros del web.

