



**Universitat Autònoma  
de Barcelona**

# ProyGes – Soporte de gestión de proyectos

Memoria del proyecto de

Ingeniería Técnica en

Informática de Gestión

realizado por

Javier Carrasco Morgado

y dirigido por

Gonzalo Vera Rodríguez

**Escuela de Ingeniería**

Sabadell, Septiembre de 2012

El abajo firmante, **Gonzalo Vera Rodríguez**,  
profesor de la Escuela de Ingeniería de la UAB,

**CERTIFICA:**

Que el trabajo al que corresponde la presente  
memoria ha sido realizada bajo su dirección  
por ***Javier Carrasco Morgado***.

Y para que conste firma la presente.  
Sabadell, ***Septiembre de 2012***.

-----  
Firmado: **Gonzalo Vera Rodríguez**

<b>Título del proyecto:</b> ProyGes – Soporte de gestión de proyectos	
<b>Autor:</b> Javier Carrasco Morgado	<b>Fecha:</b> Septiembre de 2012
<b>Tutor:</b> Gonzalo Vera Rodríguez	
<b>Titulación:</b> Ingeniería técnica en informática de gestión	
<b>Palabras clave</b> (mínimo 3) <ul style="list-style-type: none"><li>• Catalán: web, projectes, col·laboradors</li><li>• Castellano: web, proyectos, colaboradores</li><li>• Inglés: web, project, collaborators, interface</li></ul>	
<b>Resumen del proyecto</b> (extensión máxima 100 palabras) <ul style="list-style-type: none"><li>• Catalán:<p>L'aplicació té com objectiu arribar a la comunitat d'alumnes, els quals no poden tenir reunions presencials atesa la seva situació geogràfica.</p><p>L'aplicació disposa d'un seguit d'eines per poder centralitzar tota la informació que els usuaris vagin generant durant el desenvolupament dels projectes. Ha de ser una eina per facilitar l'organització i l'administració dels recursos dels usuaris de la plataforma.</p><p>El projecte que es presenta es una aplicació web, desenvolupada en llenguatges PHP, SQL, HTML, Javascript, CSS, Ajax, en un servidor Apache i base de dades MySQL.</p></li><li>• Castellano:<p>La aplicación tiene como objetivo llegar a la comunidad de alumnos, los cuales no pueden tener reuniones presenciales debido a su situación geográfica.</p><p>La aplicación dispone de un conjunto de herramientas para poder centralizar toda la información que los usuarios vayan generando durante el desarrollo de los proyectos. Ha de ser una herramienta para facilitar la organización y la administración de los recursos de los usuarios de la plataforma.</p><p>El proyecto que se presenta es una aplicación web, desarrollada en lenguajes PHP, SQL, HTML, JavaScript, CSS, Ajax, en un servidor Apache y base de datos MySQL.</p></li></ul>	

- Inglés:

The application aims to reach those students who are not able to attend their meetings due to their geographical location.

The application has a set of tools to centralize all the information that the university students generate during their projects development. It has to be a tool to facilitate the organization and management of resources among users of the platform.

The project presented is a website which has been developed in PHP, SQL, HTML, JavaScript, Ajax, CSS language, on an Apache server and MYSQL database.

## **Índice de Contenidos**

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1. Presentación y objetivos .....	1
1.2. Estado del arte .....	1
1.3. Estructura de la memoria.....	3
<b>2. ESTUDIO DE VIABILIDAD.....</b>	<b>5</b>
2.1. Introducción .....	5
2.2. Estudio de la situación actual.....	5
2.3. Objetivos del proyecto .....	6
2.4. Requisitos generales del proyecto .....	7
2.5. Alternativas y selección de la solución.....	9
2.6. Planificación del proyecto .....	12
2.7. Evaluación de riesgos .....	16
2.8. Análisis de coste – beneficio. ....	18
2.9. Conclusión .....	21
<b>3. ANÁLISIS.....</b>	<b>22</b>
3.1. Introducción .....	22
3.2. Requisitos funcionales.....	22
3.3. Requisitos no funcionales .....	27
3.4. Perfiles de usuarios .....	29
3.5. Marco tecnológico.....	30
<b>4. DISEÑO .....</b>	<b>35</b>
4.1. Introducción .....	35
4.2. Arquitectura de la aplicación .....	35
4.3. Diagrama entidad/relación .....	36
4.4. Diseño de la base de datos.....	38
4.5. Tecnología a utilizar .....	45
4.6. Patrón modelo-vista-controlador .....	45
4.7. Interfaz usuario .....	47
4.8. Planificación detallada .....	49
<b>5. IMPLEMENTACIÓN .....</b>	<b>51</b>
5.1. Introducción .....	51
5.2. Herramientas a utilizar.....	51

5.3. Organización del código .....	52
5.4. Interfaces.....	55
5.5. Pruebas.....	70
<b>6. CONCLUSIONES .....</b>	<b>72</b>
6.1. Introducción .....	72
6.2. Desviaciones del proyecto .....	72
6.3. Futuras mejoras.....	75
6.4. Valoración personal.....	76
<b>7. BIBLIOGRAFÍA Y REFERENCIAS .....</b>	<b>77</b>

## Índice de Figuras

<i>Figura 1 Teambox, conocida aplicación colaborativa en la red. ....</i>	<i>2</i>
<i>Figura 2 Moodle tiene una base de 67000 sitios registrados, que ofrecen 5,5 millones de cursos, en los que participan 54 millones de usuarios .....</i>	<i>3</i>
<i>Figura 3 Esquema del problema de la dispersión de información que puede sufrir un grupo de trabajo de varios miembros .....</i>	<i>6</i>
<i>Figura 4 Esquema lógico de la solución.....</i>	<i>12</i>
<i>Figura 5 Diagrama de Gantt Planificación .....</i>	<i>15</i>
<i>Figura 6 Diagrama de Flujo Inicio Sesión/Registrar .....</i>	<i>23</i>
<i>Figura 7 Compartir Archivos.....</i>	<i>24</i>
<i>Figura 8 Caso de Uso Proyecto .....</i>	<i>25</i>
<i>Figura 9 Caso de Uso Agenda.....</i>	<i>26</i>
<i>Figura 10 Caso de Uso Mensajería.....</i>	<i>27</i>
<i>Figura 11 Estadística uso de navegadores .....</i>	<i>28</i>
<i>Figura 12 Facebook o Wikipedia están desarrollado en PHP .....</i>	<i>30</i>
<i>Figura 13 Python es usado por Google en aplicaciones como Youtube .....</i>	<i>31</i>
<i>Figura 14 JavaScript fue desarrollado por Brendan Eich de Netscape con el nombre de Mocha, el cuál fue renombrado más tarde a LiveScript.....</i>	<i>33</i>
<i>Figura 15 Existen más de seis millones de copias de MySQL funcionando en la actualidad.....</i>	<i>34</i>
<i>Figura 16 Organización Máquina Usuario – Aplicación .....</i>	<i>36</i>
<i>Figura 17 Modelo entidad/relación.....</i>	<i>37</i>
<i>Figura 18 Modelo E/R Usuarios con entidades relacionadas directamente .....</i>	<i>38</i>
<i>Figura 19 Modelo E/R Mensajes .....</i>	<i>39</i>
<i>Figura 20 Modelo E/R Agenda .....</i>	<i>40</i>
<i>Figura 21 Modelo E/R Proyecto con entidades relacionadas directamente .....</i>	<i>40</i>
<i>Figura 22 Modelo E/R Colaboradores .....</i>	<i>41</i>
<i>Figura 23 Modelo E/R Tareas .....</i>	<i>41</i>
<i>Figura 24 Modelo E/R Discusiones .....</i>	<i>42</i>
<i>Figura 25 Modelo E/R Documentación .....</i>	<i>43</i>
<i>Figura 26 Diagrama de base de datos .....</i>	<i>44</i>
<i>Figura 27 Esquema modelo-vista-controlador .....</i>	<i>46</i>
<i>Figura 28 Esquema del diseño inicio sesión .....</i>	<i>47</i>
<i>Figura 29 Esquema del diseño web .....</i>	<i>48</i>
<i>Figura 30 PhpMyAdmin ha sido traducido a 62 idiomas.....</i>	<i>51</i>
<i>Figura 31 Ficheros de la carpeta controllers .....</i>	<i>53</i>
<i>Figura 32 Fichero de la carpeta models .....</i>	<i>53</i>
<i>Figura 33 Ficheros de la carpeta views .....</i>	<i>54</i>
<i>Figura 34 Interfaz del proyecto .....</i>	<i>55</i>
<i>Figura 35 Interfaz inicio sesión.....</i>	<i>56</i>
<i>Figura 36 Interfaz registro.....</i>	<i>56</i>
<i>Figura 37 Interfaz recuperar contraseña.....</i>	<i>57</i>
<i>Figura 38 Interfaz cambio contraseña .....</i>	<i>57</i>
<i>Figura 39 Interfaz del encabezado .....</i>	<i>58</i>

<i>Figura 40 Interfaz del pie de página</i> .....	58
<i>Figura 41 Interfaz de la barra de navegación</i> .....	59
<i>Figura 42 Interfaz barra lateral</i> .....	59
<i>Figura 43 Interfaz mi panel</i> .....	60
<i>Figura 44 Interfaz agenda</i> .....	61
<i>Figura 45 Interfaz bandeja mensajes</i> .....	62
<i>Figura 46 Interfaz mensaje</i> .....	62
<i>Figura 47 Interfaz nueva y responder mensaje</i> .....	63
<i>Figura 48 interfaz mi cuenta</i> .....	64
<i>Figura 49 Interfaz nuevo proyecto</i> .....	65
<i>Figura 50 Interfaz información proyecto</i> .....	66
<i>Figura 51 Interfaz tareas</i> .....	66
<i>Figura 52 Interfaz colaboradores</i> .....	67
<i>Figura 53 Interfaz discusiones</i> .....	68
<i>Figura 54 Interfaz discusión</i> .....	68
<i>Figura 55 Interfaz documentación</i> .....	69
<i>Figura 56 Seguridad web</i> .....	71
<i>Figura 57 Diagrama de Gantt detallado</i> .....	74

## **Índice de Tablas**

<i>Tabla 1 Amortización</i> .....	11
<i>Tabla 2 Tareas y su duración</i> .....	13
<i>Tabla 3 Probabilidad de los riesgos</i> .....	17
<i>Tabla 4 Plan de contingencia</i> .....	17
<i>Tabla 5 Recursos humanos del proyecto</i> .....	18
<i>Tabla 6 Amortización de los recursos físicos</i> .....	18
<i>Tabla 7 Asignación tareas por recurso humano y su coste</i> .....	20
<i>Tabla 8 Coste total del proyecto</i> .....	20
<i>Tabla 9 Perfil de los usuarios de la web</i> .....	29
<i>Tabla 10 Planificación detallada de las tareas</i> .....	50
<i>Tabla 11 Desviación temporal</i> .....	73



## **1. INTRODUCCIÓN**

### **1.1. Presentación y objetivos**

Como alumno me he encontrado situaciones en que mis compañeros de prácticas al no poder asistir al centro y reunirnos, se hacía difícil la comunicación y estar de acuerdo para concretar decisiones de los proyectos. Así como en mitad del desarrollo no saber que versión era la correcta ni quién la tenía por falta de organización y utilización de demasiados canales de comunicación.

El objetivo de este proyecto consiste en crear una web 2.0, donde los usuarios son quienes generan el contenido, que permitirá relacionar miembros de un mismo grupo de trabajo, para facilitar el contacto entre ellos y agilizar el proceso de organización y repartición de tareas en un proyecto.

La idea de la realización de este proyecto final de carrera es cubrir las dificultades que tienen alumnos en realizar trabajos en grupo pero sin tener contacto presencial unos con los otros. Así como la descentralización de la información por los numerosos sistemas de mensajería ya existentes.

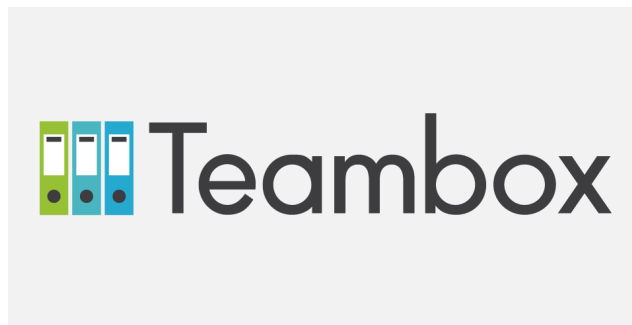
Por estos motivos se pretende crear una herramienta web en la que se recojan las distintas necesidades que se puedan presentar en el desarrollo de un trabajo en grupo y dar una solución con una herramienta colaborativa más específica de lo que se pueda encontrar hoy en día.

### **1.2. Estado del arte**

Hay numerosas grupos de herramientas para facilitar la comunicación pero no siempre son fáciles de utilizar o suficientemente específicas para completar todas las necesidades requeridas.

- **CMS (*Content Management System*):** Un sistema de gestión de contenidos es una aplicación informática usada para crear, editar, gestionar y publicar contenido digital multimedia en diversos formatos. El gestor de contenidos permite que cualquier usuario sin conocimientos de programación pueda añadir contenido en el portal web. Entre los CMS más conocidos se encuentran Joomla! o Drupal. Ambos programas libres desarrollados en PHP.
- **Aplicaciones colaborativas:** Se refiere al conjunto de programas informáticos que integran el trabajo en un solo proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo, conectadas a través de una red (internet o intranet).

Teambox o Versahub son dos ejemplos de este tipo de aplicaciones. El objetivo de la mayoría de estas aplicaciones son pequeñas empresas.



*Figura 1 Teambox, conocida aplicación colaborativa en la red.*

- **Foros/E-mails, espacios personales:** El mayor inconveniente, del que probablemente es la opción más utilizada por estudiantes, es lo repartida que queda toda la información que se pueden llegar a intercambiar a lo largo de un proyecto. Además de un coste añadido de tiempo cada vez que se quiera retomar el proyecto desde el punto donde lo dejó el propio usuario o sus compañeros.

- **LMS (*Learning Management System*)**: Es un sistema de gestión de aprendizaje web que se emplea para administrar, distribuir y controlar las actividades de formación no presencial de una institución u organización.

Las principales funciones son amplias: gestionar usuarios, recursos, así como materiales y actividades de formación, realizar evaluaciones y gestionar servicios de comunicación, entre otros.

El uso más común de este tipo de aplicaciones suele ser en Universidades utilizando, por ejemplo, Moodle por la Universidad Autònoma de Barcelona o el Proyecto Sakai por la Universidad Complutense de Madrid.



*Figura 2 Moodle tiene una base de 67000 sitios registrados, que ofrecen 5,5 millones de cursos, en los que participan 54 millones de usuarios*

Más adelante, en el punto 3.5 Marco tecnológico, se explican cada una de las tecnologías de las que se dispone para desarrollar el software, se toma una decisión y se explica porque se ha tomado esa decisión.

### **1.3. Estructura de la memoria**

La memoria está organizada de la siguiente manera: El primer capítulo es la introducción donde se resumen los objetivos esenciales del proyecto y se explica cual es el origen del mismo y que factores han contribuido a su desarrollo.

En el segundo capítulo se detalla el estudio de viabilidad del proyecto. En él, se estudiará el problema que se quiere resolver y se analizará su viabilidad técnica, los recursos necesarios y el coste que se deriva de su implementación. La conclusión de este capítulo es la de esclarecer si el proyecto es viable o no.

En el tercer capítulo, se analizan con profundidad cuales son las funcionalidades esperadas de la aplicación y bajo qué condiciones se ha de desarrollar una solución a los objetivos del proyecto. Además de analizar la tecnología que hay disponible y las alternativas que se pueden utilizar. Una vez analizado el problema se plantea un diseño de la solución.

En el capítulo de diseño, se especifica la tecnología y el lenguaje de programación elegido, el diagrama entidad relación, el diseño de la base de datos, la definición de la interfaz del usuario y los criterios elegidos para que ésta sea así. También se detallan las diferentes capas del patrón de desarrollo utilizado y la interacción entre ellas.

En el capítulo de la implementación se tratará los principales módulos de la aplicación, la estructura de los archivos y carpetas, y se explicará con detalle cada interfaz de la plataforma una vez realizadas, a partir de los esbozos del capítulo 4. El último punto de este capítulo contendrá la estrategia de pruebas utilizadas para garantizar que la aplicación sea íntegra y no tenga inconsistencias ni errores.

Por último, en el séptimo capítulo se encuentran las conclusiones del proyecto. Se analizarán si los objetivos inicialmente definidos han sido alcanzados, y las desviaciones temporales que hayan podido surgir desde el inicio hasta el final del desarrollo del proyecto, como también la valoración personal y qué mejoras puede tener la aplicación en el futuro.

## **2. ESTUDIO DE VIABILIDAD**

### **2.1. Introducción**

En este documento se mostrará la viabilidad del proyecto, así como marcando los diferentes objetivos a realizar, describiendo las partes interesadas o las restricciones técnicas.

Se analizará la situación actual y si existe solución al problema. De esta forma, a partir de la solución propuesta se hace una estimación de la planificación, un análisis de riesgos y una evaluación del coste en conjunto del proyecto para valorar el beneficio.

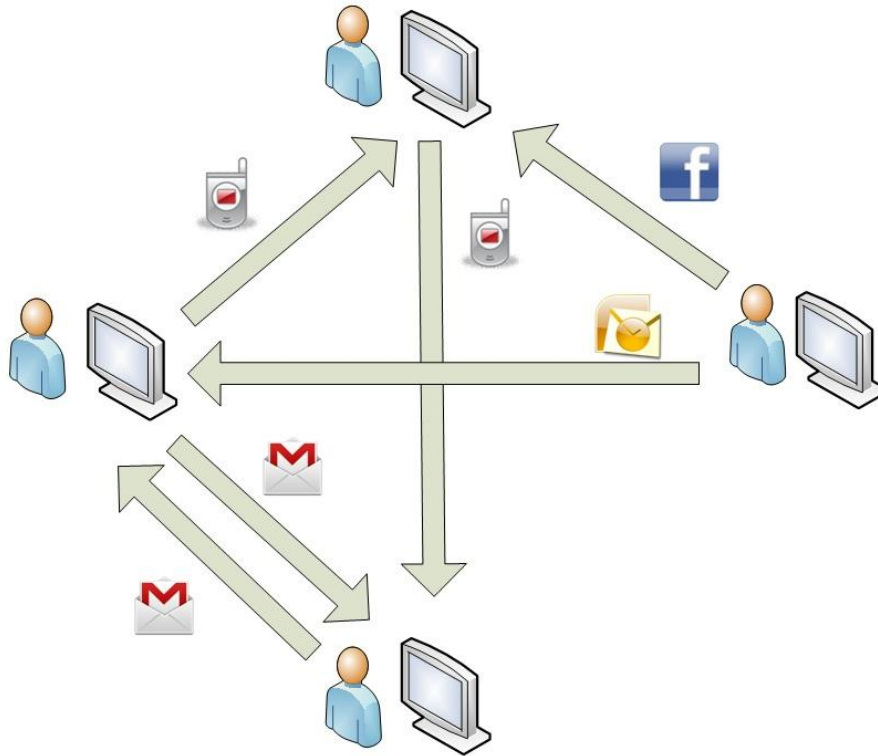
### **2.2. Estudio de la situación actual**

Actualmente en algunas universidades, especialmente las que no son a distancia, no es común disponer de una herramienta específica para la organización y colaboración de sus alumnos.

Éstas, generalmente, solo disponen de numerosas herramientas para la docencia de sus alumnos. Tomando de ejemplo la Universidad Autónoma de Barcelona, observamos algunas de ellas:

- Campus Virtual (<https://cv2008.uab.cat/>) el cual es utilizado principalmente para que profesores hagan llegar a sus alumnos el material de la asignatura, información respecto los profesores que imparten dicha asignatura o, incluso, un foro donde hacer cuestiones sobre la asignatura.
- Moodle: Es un sistema de gestión de aprendizaje, al igual que el campus virtual, se suele utilizar para colgar material didáctico, contiene un foro y se pueden realizar test on-line a través de él. Está totalmente integrado al campus virtual.
- Wikis: Es una plataforma informática que proporciona un entorno virtual colaborativo para dar soporte a diferentes actividades de la comunidad universitaria. Mediante esta aplicación, se pueden proponer listado de ejercicios y resolverlos.

Con el desarrollo de la nueva herramienta se suplirá una necesidad no cubierta por las actuales herramientas disponibles, solventando la dispersión de información que puede sufrir un grupo de trabajo de varios miembros y reduciendo el tiempo de la realización del proyecto gracias a una mejora en la eficiencia en un sistema centralizado de información del proyecto.



*Figura 3 Esquema del problema de la dispersión de información que puede sufrir un grupo de trabajo de varios miembros*

### **2.3. Objetivos del proyecto**

El objetivo de este proyecto es crear una herramienta para que les permita organizarse de una manera rápida, sencilla y que a su vez les permita un intercambio de documentos necesario para la realización de su trabajo. Así como un espacio donde los usuarios puedan compartir dudas y preguntas.

Los objetivos para llevar a cabo el proyecto son:

- Facilitar la comunicación entre compañeros haciéndola ágil y sencilla con un sistema de mensajería para cualquier usuario de la aplicación y una sección donde poder debatir temas o preguntar dudas con los demás miembros del grupo.
- Reducir el tiempo en utilizar diferentes herramientas independientes entre sí, con la creación de un sistema para compartir archivos, una agenda, un sistema de asignación de tareas y una mensajería.
- Ofrecer herramientas de utilidad para la realización de su trabajo.
- Mejorar la gestión y la organización de los alumnos.

## **2.4. Requisitos generales del proyecto**

Para definir los requisitos del proyecto hace falta conocer las funcionalidades que necesitará tener la aplicación.

En función de las características del entorno y las restricciones que tendrá la aplicación en su uso se definirán requisitos no funcionales y unas restricciones.

### **Requisitos Funcionales**

Para definir los requisitos funcionales, son el conjunto de funciones que deberá cumplir el sistema para cumplir con los objetivos. Ese conjunto de requisitos se describen a continuación:

#### **Usuarios**

- Sistema de registro para ingreso en la aplicación.
- Recuperación contraseña.
- Modificar y eliminar usuario.

## **Proyecto**

- Crear y eliminar Proyectos.
- Enviar solicitudes a futuros colaboradores proyecto.
- Creación, asignación y eliminación de Tareas asignadas a un proyecto.
- Sistema para compartir archivos entre los miembros del proyecto.
- Sección de discusiones del proyecto. Solo pueden verlo y acceder a él, los miembros del proyecto.

## **Agenda**

- Creación y eliminación de eventos.

## **Mensajes**

- Creación de mensaje.
- Responder mensaje.
- Verificación que usuario o correo electrónico existen en la aplicación.

## **Requisitos no funcionales**

Los requisitos no funcionales especifican los criterios que se deben usar para juzgar el funcionamiento del sistema, verificando como un sistema debería ser. A menudo son llamados “cualidades de un sistema”.

Los requisitos no funcionales de la aplicación son los siguientes:

- Facilidad de uso.
- Mínimo mantenimiento una vez finalizado.
- Validación de la información. Verificación en campos de obligatoriedad o longitud de caracteres.
- Buena escalabilidad.
- Correcta visualización en el navegador más utilizado del mercado.



## **Restricción del sistema**

- La aplicación se implementará en un Sistema Operativo Windows 7.
- Utilización de Software Libre.

## **2.5. Alternativas y selección de la solución**

### **Alternativa 1: Gestor de contenidos Joomla!**

Utilizar un gestor de contenidos Joomla!. Un gestor de contenido es una estructura para la creación y administración de contenidos, principalmente páginas Web. Entre las características que permiten algunos gestores de contenidos podemos destacar las siguientes:

- Creación de páginas Web sin conocimientos de programación.
- Gestión de usuarios de diferentes perfiles.
- Colaboración de varios usuarios en un mismo trabajo.
- Interacción mediante herramientas de comunicación.
- Costes de mantenimiento y evolución reducidos

Su uso puede ser un poco confuso para los usuarios que se inician. Además, puedes llegar a tener una gestión de usuarios y privilegios limitada.

Con Joomla! se pueden desarrollar páginas webs con diferentes funcionalidades rápidamente con los módulos ya predefinidos que contiene este gestor.

## **Alternativa 2: Desarrollo aplicación web**

Desarrollo de una aplicación Web con capacidad de creación de proyectos de forma colaborativa y diversas funciones que los complementen.

Esta alternativa tendría las siguientes características:

- Funcionalidad ajustada a los requisitos del usuario.
- Definición de la imagen e interfaz de usuario propia.
- Ampliación de funcionalidad ilimitada.
- Adaptación a criterios de seguridad y legales.

Se desarrollaría con el lenguaje de programación PHP, juntamente con MySQL para tratar los datos. Ambos lenguajes, son código libre igual que las herramientas para su uso.

Además de ser de los lenguajes con más soporte en la red para consultas o solución de problemas.

## **Alternativa 3: Sistema de gestión de aprendizaje Moodle**

La aplicación Moodle es un paquete de software para la creación de cursos y sitios web.

Contiene numerosos módulos de gran utilidad como los módulos de tareas, chat, consulta, encuesta, calendario, lección, blog, mensajería o foro.

Algunas de las características de Moodle serían:

- Facilidad de uso.
- Seguridad.
- Creación de grupos y subgrupos dentro de un mismo curso.
- Personalización del sitio.
- Paquete de idiomas.

## Solución propuesta

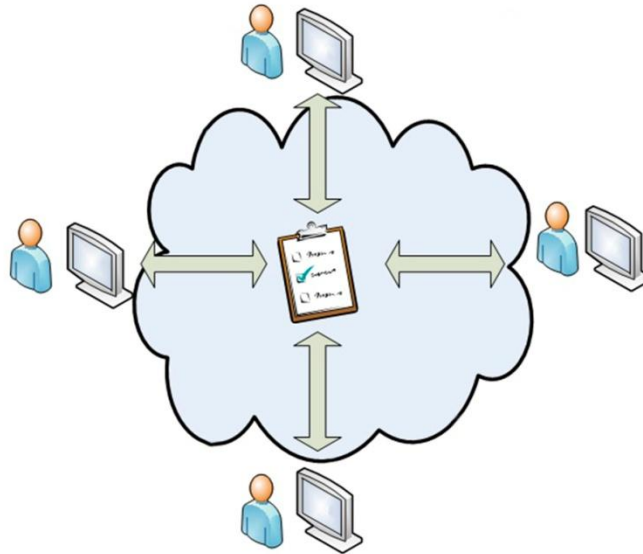
En la siguiente comparativa se ponen en común los aspectos identificados como más relevantes para la selección de la solución que más se adapta a nuestras necesidades.

	Costes Adquisición	Costes Adaptación	Ajustes a requisitos	Escalabilidad
<b>Alternativa 1: Joomla!</b>	0 €	Bajos	Medio	Media - Alta
<b>Alternativa 2: Desarrollo Web</b>	0 €	Presupuesto proyecto/Medios	Alto	Alta
<b>Alternativa 3: Moodle</b>	0 €	Bajos	Bajos - Medio	Media

*Tabla 1 Amortización*

Analizando las diferentes alternativas podemos ver que, aunque el coste de adaptación sería más bajo que las alternativa 1 y 3, ya que, vienen con gran parte de la aplicación predefinida, éstas se ajustan en menor medida a los requisitos definidos del proyecto. Además, en el caso de existir futuros requisitos adicionales en futuras fases, las alternativas 1 y 3 son menos flexibles para adaptarse a las nuevas funcionalidades. Llegando a suponer esto un coste superior en un futuro, en caso de añadir funcionalidades.

Por lo tanto, y teniendo en cuenta que la alternativa 2 se adapta al coste del presupuestado para el proyecto y que el resultado final será mucho más ajustado a los requisitos, **se decide elegir la alternativa 2.**



*Figura 4 Esquema lógico de la solución*

## 2.6. Planificación del proyecto

En la planificación del proyecto se definen las fases y las tareas que la componen, las cuales se deberán realizar durante el desarrollo del proyecto, y el tiempo estimado para cada una de estas tareas.

### Organización del proyecto

El proyecto está organizado en 4 grandes fases:

- **Planificación:** Se hace el estudio de viabilidad del proyecto determinando si el proyecto es viable técnica, temporal y económicamente.
- **Análisis de la aplicación:** Se especifican los requisitos del proyecto, se estudia cómo y con qué se puede desarrollar la aplicación.
- **Diseño de la aplicación:** Crear el diseño de las diferentes partes del que constará el desarrollo del proyecto.
- **Desarrollo y Pruebas:** Consta de las tareas relacionadas con la implementación de lo que se ha diseñado en la anterior fase.

## Tareas del proyecto

Como se puede ver en la tabla de tareas, la primera columna muestra la tarea a realizar y en la segunda, una estimación del tiempo que va a ocupar realizar la tarea. Como se puede ver en la planificación la mayoría de horas se asignan al desarrollo de la aplicación, ya que es la parte más extensa y elaborada del proyecto.

Nombre de tarea	Duración
<b>Planificación</b>	<b>26 horas</b>
Estudio Viabilidad	25 horas
Aprobación Estudio de Viabilidad	1 horas
<b>Análisis de la aplicación</b>	<b>16 horas</b>
Análisis de Requisitos	3 horas
Análisis de Datos (Base de datos)	6 horas
Documentación Análisis	6 horas
Aprobación del análisis	1 horas
<b>Diseño de la aplicación</b>	<b>25 horas</b>
Diseño de la Base de Datos	8 horas
Diseño de la interfaz	5 horas
Diseño de las pruebas	5 horas
Documentación del diseño	6 horas
Aprobación del Diseño	1 horas
<b>Desarrollo y Pruebas</b>	<b>88 horas</b>
Desarrollo de la Aplicación	81 horas
Preparación del entorno de desarrollo	3 horas
Configuración de la base de datos	3 horas
Desarrollo funcionalidades de la Aplicación	58 horas
Desarrollo de la interfaz del usuario	15 horas
Test y Pruebas	4 horas
Pruebas Unitarias	2 horas
Pruebas de integración	2 horas
Documentación del Desarrollo y Test	4 horas
Aprobación del Desarrollo y Test	1 horas
<b>Generación del documento (Memoria del Proyecto)</b>	<b>40 horas</b>
<b>Cierre del proyecto</b>	<b>6 horas</b>
<b>Defensa del Proyecto</b>	<b>9 horas</b>
<b>Total</b>	<b>210 horas</b>

Tabla 2 Tareas y su duración

## **Planificación temporal**

El proyecto tendrá una dedicación de 10 horas semanales y el total de horas dedicadas será de 210 horas.

- Fecha inicio: 23 de noviembre de 2011.
- Fecha finalización: 13 de junio 2012.

Al inicio del proyecto siempre es de gran utilidad una estimación temporal de las diferentes fases. Así siempre se puede cuantificar sobre un eje temporal el tiempo y la dedicación y, en caso de retraso, ver cuánto, fuera de lo previsto, se está desviando el proyecto.

En las siguientes imágenes se puede ver con más detalle el progreso de las tareas en las diferentes fases bajo una línea de tiempo.

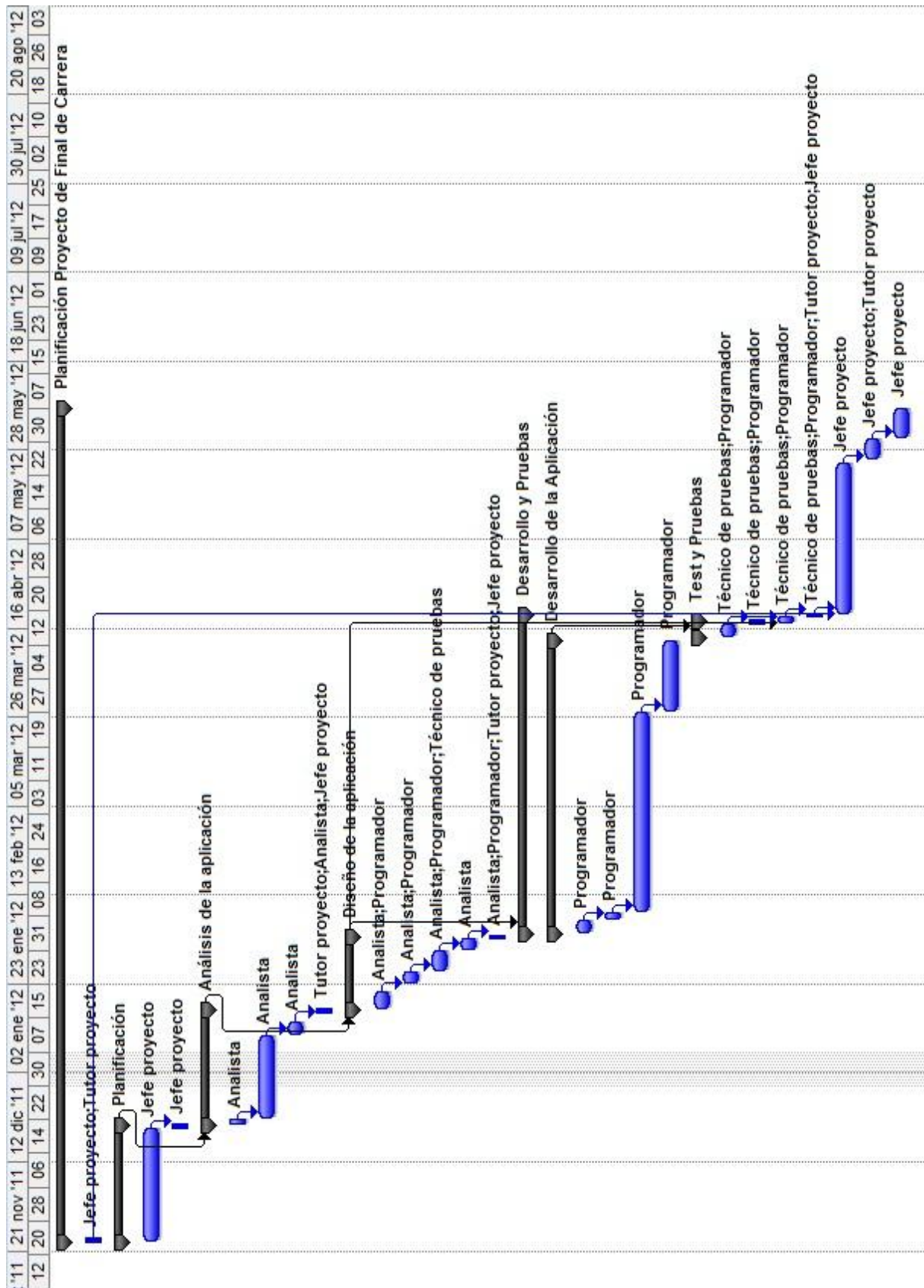


Figura 5 Diagrama de Gantt Planificación

## 2.7. Evaluación de riesgos

A continuación se hace una enumeración de riesgos existentes durante el desarrollo del proyecto y una vez finalizado éste, en el siguiente apartado se muestra una tabla donde se clasifican los riesgos según el impacto que podría tener en el éxito de la aplicación y la probabilidad de que ocurra.

### Lista de riesgos

- R1: Planificación temporal optimista. No se finaliza el proyecto en la fecha prevista.
- R2: Equipo de proyecto demasiado pequeño. Con el cual posiblemente haya un atraso de la fecha de fin del proyecto.
- R3: Cambio de requisitos del estudio de viabilidad, ya que, pueden surgir dificultades en el desarrollo de la aplicación y el resultado.
- R4: No se hace correctamente la fase de pruebas en la fase de implementación, por lo tanto puede comportar pequeñas deficiencias en la aplicación y posible insatisfacción del cliente.
- R5: Incumplimiento de alguna norma, reglamento o legislación en cualquier fase, entonces, no se cumplirán los objetivos principales y puede llegar a haber repercusiones legales.
- R6: Poca aceptación por parte del usuario final. Llame poco la atención y avoque en un olvido de la aplicación por parte del usuario a la que está destinada.



## Catalogación de riesgos

En esta tabla podemos distinguir los diferentes grados y posibilidades de impacto de los riesgos descritos en el apartado anterior.

Riesgos	Probabilidad	Impacto
R1	Alta	Importante
R2	Alta	Normal
R3	Alta	Normal
R4	Media	Importante
R5	Baja	Crítico
R6	Alta	Crítico

Tabla 3 Probabilidad de los riesgos

## Plan de contingencia

A continuación se exponen las soluciones que hacen falta adoptar para los riesgos citados anteriormente.

Riesgos	Solución a seguir
R1	Aplazar alguna funcionalidad. Tener una base del proyecto bien definida.
R2	Organización, constancia. Si no pedir aplazamiento.
R3	Reestructuración del desarrollo, modificación de la planificación y pedir aplazamiento.
R4	Diseñar la prueba con antelación y en formato automático si es posible.
R5	Revisar las normas y la legislación vigente, antes de los posibles riesgos, o consultar a un experto.
R6	Utilizar técnicas de posicionamiento en herramientas de búsquedas web. Utilización de redes sociales o foros para la promoción de la aplicación.

Tabla 4 Plan de contingencia

## 2.8. Análisis de coste – beneficio.

### Estimación del coste de personal

El siguiente cuadro detalla la relación de horas y la valoración de costes por cada recurso implicado en el desarrollo de la aplicación:

Recursos Humanos		Horas	Coste
Jefe Proyecto	50 €/h	85 h	4.250 €
Tutor Proyecto	0 €/h	10 h	0 €
Analista	40 €/h	41 h	1640 €
Programador	30 €/h	109 h	2180 €
Técnico de Pruebas	20 €/h	14 h	280 €
Total		210 h	8350 €

*Tabla 5 Recursos humanos del proyecto*

### Estimación coste de los recursos

En la tabla de a continuación podemos ver la relación de coste de los diferentes recursos no humanos que se utilizan en el momento de hacer o gestionar el proyecto.

	Coste Unitario	Periodo Amortización	Periodo utilización	Coste Amortización
Amortización PC Programador	1000 €	36 Meses	8 Meses	222,2 €
Amortización OpenOffice	0 €	N/A	N/A	0 €
Amortización MS Project	360 €	24 Meses	8 Meses	120 €
<b>Total</b>				<b>342,2 €</b>

*Tabla 6 Amortización de los recursos físicos*

## Estimación del coste de las actividades

En la siguiente imagen podemos ver la estimación del coste de cada tarea hecha a lo largo de la planificación. Los cálculos se han de hacer como base de los costes de cada recurso humano y no humano detallados anteriormente.

<b>Jefe proyecto</b>	<b>4.250 €</b>
Inicio del Proyecto: asignación y matriculación	50 €
Estudio Viabilidad	1.250 €
Aprobación Estudio de Viabilidad	50 €
Aprobación del análisis	50 €
Aprobación del Diseño	50 €
Aprobación del Desarrollo y Test	50 €
Generación del documento (Memoria del Proyecto)	2.000 €
Cierre del proyecto	300 €
Defensa del Proyecto	450 €
<b>Tutor proyecto</b>	<b>0 €</b>
Inicio del Proyecto: asignación y matriculación	0 €
Aprobación del análisis	0 €
Aprobación del Diseño	0 €
Aprobación del Desarrollo y Test	0 €
Cierre del proyecto	0 €
<b>Analista</b>	<b>1.640 €</b>
Análisis de Requisitos	120 €
Análisis de Datos (Base de datos)	240 €
Documentación Análisis	240 €
Aprobación del análisis	40 €
Diseño de la Base de Datos	320 €
Diseño de la interfaz	200 €
Diseño de las pruebas	200 €
Documentación del diseño	240 €
Aprobación del Diseño	40 €
<b>Programador</b>	<b>2.180 €</b>
Diseño de la Base de Datos	160 €
Diseño de la interfaz	100 €
Diseño de las pruebas	100 €
Aprobación del Diseño	20 €
Preparación del entorno de desarrollo	60 €
Configuración de la base de datos	60 €
Desarrollo funcionalidades de la Aplicación	1.200 €
Desarrollo de la interfaz del usuario	300 €
Pruebas Unitarias	40 €

Pruebas de integración	40 €
Documentación del Desarrollo y Test	80 €
Aprobación del Desarrollo y Test	20 €
<b>Técnico de pruebas</b>	<b>280 €</b>
Diseño de las pruebas	100 €
Pruebas Unitarias	40 €
Pruebas de integración	40 €
Documentación del Desarrollo y Test	80 €
Aprobación del Desarrollo y Test	20 €

*Tabla 7 Asignación tareas por recurso humano y su coste*

### **Análisis de coste - beneficio**

Podemos ver en la siguiente tabla el coste resultante de todos los recursos, humanos y físicos, utilizados en el desarrollo del proyecto.

<b>Tipo Gasto</b>	<b>Coste</b>
Coste de desarrollo del proyecto	8350 €
Coste de amortización del material	342,5 €
<b>Total</b>	<b>8692,2 €</b>

*Tabla 8 Coste total del proyecto*

El coste puede parecer elevado, pero se ha de tener en cuenta que el beneficio en esta aplicación es un ahorro temporal el cual a priori puede parecer intangible.

Haciendo una estimación del uso de la aplicación y del ahorro de tiempo de un usuario utilizándola, éste se puede ahorrar hasta 2 horas en una práctica de 4 semanas de durada media. Teniendo en cuenta que un crédito equivale a 25 horas de dedicación a la asignatura, el alumno podrá dedicarle, al menos, 2 horas más a la asignatura mejorando, de esta manera, la calidad en el estudio.

Si se extrapolan estas cifras a las de plazas ofertadas del grado de ingeniería informática de la UAB, 300 plazas, y suponiendo que tan solo 2 asignaturas, de primer curso, con 2 proyectos cada una, de duración de un mes cada proyecto, un alumno de éstas asignaturas ahorraría 8 horas de tiempo.

Aplicándolo a los 300 alumnos, anteriormente nombrados, sería un sumatorio de 2400 horas. Queda a juicio del lector valorar económicamente la suma de horas que, potencialmente, puede liberar una herramienta como la que se plante en este trabajo.

## **2.9. Conclusión**

Realizando el estudio podemos afirmar que se disponen de todos los medios necesarios para el desarrollo de la aplicación. Tanto el conocimiento de los lenguajes de programación como las herramientas para el desarrollo.

Por lo que respecta la planificación temporal es totalmente asumible su desarrollo en el tiempo establecido según las fases acordadas.

Por otro lado, teniendo en cuenta el ahorro de tiempo de los usuarios de la plataforma en la realización de sus trabajos y la dedicación de éste a otros asuntos de ámbito académico genera un incremento en el valor de la marca de la institución, gracias a la calidad de sus estudiantes.

Finalmente, una vez analizados estos puntos podemos concluir que el proyecto es viable, ya que se cumplen con todos los requisitos deseados y se dispone de la tecnología necesaria.

### **3. ANÁLISIS**

#### **3.1. Introducción**

En el análisis se examina el problema para llegar a una solución. Se observan las carencias, o necesidades, de los sistemas actuales y se estudia cómo mejorar esas carencias.

Con un estudio del problema podremos presentar un conjunto de especificaciones que la aplicación deberá tener.

Finalmente en este capítulo se estudiarán las opciones tecnológicas de las que se dispone para cumplir los requisitos especificados a lo largo del capítulo.

#### **3.2. Requisitos funcionales**

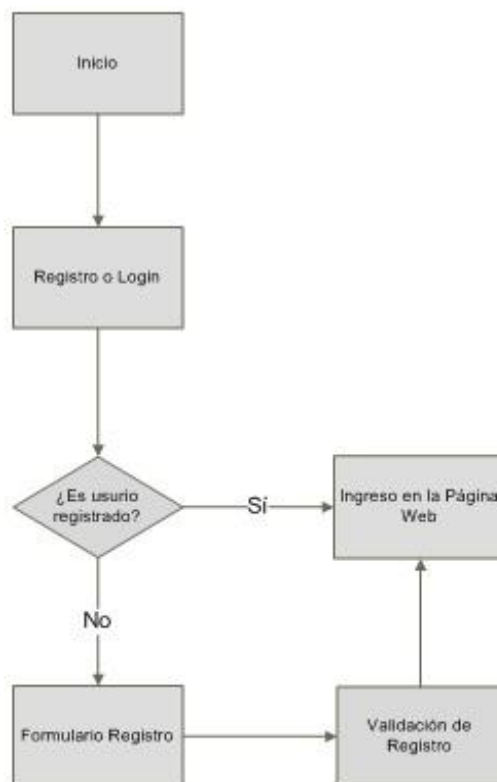
A continuación se detallan los requisitos funcionales del proyecto. En ellos se describen las funciones que deberá cumplir la aplicación en cualquier circunstancia, independientemente del usuario o de como haya sido implementado.

##### **Usuarios**

Sección referida al registro e inicio de sesión en la página web. Una vez iniciada sesión la opción de poder cambiar sus datos personales que haya facilitado o desactivar su cuenta de la aplicación.

- Registro usuarios.

El alta de usuario se realizará rellenando un formulario. El nombre de usuario y correo electrónico deberán ser únicos. Una vez validado, el usuario quedará registrado en la página y podrá hacer uso de ella.



*Figura 6 Diagrama de Flujo Inicio Sesión/Registrar*

- Recuperación contraseña.

Un usuario registrado en la aplicación, podrá recuperar su contraseña en caso de olvido, introduciendo su correo electrónico recibirá un correo con un enlace para modificar su contraseña.

- Modificar y desactivar usuario.

Cada usuario tiene acceso a sus datos personales y a la modificación de éstos, tales como nombre de usuario, dirección de correo electrónico o contraseña entre otros.

También tendrá la opción de eliminar su perfil de la base de datos de la aplicación.

## Proyecto

El espacio de mayor importancia y actividad en la aplicación. De donde contiene el mayor grueso de información que pueda tratar un usuario de la plataforma. Crear

proyecto con colaboradores, subir archivos, espacio de debate con colaboradores del proyecto.

- Crear y eliminar Proyectos.

Cada usuario tendrá la opción de crear proyectos, espacios donde invitar a otros miembros registrados en la aplicación y formar grupos de trabajo donde se podrá anotar fecha de entregas, repartición y asignación de tareas, miembros colaboradores.

Solo si se ha creado el proyecto podrá eliminar-se el proyecto. Si no, podrás darte de baja de él.

- Enviar solicitudes a colaboradores.

Después de la creación, si el usuario creador quiere invitar más colaboradores a su proyecto podrá hacerlo mediante una solicitud de participación en el proyecto a usuarios de la plataforma. El usuario que haya enviado las solicitudes de colaboración podrá hacer un seguimiento del estado de la petición si ha sido aceptada, o no.

- Creación, asignación y eliminación de Tareas.

Este apartado constará de las tareas que los colaboradores de un proyecto decidan añadir. La fecha máxima para la realización de esta tarea (si tiene), y quien es el colaborador al que le pertenece la tarea correspondiente.

Solamente quien el usuario al que ha sido asignada la tarea podrá darla por realizada y quien la ha creado podrá eliminarla.

- Compartir archivos.

Constará de un apartado en que los colaboradores podrán subir archivos para compartirlo con los demás miembros. Podrá verse el nombre del archivo subido, la fecha y el usuario que lo subió.



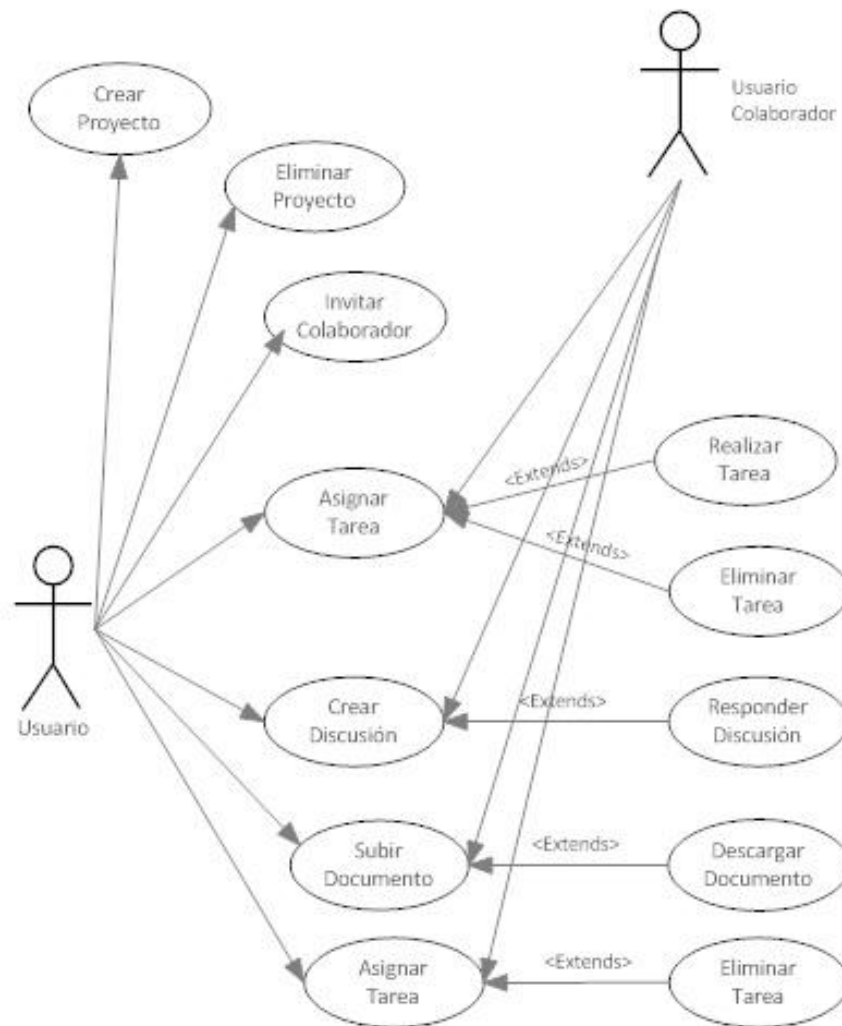
*Figura 7 Compartir Archivos*



- **Discusión Proyecto.**

Los colaboradores del proyecto podrán intercambiar dudas, sugerencias. Será un acceso exclusivo de los colaboradores del proyecto. Tan solo habrá un apartado de discusiones por proyecto, pero con tantas discusiones independientes entre sí como los componentes del proyecto necesiten crear.

Caso de Uso de la sección Proyecto:



*Figura 8 Caso de Uso Proyecto*

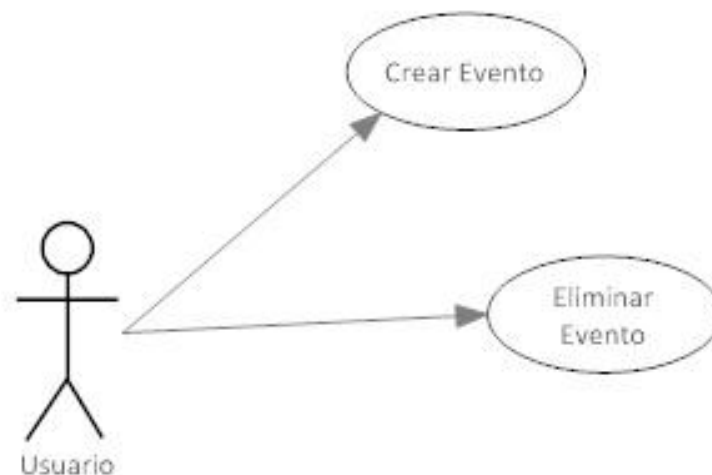
## Agenda

Apartado que permite, a modo de anotación, recordar fechas con eventos del usuario. Con un listado de todos los eventos que el usuario haya apuntado.

- Creación y eliminación evento.

Cada usuario tendrá la opción de añadir eventos a su agenda. Un evento estará compuesto del título de éste, la descripción más detallada del evento y la fecha marcada por el usuario de cuando cumple el evento.

Caso de Uso de la sección Agenda:



*Figura 9 Caso de Uso Agenda*

## Mensajes

Servicio de mensajería dentro de la web. Permite comunicarse con otros miembros de la aplicación sin necesidad de que estén en un proyecto.

- Creación Mensaje.

Formulario donde recoge Correo electrónico/Nombre Usuario al cual se le quiere enviar el mensaje, asunto y cuerpo del mensaje.

- Responder Mensaje.

Cuando se reciba un mensaje y se lea, habrá la opción de responder el mensaje poniendo al remitente como destinatario y utilizando mismo asunto de mensaje.

- Verificación usuario/correo electrónico.

Comprobación automática de que el usuario o correo electrónico que se introduce en el destinatario se encuentra registrado en la página web.

Caso de Uso de la sección Mensajes:

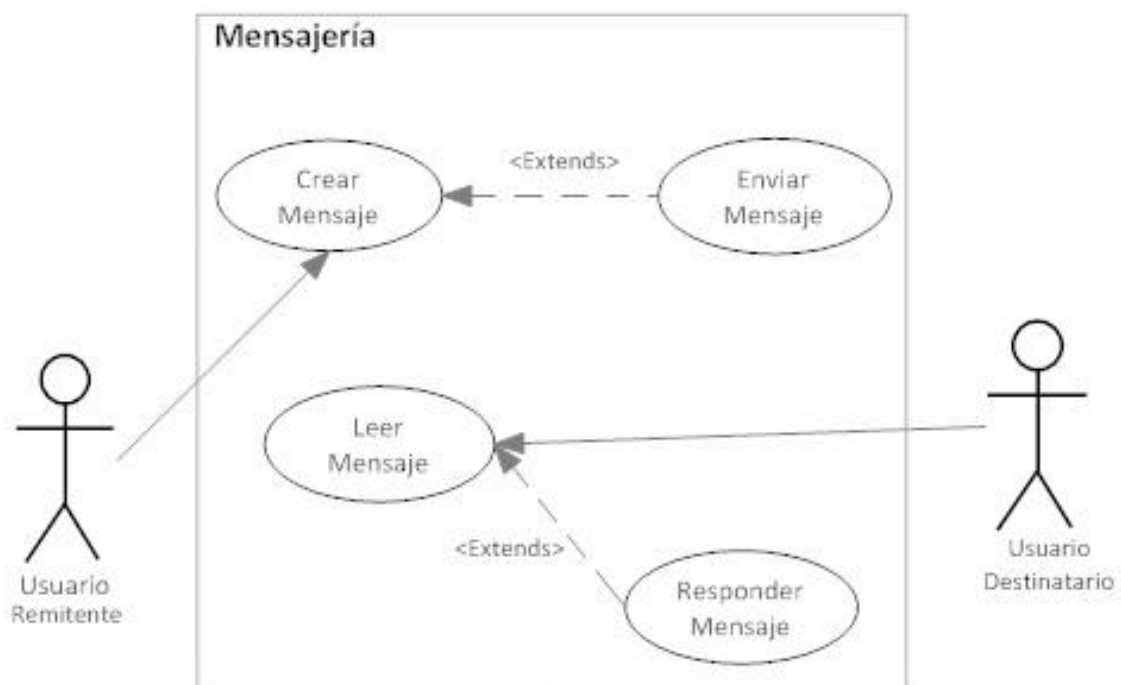


Figura 10 Caso de Uso Mensajería

### 3.3. Requisitos no funcionales

También existen un conjunto de requisitos relacionados con el entorno en el que ha de funcionar la aplicación y las características necesarias para cumplir los objetivos. Estos requisitos son los que ayudarán a hacer un mejor uso de la aplicación.

- **Facilidad de uso**

El sistema debe ser fácil de utilizar para cualquier usuario de la aplicación. Esto se consigue con una interfaz amigable y una buena distribución de las páginas de la web, siendo cualquier rincón de la aplicación accesible en no más de tres clics.

- **Navegador**

La aplicación se optimizará para utilizar en el navegador Google Chrome.

Eligiendo y centrado el desarrollo y la correcta optimización en este navegador, ya que, a día de hoy es el más utilizado y, sin duda, el navegador con más proyección y así lo demuestra su crecimiento. Con un criterio similar se ha rechazado Internet Explorer que se puede observar su claro decrecimiento

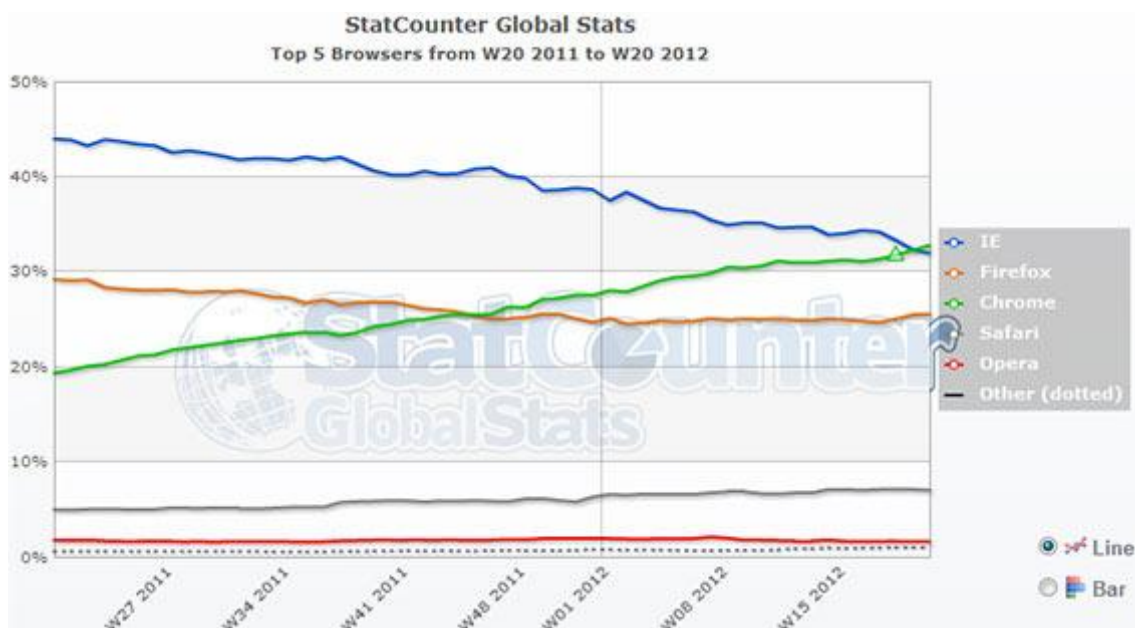


Figura 11 Estadística uso de navegadores

- **Validación de la información**

El sistema debe validar automáticamente la información contenida en los formularios de ingreso. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como la obligatoriedad de los campos o longitud permitida por el campo.

- **Escalabilidad**

El sistema debe ser construido de manera que las nuevas funcionalidades puedan implementarse afectando el código en la menor manera de lo posible.

### 3.4. Perfiles de usuarios

Para acceder a la aplicación se ha de estar registrado mediante un formulario que aparece en la pantalla de acceso a la web.

Sin dicho registro no se puede ver ningún tipo de contenido del web. Una vez registrado se tendrá total libertad para entrar en todas las secciones.

A continuación se detalla los tipos de perfiles que se existen y que permisos tiene cada uno de ellos.

Responsabilidad	Descripción
Usuario No Registrado	Sólo podrá ver la parte informativa de la web y su página de iniciar sesión.
Alumno	<p>Previamente registrado podrá hacer uso de las diferentes secciones y servicios de la página web.</p> <p>Permisos:</p> <ul style="list-style-type: none"><li>• Modificación y desactivación de la cuenta.</li><li>• Creación de proyectos con miembros de la plataforma.</li><li>• Participación en discusiones.</li><li>• Agenda.</li><li>• Almacenamiento de documentos.</li></ul>

*Tabla 9 Perfil de los usuarios de la web*

### 3.5. Marco tecnológico

Para determinar cómo, y con qué, realizar la implementación se realiza un análisis tecnológico. Analizando diferentes lenguajes y herramientas con las que se puede desarrollar una aplicación web.

Existen numerosos lenguajes para el desarrollo de una web, a continuación veremos algunos de ellos.

#### PHP

PHP es un lenguaje de programación interpretado originalmente para la creación de web dinámicas. Se utiliza principalmente para la interpretación del lado del servidor y por este motivo al navegador web ya le llega la página en formato HTML. PHP no necesita ser compilada para su ejecución. Para su funcionamiento se necesita tener instalado un servidor Web Apache o IIS con librerías de PHP.



*Figura 12 Facebook o Wikipedia están desarrollado en PHP*

Ejemplo de código PHP:

```
<?php  
  
    echo "Hola Mundo";  
  
?>
```

Además PHP cuenta con estructuras ya definidas para organizar y trabajar con los datos. Estas estructuras con las que trabajas se llaman **frameworks**.

Existen numerosos frameworks que hace que desarrollar aplicaciones con PHP sea más sencillo y rápido.

Dos de los frameworks libres más conocidos son **Codeigniter** y **CakePHP**.

CakePHP, facilita el desarrollo de las aplicaciones web, basándose sus conceptos en otro lenguaje Ruby On Rails. Se distribuye bajo la licencia MIT.

Codeigniter, es un entorno de desarrollo abierto que permite crear webs dinámicas ofreciendo numerosas librerías para facilitar la creación al usuario.

Cabe destacar que Incluso Rasmus Lerdorft, creador de PHP, expresó que le gustaba Codeigniter “porque es rápido, ligero y parece poco un entorno”.

Cuenta con un gran respaldo en la web con cantidad de tutoriales, vídeo-tutoriales y demás ejemplos de cómo hacer diferentes módulos (como Calendarios o páginas de registro).

Además ofrece soporte de Ajax (conjunto de tecnologías que permiten hacer acciones o actualizar una página web sin haberla cargado otra vez.), aunque mínimo, el suficiente para desarrollar numerosas aplicaciones vistosas.

## **Python**

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.



*Figura 13 Python es usado por Google en aplicaciones como Youtube*

Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.

Ejemplo de código de Python:

```
print "Hola mundo";
```

## **Ruby On Rails**

Es un lenguaje interpretado de muy alto nivel y orientado a objetos. Desarrollado en el 1993 por el programador japonés Yukihiro “Matz” Matsumoto. Su sintaxis está inspirada en Python y Perl.

Ruby es un lenguaje dinámico para una programación orientada a objetos rápida y sencilla.

Permite desarrollar soluciones a bajo coste y es multiplataforma.

## **HTML**

HTML es un lenguaje que su uso es, esencialmente, para la elaboración de páginas web. Este lenguaje no solo se define en la estructura, también en la semántica. Aunque al usuario se le puedan mostrar unos datos de forma idéntica utilizando etiquetas diferentes, cada etiqueta lleva un concepto asociado que ayuda a que los usuarios no-humanos, puedan entender el contenido e interpretarlo debidamente.

Este lenguaje se basa en la definición de etiquetas, escritas entre símbolos “<” y “>”. Las etiquetas pueden contener diferentes atributos para distinguirlos, definir el comportamiento o darle un cierto formato. Además, también pueden contener texto y otras etiquetas.

Estas etiquetas para saber donde comienzan y donde acaban se tienen que escribir utilizando una etiqueta de apertura y otra de cierre, quedando de la siguiente manera “<etiqueta>Texto</etiqueta>”.



## JavaScript

JavaScript al ser un lenguaje que no necesita compilación es utilizado principalmente en páginas web. Aunque su nombre, JavaScript, no proviene del lenguaje de programación Java, los dos comparten una sintaxis muy parecida inspirada en el lenguaje C.

JavaScript se ejecuta en el mismo momento que las sentencias se van descargando juntamente con el código HTML. Por lo tanto, es un lenguaje interpretado por el navegador y no puede acceder a recursos si no se hace con la ayuda de otros lenguajes.

Es un lenguaje muy utilizado en programación web ya que, permite potenciar algunos aspectos en una página Web como la validación de formularios, o aspectos visuales como acciones al mover el ratón sobre la página.



*Figura 14 JavaScript fue desarrollado por Brendan Eich de Netscape con el nombre de Mocha, el cuál fue renombrado más tarde a LiveScript*

## Diseño interfaz web

Para crear el diseño de la interfaz, lo que verán los usuarios, las opciones son más limitadas. Se puede contar con las herramientas que disponen alguno de los lenguajes anteriores, como HTML, donde estilo a las letras y a las pantallas de fondo. O, se puede complementar las opciones del HTML integrando crear hojas de estilo mediante CSS. CSS, se trata de un lenguaje formal utilizado para definir la presentación de un documento estructurado en HTML. El “World Wide Web Consortium” es el encargado

de definir las especificaciones de las hojas de estilo, que utilizaran de estándar los navegadores.

El lenguaje CSS se puede adjuntar en ficheros HTML en el mismo fichero como código añadido, con la etiqueta <Style> </Style> o con la importación de los estilos desde un fichero .css .Una misma página puede disponer de diferentes hojas de estilo, después las etiquetas o componentes se la páginas utilizaran unos o otros.

### **Sistema gestor de base de datos**

Se necesita una base de datos para almacenar toda la información con la que se trate en la aplicación. Hay numerosas gestores de bases de datos libres pero se han analizado dos posibles, MySQL y PostgreSQL.

**MySQL** es uno de los gestores de base de datos más usados del mundo ya que es muy bueno para su uso web. Se puede conectar con diversos sistemas Como Windows, Linux o Mac OS. Además, tiene un gran soporte por los propios profesionales que de MySQL.



*Figura 15 Existen más de seis millones de copias de MySQL funcionando en la actualidad.*

**PostgreSQL**, algunos la llaman la verdadera base de datos libre, otros dicen que es el único rival serio que tiene Oracle. La comunidad que la desarrolla se denomina PGDG (PostgreSQL Global Development Group) y trabajan de forma desinteresada en la ampliación de esta plataforma. Pero debido a su reducida y fragmentada comunidad muchos anteponen MySQL.

## 4. DISEÑO

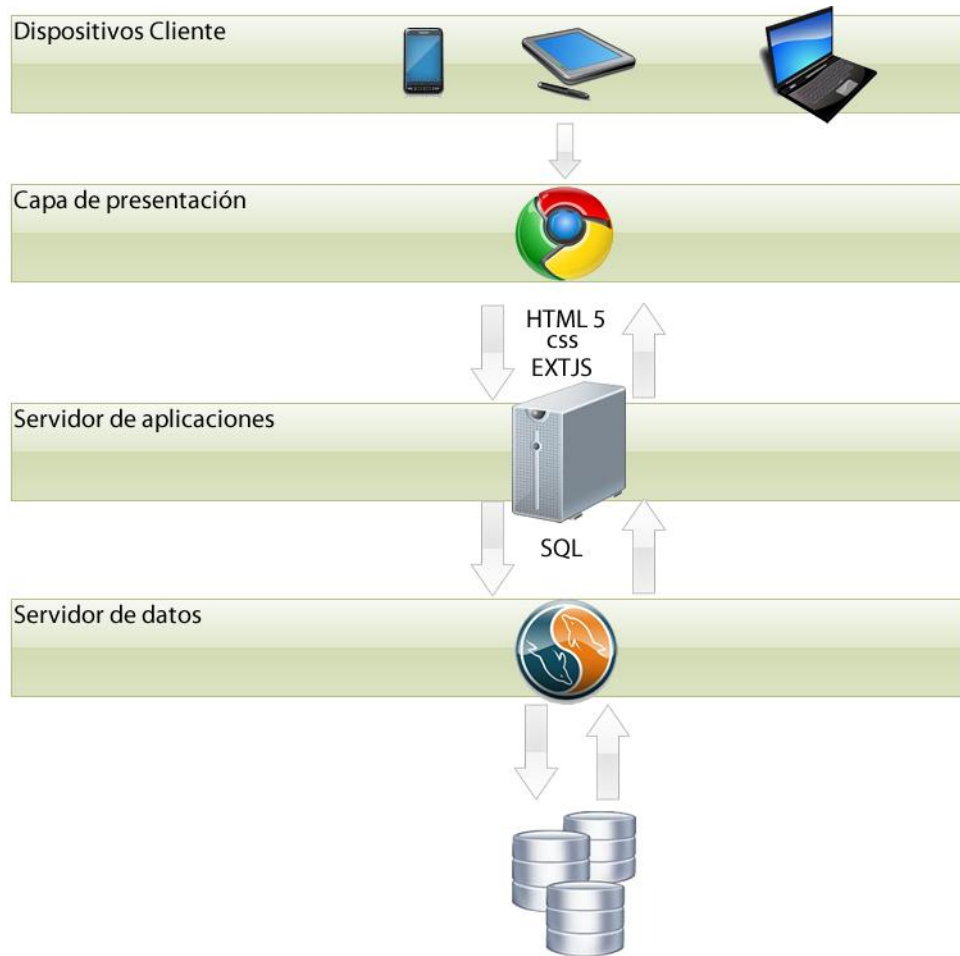
### 4.1. Introducción

Según los requisitos funcionales anteriormente descritos, se definirán los módulos de los que está compuesta la aplicación, así como el diseño de la base de datos y el modelo Entidad Relación en el que se ha basado.

### 4.2. Arquitectura de la aplicación

La arquitectura de la aplicación, estará dividida fundamentalmente en la siguientes tres capas:

- **Interfaz de Usuario**, es la capa que se mostrará al usuario. El usuario navegará por esta capa, y hará peticiones a través de ella.
- **Instrucciones**, es donde se tramitan todas las peticiones que hace el usuario en la capa anterior y conecta con la base de datos. En el siguiente capítulo, Implementación, se verá con más detalle que y como está implementada esta capa.
- **Base de datos**, almacena los datos relacionados con la aplicación web y los usuarios dados de alta en ella.



*Figura 16 Organización Máquina Usuario – Aplicación*

### 4.3. Diagrama entidad/relación

Una vez analizada la aplicación y todas sus funcionalidades que tendrá, se procede a especificar que información deberá ser guardada y cuál es la mejor manera de hacerlo. Para ello se desarrolla el diagrama de entidad relación.

A continuación se pueden distinguir las principales entidades del diagrama entidad/relación:

- Usuario
- Mensajes
- Agenda
- Proyecto
- Colaboradores
- Archivos
- Enlaces
- Discusiones
- Tareas

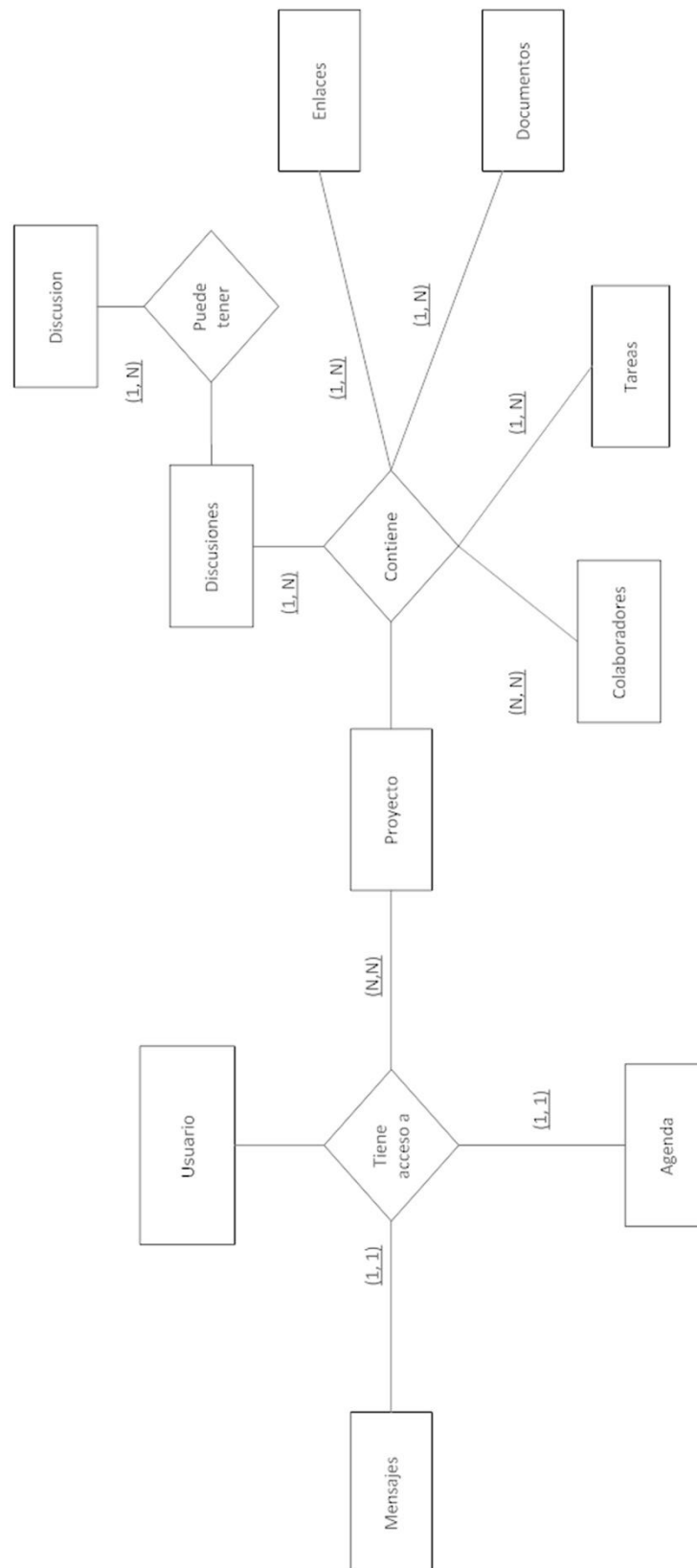


Figura 17 Modelo entidad/relación

#### 4.4. Diseño de la base de datos

La base de datos que a continuación se define contendrá la información de los usuarios. Tanto sus datos personales, como los archivos que suban de modo que puedan utilizarlo como pequeño repositorio.

Las principales tablas de la aplicación contienen la información relativa a los usuarios de la aplicación y sus proyectos. A continuación una breve explicación de cada una de las tablas que forman la base de datos de la aplicación y el modelo entidad relación con sus respectivos atributos de cada entidad.

Para hacer más claro el entendimiento de cada tabla, se han agrupado según el módulo de la aplicación a la que va acorde la tabla.

##### Usuarios

**Usuario:** La tabla del usuario. Contiene información de los usuarios de la aplicación. La información que contiene se puede clasificar en:

- *Datos Personales (Nom, Apellidos, Email, NombreUsuario)*
- *Datos de control (FechaUltimoAcceso, FechaCreacion, Contraseña)*
- *Configuración Aplicación (Activo, idUsuario).*

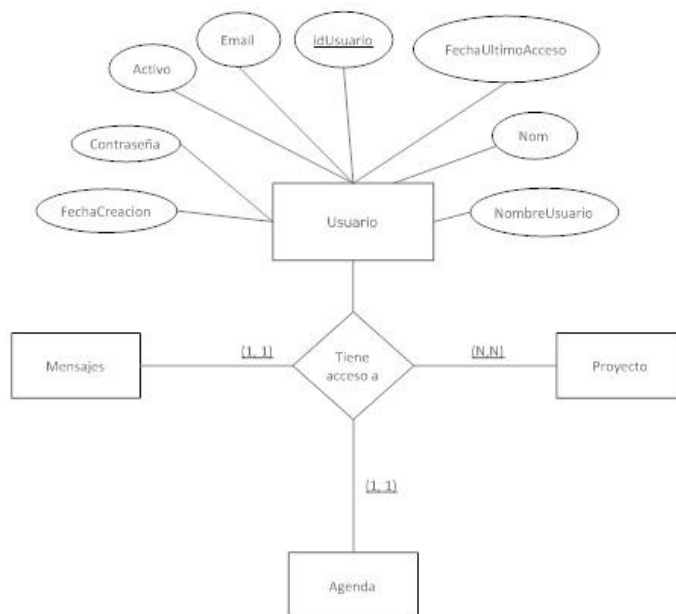


Figura 18 Modelo E/R Usuarios con entidades relacionadas directamente

## Mensajería

**Mensaje:** Tabla principal de mensajería. Contiene toda la información del mensaje (*Asunto, Cuerpo, Fecha Creación*).

Además contiene datos de la configuración de la aplicación, relativos a *Ubicación* (*Recibidos, Enviados, Eliminados*), *idUsuarioCreador*, *idMensaje* e *idProyecto*.

**Mensaje\_Destinatario:** Tabla auxiliar que contiene el destinatario del mensaje, el campo booleano que registra si un mensaje ha sido leído o no y la ubicación del mensaje.

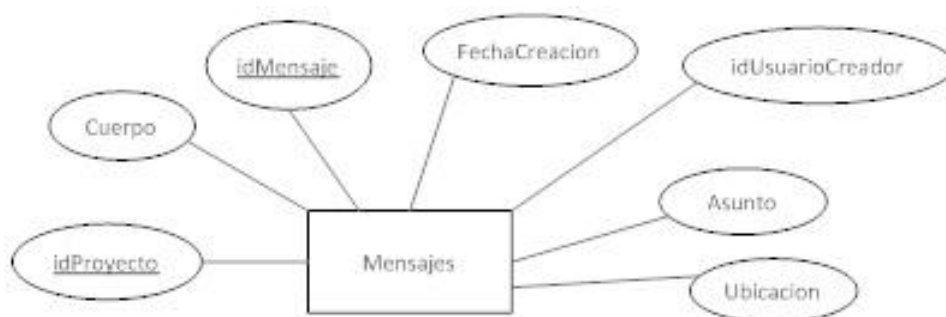


Figura 19 Modelo E/R Mensajes

## Agenda

**Evento:** El conjunto de eventos que se registran en esta tabla forman la agenda del usuario.

Contiene los datos que el usuario le interesa saber (*Nom, DataEvent, Descripcion*) y los campos que el sistema rellenará para su correcta funcionalidad (*idEvento* *idUsuarioCreador*).

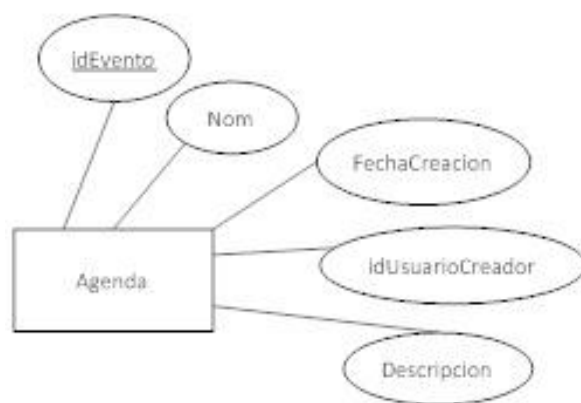


Figura 20 Modelo E/R Agenda

## Proyecto, Colaboradores y Tareas

**Proyecto:** Contiene todos los datos referentes a la información del proyecto (*Nombre, Descripción, FechaFinalizacion, FechaCreacion*). Así como identifica el creador del proyecto.

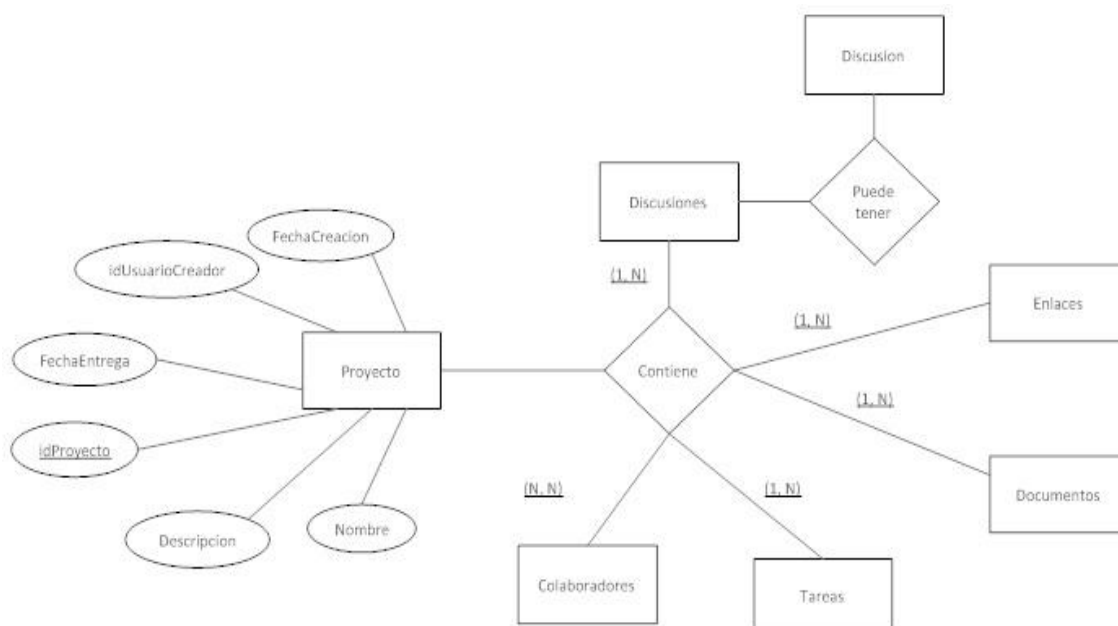


Figura 21 Modelo E/R Proyecto con entidades relacionadas directamente



**Proyecto\_colaborador:** Tabla auxiliar de Proyecto. Se define a los colaboradores de un proyecto, y la fecha de cuando se ha unido al proyecto.

**Proyecto\_invitación:** Tabla encargada de las invitaciones. Contiene la información de si el colaborador ha aceptado la petición de unirse al proyecto.



Figura 22 Modelo E/R Colaboradores

**Tarea:** Referente a Tareas dentro del apartado del módulo de Proyectos. Tiene la información de la tarea (*Descripcion*, *FechaCreacion*, *FechaFinalizacion*, *Realizado*). También define que usuario se le ha asignado la tarea y quien la ha creado.



Figura 23 Modelo E/R Tareas

## Discusiones

**Discusiones:** El conjunto de registros almacenados en esta tabla por los diferentes usuarios forman la sección discusiones.

Contiene campos de información como *título*, *cuerpo*. Y otros campos referentes a la información de las discusiones como el *idProyecto* al que hace referencia el conjunto de discusiones.

**Discussion:** Tabla que completa discusiones. Almacena las respuestas dadas en las discusiones. Almacenando el usuario que ha expuesto su respuesta y la fecha de esta respuesta.

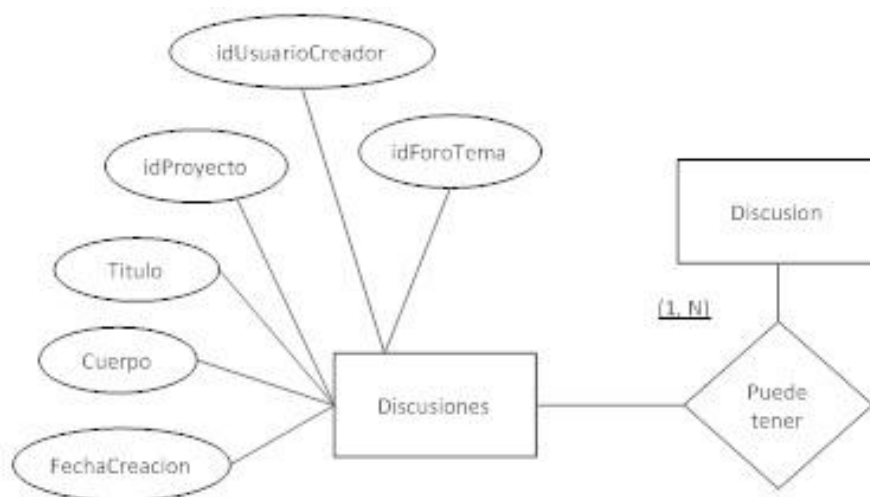


Figura 24 Modelo E/R Discusiones

## Documentación

**Documento:** Tabla que encarga de almacenar la referencia del documento que el usuario haya subido a la aplicación para compartir con sus colaboradores. En esta tabla nada es modificable directamente por el usuario, ya que el nombre de fichero recoge por defecto el que es del archivo.

El resto de campos son para la configuración del sistema como *idDocumento*, *idProyecto*, *idUsuario*, *FechaCreacion*.

**Enlace:** Recoge un enlace externo anotado por los usuarios, con información que les pueda interesar a los miembros de una web ajena.

Los campos que componen esta tabla son *idEnlace*, *idProyecto*, *Url*, *Descripción*, *idUsuarioCreador*, *FechaCreacion*.

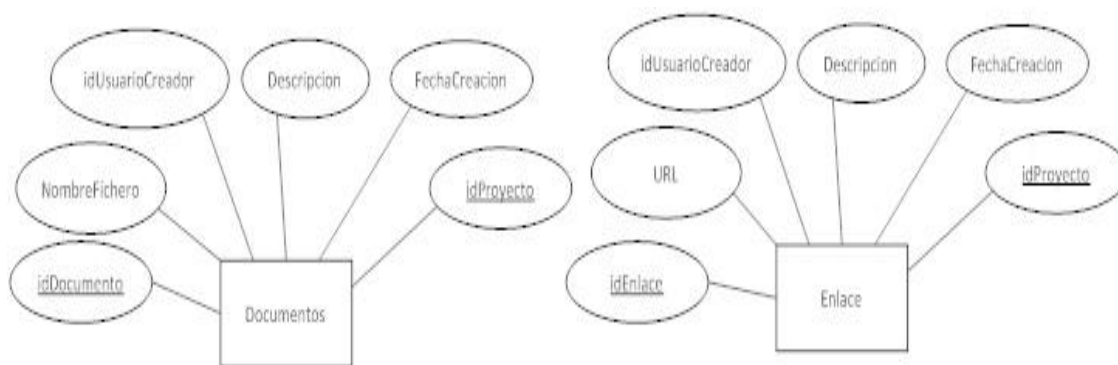


Figura 25 Modelo E/R Documentación

## Diseño base de datos

A continuación se muestra el diseño de la base de datos, utilizado para crear la base de datos del proyecto, en la cual se pueden ver las relaciones hasta ahora descritas:

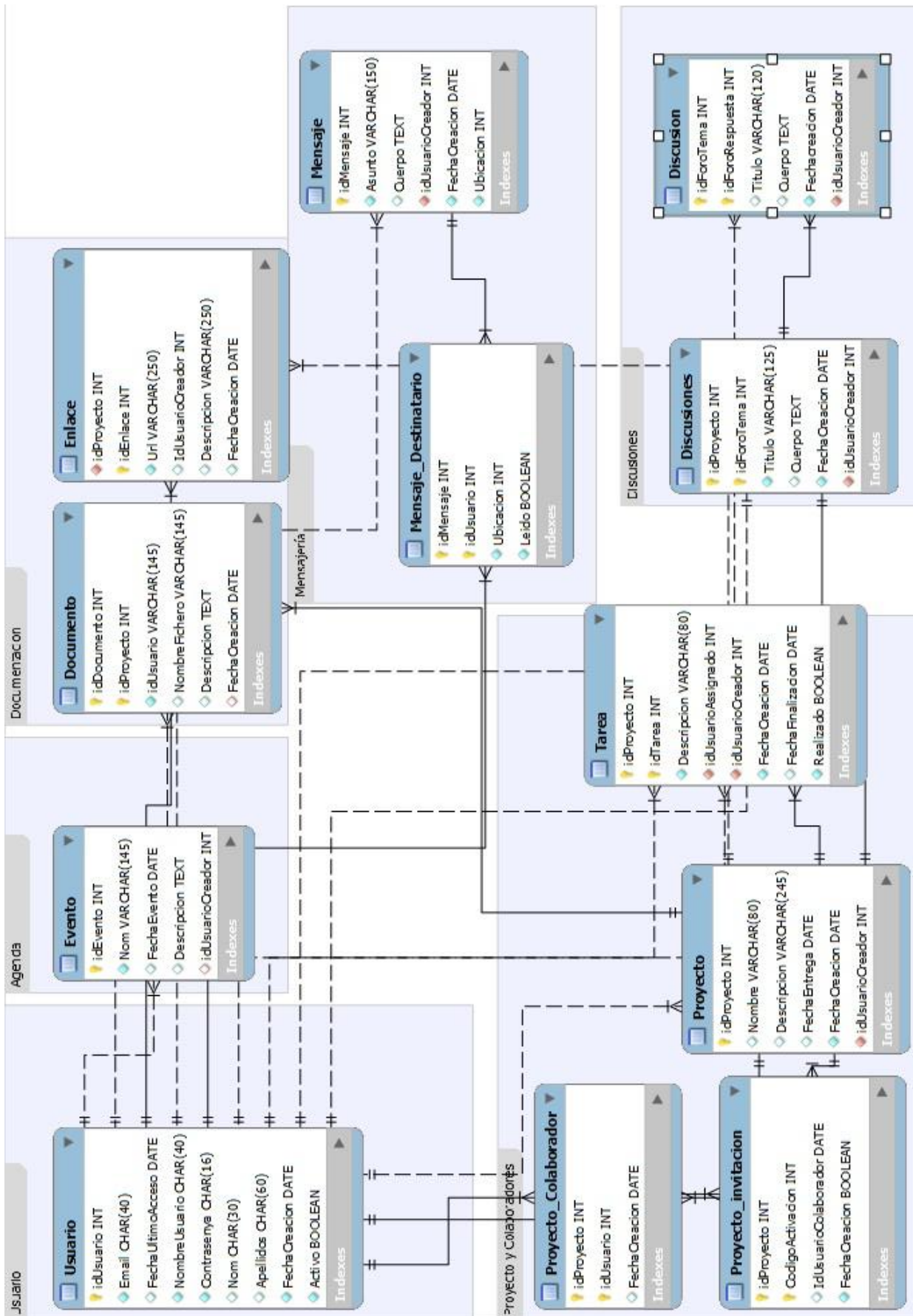


Figura 26 Diagrama de base de datos

## 4.5. Tecnología a utilizar

Después de analizar las opciones tecnológicas disponibles, 3.5 Marco tecnológico, se ha decidido finalmente utilizar PHP con el soporte del framework CodeIgniter. Debido al soporte web que dispone y con la simplicidad que supone este framework respecto a otros. Además, CodeIgniter dispone de funciones Ajax con las que hacer poder hacer páginas más rápidas, sin necesidad de grandes esperas en la carga, y JavaScript para hacer funciones visualmente más atractivas.

Por lo que respecta al gestor de base de datos se ha elegido MySQL por el soporte en la comunidad y la fiabilidad que da este gestor con el tratamiento de los datos.

Finalmente, se completará la Interfaz con HTML y hojas de estilo CSS, por su gran abanico de opciones en cuanto diseño.

## 4.6. Patrón modelo-vista-controlador

No es un lenguaje de programación sino que se trata de un patrón, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. En el framework CodeIgniter es utilizado frecuentemente.

Actualmente se está empezando a utilizar cada vez más el patrón modelo-vista-controlador (MVC) por varias razones que seguidamente explicaré.

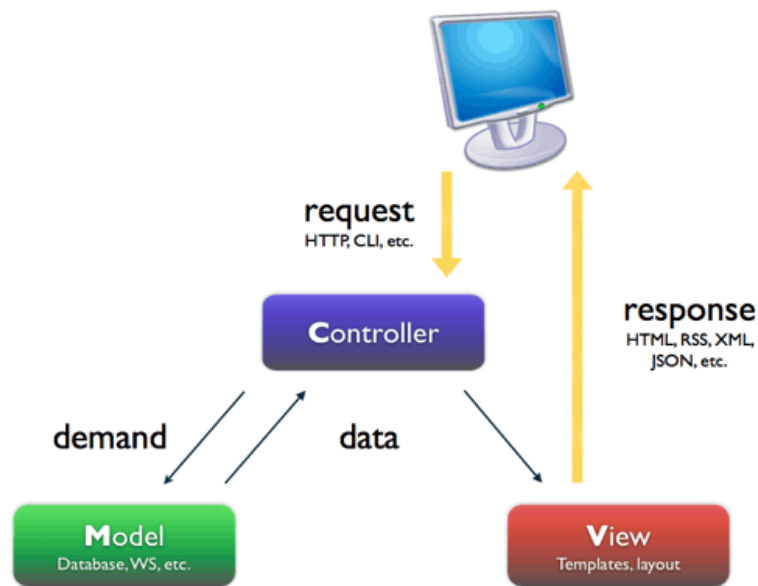
El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, interfaz de usuario y la lógica de control en tres componentes diferentes. Tal y como indica su nombre, MVC, estos tres componentes son el modelo, la vista y el controlador:

- **Modelo:** Representa las reglas de la aplicación, los datos de la aplicación. Entonces, aquí donde tendremos toda la parte lógica y el acceso a los datos.
- **Vista:** Representa la presentación de la aplicación, la interfaz del usuario. Contiene básicamente el código que se envía al navegador, es decir, al código

HTML y también puede tener código del servidor asociado, siempre y cuando se dedique a presentación.

- **Controlador:** Actúa de intermediario entre el usuario, el modelo y las vistas. Recoge las peticiones del usuario, interactúa con el modelo y decide que vista ha de mostrar los datos. Es el *motor* del patrón.

En la siguiente imagen se puede ver con claridad el funcionamiento del patrón MVC:



**Figura 27** Esquema modelo-vista-controlador

Vemos como la vista muestra al usuario las opciones por pantalla, y éste realiza una acción en la vista que es captado por el controlador y éste a su vez, como motor de este patrón, se encarga realizar las acciones necesarias pidiendo al modelo los datos requeridos y tratando estos datos para devolvérselos a la vista y que los mostrará por pantalla.

## 4.7. Interfaz usuario

La interfaz gráfica ha de ser muy intuitiva y fácil de utilizar para el usuario final, tal y como nos marca uno de los requisitos del sistema. Para ello se ha evitado numerosas pantallas muy diferentes entre ellas, y se ha optado por pantallas que cambien el mínimo de elementos posibles con el objetivo que el usuario se habitúe fácilmente a esta interfaz.

### Pantalla inicio sesión/registro

Es una pantalla básica, donde el usuario podrá ver el logotipo del proyecto, ProyGes, centrado a la izquierda y centrado a la derecha el formulario que pide las credenciales necesarias para entrar en su espacio de la aplicación.

Bajo este formulario se encuentra el link que permite el registro de la aplicación. Cargando este formulario en lugar en el que se encuentra el de inicio de sesión.



*Figura 28 Esquema del diseño inicio sesión*

## Pantalla principal

Una vez registrado accedes a la aplicación. Desde esta pantalla se puede ir a cualquier sección de la aplicación.

Esta pantalla está dividida en 5 partes.

- Encabezado: Contiene el logotipo y nombre de la empresa
- Barra Lateral: Contiene las principales secciones de la aplicación (*Mi Cuenta, Proyecto, Agenda, Mensajería*).
- Barra de navegación: Se trata de una barra para facilitar la accesibilidad del usuario a ciertas funciones de la web, como pueden ser unos accesos directos a las opciones “*Creación de proyecto*”, “*Nuevo mensaje*” y “*Nueva Tarea*”.
- Contenido: Muestra la información que queremos ver. Es donde fluye toda la información de la aplicación.
- Pie de página: Contiene el nombre del proyecto y el año de creación.

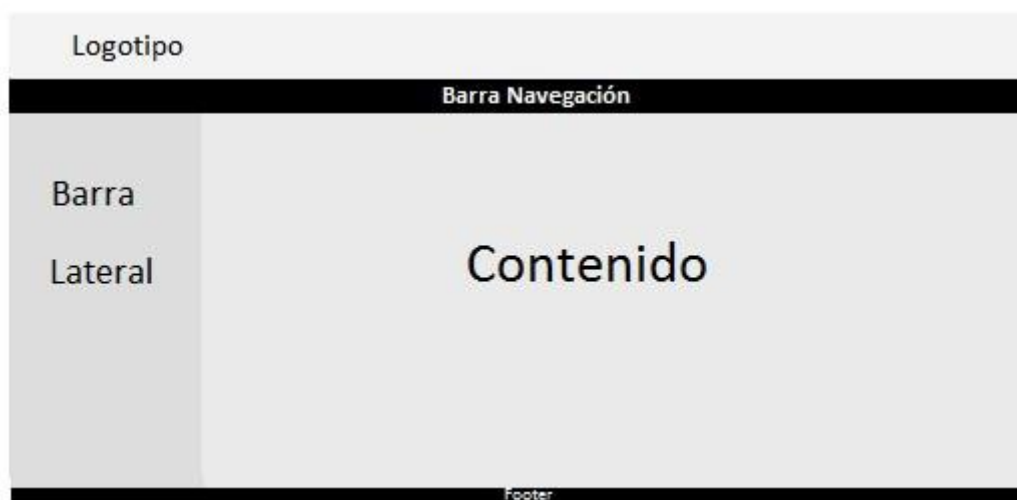


Figura 29 Esquema del diseño web



#### 4.8. Planificación detallada

Una vez analizado el proyecto y diseñado se puede hacer una planificación de tareas más detallada de la que se realiza al inicio de éste en el Estudio de viabilidad.

Nombre de tarea
<b>Planificación</b>
Estudio Viabilidad
Aprobación Estudio de Viabilidad
<b>Análisis de la aplicación</b>
Análisis de Requisitos
Análisis de Datos (Base de datos)
Documentación Análisis
Aprobación del análisis
<b>Diseño de la aplicación</b>
Diseño de la Base de Datos
Diseño de la interfaz
Diseño de las pruebas
Documentación del diseño
Aprobación del Diseño
<b>Desarrollo y pruebas</b>
<b>Desarrollo de la aplicación</b>
Preparación del entorno de desarrollo
Configuración de la base de datos
<b>Desarrollo de las funcionalidades de la aplicación</b>
<b>Desarrollo Main</b>
Desarrollo inicio usuario
Desarrollo codificación contraseña
<b>Desarrollo Mi panel</b>
<b>Desarrollo Usuario</b>
Desarrollo Registro Usuario
Desarrollo edición datos usuario
<b>Desarrollo Agenda</b>
Implementación calendario fecha
<b>Desarrollo Discusiones</b>
<b>Desarrollo Proyecto</b>
Unión de las funcionalidades

Desarrollo Inicio proyecto
<b>Desarrollo Colaboradores</b>
<b>Desarrollo Mensajería</b>
Desarrollo de envío de mensajes
Desarrollo de recibo de mensajes
Implementación sistema de bandejas
<b>Desarrollo de la interfaz del usuario</b>
Desarrollo de la Barra Lateral
Desarrollo del Contenido
Desarrollo de la Barra de Navegación
Desarrollo del Encabezado
Desarrollo del Pie de página
Desarrollo de la página de Inicio
<b>Test y pruebas</b>
Pruebas Unitarias
Pruebas de integración
Documentación del Desarrollo y Test
Aprobación del Desarrollo y Test
<b>Generación del documento (memoria del proyecto)</b>
<b>Cierre del proyecto</b>
<b>Defensa del Proyecto</b>

Tabla 10 Planificación detallada de las tareas

*Desarrollo y pruebas* ha sido la fase que más se ha detallado debido a que es la fase que más dedicación ha tenido y debido a que la clase de tareas de las que consta permiten una mayor especificación.

Dentro del *desarrollo de la aplicación* se ha añadido la *configuración de la base de datos* debido al tiempo que se le ha dedicado para adaptar la base de datos a la tecnología elegida para el desarrollo del proyecto.

Finalmente, como se muestra en la Tabla 10 Planificación detallada de las tareas, en la fase *desarrollo de las funcionalidades de la aplicación* se ha detallado cada tarea que las que consta esta fase a un nivel más cercano para saber de que está compuesta cada tarea correspondiente a esta fase.

## 5. IMPLEMENTACIÓN

### 5.1. Introducción

Este capítulo describe el resultado final de la aplicación.

Una vez realizado el análisis de la aplicación, se diseñó la estructura que debería tener el sistema para lograr los objetivos definidos, y el diseño de la apariencia a grandes rasgos.

Tan importante como un buen análisis y diseño es importante la estructuración de los ficheros de manera coherente en el servidor para facilitar el estilo de codificación.

A continuación, en este apartado hablaremos de cómo han sido estructurado los ficheros y el resultado final del diseño.

### 5.2. Herramientas a utilizar

Una vez elegidos los lenguajes con los que desarrollar la aplicación. Se ha realizado una selección de plataformas con los que tratas los diferentes lenguajes y así implementar el proyecto.

#### PhpMyAdmin

PhpMyAdmin es un software libre escrito en PHP con la intención de administrar MySQL mediante páginas web.

Actualmente esta herramienta puede crear y eliminar bases de datos. Crear, eliminar y modificar las tablas. Borrar, editar y añadir campos a las tablas. Ejecutar cualquier consulta SQL y administrar las claves primarias o foráneas de los campos.



*Figura 30 PhpMyAdmin ha sido traducido a 62 idiomas*

## **Apache**

Apache es un servidor HTTP de código libre. Es el encargado de construir y retornar las páginas web que soliciten los navegadores. Este servidor puede funcionar en diversas plataformas (Unix y Windows, entre otras). Fue desarrollado dentro del proyecto HTTP Server de la empresa Apache Software Foundation. Desde hace unos años, más del setenta por ciento de los servidores web de Internet tienen instalada Apache.

Este servidor HTTP tiene mensajes de error con mucha facilidad para ser configurados, y una base de datos de autenticación.

## **CodeLobster**

Es un editor de texto, gratuito, que soporta los lenguajes más comunes.

Ofrece un completo resaltado de la sintaxis, ayuda con los parámetros de cada función, resaltado de etiquetas y corchetes sin cerrar. Soporta código mixto (Una página HTML con código PHP en su interior).

Dispone de marcadores para acceder a las líneas más recorridas, colapso de bloques de código para una visualización más rápida, buscar y reemplazado avanzado.

## **5.3. Organización del código**

Como se ha explicado con anterioridad el código se ha desarrollado según el patrón Modelo-Vista-Controlador, dejando la organización del código muy definida. Pero debido a la cantidad de líneas de código de toda la aplicación se ha clasificado el código según las funcionalidades que se han ido comentando durante este documento.

La carpeta *controllers* contiene los controladores de la aplicación. El nombre del archivo hace referencia a la funcionalidad que corresponde, a excepción de *Proyecto.php*, que además engloba los controladores de tareas y colaboradores.

El archivo *main.php* contiene las funciones de inicio de sesión, registro y para recuperar las contraseña.

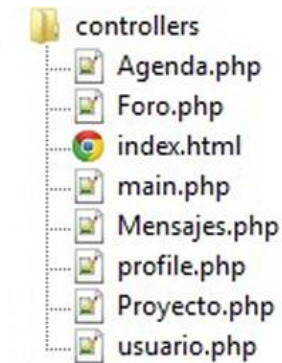


Figura 31 Ficheros de la carpeta *controllers*

La carpeta *models* guarda los ficheros del modelo. Los controladores se comunican con sus respectivos modelos y éstos recogen o insertan los datos en la base de datos.

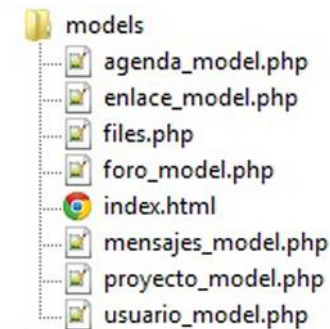
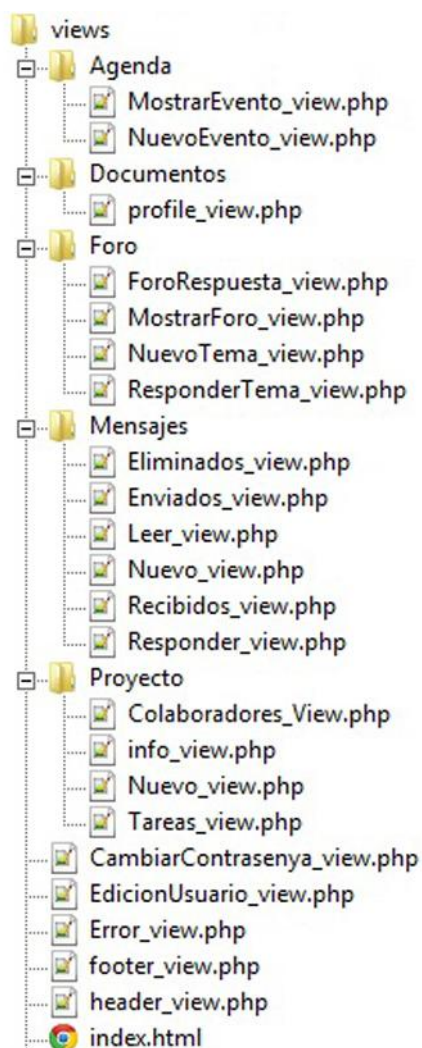


Figura 32 Fichero de la carpeta *models*

Finalmente, *views* contiene todas las vistas con las que los usuarios interactúan y mandan peticiones al controlador.



*Figura 33* Ficheros de la carpeta *views*

## 5.4. Interfaces

Tal y como se explicó brevemente en el punto *Interfaz de usuario*, se ha seguido el esquema descrito, de colores de fondos suaves y sobrios, plasmándolo en las imágenes que veremos a continuación.

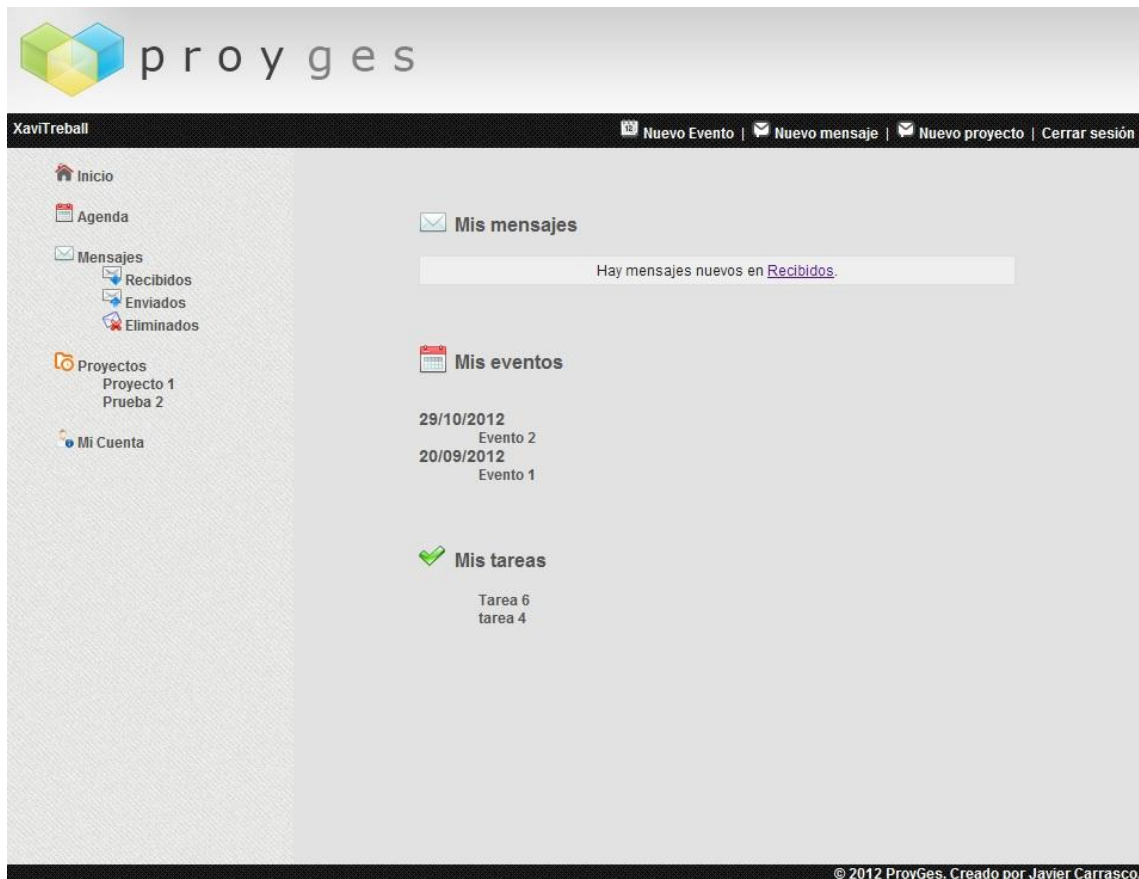


Figura 34 Interfaz del proyecto

## Interfaz de inicio sesión



The login interface features a logo on the left consisting of three overlapping cubes in green, yellow, and blue, with the text 'p r o y g e s' below it. On the right, there is a form with two input fields labeled 'Email' and 'Contraseña'. Below these fields are two links: '¿Olvidaste la contraseña?' and '¿Eres nuevo en ProyGes?'. A green 'Entrar' button is positioned to the right of the 'Contraseña' field.

Figura 35 Interfaz inicio sesión

La página más sencilla de la aplicación. Tan solo pide el email y contraseña para poder iniciar sesión. En caso de que la persona no esté registrada en la plataforma hay un enlace a la página de registro, y otro enlace en caso de haber olvidado la contraseña

## Interfaz de registro



The registration interface features the same logo as the login page. The form on the right includes input fields for 'Nombre', 'Apellidos', 'Correo electrónico', 'Nombre de usuario', and 'Contraseña'. There is a checkbox labeled 'Mostrar caracteres' and a green 'Registrate' button at the bottom right.

Figura 36 Interfaz registro

Otra pantalla similar a Inicio de sesión. Un formulario que pide algunos datos para identificarte dentro de la aplicación o para uso de la misma (como el correo electrónico como clave primaria).



## Interfaz de recordar contraseña



*Figura 37 Interfaz recuperar contraseña*

Esta pantalla permite introducir un correo electrónico. Una vez introducido haciendo clic en “Recordar” enviará un mensaje al correo de la persona con un link a la pantalla para cambiar la contraseña.

## Interfaz de cambio de contraseña



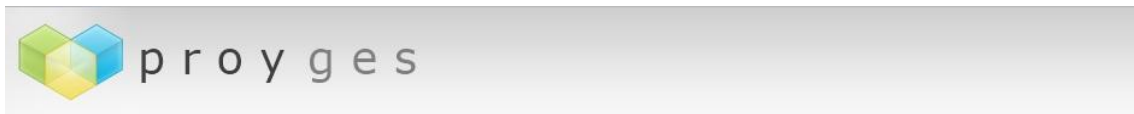
*Figura 38 Interfaz cambio contraseña*

El correo para cambiar contraseña contiene un enlace que abre la interfaz que se muestra. Esta interfaz permite cambiar la contraseña del correo electrónico de donde proceda el usuario.

## Pantallas principales

Las siguientes interfaces son en las 5 *secciones* las cuales está dividida la pantalla, y el usuario podrá añadir diferente información en alguna de ellas, siempre manteniendo la distribución clara:

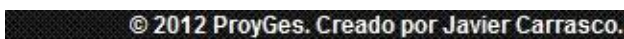
### Interfaz del encabezado



*Figura 39 Interfaz del encabezado*

Esta parte se mantiene igual independientemente del apartado en que nos encontremos. No se puede hacer nada. Muestra el Logotipo de la aplicación al lado del nombre. Los colores de fondo mantienen la estética de las pantallas de Inicio de sesión y registro.

### Interfaz del pie de página



*Figura 40 Interfaz del pie de página*

Como en el encabezado, sección sin nada más que el nombre del proyecto y el año de creación de éste. Cambiando el color de fondo siendo un negro rugoso como el de fondo.

## Interfaz de la barra de navegación



Figura 41 Interfaz de la barra de navegación

Pretende ser una barra de acceso rápido. En cualquier punto que nos encontremos de la página siempre estará a manos las opciones que ofrece (Crear un Proyecto, Crear un evento en la agenda y Crear una tarea) y la opción de Cerrar sesión para salir de la página.

## Interfaz de la barra lateral

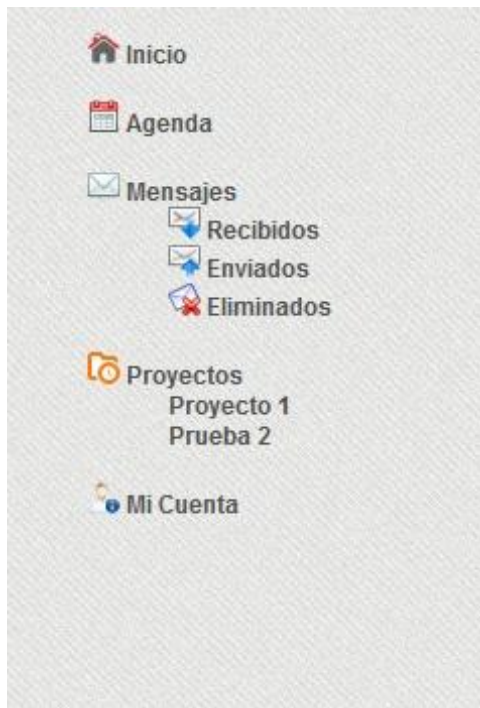


Figura 42 Interfaz barra lateral

En esta barra se podrá acceder a las funcionalidades independientes de Proyecto, exceptuando a los propios proyectos que tengamos activos.

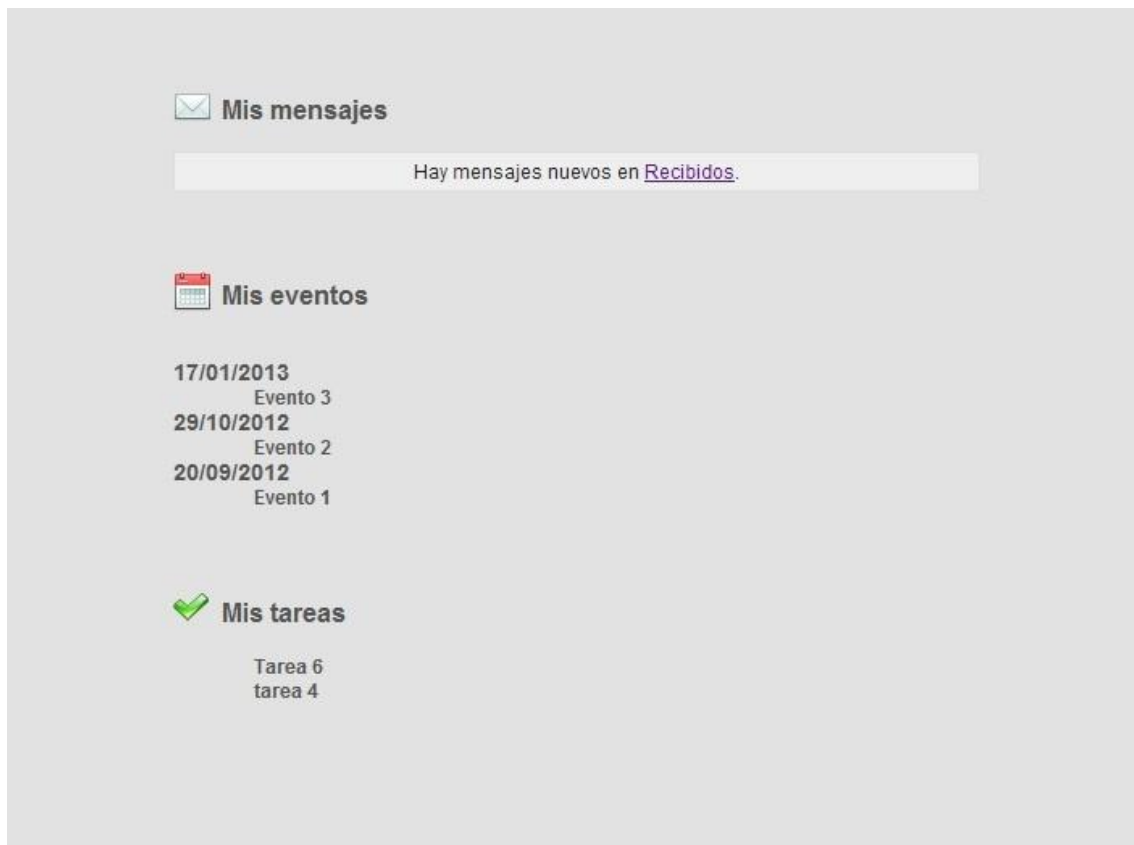
Las secciones a las que podrás acceder son Inicio, Agenda, las diferentes bandejas de Mensajería, Mi Cuenta y los Proyectos en curso.

Como muestra la imagen aparecerán en la barra lateral tantos proyectos como proyectos hayamos creado o hayamos sido invitado y aceptado.

## Interfaz del contenido

En esta capa de interfaces es donde residen la mayoría de vistas creadas en la *view*. Es la interfaz donde se refleja la mayor parte del contenido de la aplicación, desde los datos personales hasta las discusiones de proyecto. A continuación veremos cada una de estas vistas.

## Interfaz de inicio



*Figura 43 Interfaz mi panel*

Es la página de Inicio de la aplicación. Al iniciar sesión mostrará esta página. Contiene breve información de algunas secciones de la web.

- Si hay algún mensaje nuevo. En caso de haber alguno lo notificará con un enlace a la sección *Mensajes Recibidos*.
- Muestra los eventos anotados en la agenda. Tan solo la fecha anotada por el usuario y el título del evento o recordatorio.
- Muestra las tareas que hayan sido asignadas para el usuario.

### Interfaz de la sección Agenda

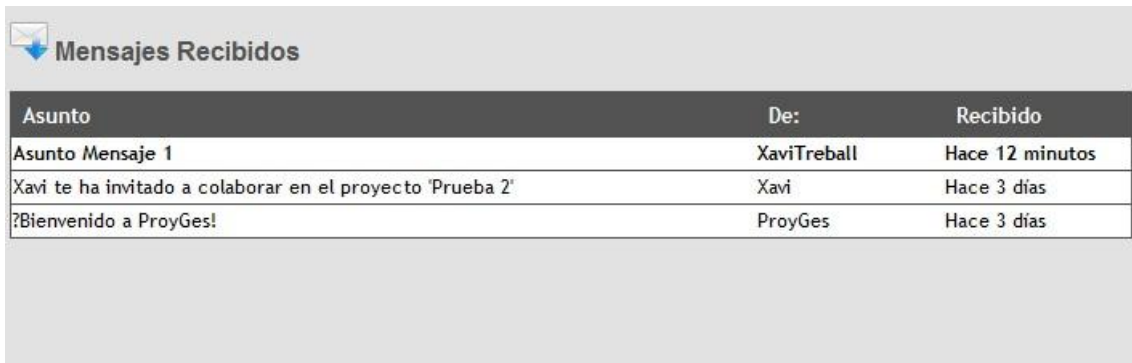


Figura 44 Interfaz agenda

En esta página muestra los eventos anotados en la agenda. Las fechas se muestran en un orden descendente siendo las fechas más lejanas en el tiempo las primeras en aparecer.

El símbolo en forma de "X" de color rojo, sirve para borrar el evento en cuestión.

## Interfaz de las bandejas de mensajes



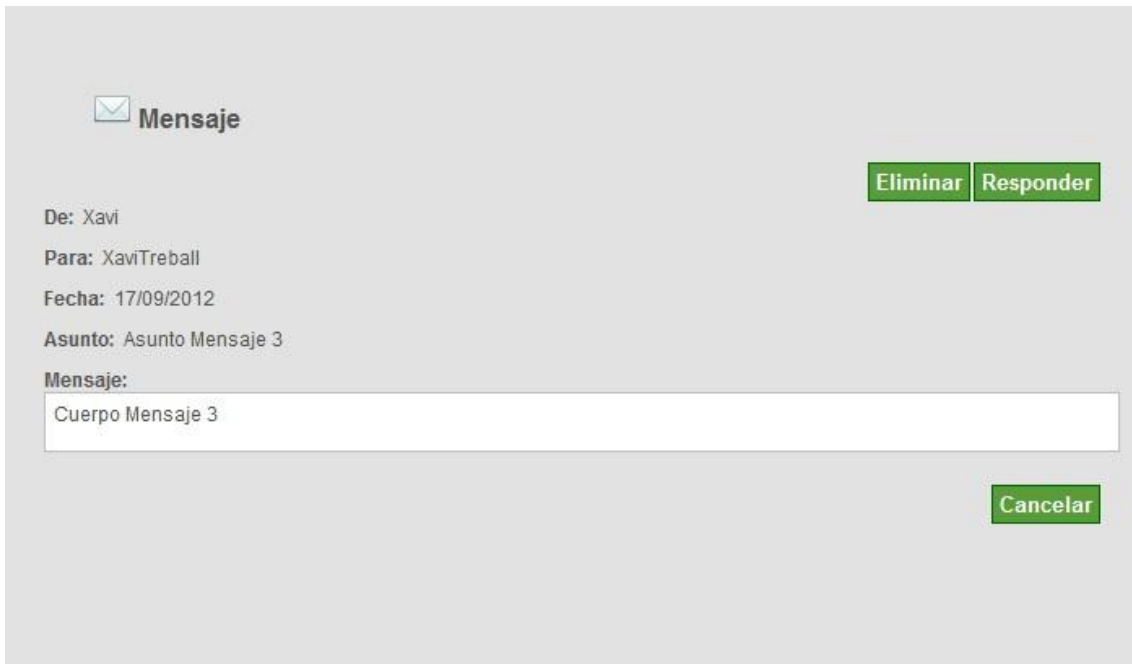
Asunto	De:	Recibido
Asunto Mensaje 1	XaviTrellall	Hace 12 minutos
Xavi te ha invitado a colaborar en el proyecto 'Prueba 2'	Xavi	Hace 3 días
¿Bienvenido a ProyGes!	ProyGes	Hace 3 días


Figura 45 Interfaz bandeja mensajes

Las tres bandejas (recibidos, enviados y eliminados) muestran un aspecto similar. Mantienen el aspecto de la página rompiéndolo para mostrar el fondo del mensaje en blanco para resaltar el mensaje seleccionado por el ratón.

Resaltando en negrita el título, el autor y la fecha, muestra los mensajes no leídos por el usuario.

## Interfaz del mensaje



 **Mensaje**

**De:** Xavi

**Para:** XaviTrellall

**Fecha:** 17/09/2012

**Asunto:** Asunto Mensaje 3

**Mensaje:**

Cuerpo Mensaje 3

**Eliminar** **Responder**

**Cancelar**

Figura 46 Interfaz mensaje

La interfaz de mensaje muestra una apariencia mucho más sencilla para que pueda el usuario ver toda la información seguida, dejando en el último campo el cuerpo del mensaje escrito por el remitente.

Con la opción en la esquina superior derecha la opción de responder el mensaje al remitente y de eliminar el mensaje (enviándolo a la bandeja de Eliminados)

### Interfaz de nuevo mensaje y responder mensaje

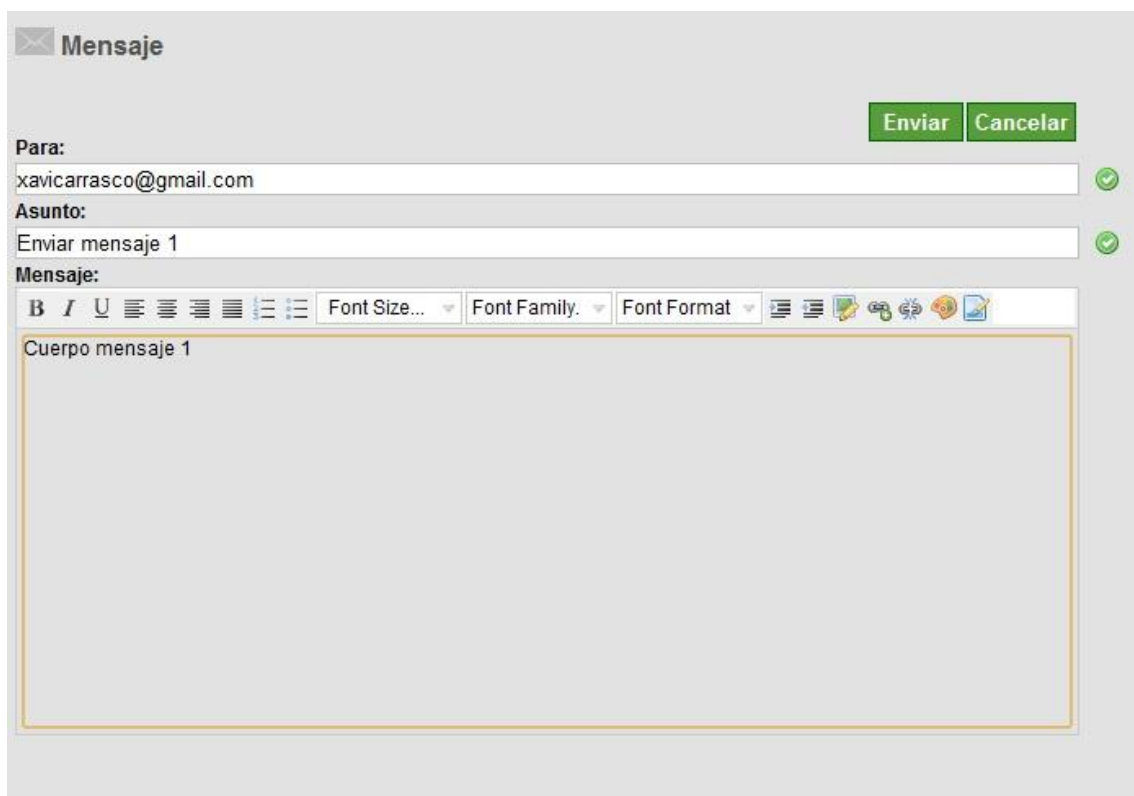


Figura 47 Interfaz nueva y responder mensaje

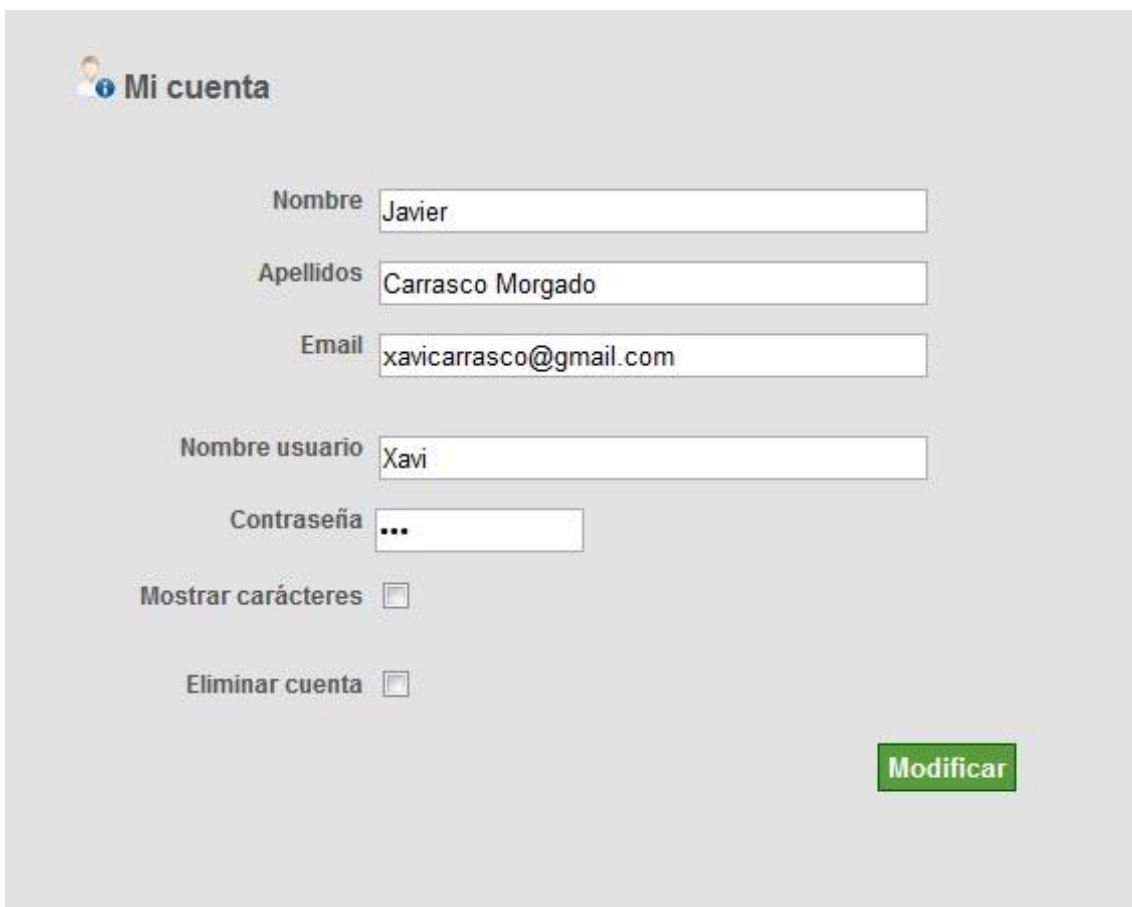
Una interfaz con tres cuadros de texto a rellenar.

- “Para:” El destinatario del mensaje. La diferencia fundamental entre *Nuevo Mensaje* y *Responder* reside en este campo. Ya que, en *Responder* ya aparece el Nombre de Usuario, o correo, del remitente para enviarle la respuesta. Sin embargo, en *Nuevo Mensaje*, hay una validación automática que nos

informará si el Nombre de Usuario, o correo electrónico, que estamos introduciendo es correcto, es decir, si se encuentra registrado en la aplicación.

- “Asunto:” El motivo del mensaje. En *Responder Mensaje* está relleno con el asunto del mensaje que respondemos.
- “Mensaje:” El contenido del mensaje. Contiene un cuadro de editor de texto para poder cambiar el estilo, color y tamaño de la letra, así como enlazar URL's e insertar imágenes.

### Interfaz de Mi cuenta



**Mi cuenta**

Nombre

Apellidos

Email

Nombre usuario

Contraseña

Mostrar caracteres ☐

Eliminar cuenta ☐

**Modificar**

Figura 48 interfaz mi cuenta

Formulario auto-rellenado con los datos que haya introducido el usuario en los campos que se requieren. Permite modificar los campos.

Casilla para desactivar la cuenta en caso que un usuario de la plataforma quisiera darse de baja.



## Interfaces de la sección Proyectos

Proyecto está compuesto de diferentes funcionalidades que dependen de él. Como es el caso de *Discusiones* y *Documento*.

A continuación repasamos la interfaz de todas ellas.

### Interfaz de Nuevo proyecto



Figura 49 Interfaz nuevo proyecto

Formulario para crear un proyecto. La descripción tan solo es vista por los miembros de un proyecto. El Nombre que se le haya dado al mensaje se enviará a los colaboradores, pero eso lo veremos en otra interfaz.

## Interfaz de la sección Información



Información | [Tareas](#) | [Colaboradores](#) | [Foro](#) | [Documentación](#)

 **ProyGes - Gestion de Proyectos**

**Descripción**  
Finalización del PFC - ProyGes. Acabar con los retoques y posibles errores que puedan dar en recoger datos y formulario.

Fecha de creación: 28/08/2012  
Fecha de entrega: 06/09/2012  
Creado por: Xavi

[Eliminar proyecto](#)

Figura 50 Interfaz información proyecto

Esta pantalla contiene la Fecha de creación del proyecto, la fecha de entrega, quien es el creador y el título y la descripción del proyecto. Además, el creador del proyecto dispone de la opción de eliminar el proyecto.

En la esquina superior derecha las interfaces del Contenido de Proyectos, hay un menú propio de Proyectos en los que acceder a los apartados de *Información*, *Tareas*, *Colaboradores*, *Discusiones* y *Documentación*.

## Interfaz de la sección Tareas



[Información](#) | [Tareas](#) | [Colaboradores](#) | [Foro](#) | [Documentación](#)

 **Tareas - ProyGes - Gestion de Proyectos**

Tarea:

Usuario:  Fecha límite:  

[Nueva tarea](#)

**Lista de tareas**

Tarea	Asignada a ..	Estado	F.Finalización
<input checked="" type="checkbox"/> Finalizar apartado de Diseño	Xavi	Realizada	23/08/2012
<input type="checkbox"/> Creación PowerPoint	Xavi	Pendiente	08/09/2012
<input type="checkbox"/> Finalizar Memoria	Xavi	Pendiente	29/08/2012
<input type="checkbox"/> Acudir Defensa Proyecto	Jordi	Pendiente	13/09/2012

Figura 51 Interfaz tareas

Recoge en la misma página:

- El formulario para rellenar que tarea hay que desempeñar, el miembro del proyecto al que hay que asignársela y la fecha límite para realizar esa tarea.
- Las tareas asignadas, a que miembro y el estado de las tareas (si han sido realizadas o no).

### Interfaz de la sección Colaboradores



E-mail	Enviada	Estado	
Jordi	29/08/2012	Aceptada	✗
Xavi	28/08/2012	Aceptada	✗

Figura 52 Interfaz colaboradores

Esta pantalla, muestra, en caso de ser el creador del proyecto, un cuadro de texto para introducir el email del colaborador al que se quiere invitar. Debajo del cuadro de texto descrito hay un listado de los colaboradores que se les ha enviado la petición y están pendientes de aceptar o si ya ha sido aceptada.

## Interfaz de la sección Discusiones



Figura 53 Interfaz discusiones

Sección donde los miembros pueden hablar y compartir dudas sobre cualquier tema creando discusiones con temáticas diferentes.

Una sencilla interfaz con el título de la discusión a tratar, quien ha creado la discusión y el momento de la creación.

## Interfaz de la Discusión



Figura 54 Interfaz discusión

Listado respuestas, por parte de los miembros de un proyecto, que ha creado una discusión.

La interfaz se mantiene muy simple, y con la sucesión de mensaje da la sensación de estar en una conversación de chat.

## Interfaz de la sección Documentación

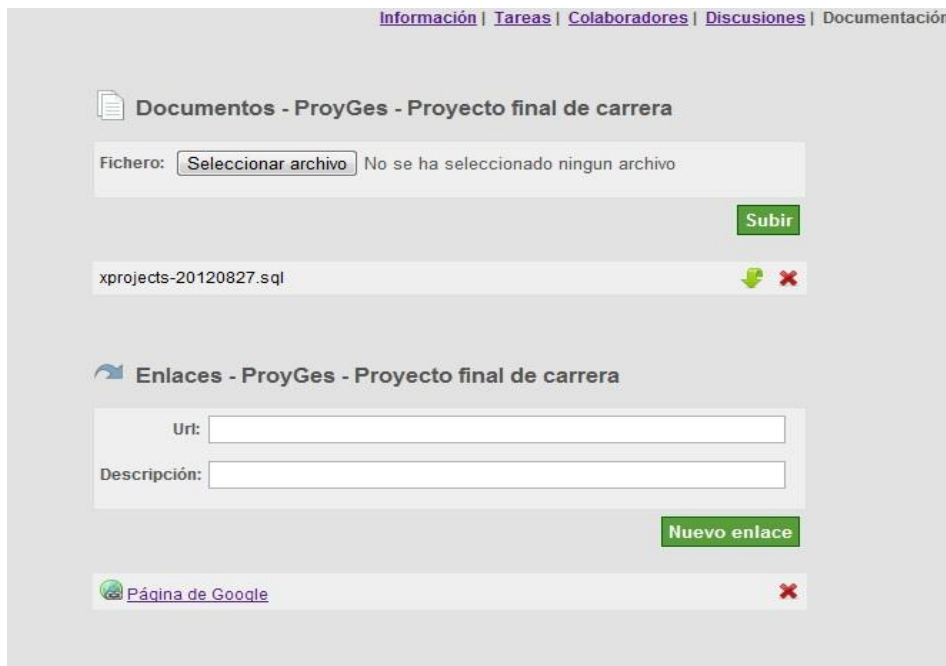


Figura 55 Interfaz documentación

Se muestra el botón para seleccionar el archivo del ordenador del usuario. Una vez subido el archivo, aparecerá en forma de listado más abajo, donde podrá ser descargado o eliminado.

## **5.5. Pruebas**

Para garantizar el correcto funcionamiento de la aplicación y la integridad de los datos que se manipulan es básico completar el ciclo de desarrollo. Hay muchos factores diferentes a tener en cuenta en el ciclo de desarrollo que minimizan la incidencia de errores y facilitan su detección.

Normalmente en el desarrollo de un proyecto, la parte de pruebas toma una parte importante del tiempo. En este caso por falta de tiempo no se le ha podido dedicar el tiempo deseado. Aún así, se han minimizado el impacto de los fallos localizado.

A continuación se repasan los distintos tipos de pruebas que se han ejecutado.

### **Pruebas de unidad**

La primera de estas pruebas es el test de todos los elementos al ser ejecutados de manera independiente y ver que su funcionamiento es el correcto en la aplicación o ver si hay algún error. Esta prueba ha sido esencial para el correcto funcionamiento de la aplicación.

### **Pruebas de integración**

Consiste en comprobar la interacción de un elemento con los otros que ya estaban presentes en la aplicación.

En este caso se han encontrado algunos elementos incompatibles entre ellos dentro de la aplicación, y se ha procedido a la rectificación de dichos errores. Gracias a este test se ha podido finalizar la aplicación sin problemas de compatibilidad entre los elementos.

En los casos que se ha encontrado error, se ha arreglado el problema modificando allí donde fallaba o buscando alternativas en otras funciones similares.

## **Pruebas de compatibilidad**

Al ser una aplicación web, se ha tenido que testear con diferentes navegadores web. Se ha testado con *Mozilla Firefox*, *Google Chrome*, *Internet Explorer* y *Opera* actualizados a la última versión disponible hasta la fecha de 01 de setiembre de 2012. Su funcionamiento en todos los navegadores es correcto. Únicamente se aprecian ligeras diferencias gráficas en Internet Explorer, pero nada funcional.

## **Prueba de seguridad**

Como para acceder a la web hace falta estar autenticado. Una de las pruebas ha sido intentar acceder a la web sin estar Registrado o Ingresado en la aplicación, introduciendo una URL de diferentes apartados de la aplicación en la barra de direcciones del navegador.

Los resultados han sido un éxito, ya que la aplicación ha impedido ver la página que estaba intentado acceder sin estar debidamente Registrado o con la sesión iniciada.



*Figura 56 Seguridad web*

## **Conclusión de resultados**

Una vez hecho las pruebas correspondientes, localizados los errores y realizados los cambios necesarios, se puede afirmar que la aplicación se ejecuta de forma estable.

## 6. CONCLUSIONES

### 6.1. Introducción

Finalmente, en este capítulo se exponen las conclusiones alcanzadas durante la realización del proyecto. Comparando la planificación temporal inicial con la real se observen desviaciones, las cuales se comentan en el siguiente apartado. Se expone las posibles ampliaciones y mejoras del proyecto para el futuro y, finalmente, la valoración personal.

### 6.2. Desviaciones del proyecto

El seguimiento del proyecto permite saber en qué fase se encuentra el desarrollo del proyecto. Es importante saber el tiempo previsto en realizar el proyecto, pero las previsiones del principio son estimaciones. Llegados a este punto, con el proyecto finalizado, podemos valorar si se ha ceñido según lo previsto o si ha habido un aumento de horas ocasionando la desviación del proyecto, como ha sido el caso.

En la siguiente tabla se detalla el tiempo previsto para cada tarea, cual ha sido realmente el tiempo de duración y la desviación que ha habido entre la planificación prevista y la real.

Nombre de tarea	Tiempo Previsto	Tiempo Real	Desviación
<b>Planificación</b>			
Estudio Viabilidad	25 horas	25 horas	0 horas
Aprobación Estudio de Viabilidad	1 hora	1 hora	0 horas
<b>Análisis de la aplicación</b>			
Análisis de Requisitos	3 horas	3 horas	0 horas
Análisis de Datos (Base de datos)	6 horas	6 horas	0 horas
Documentación Análisis	6 horas	6 horas	0 horas
Aprobación del análisis	1 hora	1 hora	0 horas
<b>Diseño de la aplicación</b>			



Diseño de la Base de Datos	8 horas	10 horas	2 horas
Diseño de la interfaz	5 horas	10 horas	5 horas
Diseño de las pruebas	5 horas	5 horas	0 horas
Documentación del diseño	6 horas	6 horas	0 horas
Aprobación del Diseño	1 hora	1 hora	0 horas
<b>Desarrollo y Pruebas</b>			
<b>Desarrollo de la Aplicación</b>			
Preparación del entorno de desarrollo	3 horas	3 horas	0 horas
Configuración de la base de datos	3 horas	3 horas	0 horas
<b>Desarrollo funcionalidades de la Aplicación</b>			
Desarrollo Usuario	5 horas	5 horas	0 horas
Desarrollo Agenda	5 horas	7 horas	2 horas
Desarrollo Discusiones	10 horas	13 horas	3 horas
Desarrollo Proyecto	20 horas	30 horas	10 horas
Desarrollo Mensajería	18 horas	30 horas	12 horas
<b>Desarrollo de la interfaz del usuario</b>			
Desarrollo de la Barra Lateral	5 horas	10 horas	5 horas
Desarrollo del Contenido	4 horas	4 horas	0 horas
Desarrollo de la Barra de Navegación	1 hora	1 hora	0 horas
Desarrollo del Encabezado	1 hora	1 hora	0 horas
Desarrollo del Pie de página	1 hora	1 hora	0 horas
Desarrollo de la página de Inicio	3 horas	6 horas	3 horas
<b>Test y Pruebas</b>			
Pruebas Unitarias	2 horas	2 horas	0 horas
Pruebas de integración	2 horas	2 horas	0 horas
Documentación del Desarrollo y Test	4 horas	4 horas	0 horas
Aprobación del Desarrollo y Test	1 hora	1 hora	0 horas
<b>Generación del documento (Memoria del Proyecto)</b>	40 horas	40 horas	0 horas
<b>Cierre del proyecto</b>	6 horas	6 horas	0 horas
<b>Defensa del Proyecto</b>	9 horas	9 horas	0 horas
<b>Total</b>	<b>210 horas</b>	<b>252 horas</b>	<b>42 horas</b>

Tabla 11 Desviación temporal

Tal y como se muestra en la tabla ha habido una considerable desviación de tiempo por lo que respecta la previsión inicial.

Esta desviación, principalmente en la fase de desarrollo de la aplicación, es debido principalmente por dos motivos: el conocimiento del lenguaje y el grado de dificultad. Ambos motivos tienen una relación directa.

En algunas fases del proyecto las carencias del conocimiento de los lenguajes utilizados se han hecho latentes, provocando que haya requerido horas de búsqueda y de formación para suplir las carencias que sufría. Prueba de ello ha sido en el desarrollo de *Mensajes y Proyectos*, pues el nivel dificultad de plasmar la idea inicial de estas dos partes ha sido mayor de los que me esperaba, retrasando el proyecto notablemente.

A continuación se muestra la planificación temporal real:

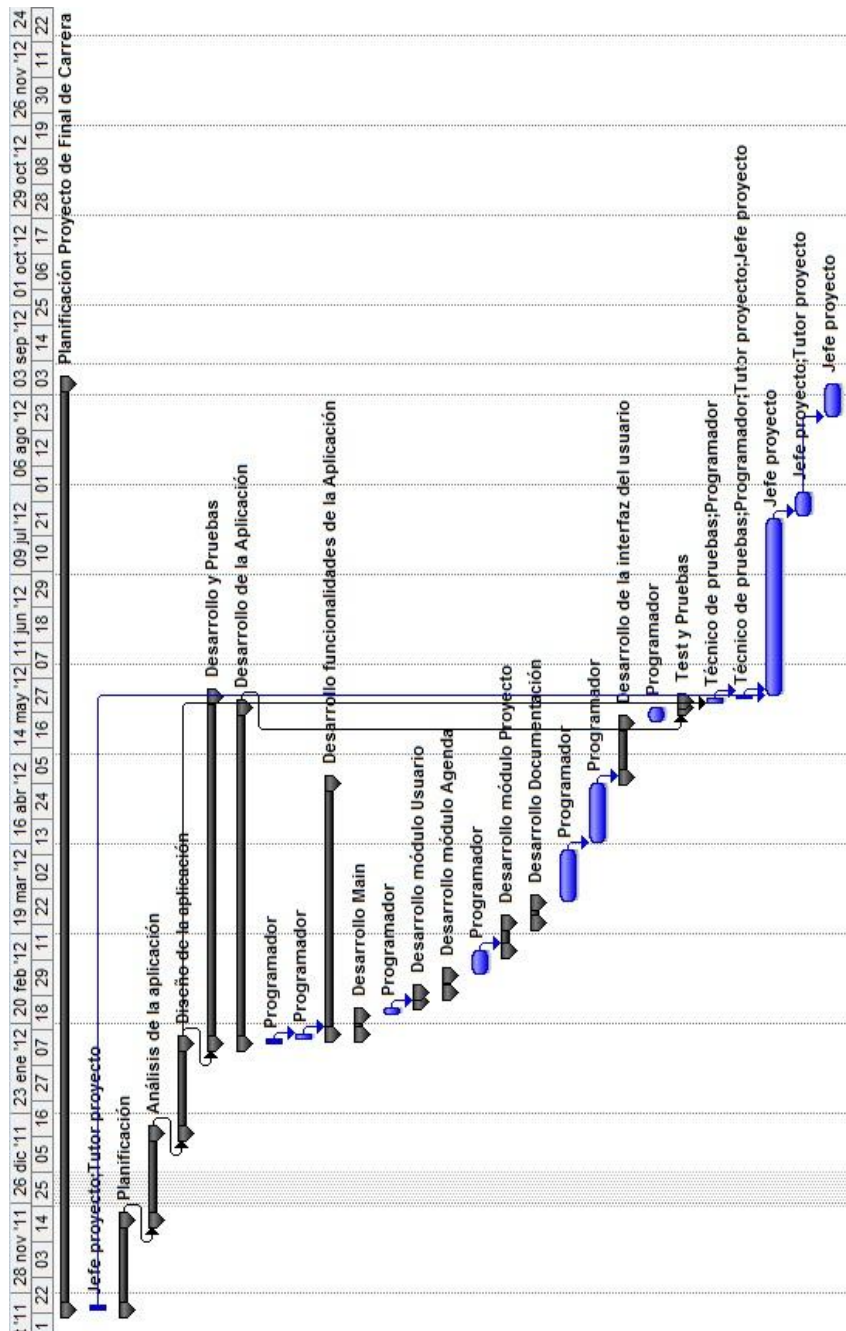


Figura 57 Diagrama de Gantt detallado

Cambiando el calendario de trabajo en verano, reduciendo las horas de trabajo diario de 2 horas a 1 hora de dedicación diaria.

Como vemos el proyecto finaliza el 12/09/2012, con su última tarea, la *defensa del proyecto*.

### 6.3. Futuras mejoras

Una vez finalizado el proyecto, y analizado todo lo desarrollado hasta este punto, el margen de mejora y ampliación es grande.

Debido a la estructuración del código, tanto como el patrón Modelo-Vista-Controlador y la organización de los ficheros, proporciona una gran escalabilidad, permitiendo que cualquier otra persona pudiera continuar con la ampliación de la aplicación.

A continuación se mientan algunas ampliaciones que podrían dar hacer crecer la aplicación:

- Buscador de usuarios de la plataforma. Actualmente, tal y como se he pensado la plataforma, es un estilo similar al correo electrónico. Si no tienes previamente el correo electrónico de la persona, no puede ponerte en contacto con ella. Con un buscador de usuarios se supliría esa carencia.
- Ampliación, o modificación, de apartados actuales. El inicio podría tener una visión más dinámica recogiendo las últimas actualidades de los proyectos en curso por el usuario, reflejando así últimas tareas añadidas, nuevos colaboradores o archivos subidos.

Poder elegir diferentes vistas del Calendario, mostrándolo en forma de agenda, como es actualmente, o en forma de calendario mensual.

- Nuevos apartados. Añadir nuevas secciones de manera que facilite la organización del usuario como un apartado de notas, donde poder hacer apuntes de aspectos referentes a sus proyectos.
- Simples mejoras. Harían falta algunas pequeñas mejoras de diseño de interfaz. Como nuevos menús, añadir atajos en la barra de navegación, o URL amigables.

- Control de Usuario. Desarrollar algún sistema para controlar que no cualquier persona puede registrarse en la web, si realmente solo se quieren personas relacionadas con una Entidad o Asignatura.
- Contratar un servicio de *hosting* que permita ampliar la capacidad de almacenado de los archivos de subida.

#### **6.4. Valoración personal**

Después de ver el resultado obtenido en la realización del proyecto, se puede decir que se han alcanzado los objetivos y los retos marcados al principio de éste.

Los conocimientos adquiridos durante el desarrollo son cuantiosos, más de los que al principio podría pensar que adquiriría. Pues el crear los problemas uno mismo y solucionarlos, siempre es enriquecedor y mucho más provechoso para uno mismo.

La experiencia ha sido muy positiva. No solamente por el resultado final, si no por el trabajo desempeñado y las horas dedicadas, haciendo algo desde cero, y paso a paso, ver como cogía forma poco a poco pero con la suficiente constancia para no abandonar y poder llegar hasta este último tramo.

## 7. BIBLIOGRAFÍA Y REFERENCIAS

### Libros

- ✓ **Craig Larman** *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificad.* Segunda edición.  
PEARSON EDUCACIÓN, S.A. 2003  
ISBN: 978-84-205-3438-1

### Referencias internet

- ✓ W3Schools, <http://www.w3schools.com> (05/09/2012)
- ✓ The PHP Groups, <http://www.php.net/manual/es/index.php> (01/09/2012)
- ✓ Factoría de Internet S.L., <http://www.webtaller.com/> (14/08/2012)
- ✓ Ellislab, INC ,[http://codeigniter.com/user\\_guide/](http://codeigniter.com/user_guide/) (01/09/2012)
- ✓ Etnassoft, [http://books.openlibra.com/pdf/CodeIgniter\\_Guia\\_Usuario\\_2\\_0.pdf](http://books.openlibra.com/pdf/CodeIgniter_Guia_Usuario_2_0.pdf)  
(01/09/2012)
- ✓ Envato, <http://net.tutsplus.com/tutorials/php/creating-a-file-hosting-site-withcodeigniter/> (12/08/2012)
- ✓ Codeacademy, <http://www.codecademy.com> (24/07/2012)
- ✓ Erach Network, <http://www.tizag.com/ajaxTutorial/ajax-javascript.php>  
(26/05/2012)
- ✓ Brian Kirchoff Production, <http://nicedit.com/download.php> (13/08/2012)
- ✓ Iconfindes ApS, <http://www.iconfinder.com/> (15/04/2012)

Javier Carrasco Morgado