



**Universitat Autònoma  
de Barcelona**

# PORTAL WEB PARA LA EMPRESA RUBEN'S INSTALACIONES Y EXCAVACIONES

Memoria del proyecto  
de Ingenieria Técnica en  
Informática de Sistemas  
realizado por  
*Jose Hernandez Carballido*  
y dirigido por  
*Marc Talló Sendra*

**Escola d'Enginyeria**

Sabadell, Junio de 2013.



**PROYECTO FIN DE CARRERA DE LA ESCUELA DE INGENIERIA**

<b>Título del proyecto:</b>	
PORTAL WEB PARA LA EMPRESA RUBEN'S INSTALACIONES Y EXCAVACIONES	
<b>Autor:</b> Jose Hernandez Carballido	<b>Fecha:</b> Junio de 2013
<b>Tutor:</b> Marc Talló Sendra	
<b>Titulación:</b> Ingeniería Técnica Informática de Sistemas	
<b>Palabras clave</b>	
<ul style="list-style-type: none"> <li>• Catalán: web, PHP, MySQL</li> <li>• Castellano: web, PHP, MySQL</li> <li>• Inglés: web, PHP, MySQL</li> </ul>	
<b>Resumen del proyecto</b>	
<ul style="list-style-type: none"> <li>• Catalán: Aquest projecte consisteix en el desenvolupament d'un portal web per a una empresa de construcció. Estará format per una part pública de cara al client que conté diversa informació de l'empresa i una secció per contactar amb l'empresa. A més, tindrà una part privada de gestió de certs aspectes de l'empresa, com pressupostos i pagaments, a més d'un mòdul d'estadístiques pel seguiment. El desenvolupament s'ha considerat necessari degut al context actual i a les mancances de l'empresa que s'han detectat.</li> <li>• Castellano: Este proyecto consiste en el desarrollo de un portal web para una empresa de construcción. Estará formado por una parte pública de cara al cliente que contiene diversa información de la empresa y una sección para contactar con la empresa. Además, tendrá una parte privada de gestión de ciertos aspectos de la empresa como presupuestos y pagos, además de un módulo de estadísticas para el seguimiento. El desarrollo se ha considerado necesario debido al contexto actual y a las carencias detectadas en la empresa.</li> <li>• Inglés: This project is the development of a web application for a construction company. It will have a non protected part faced to customers that will contain different information about the company and a contact section. In addition, it will have a protected part to manage some aspects of the company like budgets or payments, and a stats module for tracing. The development has been necessary as a consequence of the current context and the identification of some gaps in the company.</li> </ul>	

## ÍNDICE

<b>Índice de Tablas y Figuras.....</b>	<b>6</b>
<b>Capítulo 1. Introducción.....</b>	<b>7</b>
1.1. Presentación.....	7
1.2. Objetivos.....	7
1.3. Estado del arte.....	8
1.4. Motivaciones.....	9
1.5. Estructura de la memoria.....	9
<b>Capítulo 2. Estudio de viabilidad.....</b>	<b>11</b>
2.1. Introducción.....	11
2.2. Situación actual.....	11
2.3. Perfil de usuario.....	12
2.4. Objetivos.....	13
2.5. Especificaciones.....	13
2.5.1. Especificaciones funcionales.....	13
2.5.2. Especificaciones no funcionales.....	13
2.6. Recursos.....	14
2.6.1. Desarrollo.....	14
2.7. Planificación Inicial.....	15
2.7.1. Descripción de las tareas.....	15
2.7.2. Diagrama de Gantt.....	16
2.8. Costes.....	16
2.9. Riesgos.....	18
2.10. Conclusiones.....	19
<b>Capítulo 3. Fundamentos teóricos.....</b>	<b>20</b>
3.1. Lenguajes de programación.....	20
3.1.1. Lenguajes del lado cliente.....	20
3.1.2. Lenguajes del lado servidor.....	23
3.2. Sistema gestor de bases de datos.....	28
<b>Capítulo 4. Análisis del proyecto.....</b>	<b>30</b>
4.1. Introducción.....	30
4.2. Requerimientos funcionales.....	30
4.2.1. Diagramas de casos de uso.....	33
4.2.2. Diagramas de secuencia.....	36
4.3. Requerimientos no funcionales.....	43
4.3.1. Restricciones de diseño.....	43

4.3.2. Objetivos de diseño.....	44
4.4. Estructura y diseño Base de Datos.....	44
4.5. Menús de navegación.....	50
4.5.1. Menú navegación parte pública.....	50
4.5.2. Menú navegación parte privada.....	52
<b>Capítulo 5. Implementación.....</b>	<b>54</b>
5.1. Introducción.....	54
5.2. Entorno de usuario.....	54
5.3. Motor de la aplicación.....	56
5.4. Capa de datos.....	61
<b>Capítulo 6. Pruebas.....</b>	<b>63</b>
<b>Capítulo 7. Conclusiones.....</b>	<b>65</b>
7.1. Consecución de objetivos.....	65
7.2. Desviación de la planificación.....	65
7.3. Líneas de mejora y ampliación.....	66
<b>Capítulo 8. Bibliografía y fuentes de consulta.....</b>	<b>67</b>
8.1. Referencias electrónicas.....	67

## Índice de Tablas y Figuras

### Capítulo 2

Tabla 2.1. Descripción de tareas para planificación de proyecto.....	15
Tabla 2.2. Recursos humanos o equipo de proyecto.....	16
Tabla 2.3. Desglose de costes del proyecto.....	17
Figura 2.1. Diagrama de Gantt para planificación de proyecto.....	17

### Capítulo 3

Figura 3.1. Esquema programación del lado del servidor.....	23
---	----

### Capítulo 4

Tabla 4.1. Acciones del caso de uso Login.....	36
Tabla 4.2. Acciones del caso de uso Crear Usuario.....	37
Tabla 4.3. Acciones del caso de uso Editar Usuario.....	38
Tabla 4.4. Acciones del caso de uso Eliminar Usuario.....	39
Tabla 4.5. Acciones del caso de uso Consultar Usuarios.....	40
Tabla 4.6. Acciones del caso de uso Editar Presupuesto (Añadir Recurso).....	41
Tabla 4.7. Acciones del caso de uso Consultar Logs.....	42
Figura 4.1. Diagrama de casos de uso perfil Usuario no registrado.....	33
Figura 4.2. Diagrama de casos de uso perfil Encargado.....	34
Figura 4.3. Diagrama de casos de uso perfil Contable.....	34
Figura 4.4. Diagrama de casos de uso perfil Director.....	35
Figura 4.5. Diagrama de casos de uso perfil Administrador.....	35
Figura 4.6. Diagrama de secuencia del caso de uso Login.....	36
Figura 4.7. Diagrama de secuencia del caso de uso Crear Usuario.....	37
Figura 4.8. Diagrama de secuencia del caso de uso Editar Usuario.....	38
Figura 4.9. Diagrama de secuencia del caso de uso Eliminar Usuario.....	39
Figura 4.10. Diagrama de secuencia del caso de uso Consultar Usuarios.....	40
Figura 4.11. Diagrama de secuencia del caso de uso Editar Presupuesto (Añadir Recurso).....	42
Figura 4.12. Diagrama de secuencia del caso de uso Consultar Logs.....	43
Figura 4.13. Diagrama estructura base de datos.....	45
Figura 4.14. Screenshot página inicial.....	51
Figura 4.15. Screenshot listado proyectos.....	45
Figura 4.16. Screenshot listado información empresas.....	45
Figura 4.17. Screenshot sección presupuestos abiertos.....	45

## Capítulo 1. Introducción

### 1.1. Presentación

Este proyecto tiene como objetivo principal la creación de un portal web para una empresa de construcción. Con el desarrollo de este proyecto se pretenden solventar diferentes necesidades que se han detectado en la empresa debido a su crecimiento y al contexto actual. A consecuencia, se ha tomado la decisión de crear una página web de empresa de cara al cliente e implantar una solución de gestión basada en una aplicación web que a continuación procederemos a presentar.

### 1.2. Objetivos

En primer lugar, hablaremos sobre la difusión y promoción de la empresa. El proyecto tendrá como uno de sus objetivos principales potenciar la presencia de esta empresa en la red con el objetivo de darse a conocer y mediante el uso de esta página web disponer de una plataforma publicitaria que acerque a los clientes presentado los servicios y numerosos proyectos realizados. También se busca la inserción de nuevos recursos tecnológicos para una mejor automatización de tareas en la empresa como pueden ser contacto con los clientes vía email o soporte de datos mediante BBDD y entorno web, así como la posibilidad de difusión de proyectos realizados.

Otro de los objetivos de este proyecto es dar solución a la necesidad de informatizar datos de la empresa como por ejemplo de clientes o proyectos, para poder acceder a ellos y gestionarlos de manera mucho más eficiente e incluso a distancia, aprovechando las tecnologías móviles de las que disponemos hoy en día combinadas con el uso de internet.

Como último objetivo de este proyecto, se pretende desarrollar una herramienta para la creación y el control de presupuestos de las diferentes obras, tanto de las que están en curso, como de las ya finalizadas. En este ámbito se trata de establecer una herramienta funcional y sencilla, que permita todo tipo de gestiones relacionadas con los presupuestos de forma interactiva. También se integrarán datos de coste de los recursos y totales de los proyectos para facilitar en mayor medida una posterior contabilidad, con disponibilidad de algunos gráficos de seguimiento.

Todo el portal, obviamente ha de ir acompañado de un diseño cuidado y una interfaz sencilla y dinámica, realizada con las tecnologías adecuadas para facilitar el uso y la experiencia del usuario. Para dar respuesta a estos requisitos, se pretende programar la interfaz y el diseño utilizando CSS, Javascript & JQuery y el framework Bootstrap para dotar de dinamismo y fluidez de cara al usuario. Se utilizará también PHP y el sistema gestor de BBDD MySQL para interactuar con todos los datos de manera dinámica. Se contará también con un sistema de credenciales para gestionar el acceso a las diferentes secciones privadas del portal.

Cabe destacar que el proyectista desea profundizar en la utilización y el desarrollo de estas tecnologías mencionadas arriba pues claramente se están imponiendo en el panorama actual de la programación en tecnologías web, y claramente es el paso adelante necesario para asentarse en la web 2.0 y el diseño centrado en el usuario.

### 1.3. Estado del arte

Previamente a la realización de este proyecto, se han estudiado alguna de las soluciones alternativas existentes. Obviamente el estudio se ha centrado en las soluciones Open Source, sin contemplar opciones de pago por adquirir una aplicación u otras soluciones que acarreen el pago de una licencia ya sea mensual o anual por la utilización del software.

En primer lugar, hemos analizado el ERP/CRM **Dolibarr**. Se trata de un software 100% gratuito y Open Source, además basado también al 100% en plataforma web de manera que podemos acceder desde cualquier lugar en el que tengamos acceso a internet. Además el código es bastante personalizable, aunque este es un aspecto en el que no hemos profundizado. La interfaz de cara al usuario es sencilla e intuitiva.

Procedemos a listar las principales funcionalidades:

- Gestión de usuarios, grupos y permisos
- Gestión financiera (facturas, impuestos, informes de resultados, gestión de cuentas bancarias...)
- Gestión de proyectos (creación y seguimiento, organización, diagrama de gantt...)
- Gestión documental (almacenamiento documentos, gestión integrada...)
- Gestión de productos y servicios (gestión completa productos, almacenes, stock, estadísticas...)
- Gestión de terceros (gestión de clientes, proveedores...)

En resumen, nos hemos encontrado un software muy potente, que reúne prácticamente todas las funcionalidades necesarias e incluso muchas más, el problema es que se trata de un software de demasiada envergadura en comparación con la solución que tratamos de implementar.

En segundo lugar, también hemos visto OpenERP, otro ERP/CRM Open Source e implementado al 100% en tecnología web para acceder desde cualquier ordenador o dispositivo móvil conectado a internet. Se trata de una aplicación prácticamente profesional, con multitud de funcionalidades de todo tipo orientadas para la gestión de las PYMES, además de un completo módulo de contabilidad.

A continuación algunas de sus funcionalidades:

- Gestión básica de proyectos (tareas e incidencias)
- Contabilidad (facturación, proveedores, bancos, planes, asientos...)



- Almacén (albaranes, inventarios)
- Recursos (planificación, lista materiales, productos...)
- Informes (tableros, análisis ventas, análisis contabilidad...)
- Ventas (presupuestos, pedidos, facturación, contratos...)

Nos hemos encontrado con que este software está dotado de muchas funcionalidades que no nos serían útiles para nuestros objetivos.

Hemos decidido no analizar más soluciones ERP/CRM Open Source ya que nos encontramos diferentes de un corte parecido, y hemos tomado la decisión de desarrollar nuestra propia solución ajustada a los requisitos y objetivos del cliente.

#### **1.4. Motivaciones personales**

La principal motivación personal es el vínculo que tengo con la empresa Ruben's Instalaciones y Excavaciones.

Conociendo sus actividades, aspiraciones y necesidades, y todo eso unido a los conocimientos que había adquirido y los que pretendía desarrollar en cuanto a tecnologías web, dieron forma a la idea de comenzar a desarrollar este proyecto. Además, sabiendo de la necesidad que tiene cualquier empresa de introducirse en el mundo de Internet, se agrandó el interés por este proyecto.

Por último, viendo el auge y el potencial de las tecnologías web actuales combinadas con los dispositivos móviles se acabó de conformar la idea de desarrollar algunos de los aspectos de la gestión de esta empresa de manera que pudieran ser accesibles desde cualquier sitio no solo desde la oficina, además de optimizar a la empresa tiempo y calidad en cuanto a la gestión de presupuestos.

#### **1.5. Estructura de la memoria**

La presente memoria se organizará en siete capítulos, en los cuales se documentará el proceso de desarrollo del proyecto, desde su estudio y planificación inicial hasta la fase de pruebas una vez implementada la aplicación.

En primer lugar se encuentra el índice de tablas y figuras, que sirve como guía rápida de consulta de todos estos elementos que se encuentren en la memoria.

Tras este primer capítulo introductorio, se realizará estudio de viabilidad el proyecto (capítulo 2), que está compuesto por una breve introducción y descripción de los objetivos, establecimiento de las

especificaciones de nuestro proyecto y posteriormente una selección de las herramientas que utilizaremos. Realizaremos una planificación del proyecto y evaluaremos los costes y los riesgos para finalizar obteniendo las correspondientes conclusiones.

Seguidamente, el tercer capítulo analizará los requerimientos para el desarrollo de nuestro portal web, adjuntando los diversos diagramas de casos de uso y de secuencia y también analizando las restricciones de diseño.

El cuarto capítulo, tratará sobre la implementación de nuestro proyecto. En él analizaremos el proyecto desde sus diferentes capas y cómo éstas están implementadas y con qué tecnologías.

Posteriormente, en el capítulo 5, describiremos las pruebas realizadas al proyecto para comprobar la funcionalidad e integridad del mismo.

A continuación, en el sexto capítulo estarán las conclusiones, donde se tratarán las desviaciones de la planificación y futuras líneas de mejora o ampliación y por último en el último capítulo se detallan todas las fuentes de consulta y referencias electrónicas utilizadas para el desarrollo del proyecto.

## Capítulo 2. Estudio de viabilidad

### 2.1. Introducción

Al realizar un proyecto el cuál requiere una utilización de recursos humanos y en menor medida económicos, es necesaria la realización de un estudio de viabilidad con la finalidad de evaluar las garantías de éxito de nuestro proyecto. Este capítulo tratará de realizar dicho estudio para llegar a la conclusión de si es viable o no poner en marcha este proyecto y en caso afirmativo evaluar y planificar la perspectiva del mismo, abordando diferentes aspectos.

En este capítulo estudiaremos los siguientes apartados:

- Situación actual que produce el desarrollo de este proyecto cómo solución
- Objetivos que se pretenden conseguir con este proyecto
- Especificaciones tanto funcionales como no funcionales para nuestro proyecto
- Selección de los recursos y el equipo y las herramientas que utilizaremos para desarrollar
- Planificación inicial con descripción de las tareas y diagrama de Gantt temporal
- Costes que supondrá la creación de este proyecto
- Riesgos que puede correr el desarrollo del proyecto
- Conclusiones finales

### 2.2. Situación actual

Actualmente, la empresa Ruben's Instalaciones y Excavaciones no dispone de página web. Su única manera de darse a conocer es mediante Páginas Amarillas y algún directorio de empresas de su sector. Esto es un claro handicap respecto a la competencia del sector, ya que actualmente esta situación supone estar en desventaja respecto a la contratación de proyectos por ejemplo procedentes del extranjero y a nivel de difusión en general. Obviamente, la actual situación a nivel empresarial requiere estar situado en Internet y disponer de todo tipo de facilidades en cuánto a nuevas tecnologías se refiere.

En referencia a la gestión de presupuestos y el control de obras se realiza todo mediante hojas de Microsoft Excel que desempeñan el trabajo mediante fórmulas de manera bastante simple. En esta hoja, en sus respectivos campos se insertan los ítems con los que contará el presupuesto y su coste por hora y por día así como las cantidades para acabar obteniendo un cálculo del total del presupuesto. Queda claro desde el primer momento que con este sistema se pierde mucho tiempo y eficiencia a la hora de realizar presupuestos y al modificar los de las obras que ya están en marcha, combinado con la interfaz de

Microsoft Excel que en ocasiones puede ser motivo de debate. Si añadimos que para realizar estas tareas hay que estar en oficina o situado delante de un ordenador con Microsoft Excel, lo cual limita la movilidad y usabilidad de cara al usuario, detectamos otro handicap importante. Tampoco disponen de gráficas de seguimiento de presupuestos, que se podrían llegar a realizar con Excel de manera tediosa, y con este proyecto se pretende dotar de ese servicio con gráficas que al ejecutarse muestren todos los datos de los que dispone la BBDD y se puedan organizar cómodamente por intervalos. Por todo esto se pretende automatizar todo el sistema de gestión de presupuestos para una cómoda accesibilidad y gestión desde cualquier lugar.

### **2.3. Perfil de usuario**

Este portal web contará con cinco tipos de usuarios. Los dividiremos en dos grupos: usuarios no registrados y usuarios registrados.

Los usuarios no registrados, que representarán la gran mayoría, serán visitantes a la página web de la empresa. Tendrán acceso a todo el contenido publicado en la parte pública del portal web. Obviamente no tendrán acceso a la parte privada de gestión del portal web.

Dentro de los usuarios registrados distinguiremos cuatro tipos: administrador, director, contable y encargado.

El administrador será el encargado de realizar algún tipo de mantenimiento si fuera necesario y debe garantizar el correcto funcionamiento de todo el portal web. Por ello, este perfil de usuario tendrá unos credenciales que le permitirán acceso a todas las secciones y a modificar cualquier tipo de contenido o información. Tendrá permisos para cambiar los credenciales a los demás usuarios.

El director será el que mayor credenciales tendrá de todos los usuarios registrados exceptuando el administrador. Este perfil de usuario tendrá acceso a todas las secciones y a modificar prácticamente la totalidad de los contenidos.

El contable será el encargado de realizar las tareas de contabilidad derivadas de los presupuestos ya cerrados. Por ello, este perfil de usuario tendrá acceso a toda la sección de presupuestos y de empresas, inclusive también la de estadísticas de seguimiento ya que puede serle de utilidad.

Por último, existe el encargado. Este perfil es referente al encargado de obra, que tendrá acceso a los presupuestos abiertos para modificar alguno de éstos en caso de que alguna obra sufra modificaciones durante su proceso. No tendrá acceso al resto de contenidos de la parte privada del portal.

## 2.4. Objetivos

- I. Creación de un portal web de la empresa Ruben's Instalaciones y Excavaciones
- II. Facilitar el contacto de clientes o potenciales clientes con la empresa
- III. Gestionar la información relativa a empresas que trabajan con Ruben's Instalaciones y Excavaciones
- IV. Automatizar y optimizar la creación de presupuestos, así como la modificación de presupuestos ya abiertos
- V. Facilitar posterior contabilidad mediante datos de coste asociados a los presupuestos
- VI. Sistema de credenciales de usuario
- VII. Medidas de seguridad suficientes para gestionar la información
- VIII. Disponer de una interfaz atractiva, intuitiva y eficiente
- IX. Posibilidades de expansión y difusión de la empresa utilizando el ámbito de Internet

## 2.5. Especificaciones

### 2.5.1. Especificaciones funcionales

- I. Recibir e-mails de clientes o usuarios del portal web
- II. Mostrar listado de proyectos realizados por la empresa
- III. Control de usuarios mediante nombre de usuario y password
- IV. Inserción/modificación/eliminación de usuarios
- V. Creación de manera sencilla y eficiente de presupuestos para las diferentes obras
- VI. Modificación de presupuestos abiertos
- VII. Consulta de presupuestos en función de su estado
- VIII. Gestión de la información de empresas relativas a Ruben's Instalaciones y Excavaciones

### 2.5.2. Especificaciones no funcionales

- I. Cumplimiento de la LOPD en referencia a los ficheros de datos y a los derechos de los clientes
- II. Normalización de la base de datos y acceso según el estándar SQL 99 (ISO/IEC 9075:1999)
- III. Los recursos utilizados para el desarrollo del portal tienen que estar ajustados a la medida de la entidad

- IV. Usabilidad: la interfaz tiene que cumplir la ISO 9241: *Ergonomic requirements for office work with visual display terminals*
- V. Control de todas las entradas de usuarios
- VI. El control de acceso se realizará de acuerdo con la lista de usuarios y permisos.

## **2.6. Recursos**

### **2.6.1. Desarrollo**

#### **Hardware**

- Procesador: Intel Pentium i3
- Memoria RAM: 4Gb DDR3
- HDD: 320Gb
- Monitor 15.4" XWGA LCD
- Acceso a Internet

#### **Software**

- Sistema Operativo: MS Windows 7 Professional Edition
- XAMPP 1.7.4 para crear servidor en local
- Notepad ++
- Adobe Photoshop
- Visual Paradigm 8.1
- Necesitaremos un cliente FTP para alojar el proyecto en el hosting posteriormente

#### **Tecnologías de programación**

- Lenguaje de etiquetas HTML
- Hojas de estilo CSS
- Lenguaje de programación PHP 5
- Lenguaje de programación Javascript
- Librería JQuery de Javascript

- Framework Bootstrap
- MySQL como sistema gestor de bases de datos

## 2.7. Planificación inicial

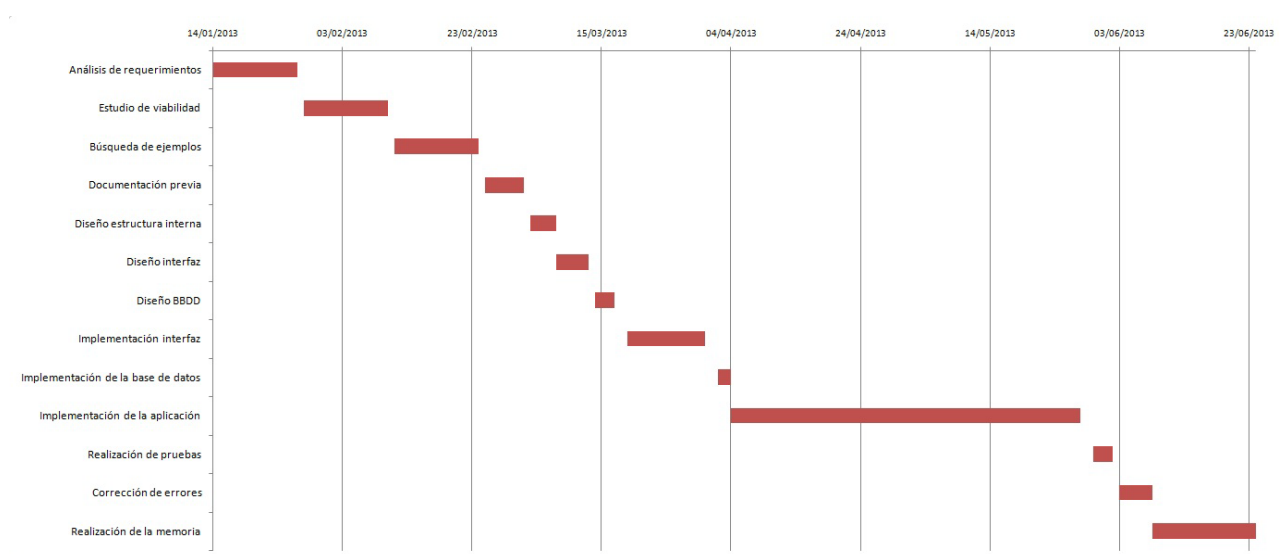
### 2.7.1. Descripción de las tareas

Dividiremos el desarrollo del proyecto en cinco etapas. A continuación se muestra una lista con la descripción de las tareas en sus correspondientes etapas y el número de horas que suponen.

<b>Etapas</b>	<b>Número de horas</b>
<b>1. Etapa de documentación</b>	<b>70 horas</b>
1.1. Análisis de requerimientos	20 horas
1.2. Estudio de viabilidad	20 horas
1.3. Búsqueda de ejemplos	20 horas
1.4. Documentación previa	10 horas
<b>2. Etapa de diseño</b>	<b>22 horas</b>
2.1. Diseño de la estructura interna del portal	8 horas
2.2. Diseño de la interfaz	8 horas
2.3. Diseño de la base de datos	6 horas
<b>3. Etapa de implementación</b>	<b>150 horas</b>
3.1. Implementación de la interfaz	25 horas
3.2. Implementación de la base de datos	5 horas
3.3. Implementación de la aplicación	120 horas
<b>4. Etapa de pruebas</b>	<b>20 horas</b>
4.1. Realización de pruebas	6 horas
4.2. Corrección de errores	14 horas

**5. Realización de la memoria****40 horas****TOTAL HORAS PROYECTO****302 horas****Tabla 2.1**

Teniendo en cuenta que la carga de trabajo será de dos horas diarias, podemos determinar una duración aproximada de 151 días de trabajo.

**2.7.2. Diagrama de Gantt****Figura 2.1****2.8. Costes**

En primer lugar, cabe destacar que este proyecto ha decidido realizarse bajo coste cero. Por este motivo, aunque no el único, se han elegido tecnologías libres para así evitar pagar un coste por licencias de uso. Véase el sub apartado **Tecnologías de programación** dentro del apartado **2.6.2. Desarrollo** para obtener un listado detallado de las mismas.

En el apartado de costes de recursos humanos, se efectuará una evaluación hipotética basada en una aproximación precio/hora que cobrarían las personas implicadas en el desarrollo del proyecto. Para un proyecto de estas características, el *project team* estaría formado por un director de proyecto, un analista y un programador que también desarrollaría la fase de pruebas. A continuación, se muestra una tabla que describe las responsabilidades derivadas de cada una de las personas que conforman el *project team* y el coste.



Cargo	Responsabilidad	Coste
Director de proyecto	Define, gestiona, planifica y controla el proyecto. Establece contacto con alguna parte interesada si fuera necesario.	45 €/hora
Analista	Colabora con el director de proyecto en el estudio de viabilidad y la planificación. Analiza la aplicación: arquitectura y metodología, especificaciones, estándares... Participa en el diseño.	35 €/hora
Programador	Diseña y desarrolla la aplicación siguiendo el análisis y la planificación previstas. Participa en la implantación y en el proceso de pruebas.	25 €/hora

Tabla 2.2

No obstante, tendremos en cuenta los costes necesarios para el desarrollo del proyecto, tanto de hardware como podría ser un ordenador, como de todo el software necesario e incluyendo también conceptos como el gasto mensual de la conexión ADSL.

A continuación, se muestra una tabla que desglosa todos los costes de forma detallada.

CONCEPTO	COSTE
<b>Costes de recursos humanos</b>	<b>8170 €</b>
Coste director de proyecto	<b>3150 €</b>
Coste analista	<b>770 €</b>
Coste programador	<b>4250 €</b>
<b>Costes de recursos hardware</b>	<b>645 €</b>
Ordenador portátil	<b>600 €</b>
Impresora	<b>45 €</b>
<b>Costes de recursos software</b>	<b>442.33 €</b>

Licencia sistema operativo Windows 7	<b>142.33 €</b>
Licencia Adobe Photoshop	<b>300 €</b>
Xampp 1.7.4	<b>0 €</b>
Notepad ++	<b>0 €</b>
Visual Paradigm 8.1	<b>0 €</b>
<b>Costes extra</b>	<b>180 €</b>
Conexión ADSL Telefónica	<b>30 €/MES X 6 MES.</b>
<b>COSTE TOTAL PROYECTO</b>	<b>9437.33 €</b>

Tabla 2.3

## 2.9. Riesgos

Debemos tener en cuenta una serie de riesgos vinculados a la creación de un proyecto que pueden alterar el resultado final, los costes totales o la cantidad de horas planificadas para el desarrollo del mismo.

Por parte del desarrollador, los factores pueden ser muy diversos. En primer lugar, se puede producir una falta de conocimiento de las herramientas a utilizar para la creación del proyecto lo cual puede alterar el desarrollo o afectar al resultado final esperado. Además, puede producir en consecuencia un incremento de las horas previstas en las diferentes fases del desarrollo.

Otro factor de riesgo puede derivarse de la seguridad de los datos ya que, al tratarse de una aplicación web, debemos tenerlo muy en cuenta. Debemos garantizar la privacidad de los datos, así como evitar intrusiones con cualquier finalidad de usuarios externos a la aplicación. También hay que proporcionar un sistema seguro de perfiles de usuarios, debido a que el nivel credenciales permitirá acceder a una serie de herramientas distintas.

No realizar correctamente una óptima fase de pruebas puede producir alteraciones en el resultado final que podrían afectar en diferentes niveles.

Otro aspecto que debemos tener en cuenta es la dificultad para acceder a las partes implicadas en el proyecto para obtener información de cualquier tipo necesaria para el desarrollo del mismo. Esto puede producir un incremento en el tiempo previamente planificado.

Por último, destacar que el incumplimiento de alguna norma o legislación vigente sería otro factor de riesgo que incluso podría tener repercusiones legales.

## 2.10. Conclusiones

Tras la conclusión del presente estudio de viabilidad, podemos concluir con la siguiente lista, un conjunto de ventajas que conllevan la realización de este proyecto.

- Difusión de la empresa en el ámbito de internet
- Automatización y optimización en la creación de presupuestos para obras
- Servicio básico de gráficos para el seguimiento del número de presupuestos y totales de dinero
- Soporte para facilitar la contabilidad posterior de los presupuestos
- Garantiza la movilidad para gestionar presupuestos desde cualquier lugar con un dispositivo móvil

Destacar también que la realización de este proyecto permite poner en práctica y ampliar en gran medida conocimientos adquiridos durante la formación en la carrera. Se utilizarán conocimientos ofrecidos en materias como Redes, Bases de Datos, Ingeniería del Software, Metodología y Tecnología de la Programación, entre otras.

Valorando estos aspectos, y teniendo en cuenta los costes y los riesgos derivados del proyecto podemos considerar este proyecto como viable.

## Capítulo 3. Fundamentos teóricos

### 3.1. Lenguajes de programación

#### 3.1.1. Lenguajes del lado cliente

##### HTML

El lenguaje HTML (*HyperText Markup Language*), definido por primera vez en 1990, es el lenguaje básico que predomina en la definición de páginas web. Se utiliza para describir la estructura y la información en forma de texto, así como incorporar ciertos elementos. Este lenguaje está formado por etiquetas delimitadas por corchetes angulares (<,>) que nos definirán los elementos que conforman la estructura básica del portal web en HTML.

Los elementos están formados por los atributos y el contenido. Los atributos, es lo que se encuentra en las etiquetas que se han citado arriba y el contenido está ubicado entre la etiqueta de apertura y la de cierre. No obstante, el lenguaje HTML tiene elementos que no tienen ni contenido ni etiqueta de cierre. Un ejemplo de estos sería el salto de línea, identificado con la etiqueta <br>.

A continuación, se detallará la estructura básica de un documento HTML.

En primer lugar, el inicio del documento se define con la etiqueta <html>, la cual le indica al navegador que lo que se encuentre después de esta etiqueta sea interpretado como código HTML.

A continuación, estaría la etiqueta <head>. Esta define la cabecera del documento HTML, dentro de la cual se puede encontrar información como el título de la página web definido con la etiqueta <title> y que aparecerá en la ventana del navegador, el vínculo a las hojas de estilo CSS que definen el diseño del sitio web mediante la etiqueta <link> y el vínculo a que queramos ejecutar al comienzo de interpretar el documento HTML usando la etiqueta <script>.

La etiqueta <body> define el contenido principal del documento HTML. En otras palabras, es la parte del documento HTML que mostrará el navegador por lo que aquí se encontrarán todos los elementos referentes al contenido.

El lenguaje HTML se distingue por su sencillez y facilidad de uso, permitiendo su aprendizaje incluso sin tener nociones de programación de ningún tipo.

##### Javascript

Javascript es un lenguaje de programación interpretado, que se utiliza principalmente en el lado del cliente.

Se define como orientado a objetos, basado en prototipos, imperativo y dinámico. No obstante, se pueden realizar implementaciones en Javascript en el lado del servidor utilizando por ejemplo *Node.js*.

Todos los navegadores modernos interpretan el código Javascript integrado dentro de las aplicaciones web. El DOM (*Document Object Model*) es una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo sobre como pueden combinarse dichos objetos y una interfaz estándar para acceder a ellos y manipularlos. Por ello, a través del DOM, es como las aplicaciones web pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML de manera dinámica utilizando el lenguaje Javascript.

Las sentencias escritas en lenguaje Javascript incluidas en una página web se encapsulan entre las etiquetas `<script>` y `</script>`.

A continuación, vamos a detallar sus principales características:

- **Imperativo y estructurado:** Javascript reúne buena parte de las estructuras de programación del lenguaje C. Las estructuras utilizadas son: secuencial, selectiva e iterativa.
- **Dinámico:** Las propiedades y los valores de los objetos en Javascript pueden ser creados, cambiados o eliminados en tiempo de ejecución.
- **Basado en prototipos:** Javascript utiliza prototipos en lugar de clases para el uso de la herencia.
- **Orientado a objetos:** Javascript dispone de fuertes capacidades de programación orientada a objetos. Se puede entender como un conjunto de objetos que cooperan para conformar una aplicación (DOM).

Veamos como se realizan las referencias a objetos en lenguaje Javascript. Supongamos este código el HTML de nuestra web:

```
<div id="container">...</div>
```

Para hacer referencia al objeto div con id="container" se puede hacer de diversas maneras:

```
document.div["container"]
```

```
document.div.container
```

```
document.getElementById("container")
```

Una vez establecida la referencia al objeto deseado, se podrían manipular las propiedades y las funciones de éste de manera dinámica y en tiempo de ejecución. Veámos un ejemplo de como sería:

```
document.getElementById("container").style.width = "300px";
```

En este ejemplo, le indicamos que la propiedad de anchura del elemento que tiene id="container" pasa a ser de 300 píxeles.

Por último, veremos cómo funciona el manejador de eventos en Javascript. Supóngase este ejemplo:

```
<div id="container" onClick="FuncionClick();">...</div>
```

En este ejemplo, se ejecuta la función FuncionClick() al producirse el evento de clicar sobre el div que tiene id="container". Esto mismo se podría realizar de la siguiente manera alternativa:

```
document.getElementById("container").onclick = FuncionClick();
```

### Librería JQuery

JQuery es una librería de Javascript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM (*Document Object Model*), manejar eventos, crear efectos y animaciones e interactuar con técnicas AJAX en las páginas web. Por esto, facilita en gran medida la modificación de contenido en una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Es software libre y de código abierto (*Open Source*). Consiste en un único fichero en lenguaje Javascript que contiene las funcionalidades comunes del árbol DOM, eventos, efectos y AJAX.

A continuación, se muestran algunos ejemplos de cómo el uso de JQuery simplifica la sintaxis de Javascript e incorpora funcionalidades muy potentes y sencillas de implementar.

La referencia a objetos se simplifica en gran medida:

```
$("#container")
```

```
$(".container")
```

El primer ejemplo hace referencia al objeto con id="container". El segundo, en cambio hace referencia a todos los objetos con class="container".

Veamos un ejemplo de la sencillez en la manipulación de objetos:

```
$(".container").removeClass("container").addClass("auxcontainer");
```

En este ejemplo, a todos los objetos con class="container" se les eliminan todas las propiedades que proporciona la class container y se les añaden las propiedades de la class auxcontainer.

A continuación, un ejemplo de cómo realizar efectos gráficos de manera muy sencilla:

```
$("#container").fadeOut("fast");
```

En este ejemplo se aplica un efecto rápido de fade que hace desaparecer el objeto con id="container".

Por último, veamos también la simplificación del manejador de eventos. Utilizaremos el mismo ejemplo anterior de Javascript:

```
<div id="container" onClick="FuncionClick();">...</div>
```

En JQuery se haría lo mismo de la siguiente forma:

```
$("#container").click(FuncionClick){
//contenido de la función FuncionClick()
};
```

### 3.1.2. Lenguajes del lado servidor

#### Programación del lado servidor

La programación del lado del servidor consiste en procesar peticiones de usuarios mediante la interpretación de scripts en el servidor web y generar dinámicamente contenido HTML como respuesta.

Un servidor web es un programa que procesa una aplicación del lado del servidor que se mantiene a la espera de peticiones de un cliente y responde a las mismas adecuadamente. La respuesta suele ser mostrada mediante una página web en el navegador generada con el contenido de la respuesta, proceso realizado por medio del protocolo HTTP. A continuación, se muestra una imagen para ilustrar mejor este proceso.

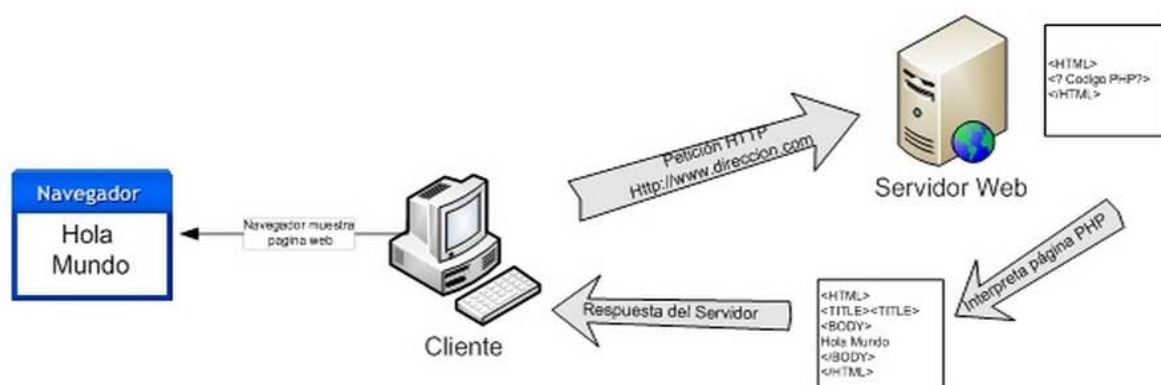


Figura 3.1

Las aplicaciones del lado del servidor son definidas como cualquier conjunto de instrucciones que tienen como finalidad ser procesadas por un servidor web para realizar alguna acción. Estas aplicaciones, son una muy buena opción para desarrollar webs, ya que, al ejecutarse en el servidor y no en la máquina del cliente éste no necesita ninguna capacidad añadida, más allá de cualquier navegador básico.

Las aplicaciones del lado del servidor están escritas mediante lenguajes de programación entre los que destacan los siguientes:

- PHP
- ASP
- JSP
- Python
- Ruby
- Perl
- CGI

Para el desarrollo de este proyecto se ha decidido utilizar el lenguaje PHP. Explicaremos los motivos que nos ha llevado a ello y compararemos este lenguaje con dos de sus competidores como son ASP y JSP, analizando algunas de sus características, ventajas e inconvenientes.

## PHP

PHP (*PHP Hypertext Pre-processor*) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en la interpretación del lado del servidor (*server-side scripting*) aunque actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluso con interfaz gráfica utilizando librerías.

El lenguaje PHP no necesita ser compilado para ejecutarse. Para su funcionamiento, necesita de *Apache*, *ISS*, o algún otro servidor web. Este lenguaje nos permite la conexión con la mayoría de sistemas gestores de bases de datos (*MySQL*, *PostgreSQL*, *Oracle*, *Microsoft SQL Server*...). Además, puede ser desplegado en casi todos los sistemas operativos y plataformas sin coste alguno.

La licencia de PHP es una licencia de software libre y de código abierto (*open source*). Esta licencia está diseñada para incentivar la distribución del código fuente libremente, cumpliendo los siguientes requisitos:

- Se incluya la declaración de los derechos de autor de la licencia PHP



- La palabra "PHP" no se use en el título de las obras derivadas
- Se incluya el siguiente anuncio bajo cualquier forma en la que se redistribuya el código:  
*This product includes PHP software, freely available from <<http://www.php.net/software>>*

Estos motivos que se han comentado, unido a las nociones que el proyectista tenía en PHP, han llevado a tomar la decisión de usar el lenguaje PHP como lenguaje de programación del lado del servidor para el desarrollo de este proyecto.

### **Características**

- Soporte mejorado para la programación orientada a objetos
- Mejoras de rendimiento
- Mejor soporte para MySQL
- Mejor soporte para XML (*eXtensible Markup Language*)
- Soporte nativo para SQLite
- Soporte integrado para SOAP (Protocolo Simple de Acceso a Objetos)
- Sintaxis basada en C, Java y Perl

### **Ventajas**

- Lenguaje de programación libre
- Multiplataforma
- Manejo de excepciones
- Biblioteca nativa de funciones
- Permite en cierta medida la programación orientada a objetos
- Utilización de módulos para expandir su potencial
- Destacada conectividad con MySQL, y con la mayoría de sistemas gestores de bases de datos
- Amplia documentación y ejemplos en página oficial ([www.php.net/manual/es/](http://www.php.net/manual/es/))

### **Inconvenientes**

- La programación orientada a objetos es aún deficiente para aplicaciones grandes
- Todo el trabajo se realiza en el lado del servidor y nada en el lado cliente. Por lo tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número
- La compresión del código puede verse afectada al mezclar sentencias HTML y PHP
- Dificulta la organización por capas de la aplicación

## ASP

ASP (*Active Server Pages*) es una tecnología del lado del servidor desarrollado por Microsoft para el desarrollo de sitios web dinámicos. Las páginas web creadas bajo este lenguaje necesitan tener instalado IIS (*Internet Information Services*), que es un servidor web y un conjunto de servicios (FTP, SMTP, NNTP y HTTP/HTTPS) para el sistema operativo Microsoft Windows, por lo que a priori, están limitadas a funcionar en este sistema operativo.

Este lenguaje no necesita ser compilado para ejecutarse. El código ASP puede ser insertado junto con código en lenguaje HTML.

Las páginas creadas con tecnología ASP funcionan en todo tipo de navegadores.

## Ventajas

- Completamente orientado a objetos
- Controles de usuario
- División entre la capa de aplicación o diseño y el código
- Facilita el mantenimiento de grandes aplicaciones
- Incremento de la velocidad de respuesta del servidor
- Mayor velocidad
- Mayor seguridad

## Inconvenientes

- Tecnología propietaria

- Mayor consumo de recursos
- Código desorganizado y poco óptimo para realizar funciones sencillas
- Hospedaje de sitios web costosos

## JSP

JSP (*Java Server Pages*) es una tecnología para desarrollar páginas web dinámicas. JSP es similar a PHP pero usa el lenguaje de programación Java, por lo tanto se trata de programación orientada a objetos. Este lenguaje necesita ser compilado, y esto se hará habitualmente en la primera petición realizada. Además, esta tecnología permite separar la parte dinámica de la estática en las páginas web. Las páginas web con tecnología JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

Para usar esta tecnología, es necesario instalar Tomcat, que es el contenedor de servlets de Java usado en la referencia oficial de la implementación de JSP. Los servlets son programas Java destinados a ejecutarse en el servidor.

El motor de las páginas JSP está basado en los servlets de Java. Éstos se encontrarán en archivos de extensión *.jsp*, que incluyen las sentencias Java que se ejecutarán en el servidor dentro de la estructura de etiquetas HTML, como se ha comentado arriba.

En primer lugar, el motor JSP llevará a cabo una fase de traducción de la página en un servlet. Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la página *.jsp* como también se ha comentado arriba. No obstante, existe la opción de pre-compilar el código para evitar el tiempo de espera del cliente la primera vez que solicite la página.

## Ventajas

- Permite separar en capas las aplicaciones web
- Multiplataforma
- Ejecución rápida de servlets
- Código bien estructurado
- Integridad con los módulos de Java

## Inconvenientes

- Complejidad de aprendizaje
- Requiere conocimiento de Java (excepciones, Tomcat, programación orientada a objetos...)
- Lento en la primera compilación en caso de no pre-compilar el código
- El hosting en Java es más caro
- Desconocimiento de la tecnología por parte de la mayoría de clientes

### 3.2. Sistema gestor de bases de datos

Un SGBD (Sistema Gestor de Bases de Datos) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos y proporciona herramientas para añadir, borrar, modificar y analizar los datos.

Además, los SGBD proporcionan métodos para asegurar la integridad de los datos, administrar el acceso de los usuarios a los datos, recuperar información del sistema y realizar copias de seguridad.

Para el desarrollo del proyecto se ha optado por MySQL como SGBD. MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

A continuación, detallaremos un listado con las principales ventajas de MySQL como sistema gestor de bases de datos que se han tenido en cuenta para ser el utilizado en este proyecto.

- Es código abierto (*Open Source*)
- Gran velocidad a la hora de realizar operaciones, lo que le convierte en uno de los sistemas gestores con mejor rendimiento
- Facilidad de configuración e instalación
- Baja probabilidad de corromper datos, incluso si los errores no son causados en el propio sistema gestor
- Escalable
- Acceso a la base de datos de forma simultanea
- Permite ser ejecutado en una máquina con escasos recursos

Además, el software XAMPP cuenta con phpMyAdmin, una herramienta para facilitar la administración de bases de datos en MySQL.

## Capítulo 4. Análisis del proyecto

### 4.1. Introducción

En este capítulo se realizará una descripción detallada del comportamiento de la aplicación, analizando en profundidad los módulos y las funcionalidades del portal. Además, se detallarán las acciones que podrán llevar a cabo los diferentes perfiles de usuario definidos en el Sistema. Como ya hemos visto en el estudio de viabilidad del capítulo 2, los diferentes tipos de usuario son los siguientes:

- Usuario no registrado o visitante
- Encargado
- Contable
- Director
- Administrador

El análisis incluye un apartado de requerimientos funcionales que mostrará diagramas de casos de uso para cada tipo de perfil de usuario y los principales diagramas de secuencia de cada módulo para ilustrar mejor su funcionamiento.

### 4.2. Requerimientos funcionales

A continuación, se describen las principales funcionalidades o requerimientos funcionales del portal web para cada perfil de usuario.

#### Requerimientos funcionales del usuario no registrado/visitante

Este perfil de usuario podrá navegar por la parte no protegida del portal web. Dispondrá de las siguientes funcionalidades:

- Ver los servicios que ofrece la empresa
- Consultar los principales proyectos realizados por la empresa
- Ver la ubicación de la empresa
- Contactar con la empresa mediante un formulario u otras vías

#### Requerimientos funcionales del perfil usuario encargado

Este usuario tendrá acceso a la parte privada de la web. Podrá acceder al módulo de presupuestos para diversas funciones que a continuación se describen:

- Consultar presupuestos
- Crear presupuestos
- Editar presupuestos
- Crear recursos/items
- Editar recursos/items

#### **Requerimientos funcionales del perfil usuario contable**

Este usuario tendrá acceso a la parte privada de la web. Podrá acceder al módulo de economía y al módulo de estadísticas. Las funcionalidades que podrá realizar son las siguientes:

- Consultar totales presupuestos
- Consultar pagos presupuestos
- Consultar estadísticas totales de pagos
- Consultar estadísticas del número de presupuestos

#### **Requerimientos funcionales del perfil usuario director**

Este usuario tendrá acceso a la parte privada del portal. Podrá acceder a todos los módulos y realizar las siguientes funcionalidades:

- Consultar presupuestos
- Crear presupuestos
- Editar presupuestos
- Crear recursos/items
- Editar recursos/items
- Consultar totales presupuestos
- Consultar pagos presupuestos

- Consultar estadísticas totales de pagos
- Consultar estadísticas del número de presupuestos
- Consultar empresas
- Crear nueva empresa
- Editar empresa
- Eliminar empresa
- Consultar usuarios
- Crear nuevo usuario
- Editar usuario
- Eliminar usuario

#### **Requerimientos funcionales del perfil usuario administrador**

Este usuario tendrá acceso a la parte privada del portal. Podrá acceder a todos los módulos y realizar las siguientes funcionalidades:

- Consultar presupuestos
- Crear presupuestos
- Editar presupuestos
- Crear recursos/items
- Editar recursos/items
- Consultar totales presupuestos
- Consultar pagos presupuestos
- Consultar estadísticas totales de pagos
- Consultar estadísticas del número de presupuestos
- Consultar empresas
- Crear nueva empresa



- Editar empresa
- Eliminar empresa
- Consultar usuarios
- Crear nuevo usuario
- Editar usuario
- Eliminar usuario
- Consultar logs conexión/desconexión de usuarios

#### 4.2.1. Diagramas de casos de uso

##### Usuario no registrado

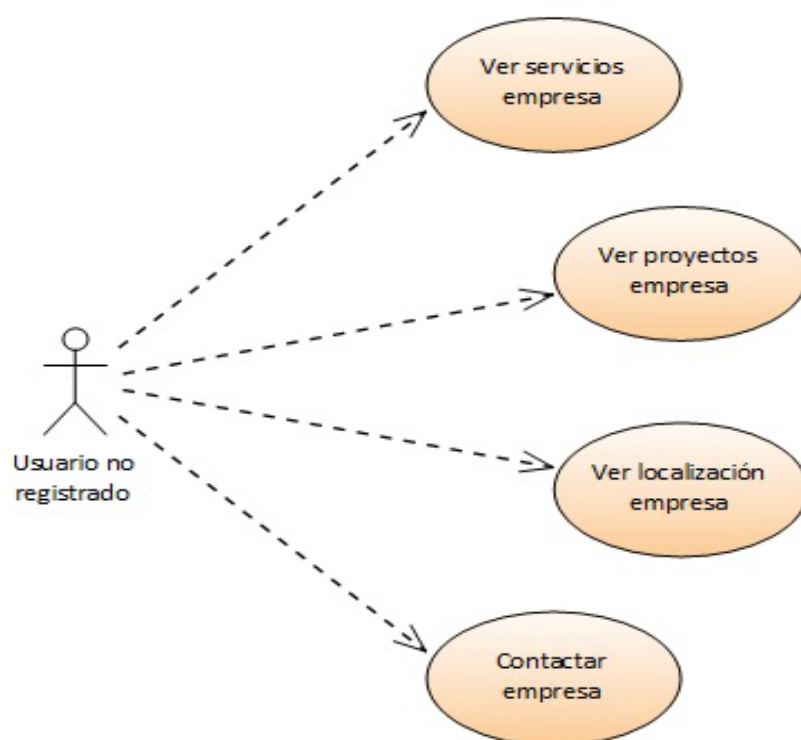


Figura 4.1

### Perfil de usuario encargado

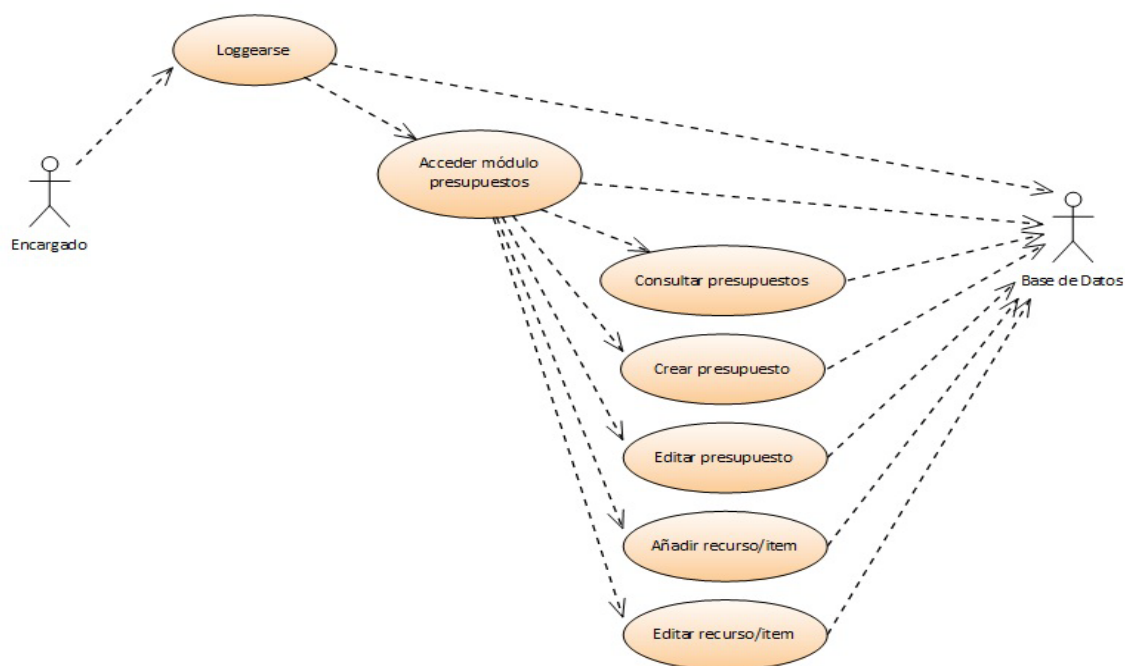


Figura 4.2

### Perfil de usuario contable

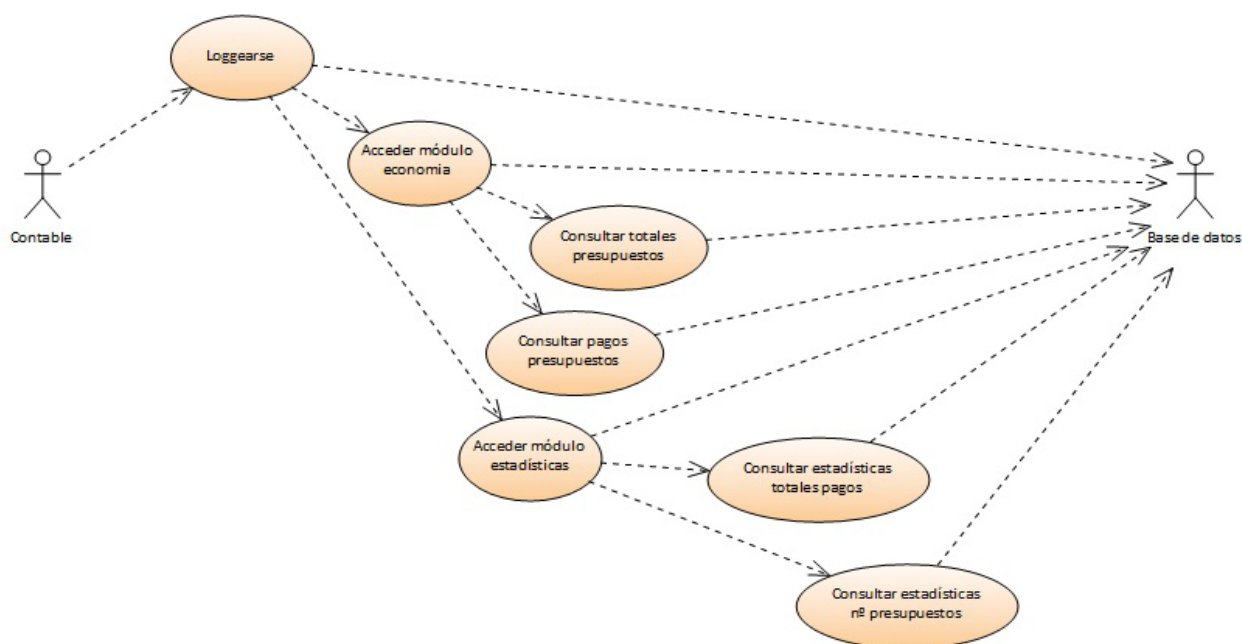


Figura 4.3

## Perfil de usuario director



Figura 4.4

## Perfil de usuario administrador

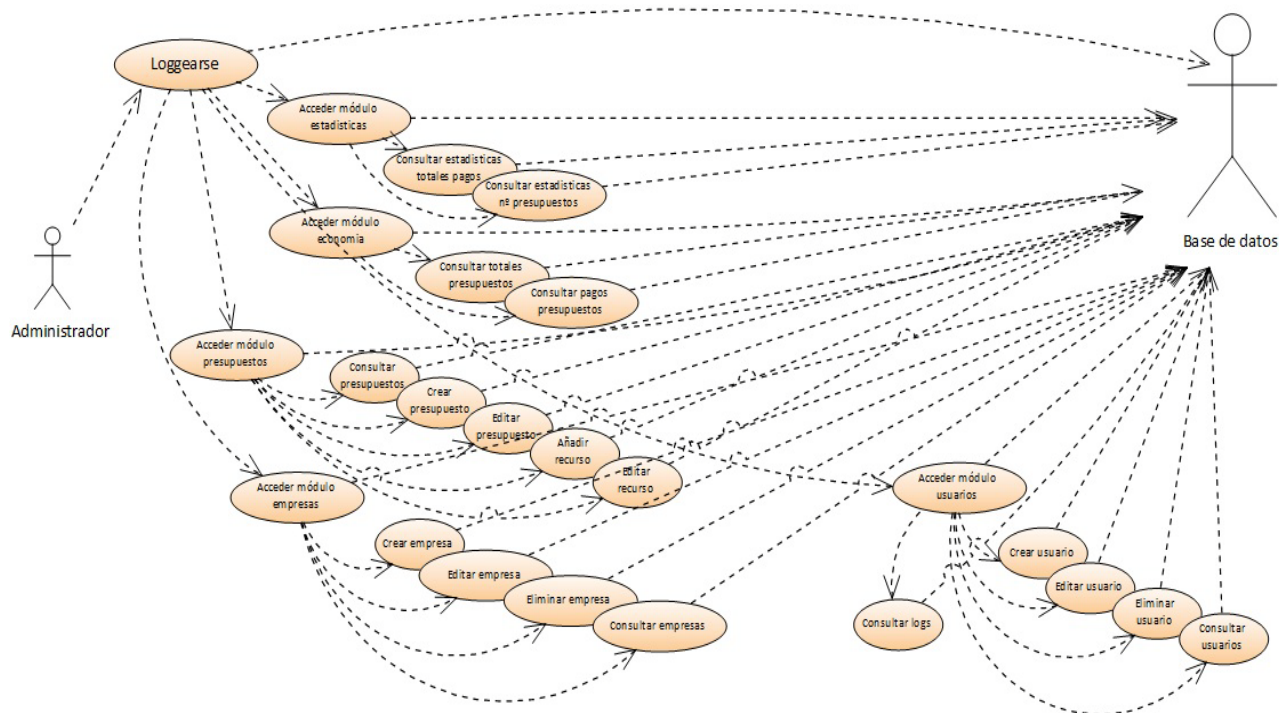


Figura 4.5

### 4.2.2. Diagramas de secuencia

A continuación, mediante diagramas de secuencia se describen una selección de casos de uso que por su funcionamiento se consideran relevantes. Se omiten aquellos que su funcionamiento es exactamente igual a otros que se describen excepto que interactúa otro módulo de la aplicación web.

#### Caso de uso: Login

- Actores: Administrador, Director, Contable y Encargado.
- Descripción: el usuario se identifica en el sistema para acceder.

Actor	Sistema
1. El usuario introduce su nombre de usuario y contraseña en el formulario de acceso.  3.El usuario accede a la parte privada del portal web.	2. El modulo de acceso comprueba si los datos son correctos en la BBDD y si lo son redirige al usuario a la parte privada.

Tabla 4.1

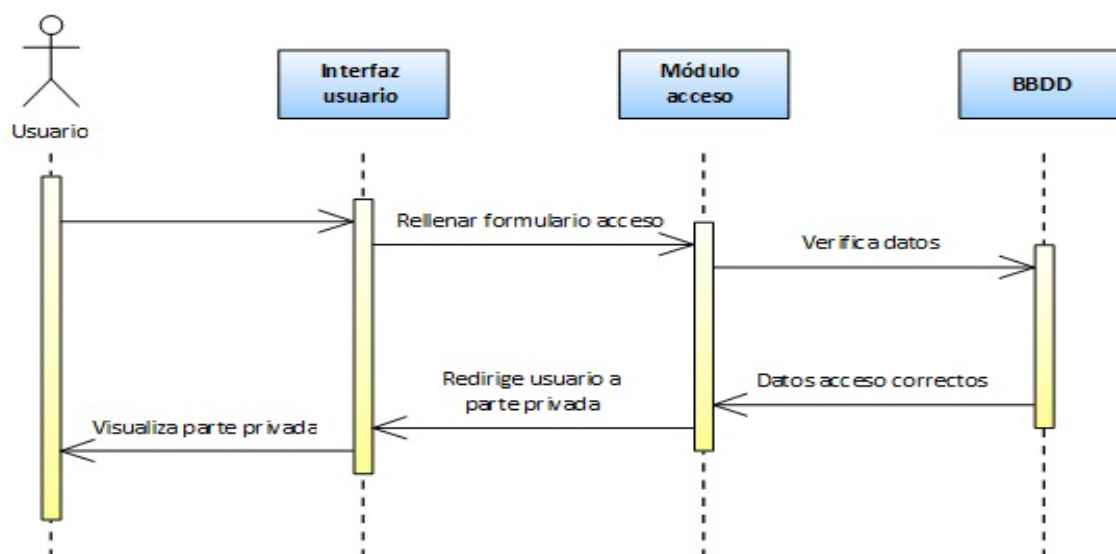


Figura 4.6

**Caso de uso: Crear Usuario**

- Actores: Administrador y Director.
- Descripción: El actor crea un nuevo usuario del sistema.

Actor	Sistema
1. El actor rellena el formulario con los datos del usuario a crear.          4. Visualiza todos los usuarios con el nuevo usuario ya insertado.	2. El modulo de usuarios añade los datos a la BBDD.  3. Se devuelve una confirmación de la inserción correcta del usuario.

Tabla 4.2

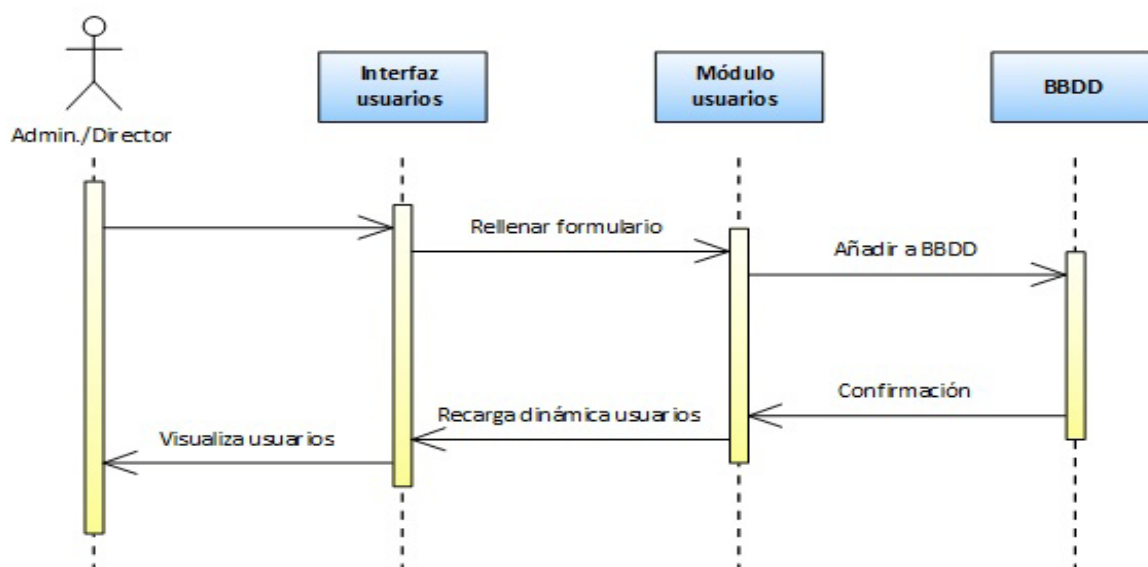


Figura 4.7

**Caso de uso: Editar Usuario**

- Actor: Administrador y Director.
- Descripción: El actor edita la información de un usuario ya existente.

Actor	Sistema
1. El actor rellena el formulario con los datos a modificar del usuario.          4. Visualiza todos los usuarios con los cambios ya realizados.	2. El modulo de usuarios edita los datos en la BBDD.   3. Se retornan los datos al modulo de usuarios.

Tabla 4.3

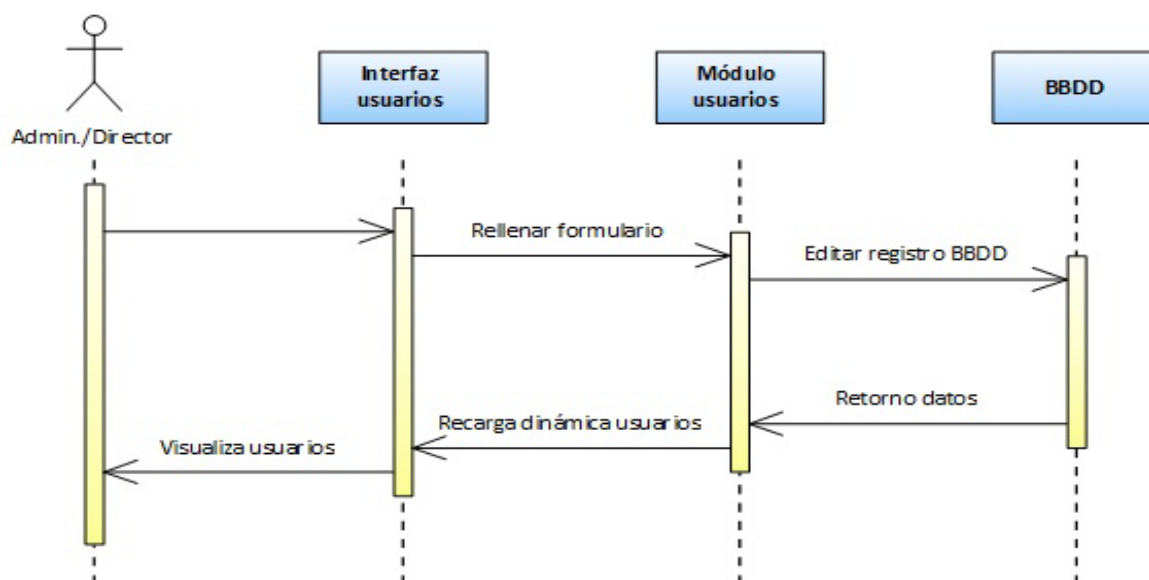


Figura 4.8

**Caso de uso: Eliminar Usuario**

- Actor: Administrador y Director.
- Descripción: El actor elimina un usuario ya existente.

Actor	Sistema
1. El actor confirma que desea eliminar el usuario seleccionado.          4. Visualiza todos los usuarios con el usuario ya eliminado.	2. El modulo de usuarios elimina el registro en la BBDD.  3. Se retornan los datos al modulo de usuarios.

Tabla 4.4

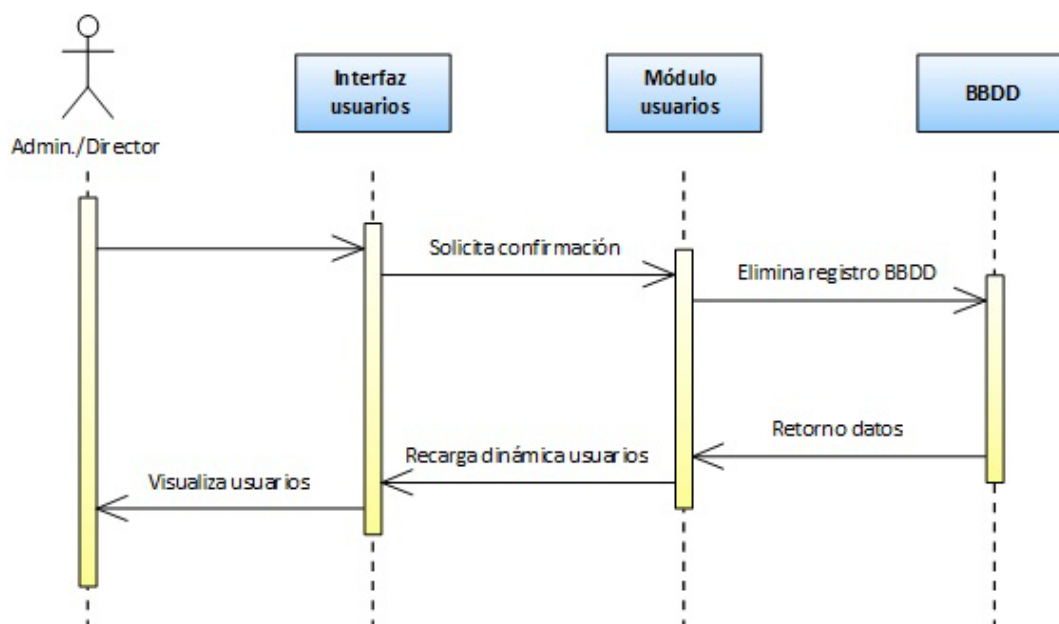


Figura 4.9

**Caso de uso: Consultar Usuarios**

- Actor: Administrador y Director.
- Descripción: El actor consulta todos los usuarios del sistema.

Actor	Sistema
1. El actor solicita los datos de los usuarios.	2. El módulo de usuarios realiza una consulta en la BBDD.
	3. Se retornan los datos al módulo de usuarios.
4. Visualiza todos los usuarios.	

Tabla 4.5

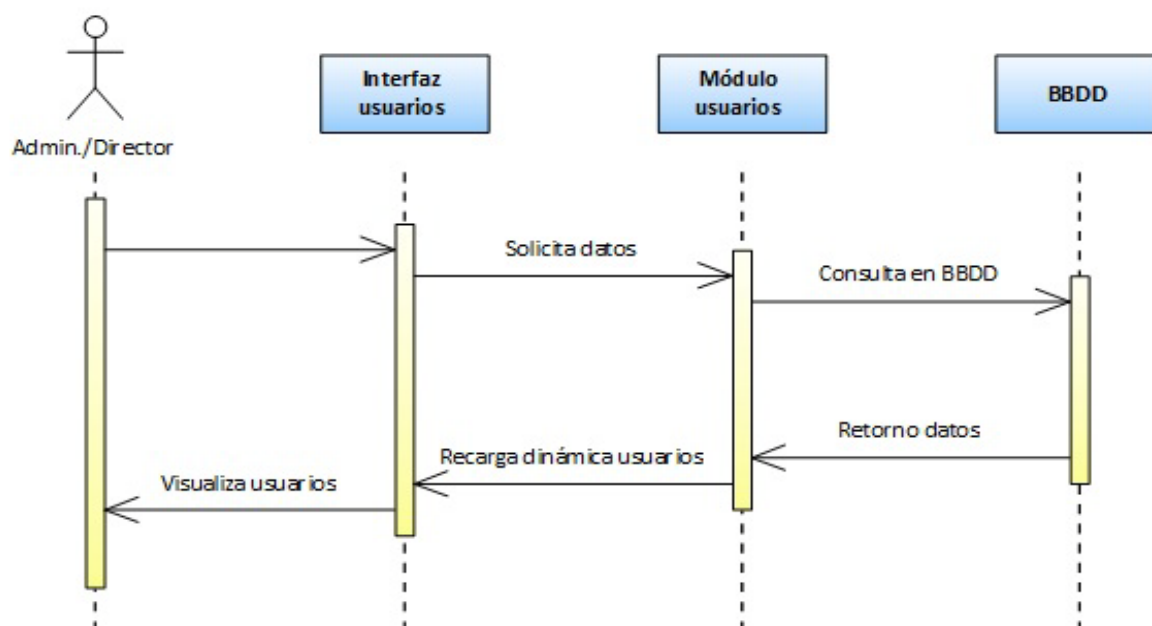


Figura 4.10



**Caso de uso: Editar Presupuestos (Añadir recurso)**

- Actor: Encargado, Administrador y Director.
- Descripción: El actor modifica un presupuesto aún no cerrado.

Actor	Sistema
<p>1. El actor consulta los presupuestos abiertos.</p> <p>4. Visualiza todos los presupuestos abiertos.</p> <p>5. El actor selecciona el presupuesto que desea modificar.</p> <p>7. Visualiza en detalle el presupuesto seleccionado.</p> <p>8. Selecciona el recurso que desea añadir al presupuesto.</p> <p>11. Visualiza el detalle del presupuesto con el recurso añadido.</p>	<p>2. El módulo de presupuestos realiza una consulta en la BBDD.</p> <p>3. Se retornan los datos al módulo de presupuestos.</p> <p>6. El módulo de presupuestos realiza una consulta en la BBDD.</p> <p>9. El módulo de presupuestos realiza una inserción del recurso en la BBDD.</p> <p>10. Se retornan los datos al módulo de presupuestos.</p>

Tabla 4.6

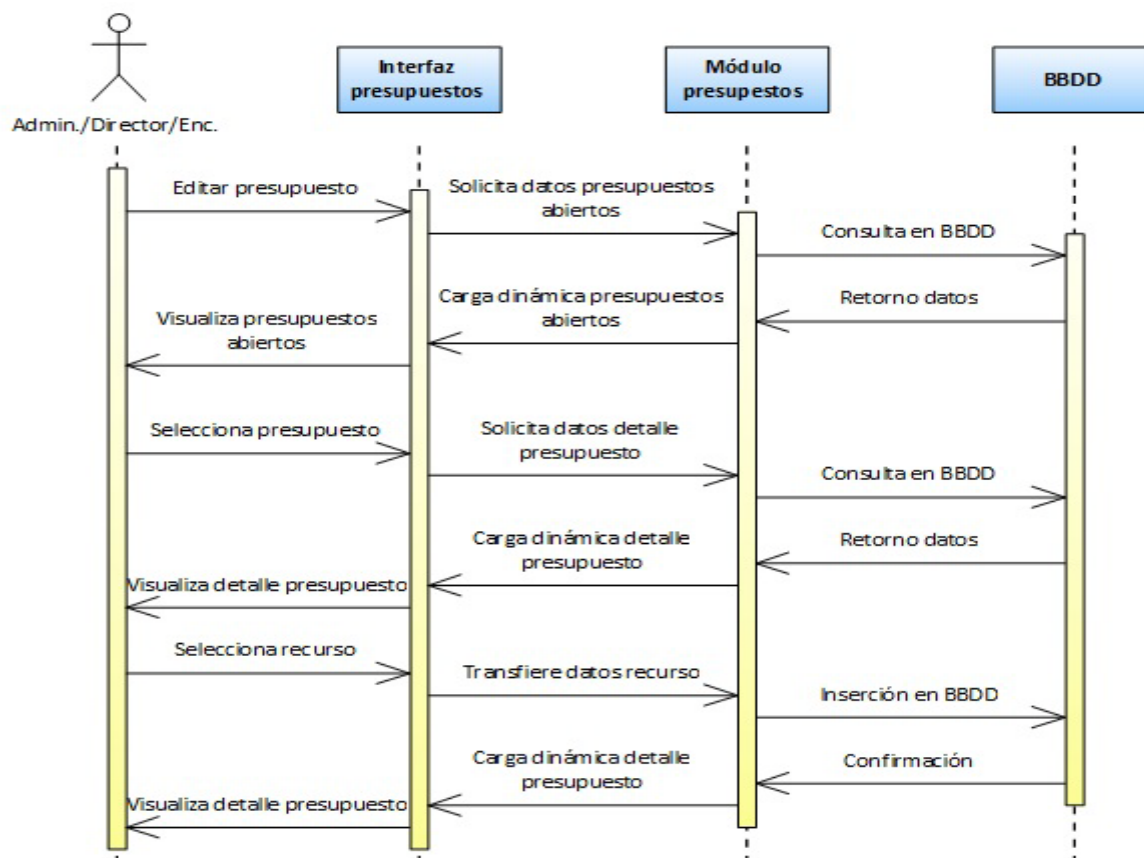


Figura 4.11

### Caso de uso: Consultar Logs

- Actor: Administrador.
- Descripción: El actor administrador consulta los logs de conexión/desconexión de los usuarios en el sistema.

Actor	Sistema
1. El actor solicita los datos de los logs.	2. El módulo de usuarios realiza una consulta en la BBDD. 3. Se retornan los datos al módulo de usuarios.
4. Visualiza todos los logs de los usuarios.	

Tabla 4.7

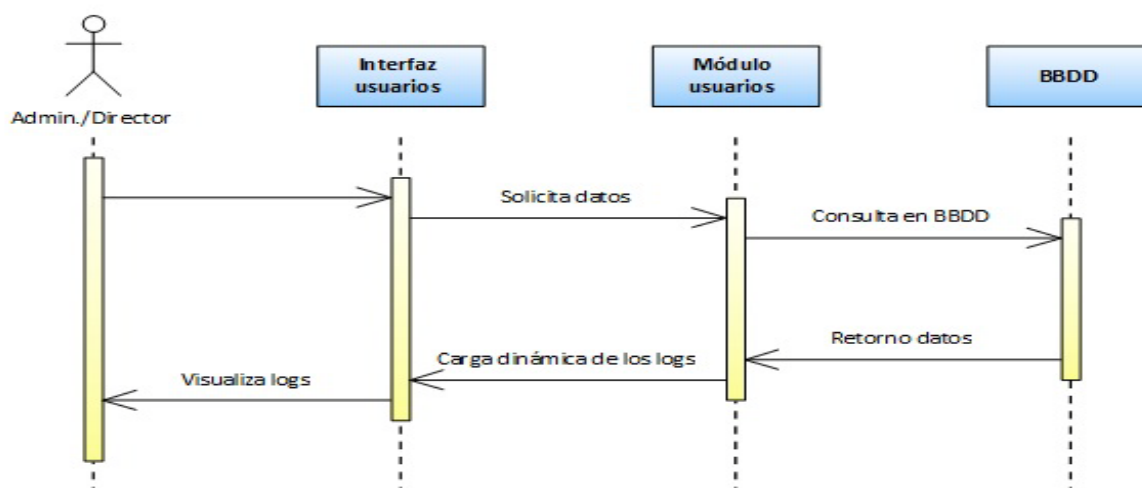


Figura 4.12

### 4.3. Requerimientos no funcionales

#### 4.3.1. Restricciones de diseño

##### Seguridad

Debido a las características del portal, ha sido necesario implementar cuatro tipos de usuario diferentes. Cada perfil de usuario tendrá acceso a unos contenidos y herramientas diferentes.

Para solventar la necesidad de gestionar el acceso a los contenidos y herramientas, la aplicación trabajará con sesiones en PHP para comprobar qué tipo de usuario está loggeado y que contenidos o herramientas se le pueden mostrar.

Si algún usuario intenta visualizar mediante la url del navegador contenidos a los que no tiene acceso, se notificará que carece de permisos para acceder y obviamente no se mostrará el contenido.

Utilización de la función `mysql_real_escape_string()` de PHP para el login de usuarios y de esta forma evitar ciertos ataques de SQL injection para poder loggearse a la parte privada sin permiso.

##### Prevención de errores en los formularios

Diversas funcionalidades de la parte privada del portal web requieren de la utilización de formularios para interactuar con la base de datos en distintas operaciones. Por tanto, será necesario comprobar cuántos campos sean necesarios para no comprometer y garantizar la integridad de la base de datos.

### 4.3.2. Objetivos de diseño

#### Programación estructurada

El portal web que desarrollamos en este proyecto es susceptible a sufrir mejoras o ampliaciones en las funcionalidades o contenidos. Por este motivo, es conveniente que toda la programación sea lo más estructurada posible y de manera modular para facilitar al máximo la comprensión del código fuente de cara al mantenedor del portal.

La utilización de CSS (Cascading Style Sheets) permite realizar cambios en el diseño del portal de manera sencilla. Además, la utilización de la librería JQuery de Javascript nos permite una gran flexibilidad a la hora de realizar cambios de manera dinámica en el diseño y la modificación o inserción de efectos de manera muy sencilla.

También se han incorporado las funciones Javascript relativas a la operatividad de la aplicación en scripts aparte, obviamente con el objetivo de modularizar el código y estructurarlo de la manera más clara posible.

#### Accesibilidad

De cara al usuario, debe facilitarse al máximo el acceso a las diferentes funcionalidades de las que dispone, intentando que la utilización del portal sea lo más intuitiva y sencilla posible, de manera que el usuario tenga a su alcance el mayor número de funcionalidades en el menor número de clics posible.

#### Claridad

Se debe utilizar un diseño que presente los datos de forma clara al usuario, facilitando la comprensión de los mismos. Además, debe tratarse de un diseño formal ya que desarrollamos un portal web para una empresa.

#### Tecnología

Se busca utilizar tecnologías de última generación que permitan mejorar la experiencia del usuario.

### 4.4. Estructura y diseño Base de Datos

La estructura de la base de datos está compuesta por un total de 13 tablas diseñadas en función de las necesidades del proyecto. Cada tabla contendrá campos de diferentes tipos donde almacenaremos la información deseada. Además, existirán relaciones entre las diferentes tablas, para facilitar el uso de la base de datos y crear consultas complejas.

A continuación se muestra un diagrama de la base de datos y su estructura con las tablas y sus respectivas relaciones, para ilustrar mejor la explicación posterior de todas las entidades que la conforman.

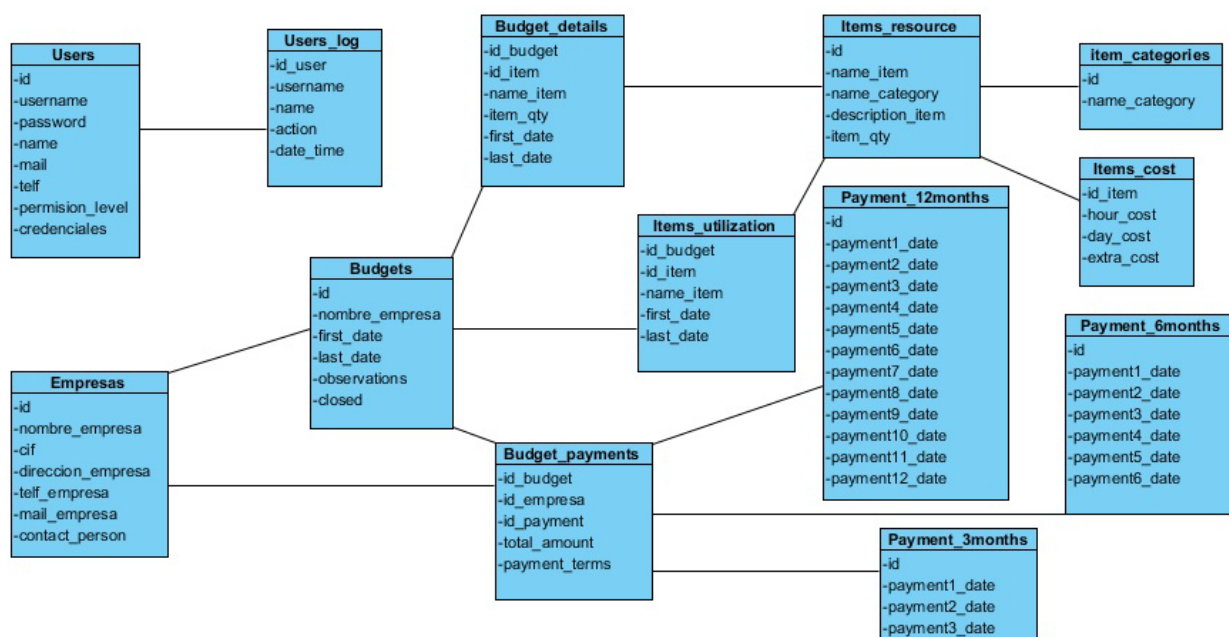


Figura 4.13

A continuación, una descripción de las diferentes tablas y campos más importantes que la componen.

### Tabla Users

Tabla donde almacenaremos la información referente a los usuarios. Cuenta con los siguientes campos:

- id: clave primaria e identificador único. Tipo entero y auto-incremental.
- username: nombre que tendrá el usuario para loggarse. Tipo varchar.
- password: contraseña que tendrá el usuario para loggarse encriptada en MD5. Tipo varchar.
- name: nombre real del usuario. Tipo varchar.
- mail: e-mail del usuario. Tipo varchar.
- telf: teléfono del usuario. Tipo varchar.
- permission\_level: número que indica el nivel de credenciales del usuario. Tipo entero.
- credenciales: permisos de que dispone el usuario. Tipo varchar.

### Tabla Users\_log

Tabla donde almacenaremos la actividad de login y logout de los diferentes usuarios. Cuenta con los siguientes campos:

- `id_user`: clave foránea al atributo *id* de la tabla Users. Tipo entero.
- `username`: nombre de usuario del que hace el login. Tipo varchar.
- `name`: nombre real del usuario. Tipo varchar.
- `action`: acción que lleva a cabo el usuario login/logout.
- `date_time`: día y hora en que se realiza la acción. Tipo datetime.

### Tabla Empresas

Tabla donde almacenaremos la información relativa a las empresas que encargan proyectos. Cuenta con los siguientes campos:

- `id`: clave primaria e identificador único. Tipo entero y auto-incremental.
- `nombre_empresa`: especifica el nombre de la empresa. Tipo varchar.
- `cif`: contiene el cif de la empresa. Tipo varchar.
- `direccion_empresa`: dirección de la empresa. Tipo varchar.
- `telf_empresa`: teléfono de la empresa. Tipo varchar.
- `mail_empresa`: e-mail de la empresa. Tipo varchar.
- `contact_person`: nombre de la persona de contacto de la empresa. Tipo varchar.

### Tabla Budgets

Tabla que contiene información de los presupuestos de forma genérica. Cuenta con los siguientes campos:

- `id`: clave primaria e identificador único. Tipo entero y auto-incremental.
- `id_empresa`: clave foránea al atributo *id* de la tabla empresas. Tipo entero.
- `first_date`: fecha en la que comienza el proyecto. Tipo date.
- `last_date`: fecha en la que finaliza el proyecto. Tipo date.
- `observaciones`: observaciones sobre el proyecto. Tipo varchar.

### Tabla Budget\_details

Tabla que contiene información del detalle de los presupuestos. Cuenta con los siguientes campos:

- id\_budget: clave foránea al atributo *id* de la tabla budgets. Tipo entero.
- id\_item: clave foránea al atributo *id* de la tabla items. Tipo entero.
- name\_item: nombre del item/recurso utilizado. Tipo varchar.
- item\_qty: cantidad utilizada del item/recurso. Tipo entero.
- first\_date: fecha en la que empieza la utilización del item/recurso. Tipo date.
- last\_date: fecha en la que finaliza la utilización del item/recurso. Tipo date.

### Tabla Budget\_payments

Tabla que contiene información sobre los pagos de un proyecto. Cuenta con los siguientes campos:

- id\_budget: clave foránea al atributo *id* de la tabla budgets. Tipo entero.
- id\_empresa: clave foránea al atributo *id* de la tabla empresas. Tipo entero.
- id\_payment: id\_empresa: clave foránea al atributo *id* de las tablas payment\_3months/ payment\_6months/ payment\_12months dependiendo del valor del atributo payment\_terms de esta tabla. Tipo entero.
- total\_amount: coste total del proyecto. Tipo float.
- payment\_terms: número de plazos a pagar el proyecto. Tipo entero.

### Tabla Items\_utilization

Tabla que contiene la información sobre la utilización de los items/recursos. Cuenta con los siguientes campos:

- id\_budget: clave foránea al atributo *id* de la tabla budgets. Tipo entero.
- id\_item: clave foránea al atributo *id* de la tabla items\_resource. Tipo entero.
- name\_item: nombre del item/recurso utilizado. Tipo varchar.
- first\_date: fecha en la que empieza la utilización del item/recurso. Tipo date.
- last\_date: fecha en la que finaliza la utilización del item/recurso. Tipo date.

### Tabla Items\_resource

Tabla que contiene información sobre los items/recursos de los que dispone la empresa. Cuenta con los siguientes campos:

- id: clave primaria e identificador único. Tipo entero y auto-incremental.
- name\_item: especifica el nombre del item/recurso. Tipo varchar.
- name\_category: nombre de la categoria a la que pertenece el item/recurso. Tipo varchar.
- description\_item: descripción del item/recurso. Tipo varchar.
- item\_qty: cantidad disponible de este item/recurso por la empresa. Tipo entero.

### Tabla Item\_categories

Tabla que contiene las distintas categorías en las que se clasificarán los items/recursos disponibles. Cuenta con los siguientes campos:

- id: clave primaria e identificador único. Tipo entero y auto-incremental.
- name\_categoria: nombre de la categoria para clasificar los items/recursos. Tipo varchar.

### Tabla Items\_cost

Tabla que contiene información sobre los costes de los diferentes items/recursos. Cuenta con los siguientes campos:

- id\_item: clave foránea al atributo *id* de la tabla items\_resource. Tipo entero.
- hour\_cost: coste por hora del item/recurso. Tipo float.
- day\_cost: coste por día del item/recurso. Tipo float.
- extra\_cost: coste extraordinario del item/recurso. Tipo float.

### Tabla Payment\_3months

Tabla que dispondrá de la información de las fechas de los pagos por los proyectos financiados a 3 meses. Cuenta con los siguientes campos:

- id: clave primaria e identificador único. Tipo entero y auto-incremental.
- payment1\_date: fecha en que se debe realizar el primer pago. Tipo date.



- payment2\_date: fecha en que se debe realizar el segundo pago. Tipo date.
- payment3\_date: fecha en que se debe realizar el tercer pago. Tipo date.

### **Tabla Payment\_6months**

Tabla que dispondrá de la información de las fechas de los pagos por los proyectos financiados a 6 meses.

Cuenta con los siguientes campos:

- id: clave primaria e identificador único. Tipo entero y auto-incremental.
- payment1\_date: fecha en que se debe realizar el primer pago. Tipo date.
- payment2\_date: fecha en que se debe realizar el segundo pago. Tipo date.
- payment3\_date: fecha en que se debe realizar el tercer pago. Tipo date.
- payment4\_date: fecha en que se debe realizar el cuarto pago. Tipo date.
- payment5\_date: fecha en que se debe realizar el quinto pago. Tipo date.
- payment6\_date: fecha en que se debe realizar el sexto pago. Tipo date.

### **Tabla Payment\_12months**

Tabla que dispondrá de la información de las fechas de los pagos por los proyectos financiados a 12 meses.

Cuenta con los siguientes campos:

- id: clave primaria e identificador único. Tipo entero y auto-incremental.
- payment1\_date: fecha en que se debe realizar el primer pago. Tipo date.
- payment2\_date: fecha en que se debe realizar el segundo pago. Tipo date.
- payment3\_date: fecha en que se debe realizar el tercer pago. Tipo date.
- payment4\_date: fecha en que se debe realizar el cuarto pago. Tipo date.
- payment5\_date: fecha en que se debe realizar el quinto pago. Tipo date.
- payment6\_date: fecha en que se debe realizar el sexto pago. Tipo date.
- payment7\_date: fecha en que se debe realizar el séptimo pago. Tipo date.
- payment8\_date: fecha en que se debe realizar el octavo pago. Tipo date.

- payment9\_date: fecha en que se debe realizar el noveno pago. Tipo date.
- payment10\_date: fecha en que se debe realizar el décimo pago. Tipo date.
- payment11\_date: fecha en que se debe realizar el undécimo pago. Tipo date.
- payment12\_date: fecha en que se debe realizar el duodécimo pago. Tipo date.

## 4.5. Menús de navegación

Uno de los módulos más importantes del portal serían los menús de navegación, con los que el usuario podrá moverse por el portal web y proceder a la búsqueda de información o funcionalidades.

Nos encontraremos con 2 módulos, el menú de navegación de la parte pública y el menú de navegación de la parte privada.

En este apartado también se muestran algunas captura de pantalla para dar una idea visual del proyecto y de las diferentes secciones que se van analizando.

### 4.5.1. Menú navegación parte pública

Está formado por cuatro apartados: inicio, obras y servicios, situación, contacto.

#### **Inicio**

Es la sección inicial de la web, la primera que nos encontramos al entrar al portal web. Contiene algo de información sobre la empresa.

#### **Obras y servicios**

Esta sección contiene información más detallada sobre los servicios que ofrece la empresa. También podemos consultar un listado con los principales proyectos realizados por la empresa.

#### **Situación**

En esta sección podemos encontrar un mapa que nos muestra la ubicación de la empresa así como su dirección.

#### **Contacto**

En esta sección se muestran las diferentes vías de contacto con la empresa. También contiene un formulario para enviar consultas directamente al mail de la empresa.



Figura 4.14



Figura 4.15

#### **4.5.2. Menú navegación parte privada**

Está formado por cinco secciones: usuarios, presupuestos, economía, estadísticas y empresas.

##### **Usuarios**

En este módulo podremos gestionar todo lo relativo a usuarios. Las funcionalidades de insertar, modificar y eliminar usuario se encuentran dentro de esta sección.

##### **Presupuestos**

Este módulo es el más extenso. En él se encuentran diversas funcionalidades relacionadas con los presupuestos. Se incluye la creación de presupuestos y modificación de los mismos. También la inserción, modificación y eliminación de los recursos/items. Se podrán insertar nuevas categorías para los items/recursos. También se podrán cerrar los presupuestos y establecer las fechas de los pagos de los mismos.

##### **Economía**

En esta sección se podrán consultar los totales de los proyectos y funcionalidades de filtrado de diversos tipos.

##### **Estadísticas**

Este módulo será el encargado de proveer de un servicio de estadísticas de seguimiento relativas al número de proyectos mensuales y al total de dinero acumulado mensual.

##### **Empresas**

En esta sección se gestionará toda la información relativa a empresas. La inserción, modificación y eliminación de empresas serán funcionalidades que se encontrarán en este módulo.










 <span>Usuarios</span> <span>Presupuestos</span> <span>Economía</span> <span>Estadísticas</span> <span>Empresas</span> <span>Salir</span>						
Nombre Empresa	CIF	Dirección	Teléfono	E-mail	P. contacto	
AUTOVALLS S.L.	222222222	C/ Pineda, 57 Vallis	333333333	asñlkjdfñ@ass.com	Juan Pedro	 
GLS	555555555	C/ Pedret 3, Selva del Camp	444444444	asdfasdfsdf@sss.com	Maria Casado	 
CRUVASA	444444444	C/ Lledoners, 23	555555555	lkjdfalslkja@ajs.com	Marti Roca	 
ESTRUGASA	999999999	C/ Mancomunitat 3, Perafort	656565656	jñlkasdjñls@las.com	Joan Martos	 

Figura 4.16




 <span>Usuarios</span> <span>Presupuestos</span> <span>Economía</span> <span>Estadísticas</span> <span>Empresas</span> <span>Salir</span>						
Presupuestos abiertos				Presupuestos cerrados		
Empresa	Fecha inicio	Fecha fin	Descripción			
AUTOVALLS S.L.	2013-01-07	2013-03-04	nivelación y asfaltado de accesos y parking			
ESTRUGASA	2013-03-04	2013-05-02	trabajos varios			
GLS	2012-09-18	2012-10-25	reforma integral nave industrial			

Figura 4.17

## Capítulo 5. Implementación

### 5.1. Introducción

En este capítulo se abordan las soluciones implementadas para las diferentes funcionalidades del portal web. Para ello, describiremos los detalles más importantes de las capas que forman la aplicación.

En primer lugar, hablaremos del entorno de usuario y de diversos aspectos que lo conforman. Seguidamente, detallaremos el motor de la aplicación, con detenimiento en algunas funcionalidades importantes para entender el funcionamiento de la aplicación adjuntando porciones de código de algunas de ellas para mayor detalle. Por último, analizaremos y describiremos los aspectos más relevantes de la implementación de la capa de datos de la aplicación.

### 5.2. Entorno de usuario

La capa de entorno de usuario es la encargada de realizar la comunicación con el usuario de la aplicación y de mostrar todo el contenido que visualizará el mismo.

Las diferentes interfaces que se encontrará el usuario se han codificado utilizando el lenguaje de etiquetas HTML para todos los elementos estáticos, y utilizando CSS (Cascading Style Sheets) para dar diseño a los mismos y estructurar la información utilizando capas.

El diseño de la aplicación se ha construido desde cero. El archivo *estilos.css* contiene la implementación de todo el diseño del portal web, el cuál es enlazado tanto desde la parte publica de la web como desde la parte privada. En este archivo *estilos.css* se especifica el formato y estilo de los diferentes elementos que aparecen en la aplicación web (tamaño, color, tipo de letra, bordes, propiedades de display, etc). En él también se incluye el posicionamiento de todos los elementos de la interfaz formando la estructura de diseño definida previamente.

Cabe destacar que todas estas propiedades que se encuentran en el archivo *estilos.css* se modifican dinámicamente durante la navegación en la aplicación web. Esto se realiza mediante el uso de la librería JQuery de Javascript, que nos permite mediante los selectores, modificar las propiedades de cualquier elemento en el momento en que, por ejemplo, se ejecute algún evento y también aplicar efectos a los distintos elementos.

Con el uso de Javascript y JQuery gestionaremos como se muestran todos los contenidos de nuestra aplicación web o los cargaremos dinámicamente, de manera que obtenemos un código más compacto y robusto garantizando mayor rapidez al usuario.

También se utiliza el framework Bootstrap, para aprovechar el potencial que proporciona con la diversidad de clases de estilo y funciones ya creadas de uso relativamente sencillo.

## Código fuente

### Propiedades CSS del elemento body dentro de *estilos.css*

```
body{
    background-image: url(../img/fondogrisgrande.png);
    background-repeat: repeat-x;
    padding: 0px;
    margin-top: -3px;
    background-color: #FFFFFF;
}
```

### Uso de JQuery para mostrar formulario de login dinámicamente al hacer clic en el icono de zona privada

```
$('#candy').click(function() {
    $('#formaccesoprivado').slideToggle('fast');
});
```

### Uso de JQuery para aplicar efecto de movimiento sobre icono facebook situar ratón encima

```
$('#facebook').mouseover(function() {
    $('#facebook').animate({'margin-left': '+=17px'}, 'fast');
});
$('#facebook').mouseleave(function() {
    $('#facebook').animate({'margin-left': '-=17px'}, 'fast');
});
```

### Uso de Bootstrap para abrir una ventana modal de inserción de usuario al hacer clic en su icono

```
<a href="#myModal2" role="button" data-toggle="modal" id="butadduser"></img></a>

<div id="myModal2" class="modal hide fade" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" >
    <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-hidden="true" >x</button>
        <h4 id="myModalLabel">Añadir nuevo usuario</h4>
    </div>
    <div class="modal-body" id="modal-body2">
        <div class="adduser_form">
            <form name="form_adduser" class="form_adduser" method="post" action="">
                Nombre usuario<br>
```

```

<div style="height:2px;"></div>
<input type="text" style="height:25px;" name="username" class="username" size="20"/>
Password<br>
<div style="height:2px;"></div>
<input type="password" style="height:25px;font-size:19px;" name="pass" class="pass"
size="20"/>
Nombre y apellidos<br>
<div style="height:2px;"></div>
<input type="text" style="height:25px;" name="name" class="name" size="60"/>
Mail<br>
<div style="height:2px;"></div>
<input type="text" style="height:25px;" name="mail" class="mail" size="60" />
Telefono<br>
<div style="height:2px;"></div>
<input type="text" style="height:25px;" name="telf" class="telf" size="15"/>
Credenciales<br>
<div style="height:2px;"></div>
<select name="level" class="level" >
    <option>Encargado</option>
    <option>Contable</option>
    <option>Director</option>
</select><br>
<div class="confade">
    </img>
    <div id="textfade">
        Usuario<br>
        Registrado
    </div>
</div>
<button class="btn btn-small" type="button" onclick="afegirUsuari(this);"
id="afegir">Añadir</button>
</form>
</div>
</div>
<div class="modal-footer">
    <button class="btn btn-danger" data-dismiss="modal" aria-hidden="true" >Cerrar</button>
</div>
</div>

```

### 5.3. Motor de la aplicación

El motor de la aplicación se encarga de enlazar el entorno de usuario y la capa de datos. Para ello principalmente utilizaremos el lenguaje PHP, que es un lenguaje de programación interpretado, por el cual será ejecutado por medio de un intérprete en el lado del servidor. Este lenguaje nos permite la manipulación de la información obtenida de la capa de datos, interactuando con el sistema de base de datos MySQL accediendo a la información que en ella se almacena. También nos permite embeber código HTML para así poder formatear toda la información procedente de la capa de datos de cara a mostrarla al usuario.



Diferentes funcionalidades de la aplicación web, como son la inserción, modificación o eliminación entre otras, se encuentran en archivos *.php* distintos, de manera que serán ejecutadas mediante una petición AJAX y posteriormente poder cargar el contenido dinámicamente donde sea necesario. Estas funcionalidades, interactúan con la capa de datos mediante formularios, para enviar la información necesaria para insertar, modificar, eliminar y para realizar consultas que seleccionen los datos necesarios y los devuelvan a la interfaz de usuario. Explicaremos el procedimiento para este último caso, por ejemplo, cuando solicitamos mostrar los usuarios registrados del portal web. En primer lugar, la aplicación realizará una consulta de selección de información a la base de datos, en este caso seleccionaremos todos los registros que hay en la tabla *users* (ver **4.4. Diseño y estructura Base de datos** para más información). Seguidamente, se procesarán los registros devueltos de la consulta realizada y se preparará la información para ser interpretada por el entorno de usuario. Las consultas son funcionalidades específicas del lenguaje PHP.

Para la diferenciación de distintos perfiles, se utiliza un campo de la tabla *users* llamado *permission\_level* que contendrá un índice que marcará el nivel de credenciales del usuario loggeado. De esta forma, al intentar acceder a las distintas funcionalidades, se llamará a la función *permisosUsers()* que devolverá el valor del campo *permission\_level* que resulte del registro que devuelva la consulta sobre la tabla *users*, que buscará el usuario cuyo *username* coincida con el parámetro `$_SESSION["usr"]` que es la variable de sesión que contiene el nombre del usuario que está loggeado en ese momento. Esta variable de sesión se mantiene durante todo el tiempo que el usuario está loggeado en el aplicativo. En función del valor que devuelva la función *permisosUsers()* se permitirá o se denegará el acceso a una funcionalidad.

## Autentificarse

La funcionalidad de autentificarse es la que permite tener un control de seguridad de la aplicación web. Dicha funcionalidad está implementada en el archivo *acceso.php* y puede ser ejecutada desde cualquier parte no privada de la aplicación web utilizando el formulario de acceso que se desplegará al clicar en el icono de zona privada.

Esta funcionalidad está implementada con el uso de variables de sesión. El usuario que disponga de los datos para realizar el login, introducirá su nombre de usuario y el password en el formulario de acceso. Con el nombre de usuario, ejecutaremos una consulta a la base de datos para saber si el nombre de usuario y el password existen y en caso afirmativo, nos permitirá el acceso a la zona privada. De lo contrario, el usuario permanecerá en la zona pública de la web. Al acceder, esta información es almacenada en las variables de sesión que se mantienen todo el tiempo que el usuario permanece conectado como ya hemos comentado. También se insertará la información con la hora del login en la tabla *users\_log* que será la tabla encargada

de llevar un seguimiento de las acciones de acceso y desconexión de los distintos usuarios. Las variables de sesión solamente se destruirán en caso de que el usuario proceda a desloggearse.

### Consultar usuarios

La funcionalidad de **consultar usuarios** se encuentra en la parte privada de la web. Está implementada en el archivo *userstab.php* y se ejecutará cuando accedamos a la sección de *Usuarios* de la zona protegida, en caso de tener los permisos necesarios.

Esta funcionalidad, ejecuta una consulta a la base de datos que nos devuelve todos los registros que hay en la tabla *users*. Seguidamente, procedemos a formatear toda la información de los registros devueltos, en forma de tabla, para una mejor visualización del usuario.

### Insertar usuario

La funcionalidad de **insertar usuario** también se encuentra en la parte privada de la web. Está implementada en el archivo *adduser.php* y se ejecutará al clicar sobre el icono de insertar usuario que está dentro de la sección de usuarios.

Esta funcionalidad, cargará una ventana modal que mostrará un formulario en el que se deberán añadir todos los campos con la información necesaria para registrar un usuario. En caso de que la información introducida sea correcta, ésta se enviará al archivo *adduser.php* mediante una petición AJAX y el método post. El archivo *adduser.php* recibirá todos los campos y ejecutará una sentencia a la base de datos que insertará en la tabla *users* los campos correspondientes procedentes del formulario.

### Editar usuario

La funcionalidad de **editar usuario** se encuentra en la parte privada de la aplicación web. Está implementada en el archivo *edituser.php* y se ejecutará al clicar en el icono de editar perteneciente a cada usuario cuando consultamos los usuarios.

Esta funcionalidad también cargará una ventana modal que mostrará un formulario con todos los campos de información del usuario seleccionado pero estos ya mostrarán la información guardada en la base de datos. De esta manera, solo cambiaremos la información de los campos que queramos editar, el resto de campos no se modifican. Si la información que cambiamos es correcta, ésta se enviará al archivo *edituser.php* mediante una petición AJAX Y el método post. El archivo *edituser.php* recibirá todos los campos procedentes del formulario y ejecutará una sentencia a la base de datos para actualizar en la tabla *users* el registro cuya id coincida con la id del usuario que recibe *edituser.php*, que es un campo de `type="hidden"` que también se pasa desde el formulario.

## Eliminar usuario

La funcionalidad de **eliminar usuario** es de la parte privada de la web y está implementada en el archivo *deleteuser.php*. Se ejecutará al clicar en el icono de eliminar perteneciente a cada usuario cuando consultamos los usuarios.

Esta funcionalidad cargará una ventana modal que mostrará un mensaje de confirmación para eliminar el usuario seleccionado. En caso afirmativo, mediante una petición AJAX, se pasará al archivo *deleteuser.php* un campo de `type="hidden"` con la id del usuario a eliminar mediante el método post. El archivo *deleteuser.php* recibirá el campo id del usuario y ejecutará una sentencia a la base de datos para eliminar el registro de la tabla *users* cuya id coincida con la id de usuario que recibe el archivo *deleteuser.php*.

## Generación de estadísticas

La funcionalidad de generar estadísticas también se encuentra en la parte privada de la web. Se ejecutará al clicar en el icono de estadísticas de presupuestos o el de estadística económicas dentro del módulo de estadísticas, siempre y cuando dispongamos de los credenciales adecuados.

Los datos para generar las estadísticas se encuentran en un archivo .php, el cual realiza una consulta a la base de datos y mediante la instrucción *mysql\_fetch\_array* dentro de un bucle *while* procesa toda la información obtenida de la consulta.

Los datos procesados en este archivo.php son llamados dinámicamente mediante una petición *AJAX* y son añadidos a la configuración de la gráfica antes de que ésta sea generada.

## Detalles de presupuesto

Para acceder a los detalles de un presupuesto, presionaremos el icono de detalles de presupuesto que se encuentra a la derecha de cada presupuesto.

Si disponemos de credenciales, se pasará el parámetro *id* de presupuesto a otro archivo .php que realizará una consulta a la tabla *budget\_details* de la base de datos con el parámetro *id* de presupuesto que ha recibido. Mediante la instrucción *mysql\_fetch\_array* en un bucle *while* se procesaran todos los datos devueltos por la consulta y se cargarán dinámicamente en el *div* correspondiente.

## Cerrar presupuestos

No se podrá cerrar ningún presupuesto sin los credenciales requeridos para ello. Al realizar esta funcionalidad, de entrada cambiaremos el valor del atributo *closed* del presupuesto seleccionado mediante una consulta a la tabla *budgets* de la base de datos y tomará el valor 1. Además, se establecerá en cuantos

pagos se cobrará el presupuesto y las fechas de los mismos y estos datos se introducirán en las tablas correspondientes en función de los pagos.

## Otras funcionalidades

En base a la lógica que se ha seguido en la implementación de las funcionalidades explicadas arriba, se han implementado el resto de funcionalidades aplicadas a información de empresas, presupuestos, etc. en vez de aplicadas a información de usuarios.

## Código fuente

### Archivo *acceso.php* que permite el login de los usuarios a la parte privada

```
<?
error_reporting(0);
session_start();

$username = $_POST["usuario"];
$password = $_POST["password"];

//CONEXION A LA BASE DE DATOS
$link = mysql_connect("localhost","root","");
mysql_select_db("projecte",$link);

$query = sprintf("SELECT * FROM users WHERE username = '%s' AND password =
'%s'",mysql_real_escape_string($username),mysql_real_escape_string(md5($password)));
$result = mysql_query($query);
if(mysql_num_rows($result)!=0)
{
    $_SESSION["usr"] = $_REQUEST["usuario"];
    $_SESSION["pass"] = $_REQUEST["password"];
    session_register("SESSION");
    $action = 'login';

    $query2 = sprintf("SELECT * FROM users WHERE username = '%s' AND password =
'%s'",mysql_real_escape_string($username),mysql_real_escape_string(md5($password)));
    $result2 = mysql_query($query2);
    while ($user1 = mysql_fetch_array($result2))
    {
        $id=$user1['id'];
        $name=$user1['name'];
        $insert = mysql_query ("INSERT INTO users_log (id_user,username,name,action) VALUES
('.$id.','.$username.','.$name.','.$action.')",$link);
    }
    header("location:gestion.php");
}
else
{
    header("location:projecte.php");
}
```

```
?>
```

### Recepción de datos y sentencia PHP para insertar usuario en la base de datos

```
<?php
```

```
$username=$_POST['username'];
$pass=md5($_POST['pass']);
$name=$_POST['name'];
$mail=$_POST['mail'];
$telf=$_POST['telf'];
$level=$_POST['level'];
$credenciales=$_POST['credenciales'];

$link = mysql_connect("localhost","root","");
mysql_select_db("projecte",$link);
$insert = mysql_query ("INSERT INTO users (username,password,name,mail,telf,permission_level,credenciales) VALUES ('".$username."','".$pass."','".$name."','".$mail."','".$telf."','".$level."','".$credenciales."')",$link);
```

```
?>
```

### Recepción de datos y sentencia PHP para eliminar usuario de la base de datos

```
<?php
```

```
$id=$_POST['id'];

$link = mysql_connect("localhost","root","");
mysql_select_db("projecte",$link);

$delete = mysql_query ("DELETE FROM users WHERE id='".$id'", $link);
```

```
?>
```

## 5.4. Capa de datos

La capa de datos será la encargada de almacenar toda la información de la aplicación: usuarios, presupuestos, empresas, etc. Tanto el acceso como la gestión de esta información se efectuará a través de consultas y sentencias interpretadas por MySQL. Además utilizaremos el gestor phpMyAdmin que facilita el tratamiento y la visualización de la información almacenada, y nos permite realizar modificaciones en las tablas de nuestra base de datos de manera sencilla. MySQL viene integrado en el paquete XAMPP además del intérprete de PHP para ejecutar código del lado del servidor.

Para el acceso y la gestión de la información de la base de datos, en primer lugar se debe establecer la conexión con la base de datos. Con las siguientes dos sentencias establecemos la conexión:

```
$link = mysql_connect("localhost","root","");  
mysql_select_db("projecte",$link);
```

Una vez realizada la conexión, el motor de la aplicación puede acceder a la información de la base de datos mediante consultas y sentencias y retornar la información específica formateada mediante PHP.

Para más información sobre la capa de datos, ver **4.4. Diseño y Estructura Base de datos.**

## Capítulo 6. Pruebas

Una vez finalizada la implementación de todo el proyecto, es necesario realizar un juego de pruebas para confirmar el correcto funcionamiento del sistema y detectar errores que puedan haberse producido en cualquiera de las fases del proyecto.

A continuación, explicaremos las pruebas realizadas en este proyecto.

### Pruebas de compatibilidad

Puesto que el proyecto desarrollado es una aplicación web, el acceso debe realizarse mediante un navegador, por lo tanto, se ha procedido a realizar una prueba de funcionamiento de la aplicación con diferentes navegadores que pueden ser utilizados.

El portal web se ha diseñado e implementado para funcionar de manera óptima tanto en *Google Chrome* como en *Mozilla Firefox* dado que son los navegadores más utilizados actualmente y los más eficientes de cara al usuario.

El funcionamiento de la aplicación web es el adecuado en estos dos navegadores en cuanto a rendimiento. En referencia al diseño, podemos apreciar pequeños cambios entre ellos. Esto viene producido porque Google Chrome aplica por defecto un zoom a los elementos que hace que éstos se vean un poco más pequeños respecto a su tamaño real. Es prácticamente inapreciable y no provoca ningún desajuste importante en el diseño en uno u otro navegador.

No obstante, el resultado en Internet Explorer no es óptimo. Principalmente en el apartado de diseño es donde se aprecian los mayores desajustes, sobretodo en los tamaños de las imágenes y el posicionamiento, por lo tanto para la presentación de este proyecto no habrá compatibilidad con éste navegador, ya que habría que crear un .css alternativo que se ejecutase en caso de que el navegador utilizado fuera Internet Explorer y no ha sido posible realizar por falta de tiempo. En referencia al funcionamiento, también se aprecian errores derivados de las cargas dinámicas, por lo que habría que ejecutar un archivo de javascript diferente al usado en Google Chrome y Mozilla Firefox para solventar esos errores.

### Pruebas de seguridad

La importancia de realizar un juego de pruebas de seguridad en una aplicación que dispone de diferentes perfiles de usuario es muy importante, puesto que cada perfil de usuario tendrá permisos para acceder a funcionalidades y contenidos distintos. Es necesario comprobar que el control de accesos que se ha implementado permita una correcta gestión e identificación de usuarios.

Se han realizado pruebas de acceso a contenidos restringidos, tratando de visualizar contenidos restringidos de forma directa mediante la URL del navegador. Todas las pruebas fueron exitosas, mostrando mensajes de falta de permisos para acceder y ocultando los contenidos restringidos.

También se deberían realizar pruebas de inyección SQL a fin de evitar intrusiones de personas ajenas a la aplicación web. No se han podido realizar debido a la falta de tiempo y del poco conocimiento de estas técnicas.

### **Pruebas de integración**

Con las pruebas de integración se comprueba la correcta unión de los diferentes módulos que forman la aplicación web. Se trata de testear que la interacción entre diferentes módulos no implique un mal funcionamiento de los mismos.



## Capítulo 7. Conclusiones

### 7.1. Consecución de objetivos

El principal objetivo del presente proyecto era la realización de un portal web, que permitiera a Ruben's Instalaciones y Excavaciones la posibilidad de difusión en Internet. Además también se pretendía crear un conjunto de herramientas y funcionalidades para gestionar de manera más óptima diversos aspectos de la empresa, principalmente los presupuestos. Para cumplir estos objetivos se han diseñado e implementado las correspondientes funcionalidades cumpliendo con los requerimientos no funcionales preestablecidos.

La gestión de los presupuestos se ha optimizado tratando de flexibilizar la aplicación al máximo de cara al usuario. Disponemos de todo un conjunto de funcionalidades de manera dinámica, como son la creación y edición de los presupuestos realizables en el menor número de clics posibles, así como el establecimiento de las fechas de pago a la hora de cerrar un presupuesto. También la gestión de los recursos y sus categorías para crear los presupuestos se ha realizado de manera óptima. Podemos afirmar que estos objetivos también se han cumplido.

Respecto a la gestión de usuarios, la inserción, modificación, eliminación y consulta de los mismos se realiza de manera sencilla y óptima, por tanto hemos cumplido otro de los objetivos.

Para el correcto funcionamiento de la aplicación se ha creado un sistema de autenticación de usuarios, así como un sistema de perfiles de usuario mediante credenciales. Esto permite cumplir con los objetivos de seguridad y control de acceso de los usuarios, así como del acceso a contenidos y herramientas dependientes del perfil de usuario.

Se ha cumplido también con el objetivo de consultar los pagos realizados de manera sencilla, para así facilitar en cierta medida el trabajo a los encargados de contabilidad.

Se ha integrado también un módulo de estadísticas de seguimiento del número de presupuestos y de los totales de dinero que suponen, organizados de manera mensual.

Por último, otro objetivo cumplido es la gestión de la información relativa a las empresas con las que se trabaja de manera óptima.

### 7.2. Desviación de la planificación

El presente proyecto se ha desarrollado aproximadamente en el plazo de tiempo establecido en el apartado **2.7. Planificación inicial**. No obstante, la distribución de las horas diarias dedicadas al proyecto ha sufrido

variaciones durante el transcurso del desarrollo. Las principales causas han sido la actividad laboral y un aumento del tiempo de aprendizaje estimado necesario para la realización de diferentes tareas del proyecto, que por un lado han supuesto una dificultad para mantener una implicación en las horas estimadas para la realización de este proyecto, y por otro lado se han necesitado más horas de las previstas para conseguir un desarrollo óptimo del proyecto tal y como se pretendía.

La inexperiencia en proyectos de esta envergadura, unido a los motivos explicados arriba, han producido desviaciones de la planificación que han hecho peligrar la consecución del proyecto en los plazos estimados en la planificación inicial.

### 7.3. Líneas de mejora y ampliación

Este proyecto es susceptible a mejoras o ampliaciones de las funcionalidades que lo integran.

Una de las principales mejoras consiste en dar soporte a navegadores Internet Explorer ya que como hemos explicado en las **Pruebas de compatibilidad** dentro del **Capítulo 5. Pruebas** no se ha cumplido con este objetivo. Internet Explorer es un navegador bastante utilizado a pesar de su mal funcionamiento y compatibilidad, por ello, esta mejora cobra especial importancia.

Otra mejora a realizar podría ser una mejora o ampliación del modulo de estadísticas para obtener mayores datos, como por ejemplos, número de proyectos encargados por una empresa durante un año o estadísticas globales del número de proyectos encargados filtrados por empresas mensualmente.

Se podría tener en cuenta también la inclusión de un módulo referente a los proveedores, con información de los mismos y facturas emitidas por ellos con fechas de pago, para facilitar en mayor medida la labor de los encargados de contabilidad.

Por último, se podría mejorar el sistema de logs de usuario capturando más acciones dentro de la aplicación web y no limitarse solo al login/logout.

## Capítulo 8. Bibliografía y fuentes de consulta

### 8.1. Referencias electrónicas

#### Páginas web

[<http://www.w3schools.com/>] - Página web en inglés que contiene todo tipo de tutoriales relativos a la programación web (HTML, CSS, ASP, XML, Javascript, PHP...). Contiene ejemplos y un editor *Try it Yourself* para realizar pruebas y ver el resultado en el navegador.

[<http://twitter.github.io/bootstrap/>] - Página web en inglés del framework Bootstrap. En ella podemos descargar los archivos necesarios para su uso, consultar toda la documentación, gran cantidad de ejemplos, etc.

[<http://www.w3resource.com/>] - Página web en inglés dedicada a tutoriales de desarrollo web de todo tipo. Se encuentran clasificados por categorías (Frontend, Backend, SQL, NoSQL, APIs...).

[<http://www.desarrolloweb.com/>] - Página web en español dedicada al mundo del desarrollo web. Dispone de gran cantidad de material como tutoriales, monográficos o mini cursos. Además cuenta con un foro bastante activo.

[<http://php.net/manual/es/index.php>] - Página web en español con toda la documentación del lenguaje de programación PHP.

[<http://api.jquery.com/>] - Página web en inglés de la API JQuery. Podemos descargar los archivos necesarios para su uso, consultar toda la documentación, plugins, etc.

[<http://rhinoslider.com/demo-download/>] - Página web de un slider JQuery de nombre Rhinoslider utilizado en el proyecto. Desde la misma web, podemos generar un slider con los parámetros deseados y descargar los archivos que se deberán incorporar en el proyecto para su funcionamiento.

[<http://css-tricks.com/>] - Página web en inglés dedicada al mundo del CSS (*Cascading Style Sheets*). Podemos encontrar ejemplos de todo tipo y un foro bastante activo.

[<http://www.asostips.com/configuracion-instalacion-Apache-PHP-MySQL-windows/>] - Página web en español que muestra los pasos iniciales y de configuración de la aplicación XAMPP.

[<http://icons8.com/download-huge-windows8-set/>] - Página web de la cual se ha descargado el set de iconos utilizados para el desarrollo del proyecto.

## Tutoriales y video-tutoriales

[<http://www.youtube.com/user/OutKast>] - Canal de youtube del autor de Videotutoriales.com en el que podemos encontrar gran cantidad de tutoriales sobre tecnologías de desarrollo web.

[<http://www.w3resource.com/twitter-bootstrap/tutorial.php>] - Tutorial en inglés sobre el framework Bootstrap en el que podemos encontrar bastantes ejemplos prácticos.

## Foros

[<http://stackoverflow.com/>] - Foro en idioma inglés pensado para programadores. No requiere de registro, cualquiera puede preguntar y cualquiera puede responder. Ideal para buscar soluciones a casos concretos de problemas de programación en el desarrollo.

[<http://www.forosdelweb.com/>] - Foro en español sobre el mundo del desarrollo web. Contiene diferentes secciones para clasificar los hilos.

[<http://foros.cristalab.com/>] - Foro de la web de cristalab. Se ha consultado por problemas de programación durante el desarrollo. Nos encontramos diversos subforos, los de principal interés para el desarrollo web serían el de desarrollo y el de diseño.