



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE SEGOVIA**

**Grado en Ingeniería Informática  
de Servicios y Aplicaciones**

---

**Título del Trabajo Fin de Grado**

---

**Alumno: Nombre y Apellidos**

**Tutor/a/es: Nombres y Apellidos**



**Título del TFG**

Nombre y Apellidos del Autor



# Índice general

Lista de figuras	III
Lista de tablas	V
Resumen	XI
<b>I Memoria del Proyecto</b>	<b>1</b>
<b>1. Descripción del proyecto</b>	<b>3</b>
1.1. Introducción . . . . .	3
1.2. Objetivos del trabajo . . . . .	3
1.3. Entorno de aplicación . . . . .	4
<b>2. Metodología</b>	<b>5</b>
2.1. Proceso de desarrollo . . . . .	5
2.2. Herramientas utilizadas . . . . .	6
2.3. Arquitectura . . . . .	6
2.4. Definición de siglas y abreviaturas . . . . .	6
<b>3. Planificación</b>	<b>9</b>
3.1. Estimación del esfuerzo . . . . .	9
3.2. Planificación temporal . . . . .	9
3.3. Presupuesto económico . . . . .	9
3.3.1. Hardware y software . . . . .	10
3.3.2. Recursos humanos . . . . .	10
3.3.3. Presupuesto total . . . . .	10
<b>4. Conclusiones</b>	<b>13</b>
<b>II Documentación técnica</b>	<b>15</b>
<b>5. Análisis</b>	<b>17</b>

5.1. Requisitos . . . . .	17
5.2. Atributos de calidad . . . . .	18
<b>6. Diseño</b>	<b>19</b>
6.1. Diseño de datos . . . . .	19
6.2. Diagramas de clase y de secuencia . . . . .	19
<b>7. Implementación</b>	<b>21</b>
<b>8. Pruebas</b>	<b>23</b>
 <b>III Manuales de la Aplicación</b>	 <b>25</b>
<b>9. Manual de Instalación</b>	<b>27</b>
<b>10. Manual de Usuario</b>	<b>29</b>
10.1. Manual de Usuario . . . . .	29
10.2. Manual de Administración . . . . .	29
 <b>IV Apéndices</b>	 <b>31</b>
<b>A. Anexos</b>	<b>33</b>
A.1. Información complementaria . . . . .	33
A.2. Diagramas y tablas . . . . .	33
<b>B. Contenido del CD</b>	<b>35</b>
<b>Bibliografía</b>	<b>37</b>

# Índice de figuras

2.1. Modelo de desarrollo iterativo . . . . .	5
2.2. Arquitectura lógica . . . . .	7
2.3. Arquitectura física . . . . .	7
3.1. Diagrama de Gantt . . . . .	10





# Índice de cuadros

3.1. Presupuesto total . . . . .	11
5.1. CU-01. Solicitar registro . . . . .	18



*Dedicado a  
mi familia*



# Agradecimientos

Muchas gracias a todos



# Resumen

Resumen que aparecerá en UVADOC.

**Palabras claves:** separadas por comas.





# Parte I

## Memoria del Proyecto



# Capítulo 1

## Descripción del proyecto

Se tratará brevemente de explicar cómo se organiza la Memoria del Trabajo Fin de Grado (TFG), del posible contenido de cada uno de los capítulos y secciones, así como de contenidos mínimos exigibles y algunas recomendaciones prácticas.

### 1.1. Introducción

Se hará una introducción general, en la que se explicará la motivación que condujo a la elección del tema del TFG, se dará una idea general del propósito del trabajo, y se resaltará el componente de originalidad del mismo.

Asimismo, se hará una descripción general de entorno de la aplicación, es decir, en qué organización o empresa se integrará la aplicación que se desarrolla (si la hubiere), los fines y tareas de tal organización, etc.

Por último, se dará una visión general de los contenidos de la Memoria.

### 1.2. Objetivos del trabajo

Se detallarán en esta sección los objetivos generales del TFG, tanto si se trata de un estudio teórico, como si se trata de desarrollar una aplicación de software.

En este segundo caso, se describirá la funcionalidad básica, y se explicará qué necesidades se cubren con la aplicación desarrollada, y de la organización en la que se integrará (si procede).

Puede haber diversos tipos de objetivos:

- A nivel personal.
- A nivel teórico.
- A nivel práctico.
- Si se aplicará o no en un contexto real.

Se explicará también el alcance de la aplicación, sus restricciones y limitaciones, así como las perspectivas del trabajo.

### 1.3. Entorno de aplicación

En esta sección se tratarán los antecedentes y el “estado del arte”.

Se dará una panorámica de trabajos y aplicaciones tecnológicas existentes en el mismo área de trabajo, sus ventajas e inconvenientes, y se hará un análisis crítico que justifique el enfoque del TFG.

# Capítulo 2

## Metodología

En este capítulo se detallarán las cuestiones metodológicas, es decir, las metodologías y herramientas que se han utilizado para plantear el trabajo.

### 2.1. Proceso de desarrollo

Este apartado se refiere a si se ha utilizado un modelo de desarrollo iterativo, de prototipos, en cascada, etc (ciclo de vida del software). Por ejemplo, el siguiente diagrama refleja el modelo de desarrollo iterativo:

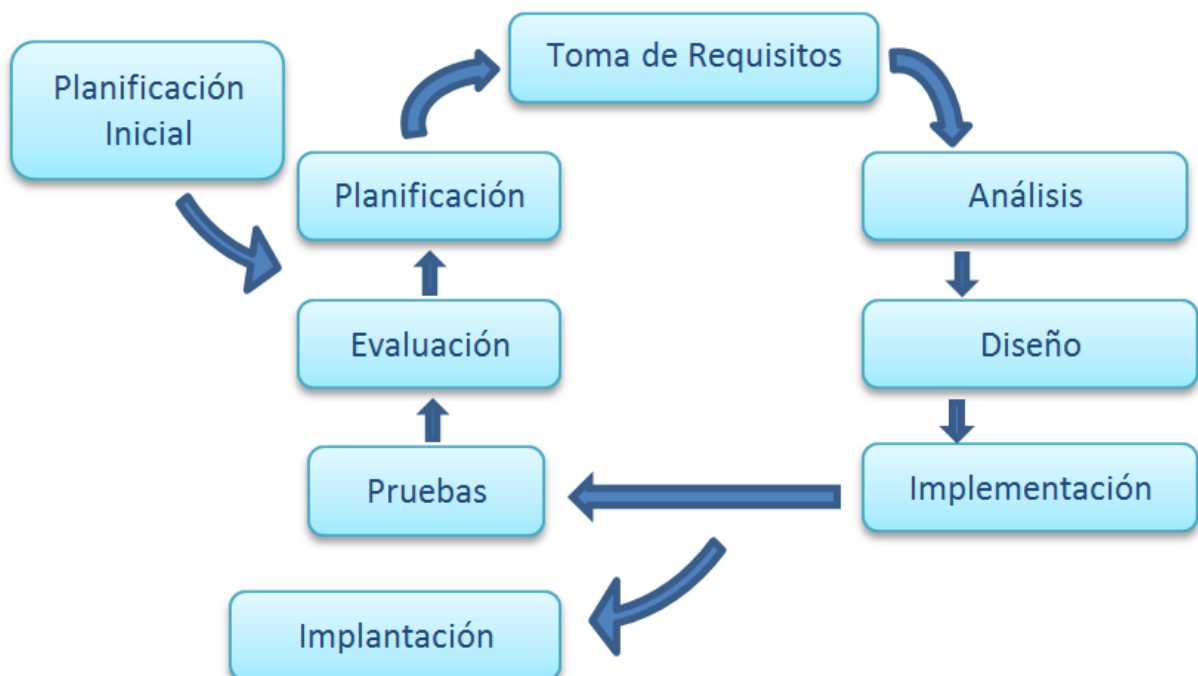


Figura 2.1: Modelo de desarrollo iterativo

También se especificará si se ha usado el paradigma de programación estructurada o programación orientada a objetos.

Asimismo, se explicará si se ha usado alguna metodología especial (metodologías ágiles, TDD, etc).

### 2.2. Herramientas utilizadas

En esta sección se entrará más en detalle en las tecnologías específicas que se han empleado para desarrollar la aplicación: lenguajes de programación empleados, gestor de bases de datos, herramientas usadas en la planificación y en la generación de documentación, sistema operativo, etc.

Asimismo, se defenderán los criterios por los que se han seleccionado las herramientas elegidas, de entre otras posibles, haciendo referencia a posibles ventajas e inconvenientes.

### 2.3. Arquitectura

Se explicará en este apartado la arquitectura de la aplicación, tanto a nivel lógico como físico. Por ejemplo, el diagrama 2.2 ilustra la arquitectura lógica cliente-servidor.

Sin embargo, el diagrama 2.3 representa una posible arquitectura física correspondiente a dicho modelo lógico.

### 2.4. Definición de siglas y abreviaturas

Es interesante mostrar una tabla o lista de abreviaturas y siglas (acrónimos) de uso extendido en la Memoria, pero de los que es posible que el lector de la misma no tenga conocimiento o sean ambiguos.

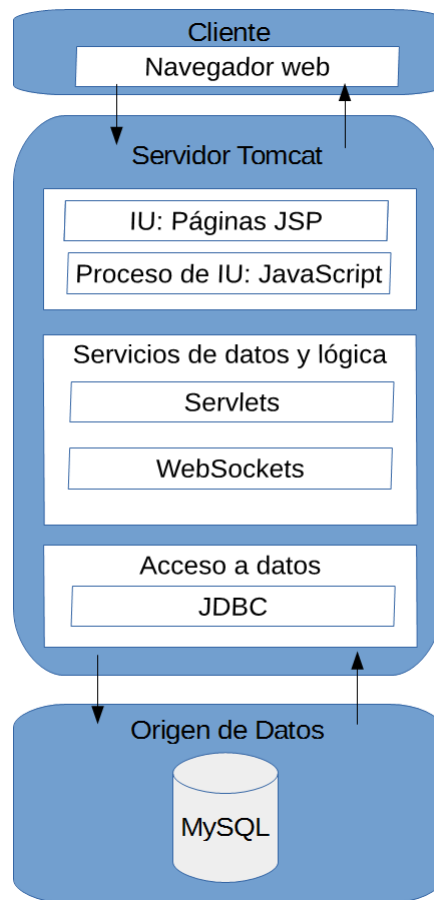


Figura 2.2: Arquitectura lógica

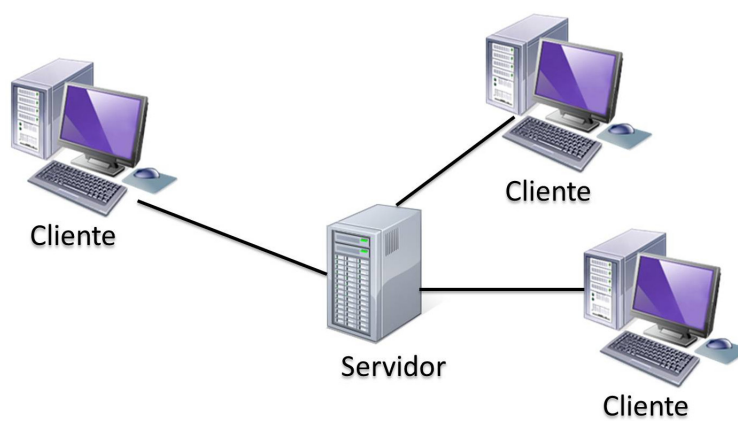


Figura 2.3: Arquitectura física





# Capítulo 3

## Planificación

En este capítulo se abordarán las cuestiones relativas a la planificación del trabajo.

### 3.1. Estimación del esfuerzo

Para la planificación temporal y de costes pueden aplicarse al menos dos enfoques: el primero se basa en la estimación del esfuerzo mediante **puntos de función** y su transformación en líneas de código, con el fin de calcular el coste final mediante el método *COCOMO II*.

El segundo método consiste en la estimación de **puntos de Caso de Uso**, que se basa en un análisis previo de los requisitos funcionales de la aplicación.

Cualquiera que sea el método elegido, conviene justificar las ventajas e inconvenientes, aplicándolo al caso concreto del TFG.

### 3.2. Planificación temporal

En la planificación de tareas se tendrán en cuenta los objetivos y requisitos de la aplicación, así como la estimación del esfuerzo, pero también puede tenerse en cuenta la distinción en tareas de análisis, diseño, implementación y pruebas, en cada una de las posibles iteraciones del ciclo de vida del proyecto, y el marco temporal (plazo de entrega del proyecto).

El correspondiente diagrama de Gantt permite observar de forma gráfica la distribución temporal de las tareas:

### 3.3. Presupuesto económico

Para la estimación del presupuesto se tendrán en cuenta las herramientas utilizadas (hardware y software, con los factores de impacto que correspondan a la duración del proyecto), junto con los recursos humanos necesarios, según la planificación de tareas, y el

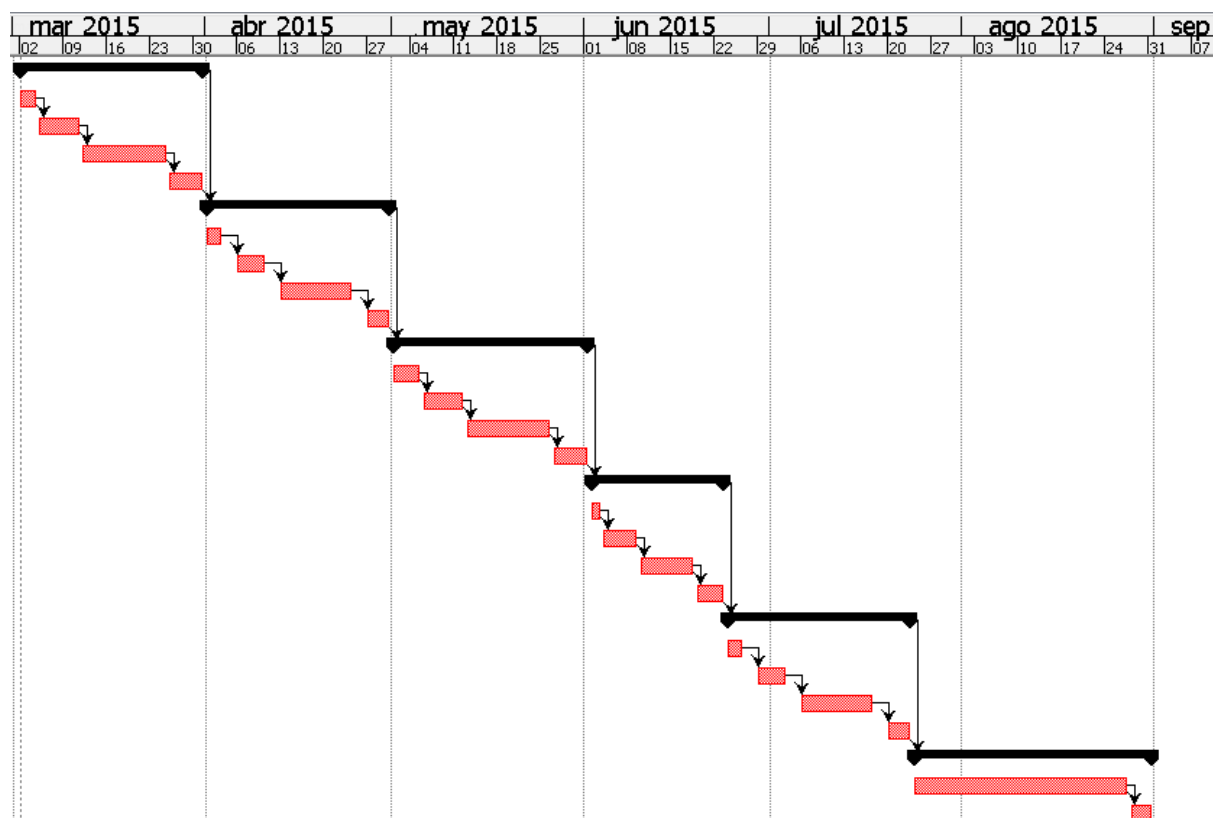


Figura 3.1: Diagrama de Gantt

tipo de rol (analista, programador, etc) correspondiente a cada tarea. Para ello conviene tener una tabla actualizada del coste por hora de cada tarea del proyecto, además del estudio previo de la planificación de tareas.

### 3.3.1. Hardware y software

Valoración del hardware y software utilizado en el proyecto.

### 3.3.2. Recursos humanos

Valoración de los recursos humanos

### 3.3.3. Presupuesto total

Se resumirán los conceptos económicos, por ejemplo en una tabla, con las estimaciones realizadas y el coste real, si procede:

**Nota:** es interesante analizar la discrepancia (si es que existe) entre el presupuesto y el coste real, y tratar de explicar sus causas.

	Hardware	Software	RRHH	Total
<b>Estimación I</b>	650,75 €	500	60000 €	61150,75 €
<b>Estimación II</b>	650,75 €	500	40000 €	41150,75 €
<b>Coste real</b>	650,75 €	400	50000 €	51050,75 €

Cuadro 3.1: Presupuesto total



## Capítulo 4

### Conclusiones

Este capítulo debe analizar los aspectos relativos a la consecución de los objetivos propuestos, y el grado de satisfacción de los mismos.

Asimismo, se puede abordar un análisis crítico de las ventajas e inconvenientes de las soluciones propuestas. en el trabajo.

Por otra parte, conviene resaltar la el grado de originalidad del TFG, así como el valor añadido que supone el TFG con respecto a los estudios de grado; dicho de otra forma, qué cosas nuevas se han aprendido y trabajado en el proyecto que no se habían aprendido a lo largo de la carrera.

Por último, es necesario proponer posibles mejoras y futuras ampliaciones del trabajo.



## Parte II

# Documentación técnica





# Capítulo 5

## Análisis

### 5.1. Requisitos

En este apartado se describen los requisitos del sistema que se va a desarrollar: requisitos funcionales, de interfaz de usuario, de información, etc. Asimismo, se especificará si la aplicación debe ser o no multiplataforma (o multilingüe), si se establecen restricciones de uso o de otro tipo (por ejemplo heredadas del entorno organizativo en el que se integrará), etc.

Para una mejor descripción de los requisitos, se usan los llamados casos de uso, basados en la identificación de actores y tareas.

A modo de ejemplo, este sería un caso de uso para la tarea “solicitar registro” en una aplicación WEB, que deba ser validado por un administrador:

<b>Nombre e ID del CU</b>	CU-01. Solicitar registro
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario envía sus datos de usuario al sistema, generando una solicitud de alta que deberá ser gestionada por un administrador.
<b>Precondiciones</b>	PRE-1. El usuario no está identificado en el sistema.
<b>Postcondiciones</b>	POST-1. La solicitud queda almacenada en el sistema.
<b>Flujo normal</b>	<p>FN1 El actor introduce sus datos de usuario e indica al sistema que quiere solicitar el registro.</p> <p>FN2 El sistema comprueba los datos introducidos.</p> <p>FN3 Si los datos son correctos, el sistema almacena la solicitud de registro e informa al actor del resultado.</p>
<b>Flujo alternativo 1</b>	<p>FA3 Si los datos son incorrectos se informa al usuario del error y no se procede a almacenar la solicitud.</p>

<b>Flujo alternativo 2</b>	FA3 Si no hay otros usuarios, el usuario pasa automáticamente al estado de activación y tendrá el rol de administrador.
<b>Excepciones</b>	E1 El usuario ha dejado campos requeridos sin rellenar. E2 El usuario o correo electrónico ya existen. E3 El usuario está bloqueado.
<b>Prioridad</b>	Alta
<b>Otra info</b>	El primer usuario registrado en la aplicación será el que adquiera el rol de administrador.

Cuadro 5.1: CU-01. Solicitar registro

## 5.2. Atributos de calidad

Se analizarán aquí los posibles indicadores de calidad de la aplicación desarrollada, a saber:

**Rendimiento:** tiene que ver con los tiempos de espera y de ejecución, así como la gestión de la memoria y posibles problemas de concurrencia.

**Seguridad:** está relacionado con la seguridad del sistema, la privacidad de datos, etc.

**Robustez:** tiene que ver con el comportamiento, la fiabilidad y la estabilidad de la aplicación ante posibles incidencias en tiempo real.

# Capítulo 6

## Diseño

Los aspectos de diseño depende mucho del paradigma elegido (programación orientada a objetos o no, por ejemplo).

### 6.1. Diseño de datos

En primer lugar podemos ver el modelo de datos que se usará para diseñar la base de datos, si esta es necesaria. Así, habrá que mostrar el diagrama entidad-relación, el modelo relacional, y el diccionario de datos, a nivel lógico, y por otra parte el diseño físico de la base de datos, y los perfiles de acceso a la misma por parte de los usuarios (si procede).

### 6.2. Diagramas de clase y de secuencia

Si la programación es orientada a objetos, es necesario especificar los diagramas de clases que se usan.

Los diagramas de secuencia describen la secuencia de pasos con que un usuario interacciona con el sistema, y están relacionados con los casos de uso de la etapa de análisis.

Es interesante también especificar si se ha hecho uso de diseño de patrones.



# Capítulo 7

## Implementación

En este capítulo se mostrarán los detalles más relevantes en cuanto a la implementación del sistema.



# Capítulo 8

## Pruebas

Este capítulo es muy importante, y muestra las pruebas que se han realizado para demostrar el funcionamiento del sistema.

Se referirán las estrategias de prueba utilizadas, y cuándo se han efectuados dichas pruebas.

Se realizarán pruebas de caja negra y de caja blanca, unitarias, de integración, de rendimiento, de seguridad, etc.

Se realizará una buena batería de pruebas para probar la funcionalidad de la aplicación, que está determinada por los requisitos de la misma, y se mostrarán los resultados (si estos son muchos, se pueden dejar los menos relevantes para el contenido del CD o para los Anexos de la Memoria).





## Parte III

### Manuales de la Aplicación



# Capítulo 9

## Manual de Instalación

Este capítulo se dedica a explicar los prerequisites técnicos, sistema operativo, etc, necesarios para instalar la aplicación desarrollada en el TFG.

Se explicará también todo lo necesario para conseguir e instalar el software de terceros que se precise con carácter previo.

Si se trata de una aplicación WEB, explicar cómo instalar y configurar los servidores necesarios para que funcione dicha aplicación, y cómo se realiza el despliegue de la misma.

Si se trata de una aplicación de escritorio, explicar el proceso de instalación y configuración en las plataformas correspondientes.



# Capítulo 10

## Manual de Usuario

Este capítulo final se dedica a desarrollar el manual del usuario final, así como del usuario administrador, si fuese necesario.

### 10.1. Manual de Usuario

Se recomienda que contenga la información relevante, lo más completa y clara posible, como si fuese el manual de un producto comercial, dirigido a un usuario no avanzado. Esta documentación pudiera coincidir (de hecho, se recomienda) con la AYUDA del programa.

### 10.2. Manual de Administración

Este manual sería mucho más técnico, pero asimismo igual de claro y completo, y estaría dirigido al administrador de la aplicación.



Parte IV

Apéndices





# Apéndice A

## Anexos

### A.1. Información complementaria

En los anexos se pueden incluir todas aquellas cosas que pueden servir, a nivel teórico o práctico, para ayudar al lector a profundizar en ciertos aspectos de la memoria, como los prerrequisitos teóricos, descripción de alguna herramienta en particular, etc.

### A.2. Diagramas y tablas

También se pueden incluir aquí las tablas, los casos de uso, o los resultados de prueba que no se hayan incluido en el capítulo correspondiente, para una mayor claridad en la lectura de la documentación técnica.



## Apéndice B

### Contenido del CD

Explicar el contenido del CD que se entrega junta a la Memoria, junto con el árbol de directorios correspondiente, para facilitar la navegación por los archivos del mismo.



# Bibliografía

- [1] P. Bourque y R.E. Fairley. *Guide to the Software Engineering Body of Knowledge, Version 3.0*. ACM-IEEE. 2014.
- [2] *Function Point Languages Table*. Quantitative Software Management. Agosto 2015.  
URL: <http://www.qsm.com/resources/function-point-languages-table>.
- [3] R. Pressman. *Ingeniería del Software*. McGraw-Hill, 2014.
- [4] Adi Shamir. “How to share a secret”. En: *Communications of the ACM* 22 (1979).

# Índice alfabético

Bases de Datos, 6, 19

Programación Orientada a Objetos, 6, 19