

Memoria del trabajo de fin de grado

# **Plantilla LaTeX para la redacción de la memoria del trabajo de fin de grado**

Sergio Barrachina Mir

Trabajo de fin de grado

Grado en Ingeniería Informática

Universidad Jaume I

25 de abril de 2013

Proyecto dirigido por: Sergio Barrachina Mir





A Laura



# Agradecimientos

En primer lugar, me gustaría agradecer a mi director/a de trabajo de fin de grado, Nombre y Apellidos, su...

Más agradecimientos.

Por último, me gustaría dar las gracias a Sergio Barrachina Mir por su «Plantilla L<sup>A</sup>T<sub>E</sub>X para la redacción de la memoria del trabajo de fin de grado», que he utilizado como punto de partida para la redacción de esta memoria.



# Resumen

Este documento pretende servir como guía para la redacción de la memoria del trabajo fin de grado de los estudiantes de la Universidad Jaume I, y en particular, para los trabajos de las titulaciones de informática. El objetivo es el de proporcionar un modelo que se pueda utilizar como referencia. En ningún caso debe verse como un requisito que deba seguirse a rajatabla.

En cada apartado de este documento se muestra una breve indicación de cuál debería ser el contenido de dicho apartado.

Así, el resumen debe dar una idea, clara y concisa, de la justificación —necesidad, oportunidad— del trabajo presentado, de sus objetivos generales y de lo que se ha conseguido durante su desarrollo.

El lector, después de leer el resumen, debe saber qué va a encontrar en el resto de la memoria.





# Índice general

<b>Índice general</b>	<b>v</b>
<b>1 Introducción</b>	<b>1</b>
1.1. Motivación del trabajo . . . . .	1
1.2. Objetivos del trabajo . . . . .	1
1.3. Entorno y estado inicial . . . . .	2
<b>2 Planificación y evaluación de recursos</b>	<b>3</b>
2.1. Planificación . . . . .	3
2.2. Evaluación de recursos . . . . .	3
<b>3 Análisis y diseño del sistema</b>	<b>7</b>
3.1. Análisis de requisitos . . . . .	7
3.2. Diseño del sistema . . . . .	9
3.3. Arquitectura del sistema . . . . .	9
3.4. Diseño de la interfaz (en su caso) . . . . .	10
<b>4 Desarrollo y resultados del trabajo</b>	<b>11</b>
4.1. Desarrollo del trabajo . . . . .	11
4.2. Resultados . . . . .	12
<b>5 Conclusiones y trabajo futuro</b>	<b>13</b>
5.1. Conclusiones . . . . .	13
5.2. Trabajo futuro . . . . .	13
<b>Bibliografía</b>	<b>15</b>
<b>A Otras consideraciones</b>	<b>17</b>
A.1. Bibliografía . . . . .	17
A.2. Apéndices . . . . .	17
<b>B Código fuente</b>	<b>19</b>



# Introducción

## Índice

---

1.1. Motivación del trabajo . . . . .	1
1.2. Objetivos del trabajo . . . . .	1
1.3. Entorno y estado inicial . . . . .	2

---

Este capítulo debe reflejar de forma muy clara qué se va a realizar durante el desarrollo del trabajo. El punto fundamental son los objetivos, pero es interesante comentar también la necesidad, idea, etc., que motiva el trabajo y el estado desde el cuál se va a empezar a trabajar [2].

### 1.1. Motivación del trabajo

En este apartado se debe describir el proceso de gestación del trabajo, es decir, las circunstancias que han motivado el desarrollo del trabajo. Esta introducción puede ser genérica, comentando todo aquello que hace que el trabajo sea útil y oportuno.

Resumiendo, este apartado debe dejar claro por qué se ha hecho el trabajo.

### 1.2. Objetivos del trabajo

Este apartado es fundamental para que el lector pueda comprender qué se va a realizar como trabajo. Todo lo que se pretende conseguir con el trabajo debe describirse con la mayor objetividad y precisión posibles.

También se debe indicar en este apartado si el trabajo forma parte de un trabajo mayor, si se complementa con otros, etc.

En definitiva, los lectores de la memoria, al terminar de leer los objetivos del trabajo, deberán saber claramente qué pretende alcanzar individualmente el proyectando.

### **1.3. Entorno y estado inicial**

Todo trabajo se desarrolla partiendo de unas circunstancias de trabajo conocidas a priori y de un estado inicial. Este apartado debe reflejar ambos. En primer lugar, se debe describir cuál es la forma de trabajar, el equipo técnico y humano, etc. Y en segundo lugar, se debe dejar constancia de todas aquellas decisiones impuestas o seleccionadas a priori. Aquellas que no vayan a ser revisadas durante el desarrollo del trabajo.

# Planificación y evaluación de recursos

## Índice

---

2.1. Planificación . . . . .	3
2.2. Evaluación de recursos . . . . .	3

---

Este capítulo contiene la parte más técnica del trabajo. Todo trabajo de ingeniería debe poder ser comprensible y evaluable en base a información objetiva siguiendo un mismo patrón. Tal información debe aparecer en este capítulo.

### 2.1. Planificación

En este apartado se debe mostrar la planificación temporal detallada del trabajo, incluyendo todas sus tareas y subtareas, las dependencias entre ellas y las posibles dependencias con otros trabajos o eventos externos.

También es conveniente mostrar un resumen de dicha información en forma de diagrama de Gantt (ver Figura 2.1).

### 2.2. Evaluación de recursos

El proyectando debe evaluar los costes del trabajo que se dispone a emprender. Estos costes, humanos y de equipamiento, deben cuantificarse para que el trabajo pueda valorarse y para que, en un caso real, se pueda evaluar la viabilidad económica del trabajo.

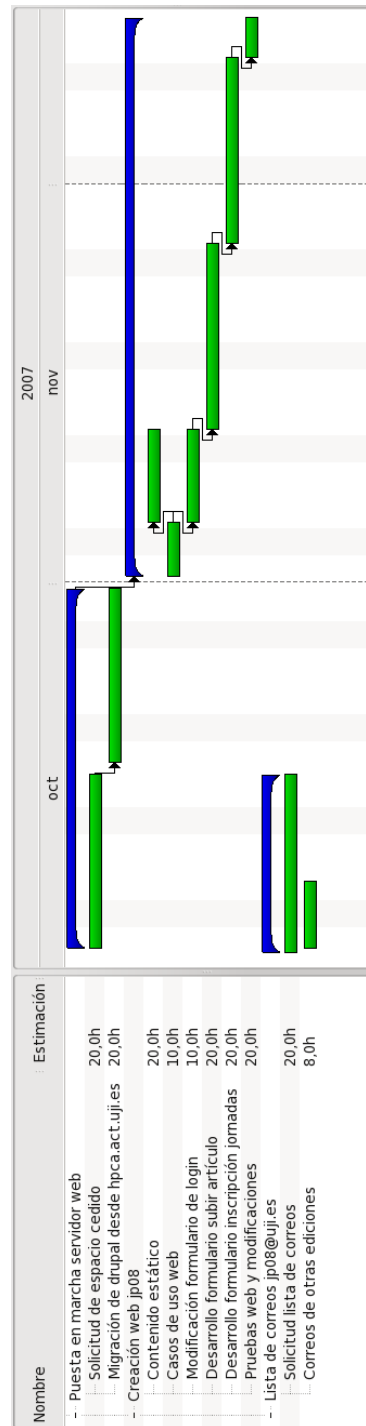


Figura 2.1: Ejemplo de diagrama de Gantt (realizado con Kplato)

En este apartado, el de evaluación de recursos, se deben describir los recursos humanos y de equipamiento necesarios para desarrollar e implantar el trabajo. Así como el coste estimado de dichos recursos.





# Análisis y diseño del sistema

## Índice

---

3.1. Análisis de requisitos . . . . .	7
3.2. Diseño del sistema . . . . .	9
3.3. Arquitectura del sistema . . . . .	9
3.4. Diseño de la interfaz (en su caso) . . . . .	10

---

En este capítulo se presentan el análisis de requisitos, el diseño y la arquitectura del sistema propuesto, así como, en su caso, el diseño de la interfaz del trabajo.

### 3.1. Análisis de requisitos

Para llevar a cabo un trabajo es necesario realizar un análisis previo de los requisitos de dicho trabajo. En este apartado se deben detallar los requisitos funcionales y no funcionales del trabajo presentado.

En [3] se puede ver un ejemplo en el que dado un problema determinado, la creación de una tienda en línea de música, se obtienen algunos de sus requisitos funcionales y no funcionales. Dicho ejemplo también permite ver claramente la diferencia entre los objetivos del trabajo, que aparecen como parte de la descripción del problema que se quiere resolver, y los requisitos funcionales y no funcionales de dicho trabajo.

### 3.1.1. Requisitos funcionales

Un requisito funcional define una función del sistema que se quiere desarrollar. Dicha función se describe como un conjunto de entradas, su comportamiento y sus salidas [4, 5].

Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulaciones de datos y procesos y cualquier otra funcionalidad específica que defina lo que se supone que un sistema debe conseguir.

De forma general, los requisitos funcionales se refieren a lo que el sistema debe hacer y se pueden expresar de la forma «el sistema hará <requisito>». El plan para implementar los requisitos funcionales se detallará en el diseño del sistema (Apartado 3.2).

Utilizando el ejemplo de [3], el de una tienda en línea de música, la descripción de dos de sus requisitos funcionales podría hacerse de la forma mostrada en el siguiente párrafo y en los cuadros citados por él.

El sistema propuesto debe permitir la compra de discos utilizando créditos. Los requisitos funcionales relacionados con la consulta y la compra de créditos se muestran en los Cuadros 3.1 y 3.2.

---

Entrada: Consulta de créditos disponibles

---

Salida: Créditos del usuario

---

Cada usuario posee un número de créditos. Cuando el usuario solicita información sobre el número de créditos que posee, se le debe devolver dicho valor.

---

Cuadro 3.1: Requisito funcional «CRED1. Consulta de créditos»

---

Entrada: Número de créditos a comprar

---

Salida: Resultado de la transacción: correcta o fallida

---

El usuario solicita la compra de un número dado de créditos. Se intenta realizar el cargo correspondiente en el medio de pago previamente seleccionado por el usuario. Si el cobro se realiza correctamente, se incrementa el número de créditos del usuario y se le avisa de que la operación se ha realizado correctamente. En el caso de que haya algún fallo en el proceso de cobro, se informa al usuario del error.

---

Cuadro 3.2: Requisito funcional «CRED2. Compra de créditos»

### 3.1.2. Requisitos no funcionales

Los requisitos no funcionales imponen condiciones en el diseño o en la implementación (p.e., para cumplir con restricciones de rendimiento, seguridad o fiabilidad) [4, 5].

De forma general, los requisitos no funcionales definen cómo se supone que es un sistema y se pueden expresar de la forma «el sistema será/estará <requisito>». El plan para implementar los requisitos no funcionales se detallará en la arquitectura del sistema (Apartado 3.3).

Un listado bastante completo de requisitos no funcionales se puede consultar en [5]. Algunos de los requisitos no funcionales que aparecen en dicho listado son: accesibilidad, política de copia de seguridad, certificación, documentación, extensibilidad, interoperabilidad, código abierto o no, rendimiento, compatibilidad entre plataformas, portabilidad, calidad, escalabilidad y usabilidad.

Un requisito no funcional sería por ejemplo: el sistema estará documentado mediante un manual en formato pdf que describirá cómo administrarlo y usarlo.

## 3.2. Diseño del sistema

Este apartado debe presentar el diseño (lógico o de funcionamiento) del sistema que se va a llevar a cabo. Dependiendo del caso, dicho diseño es conveniente ilustrarlo por medio de los siguientes tipos de diagramas [1]:

- Diagramas de casos de uso.
- Diagramas de clases.
- Diagramas de actividades.
- Diagramas de interacción.

## 3.3. Arquitectura del sistema

En este apartado se describe la arquitectura del sistema proyectado. Aspectos que se pueden incluir son, por ejemplo, equipamiento hardware y software necesario, interconexión de los componentes, requisitos mínimos para el funcionamiento —memoria RAM, disco duro, etc.—, documentación proporcionada, etc.

### **3.4. Diseño de la interfaz (en su caso)**

En el caso de que el trabajo desarrollado proporcione una interfaz de usuario diseñada por el proyectando, este apartado debe describir el diseño de dicha interfaz.

## Desarrollo y resultados del trabajo

### Índice

---

4.1. Desarrollo del trabajo . . . . .	11
4.2. Resultados . . . . .	12

---

El trabajo desarrollado y los resultados obtenidos deben explicitarse en este capítulo. Todas las posibles desviaciones con respecto a la planificación inicial deben detallarse y justificarse. De esta forma, los lectores de la memoria podrán entender el porqué de las posibles discrepancias entre los objetivos del trabajo, la planificación que supuestamente debía permitir su obtención y los resultados finales alcanzados.

### 4.1. Desarrollo del trabajo

En este apartado se debe comentar lo más relevante del desarrollo del trabajo. Se puede seguir un orden cronológico, según la planificación, un orden según las tareas del trabajo o según la importancia de los hitos conseguidos. Es fundamental que este apartado enumere los hitos que se han ido consiguiendo y los problemas que hayan producido modificaciones con respecto a los objetivos y plan de trabajo iniciales.

Se debe aportar únicamente la información técnica que sea necesaria para que el lector pueda comprender claramente el trabajo. Dicha información, además, deberá estar presentada de tal forma que sea fácil de entender por alguien que no esté especialmente versado en el tema del trabajo.

Esto implica que la mayor parte de la información técnica, aquella que pueda no ser comprensible o interesante para los evaluadores del trabajo, de-

berá estar en los apéndices apropiados, y ser referenciada desde aquí cuando sea necesario.

## **4.2. Resultados**

En este apartado se describen los resultados del trabajo, siempre haciendo referencia a los objetivos iniciales. El proyectando deberá centrarse en lo hitos conseguidos y, de nuevo, huir de un exceso de detalles técnicos.

También se debe comentar en este apartado las aplicaciones o usos reales o posibles de los resultados del trabajo, así como toda la divulgación que se haya hecho o se vaya a hacer de éstos —repositorios de software, páginas web, etc.

## Conclusiones y trabajo futuro

### Índice

---

5.1. Conclusiones . . . . .	13
5.2. Trabajo futuro . . . . .	13

---

En este capítulo se muestran las conclusiones del trabajo, así como sus futuras ampliaciones.

### 5.1. Conclusiones

En este apartado el estudiante puede comentar libremente las experiencias, profesionales o personales, que haya tenido durante el desarrollo del trabajo, la adecuación del trabajo a los estudios cursados, etc. Este apartado debe ser considerado como un apartado de opinión, cuyo contenido no va a ser evaluado —aunque sí que pueden evaluarse sus aspectos formales.

### 5.2. Trabajo futuro

Si se cree que el trabajo debería o podría continuarse es conveniente incluir este apartado para indicar cómo y en qué direcciones. También se debe indicar si el propio proyectando tiene pensado hacerlo en el futuro.

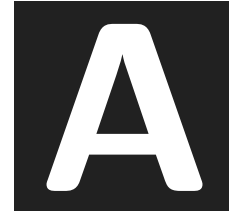




# Bibliografía

- [1] Dan Pilone y Neil Pitman, *UML 2.0 in a nutshell*, O'Reilly, 2005.
- [2] Germán Fabregat Lluca, *Guía para la redacción de las memorias*,  
[http://mermaja.act.uji.es/itis/IS31/guia\\_memoria.pdf](http://mermaja.act.uji.es/itis/IS31/guia_memoria.pdf)
- [3] Javier J. Gutiérrez. *Un ejemplo de requisitos*  
[http://www.lsi.us.es/~javierj/cursos\\_ficheros/02.Unejemploderequisitos.pdf](http://www.lsi.us.es/~javierj/cursos_ficheros/02.Unejemploderequisitos.pdf)
- [4] Wikipedia, *Functional requirements*  
[http://en.wikipedia.org/wiki/Functional\\_requirements](http://en.wikipedia.org/wiki/Functional_requirements)
- [5] Wikipedia, *Non-functional requirements*  
[http://en.wikipedia.org/wiki/Non-functional\\_requirement](http://en.wikipedia.org/wiki/Non-functional_requirement)





## Otras consideraciones

Este apéndice se muestra simplemente para comentar ciertos aspectos que no se han tratado en el resto de la plantilla y como ejemplo del formato que deben tener los apéndices de la memoria. En definitiva, que la memoria del trabajo no tiene por qué tener un apéndice titulado «otras consideraciones».

### A.1. Bibliografía

En la bibliografía deben aparecer todas las referencias a la documentación externa utilizadas para la redacción de la memoria (y, por tanto, para la realización del trabajo). Salvo que se utilice un comando que existe para ello, y que no te voy a decir, solo aparecerán en la bibliografía aquellas entradas que hayan sido citadas desde la memoria.

Las entradas de la bibliografía deben estar ordenadas alfabéticamente.

### A.2. Apéndices

La información técnica del trabajo debe presentarse en los Apéndices. Por información técnica se entiende: el código fuente (ver Apéndice B), las capturas de pantalla, las fotos del sistema implementado, los manuales desarrollados, etc. Por regla general, cada uno de las partes anteriores dará lugar a un apéndice separado.





## Código fuente

Si tu trabajo está relacionado con la informática y has desarrollado una aplicación, puedes mostrar partes del código fuente de dicha aplicación en algunos apartados de la memoria y en uno o varios apéndices. Lo recomendable es que en el cuerpo de la memoria solo muestres aquellas partes del código sobre los que quieras discutir aspectos concretos de la implementación. El código completo de la aplicación que hayas desarrollado, si quieres incluirlo en la memoria, deberá ir en uno o en varios apéndices.

Una de las opciones disponibles en  $\text{\LaTeX}$  para mostrar fragmentos de código es el paquete «`listings`»<sup>1</sup>. Dicho paquete permite cambiar el tipo de letra utilizado para representar distintas partes del código, como pueden ser las palabras reservadas, los comentarios y las cadenas de texto.

Para ilustrar cómo se puede utilizar el paquete «`listings`», se muestra a continuación el código fuente del programa «Hello World!» en varios lenguajes de programación<sup>2</sup>. El código fuente del primer ejemplo está en el fichero `tex` de este apéndice, mientras que el resto se leen directamente de sus respectivos ficheros fuente (ésta última forma es la más recomendable).

Como se van a mostrar códigos en C, C++, Java y Python, en el fichero «`memoria_cabecera.tex`» de esta memoria se han cargado los lenguajes C, C++, Java y Python con la instrucción:

```
«\lstloadlanguages{C, C++, Java, Python}».
```

En el caso de necesitar otros lenguajes, será necesario modificar dicha línea para indicar qué lenguajes se deben cargar. La lista de lenguajes soportados se puede consultar en la documentación del paquete `listings`.

---

<sup>1</sup>*The Listings Package*, <http://www.ctan.org/pkg/listings>.

<sup>2</sup>Los programas se han copiado de *The Hello World Collection*, donde se pueden ver programas «Hello World!» en más de 400 lenguajes de programación.

## Hello world! en C

```
1 /* Hello World in C, Ansi-style */
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 int main(void)
7 {
8     puts("Hello_World!");
9     return EXIT_SUCCESS;
10 }
```

## Hello world! en C++

```
1 // Hello World in ISO C++
2
3 #include <iostream>
4
5 int main()
6 {
7     std::cout << "Hello_World!" << std::endl;
8 }
```

## Hello world! en Java

```
1 // Hello World in Java
2
3 class HelloWorld {
4     static public void main( String args[] ) {
5         System.out.println( "Hello_World!" );
6     }
7 }
```

## Hello world! en Python

```
1 # Hello World in Python
2 print "Hello_World!"
```