

UNIVERSIDAD REY JUAN CARLOS

**E.T.S. DE INGENIERÍA INFORMÁTICA**

PROYECTO FIN DE GRADO

**GRADO EN INGENIERÍA DE COMPUTADORES**

# Aplicación móvil de mejora de rendimiento de baloncesto

**Autor:** Diego Rangel Sanz

**Director:** Director Proyecto

**Codirectora:** Codirector Proyecto

Madrid, 1 de marzo de 2024

*Aplicación móvil de mejora de rendimiento de baloncesto*

Proyecto Fin de Grado, 1 de marzo de 2024

**Autor:** Diego Rangel Sanz

**Director:** Director Proyecto

**Codirectora:** Codirector Proyecto

**E.T.S. de Ingeniería Informática**

Campus de Móstoles, Calle Tulipán s/n.

28933 Móstoles, Madrid

---

Si deseas citar este trabajo, la entrada completa en BibTeX es la siguiente:

```
@mastersthesis{citekey,  
  title = {Aplicación móvil de mejora de rendimiento de baloncesto},  
  author = {Proyecto, A. \& Proyecto, D. \& Proyecto, C.},  
  school = {E.T.S. de Ingeniería Informática},  
  year = {2024},  
  month = {3},  
  type = {Proyecto Fin de Grado}  
}
```

---

Esta obra está bajo una licencia [Creative Commons «Atribución-NoComercial-CompartirIgual 4.0 Internacional»](https://creativecommons.org/licenses/by-nc-sa/4.0/). Obra derivada de <https://github.com/scaverod/TFG-Report-Template.git>.



Todo cambio respecto a la obra original es responsabilidad exclusiva del presente autor.

# Agradecimientos

---

Aquí los agradecimientos que quieras dar. Y si no quieres, borras la entrada `\acknowledgements` de `report.tex` y ya está.

# Resumen

---

Este trabajo de fin de grado presenta el diseño y desarrollo de una aplicación móvil destinada a mejorar el rendimiento de los jugadores de baloncesto, denominada *Hoop Finder*. La aplicación se enfoca en ofrecer herramientas y recursos para el entrenamiento físico, técnico y táctico, adaptados a las necesidades específicas de los usuarios. Se emplean técnicas de análisis de datos para proporcionar un enfoque interactivo y personalizado que motive a los jugadores a alcanzar sus objetivos de mejora.

El desarrollo de la aplicación se basa en proporcionar herramientas para la mejora del tiro del jugador y entrenamiento en equipo. Además, mediante el uso del GPS, los jugadores podrán realizar un seguimiento de su entrenamiento. Se implementan características como un campo de tiro que rastrea la posición del jugador o jugadas de ataque en equipo. Además, se realiza un estudio estadístico para proporcionar al jugador toda la información necesaria para monitorizar sus entrenamientos.

El resultado final es una aplicación móvil intuitiva y completa que ofrece a los jugadores de baloncesto una plataforma integral para mejorar su rendimiento individual y colectivo. Se espera que esta herramienta contribuya significativamente al desarrollo y perfeccionamiento de habilidades en jugadores de todos los niveles, desde amateurs hasta profesionales, y promueva una cultura de aprendizaje continuo en el ámbito del baloncesto.

**Palabras clave:** Aplicación móvil para baloncesto; Mejora del tiro; Seguimiento mediante GPS; Rendimiento individual y colectivo

# Abstract

---

This bachelor's degree final project presents the design and development of a mobile application aimed at improving the performance of basketball players, named *Hoop Finder*. The application focuses on providing tools and resources for physical, technical, and tactical training, tailored to the specific needs of users. Data analysis techniques are employed to provide an interactive and personalized approach that motivates players to achieve their improvement goals.

The development of the application is based on providing tools for improving shooting skills and team training. Additionally, using GPS, players will be able to track their training. Features such as a shooting range that tracks the player's position or team attack plays are implemented. Furthermore, a statistical study is conducted to provide the player with all the necessary information to monitor their training.

The result is an intuitive and comprehensive mobile application that offers basketball players a comprehensive platform to improve their individual and collective performance. It is expected that this tool will significantly contribute to the development and refinement of skills in players of all levels, from amateurs to professionals, and promote a culture of continuous learning in the field of basketball.

**Keywords:** Basketball mobile application; Shooting improvement; GPS tracking; Individual and collective performance

# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.2	Motivación . . . . .	2
1.3	Justificación . . . . .	3
1.4	Estructura de la memoria . . . . .	3
<b>2</b>	<b>Configuración de la memoria</b>	<b>5</b>
2.1	¿Cómo empiezo a escribir la memoria? . . . . .	5
2.2	¿Cómo estructurar la memoria? . . . . .	9
<b>3</b>	<b>Componentes de la plantilla</b>	<b>11</b>
3.1	Columnas . . . . .	11
3.2	Ecuaciones . . . . .	11
3.3	Elementos flotantes . . . . .	13
3.4	Enlaces de hipertexto . . . . .	21
3.5	Fórmulas matemáticas . . . . .	21
3.6	Glosario . . . . .	21
3.7	Notas . . . . .	25
3.8	Referencias cruzadas . . . . .	25

---

3.9	Referencias bibliográficas . . . . .	27
3.10	Referencias cruzadas . . . . .	28
3.11	Referencias a recursos externos . . . . .	28
<b>4</b>	<b>Licencia</b>	<b>29</b>
<b>A</b>	<b>Escuelas y títulos</b>	<b>30</b>
A.1	Escuelas . . . . .	30
A.2	Titulaciones . . . . .	30
<b>B</b>	<b>¿Cómo ampliar la plantilla?</b>	<b>32</b>
<b>C</b>	<b>Lista de paquetes incluidos</b>	<b>33</b>

# Índice de figuras

---

3.1	Vault Boy approves that . . . . .	18
3.2	Todos los Vault Boy . . . . .	20
A.1	Logo de la ETSISI utilizado en la cubierta trasera de la memoria .	31



# Índice de tablas

---

3.1	Opciones para los elementos flotantes de $\text{\LaTeX}$ . . . . .	14
3.2	Comandos para incluir términos del glosario en el texto de la memoria . . . . .	23
3.3	Comandos específicos para controlar la presentación de acrónimos . . . . .	24
A.1	Relación entre el código de la plantilla y la escuela a la que se refiere . . . . .	30

# Índice de listados

---

2.1	Primeras líneas del fichero <code>report.tex</code> . . . . .	6
2.2	Inclusión del fichero de referencias bibliográficas <code>references.bib</code> . . . . .	6
2.3	Configurando autor, título del proyecto y director . . . . .	7
2.4	Configurando codirector . . . . .	7
2.5	Cómo se incluyen los capítulos y los apéndices . . . . .	8
3.1	Ejemplo de inserción de fórmulas en línea . . . . .	11
	<code>sources/adding-blocks.tex</code> . . . . .	14
	<code>sources/adding-blocks.tex</code> . . . . .	14
	<code>sources/adding-blocks.tex</code> . . . . .	15
	<code>sources/adding-blocks.tex</code> . . . . .	15
	<code>sources/snippets.py</code> . . . . .	16
	<code>sources/snippets.py</code> . . . . .	16
	<code>sources/adding-blocks.tex</code> . . . . .	17
3.2	Función para determinar cuando una palabra <code>w1</code> es anagrama de otra palabra <code>w2</code> . . . . .	17
3.3	Inserción de una figura . . . . .	18
3.4	Inserción de varias subfiguras . . . . .	19
3.5	Código para crear una entrada en el glosario . . . . .	22
3.6	Especificando el plural para un término del glosario . . . . .	23

---

3.7	Entrada genérica de una sigla o acrónimo en el glosario . . . . .	23
3.8	Entrada de <code>rpg</code> en <code>glossaries.tex</code> . . . . .	25
3.9	Referenciando una figura y su página . . . . .	26
3.10	Estructura general de una referencia . . . . .	27

# 1.

# Introducción

---

La introducción a este Proyecto de Fin de Grado (PFG) marca el comienzo de un viaje que ha sido concebido con el propósito de abordar una necesidad creciente en el ámbito del baloncesto: la mejora del rendimiento de los jugadores a través de la tecnología móvil. En un contexto donde el desarrollo atlético se fusiona cada vez más con la innovación tecnológica, surge la aplicación móvil *Hoop Finder*, diseñada para proporcionar a los jugadores de baloncesto herramientas y recursos que optimicen su entrenamiento y potencien sus habilidades tanto individuales como colectivas.

La importancia de este proyecto radica en su capacidad para adaptarse a las demandas específicas de los usuarios, ofreciendo un enfoque interactivo y personalizado que se alinea con los avances en el análisis de datos deportivos. A través de la combinación de técnicas de análisis de datos y el aprovechamiento del GPS integrado en los dispositivos móviles, *Hoop Finder* aspira a revolucionar la forma en que los jugadores monitorean y mejoran su desempeño en la cancha.

En este contexto, esta introducción busca generar expectativas al destacar la relevancia de la aplicación para el desarrollo del baloncesto, planteando interrogantes sobre cómo esta herramienta puede influir en la evolución del juego y en el crecimiento de los jugadores. Además, se establecerán los objetivos y alcances del proyecto, así como las suposiciones y limitaciones que guiarán el desarrollo y la evaluación de *Hoop Finder*.

La finalidad principal de *Hoop Finder* es proporcionar a los usuarios una plataforma que les permita alcanzar sus objetivos de mejora de manera efectiva y eficiente. Esto se logra mediante la implementación de funciones avanzadas, como el seguimiento mediante GPS, herramientas para el desarrollo del tiro y el entrenamiento en equipo. Al enfocarse en el diseño y desarrollo de una aplicación móvil que combine tecnología de vanguardia con las necesidades reales de los jugadores de baloncesto, se espera que *Hoop Finder* contribuya significativamente al avance y la profesionalización de este deporte.

## 1.1. App existentes

Información recopilada de todas las apps existentes

## 1.2. Conceptos básicos del baloncesto

Para comprender adecuadamente el funcionamiento de la aplicación, es crucial tener claros los conceptos básicos de este deporte. El baloncesto es un juego de equipo que se practica en una cancha rectangular con dos aros en cada extremo. El objetivo principal es encestar la pelota en el aro contrario mientras se evita que el equipo rival haga lo mismo. Los jugadores pueden moverse driblando la pelota (botándola), pasarla entre sí y realizar tiros a la canasta. La defensa busca interceptar los pases, bloquear tiros y recuperar la posesión de la pelota. Es fundamental entender estos principios para aprovechar al máximo las funcionalidades y consejos que ofrece la aplicación de entrenamiento de baloncesto.

## 1.3. Motivación

Mi motivación para elegir este tema para mi PFG ha sido una combinación de intereses personales y profesionales. Desde hace muchos años, he sido un apasionado del baloncesto y he dedicado gran parte de mi tiempo libre al entrenamiento y participación en este deporte. Durante mi trayectoria como jugador, he sido testigo de las diversas áreas en las que la tecnología puede tener un impacto significativo en la mejora del rendimiento individual y colectivo en el baloncesto.

Además, como estudiante de ingeniería informática, siempre he estado fascinado por el potencial de las aplicaciones móviles para resolver problemas prácticos y mejorar la calidad de vida de las personas. La idea de desarrollar una aplicación móvil que combine mi pasión por el baloncesto con mi experiencia en tecnología móvil me pareció extremadamente emocionante y

motivadora.

## 1.4. Estructura de la memoria

Cómo se organiza y estructura el proyecto en su totalidad. Esta sección presenta un resumen de los diferentes capítulos que conforman la memoria, así como una **muy breve** descripción de su contenido y propósito. Debéis utilizar los comandos `\ref` o `\cref` para referenciar a los capítulos, secciones, figuras, tablas, etc. Por ejemplo, «En el capítulo 1 se presenta la introducción del proyecto.» o «En la sección 2.2 se describen los objetivos del proyecto.». Esto es importante para que el lector pueda navegar por la memoria de forma más sencilla.

Proporciona al lector una visión general de la estructura y el flujo del trabajo, permitiéndole comprender la secuencia lógica de cómo se desarrolla el trabajo o investigación.

## 2.

# Objetivos

---

El TFG se centra en el desarrollo de una aplicación móvil de entrenamiento de baloncesto que utiliza la tecnología GPS para proporcionar una herramienta efectiva para mejorar las habilidades individuales y el rendimiento en equipo de los jugadores. Este proyecto busca integrar funciones de seguimiento de posición para ofrecer ejercicios personalizados en un campo de tiro, así como también facilitar la práctica de jugadas en equipo mediante la simulación de situaciones de juego reales. Los objetivos del TFG se dividen en generales y específicos, abarcando desde el diseño de la interfaz de usuario hasta la implementación de características que aseguren una experiencia de entrenamiento completa y efectiva

### 2.1. General

Los objetivos generales de este Trabajo de Fin de Grado se centran en el desarrollo de una aplicación móvil de entrenamiento de baloncesto que se destaque por su efectividad y accesibilidad para los usuarios. En primer lugar, se busca crear una aplicación que sea intuitiva y fácil de usar, con una interfaz de usuario amigable que permita a los jugadores acceder a las diferentes funcionalidades de entrenamiento de manera sencilla. Este aspecto es crucial para garantizar que la aplicación sea accesible para jugadores de todos los niveles de habilidad, desde principiantes hasta avanzados, y para que puedan aprovechar al máximo las herramientas de entrenamiento disponibles.

Además, se pretende integrar de manera efectiva la tecnología GPS en la aplicación, permitiendo el seguimiento preciso de la ubicación de los jugadores durante las sesiones de entrenamiento. Esto facilitará la personalización de los ejercicios según la posición del jugador en el campo, lo que garantizará un entrenamiento más específico y relevante para las necesidades individuales de cada usuario.

Por último, se aspira a fomentar el aprendizaje en equipo mediante la implementación de funciones que permitan practicar jugadas en conjunto, simulando situaciones de juego reales. Esto no solo mejorará la comprensión del juego por parte de los jugadores, sino que también promoverá la cooperación y la colaboración entre ellos, aspectos fundamentales en el desarrollo de habilidades de juego colectivo en el baloncesto.

En resumen, los objetivos generales de este proyecto se centran en crear una aplicación móvil de entrenamiento de baloncesto efectiva, accesible y que fomente el aprendizaje tanto a nivel individual como en equipo.

## 2.2. Específico

Los objetivos específicos delineados para este proyecto de Trabajo de Fin de Grado se enfocan en aspectos concretos que contribuirán al logro de los objetivos generales establecidos. En primer lugar, se planea diseñar una interfaz de usuario que no solo sea intuitiva, sino que también ofrezca una experiencia fluida y atractiva para los usuarios. Esto implica no solo la disposición visual de los elementos en pantalla, sino también la implementación de controles y funciones que sean fáciles de entender y utilizar, independientemente del nivel de habilidad del usuario.

Además, se buscará integrar la tecnología GPS de manera efectiva en la aplicación, desarrollando algoritmos que permitan el seguimiento preciso de la ubicación de los jugadores en tiempo real. Aunque la aplicación no ajustará dinámicamente el entrenamiento, la tecnología GPS seguirá siendo fundamental para ubicar a los jugadores en un campo virtual y ofrecer ejercicios relevantes para diferentes posiciones en la cancha.

También se pretende desarrollar una amplia base de datos de ejercicios de entrenamiento, centrándose especialmente en el ataque, abarcando una variedad de habilidades como movimientos sin balón, selección de tiros y trabajo en equipo para generar oportunidades de anotación. Esta base de datos servirá como una fuente de recursos para los usuarios, permitiéndoles acceder a una amplia gama de opciones de entrenamiento y personalizar su experiencia según sus necesidades y objetivos específicos de ataque.



Finalmente, se explorará la implementación de herramientas de análisis de rendimiento que permitan a los usuarios evaluar su progreso y áreas de mejora en diferentes aspectos del juego ofensivo. Esto podría incluir métricas como la efectividad en la ejecución de jugadas, la precisión de los pases y la capacidad para crear oportunidades de anotación. En conjunto, estos objetivos específicos se diseñan para contribuir al desarrollo de una aplicación de entrenamiento de baloncesto integral y efectiva, que satisfaga las necesidades de los jugadores y promueva su mejora continua en el juego, especialmente en el aspecto ofensivo.

## **3.** **Desarrollo del Software**

---

Escribir todo el proceso de desarrollo realizado

### **3.1. Metodologia**

Metodologia seguida en el desarrollo en este caso la metodologia agil scrum

### **3.2. Prototipos**

Explicar toda la fase del desarrollo del diseño y prototipos de la aplicación

### **3.3. Herramientas utilizadas**

GitHub, Figma, IntelliJ...

## 4.

# Resultados

---

Escribir los resultados conseguidos en el desarrollo y explicar el funcionamiento final de la aplicación

# 5. Conclusiones y trabajos futuros

---

Conclusiones finales del PFG

**6.**

**Anexo**

---

## 7. Configuración de la memoria

---

La estructura de esta plantilla está dividida en varios ficheros. Aunque puede parecer complicado, en realidad no es tanto:

- `./appendices/`: Los fuentes de los capítulos de apéndices, vamos, los que van al final y que se usan como adenda a la memoria como tal.
- `./chapters/`: Los fuentes de los capítulos que forman parte del cuerpo de la memoria.
- `./figures/`: Las figuras (imágenes, diagramas) que se usarán a lo largo de todo el documento.
- `./frontmatter/`: Los fuentes de todo aquello que se incluye antes del cuerpo de la memoria. Por ahora, todas las entradas del glosario.
- `./sources`: Ficheros con fuentes que se incluyen dentro de listados de fuentes del documento.
- `./references.bib`: Los fuentes en `BIBTEX` de la bibliografía que se referencia en la memoria en la memoria.
- `./report.tex`: El fichero principal a partir del cual se compila todo el proyecto.
- `./upm-report`: El directorio que tiene todo el contenido que hace que la memoria se vea así. Si tocas aquí, que sea con mimo y cariño, porque es muy fácil cargárselo todo.

### 7.1. ¿Cómo empiezo a escribir la memoria?

Por el principio, es decir, con el fichero `report.tex`. Veamos la primera línea del fichero (listado 2.1).

**Listado 7.1.** Primeras líneas del fichero `report.tex`

---

```
\documentclass[%  
    school=etsii,%  
    type=pfg,%  
    degree=GIC,%  
    authorsex=m,%  
    directorsex=m,%  
{urjc-report}
```

---

En este punto es donde se configura gran parte de la plantilla. Los parámetros y sus opciones son las siguientes:

- `school`: La escuela a la que pertenece el estudiante. La idea de la plantilla es que se use a lo largo de todas las escuelas de la UPM, y que cada una de ellas tenga su propia configuración. La escuela determinará, entre otras cosas, direcciones y colores principales. Las opciones se describen en el [Apéndice A](#).
- `type`: El tipo de memoria. Modifica algunos textos, incluida la portada. Puede tomar los valores `pfg` ([Proyecto Fin de Grado](#)) y `pfm` ([Proyecto Fin de Máster](#)).
- `degree`: El grado al que aspira el estudiante. De momento sólo están definidos los grados que se imparten en la ETSISI (también en el [Apéndice A](#)).
- `authorsex`: Puede ser `m` (masculino) o `f` (femenino), y sirve para modificar algunos textos relacionados con el sexo del estudiante.
- `directorsex`: Similar al parámetro `authorsex`, pero para el director del proyecto.

Tras esta configuración, se incluye el fichero de referencias bibliográficas (listado [2.2](#)):

**Listado 7.2.** Inclusión del fichero de referencias bibliográficas `references.bib`

---

```
\addbibresource{references.bib}
```

---

El tema de las referencias bibliográficas se explica en el [Sección 3.9](#) del [Capítulo 3](#). En principio no habría que tocar nada (bueno sí, añadir las fuentes y referenciarlas), pero si las referencias se tienen en otro fichero, bastaría con cambiar el nombre al de dicho fichero.

Los cinco siguientes comandos, como se ve en el listado [2.3](#), indican el título del proyecto, el nombre del autor y su entrada en la bibliografía, y el nombre del director y su entrada en la bibliografía.

**Listado 7.3.** Configurando autor, título del proyecto y director

---

```
\title{Aplicación movil de mejora de rendimiento de baloncesto}  
\author{Diego Rangel Sanz}  
\bibauthor{Proyecto, A.}  
\director{Director Proyecto}  
\bibdirector{Proyecto, D.}
```

---

El único misterio es eso de las entradas bibliográficas para el autor y el director. No es más que los nombres que aparecen más adelante cuando se indica «cómo citar el proyecto». El día que aprenda cómo hacerlo automáticamente, será una configuración que desaparezca<sup>1</sup>.

Para incluir un codirector o codirectora, en el caso de que el trabajo lo requiera, simplemente descomenta la línea que verás a continuación (listado [2.4](#)).

**Listado 7.4.** Configurando codirector

---

```
\codirector[f]{Codirector Proyecto}{Proyecto, C.}
```

---

El primer parámetro dentro de los corchetes [f/m] indica el género del codirector (m para masculino, f para femenino). A continuación, proporciona el

---

<sup>1</sup>Bueno, y si tú, queridísimo lector o lectora sabes cómo hacerlo, hazme un *pull request* al repositorio de la plantilla: <https://github.com/scaverod/TFG-Report-Template.git>.



nombre completo del codirector y el nombre para la referencia bibliográfica. Recuerda que esta configuración es opcional, si no se necesita un codirector, simplemente deja esta línea comentada. El documento generará automáticamente la portada y el contenido sin incluir la sección del codirector.

Tras ello, empieza el primer contenido de verdad: **resumen** y **abstract**, cada uno con sus palabras clave asociadas. Ambos dos son obligatorios y se añaden con la macro `\abstract`, donde se especificarán el idioma (`spanish` o `english`) y el contenido. De la misma manera, las palabras clave se añaden con la macro `\keywords`. Ni que decir tiene que ambos deben tener el mismo contenido, uno en español y el otro en inglés. Y además es obligatorio (según la normativa de la ETSISI).

Existe la opción de añadir agradecimientos a través de la macro `\acknowledgements`. Es opcional, así que si no se pone no se renderiza en el documento final, pero es algo bonito y a las abuelas les encanta aparecer ahí. Y las abuelas son de lo más bonito que existe en este mundo, así que cuidadlas.

Y ahora sí, se empieza con el grueso del documento. Tras incluir el glosario, del que se hablará en la sección 3.6 del 3, se comenzarán a incluir uno tras otro todos los capítulos de los que se compone nuestra memoria, tal y como se muestra en el listado 2.5.

#### Listado 7.5. Cómo se incluyen los capítulos y los apéndices

---

`achieve their improvement goals.`

`The development of the application is based on providing tools for improving shooting skills and team training. Additionally, using GPS, players will be able to track their training. Features such as a shooting range that tracks the player's position or team attack plays are implemented. Furthermore, a statistical study is conducted to provide the player with all the necessary information to monitor their training.`

`The result is an intuitive and comprehensive mobile application that offers basketball players a`

```
comprehensive platform to improve their
individual and collective performance. It is expected that
this tool will significantly contribute to the
development and refinement of skills
in players of all levels, from amateurs to professionals,
and promote a culture of continuous learning in the
field of basketball.
}
\keywords{english}{Basketball mobile application; Shooting
improvement; GPS tracking; Individual and collective
performance}
```

---

La macro `\appendix` del medio indica a partir de qué punto se añaden los apéndices. No son obligatorios, ni mucho menos, pero en algunos **Proyectos Fin de Grado (PFGs)** y **Proyectos Fin de Máster (PFMs)** se suelen incluir para dar información adicional de contexto que no es el objetivo de la memoria, pero sí interesante para complementar. Por ejemplo, en un **PFM** para el estudio del comportamiento de conductores al volante, uno de los apéndices podría ser cada uno de los formularios que se le ofrecieron para rellenar a cada uno de los conductores de dicho estudio.

Y ya estaría todo. Resumiendo, hay que configurar la plantilla, poner el autor, título y director del proyecto e incluir los capítulos y apéndices que queramos.

## 7.2. ¿Cómo estructurar la memoria?

La respuesta rápida es «como buenamente quieras/puedas». En realidad la estructura de la memoria va a depender del tipo de trabajo desarrollado, pero con carácter general, los trabajos suelen seguir ciertas estructuras.

Un **PFG** es un trabajo cuyo propósito es demostrar que se han llegado a adquirir las competencias asociadas con la titulación cursada. Con esto queremos decir que, a diferencia de otros tipos de trabajo académico, en éste no es necesario realizar aportaciones originales al estado de la cuestión.

Una estructura típica de TFG es la siguiente:

1. Resumen
2. Introducción
3. Estado de la cuestión
4. Metodología
5. Resultados y Discusión
6. Conclusiones
7. Apéndices
8. Referencias bibliográficas
9. Glosario

Un **PFM**, a diferencia de un **PFG** trata de profundizar más en un campo concreto de una disciplina, por lo que tiene a ser más extenso y mucho más específico.

En términos generales, la estructura es similar. Sin embargo es de esperar que el nivel de exigencia sea mayor, ya que el estudiante que lo realiza debe demostrar que es un titulado superior. Esto se nota más en la fase de documentación, ya que al tratar de profundizar en un tema más específico, el trabajo de contextualizar y argumentar es más tedioso.

Se pueden identificar dos tipos de proyectos diferentes, aquellos que podríamos catalogar de *profesionales*, con enfoque a la innovación o mejora en un área profesional concreta, y aquellos *de investigación*, más enfocados a la búsqueda de nuevo conocimiento en el área, y que suelen ser el comienzo de la carrera investigadora.

## 8. Componentes de la plantilla

---

En este capítulo hablaremos de los componentes principales con los que trabajaremos en nuestra memoria.

### 8.1. Columnas

TBD

### 8.2. Ecuaciones

La facilidad de composición de ecuaciones es una de las cosas que más atrae de  $\text{\LaTeX}$  a muchos autores.  $\text{\LaTeX}$  mantiene dos renderizadores diferentes, uno para el texto y otro para las ecuaciones, denominados modo párrafo y modo matemático<sup>1</sup>. El modo párrafo es el modo por defecto y no se le llama explícitamente. Al modo matemático, sin embargo, se le invoca de varias maneras diferentes.

#### 8.2.1. Modo en párrafo

La forma más común es la forma “en línea”, donde el texto para el modo matemático se encierra entre dos signos  $\$$ . Por ejemplo, veamos la frase del listado 3.1.

**Listado 8.1.** Ejemplo de inserción de fórmulas en línea

---

<sup>1</sup>Existe un tercer modo, denominado *LR mode* o *left-to-right mode*, raramente utilizado y que no trataremos aquí

---

El pequeño teorema de Fermat dice que si  $p$  es un número primo, entonces, para cada número natural  $a$ , con  $a > 0$ ,  $a^p \equiv a \pmod{p}$

---

La frase quedaría como sigue:

El pequeño teorema de Fermat dice que si  $p$  es un número primo, entonces, para cada número natural  $a$ , con  $a > 0$ ,  $a^p \equiv a \pmod{p}$

### 8.2.2. Ecuaciones en bloque

Cuando en lugar de poner una ecuación dentro de un párrafo existente la queremos insertar en su propio espacio independiente hacemos uso de los entornos `equation` o `align`, dependiendo de si queremos una o más ecuaciones en el bloque, respectivamente. Por ejemplo, en el caso de una única ecuación, sería similar al ejemplo siguiente:

---

<pre>\begin{equation}   \int_a^b f'(x) \, dx = f(b) - f(a) \end{equation}</pre>	$\int_a^b f'(x) \, dx = f(b) - f(a) \quad (8.1)$
---	--

---

Sin embargo, en el caso de que quisiésemos más de una ecuación en el mismo bloque haríamos uso del carácter `&` para indicar en qué punto se alinean las ecuaciones; por ejemplo:

---

<pre>\begin{align}   u &amp;= \arctan x \\   du &amp;= \frac{1}{1+x^2} dx \end{align}</pre>	$u = \arctan x \quad (8.2)$ $du = \frac{1}{1+x^2} dx \quad (8.3)$
---	---

---

Por último, si no tenemos por qué referenciarlas en el texto, podemos hacer

uso de los entornos `equation*` y `align*`

---

```
\begin{equation*}
  \int_a^b f'(x) \, dx = f(b) - f(a)
\end{equation*}
```

---

$$\int_a^b f'(x) \, dx = f(b) - f(a)$$

## 8.3. Elementos flotantes

Vamos a hablar un poco de los elementos denominados «flotantes» o *floats*. Éstos son bloques de contenido que «flotan» por la página hasta que  $\text{\LaTeX}$  los coloca donde considera a través de ciertos algoritmos.

Lo general es **declarar el elemento flotante inmediatamente después del párrafo donde se ha referenciado** (las referencias cruzadas las vemos en la [Sección 3.8](#) de este capítulo), y después dejar que  $\text{\LaTeX}$  elija el mejor sitio. Y si después de haber escrito todo el documento hay algo que no cuadre, pues ahí modificarlo.

«¿Y si no hago referencia a un elemento flotante, dónde lo pongo?» Bueno, pues en general no se pone. Si algo no es nombrado, suele ser porque no aporta información, y si no la aporta, pues para qué lo vamos a meter. Ojo, que lo mismo existe algún caso donde sí tiene sentido, pero es muy raro.

Aunque más adelante veremos los diferentes tipos de *floats*, en caso de que queramos modificar su comportamiento todos tienen los especificadores de posición indicados en la [tabla 3.1](#):

### 8.3.1. Código fuente

Para la gestión de los listados de código fuente se utiliza el paquete `listings`. El estilo usado es una modificación<sup>2</sup> de `Solarized` desarrollado por Ethan Schoonover.

---

<sup>2</sup>En realidad la modificación es cambiar el fondo de crema a blanco

**Tabla 8.1.** Opciones para los elementos flotantes de  $\text{\LaTeX}$ 

Especificador	Acción
H	Colocar exactamente en el sitio indicado
h	Colocar aproximadamente en el sitio indicado
t	Colocar al comienzo de la página
b	Colocar al final de la página
p	Colocar en una página exclusiva para elementos “flotantes”
!	Forzar las opciones obviando los mecanismos internos de $\text{\LaTeX}$

Existen muchas formas diferentes de incluir listados de código en una memoria. Aquí introducimos los más comunes.

### Código dentro del propio párrafo

TBD

### Bloques de código

Insertar código en párrafos no es tan común como insertar bloques enteros. Para ello haremos uso del entorno `\lstlisting`. Por ejemplo:

---

```
\begin{lstlisting}
from collections import Counter

def is_anagram(w1, w2):
    return Counter(w1) == Counter(w2)
\end{lstlisting}
```

---

Nos daría el siguiente resultado:

---

```
from collections import Counter
```

```
def is_anagram(w1, w2):  
    return Counter(w1) == Counter(w2)
```

---

Una cosa que hay que tener en cuenta es que dentro de un entorno `lstlisting` se ignoran todos los comandos de  $\text{\LaTeX}$ <sup>3</sup> y el texto se imprime tal y como se ha introducido. Esto incluye los tabuladores y espacios de principio de línea.

Al igual que en el código de párrafo, también podemos especificar en qué lenguaje está escrito el código para que se resalten en éste las palabras reservadas. Por ejemplo:

---

```
\begin{lstlisting}[language=python]  
from collections import Counter  
  
def is_anagram(w1, w2):  
    return Counter(w1) == Counter(w2)  
\end{lstlisting}
```

---

Nos daría como resultado el siguiente bloque de código:

---

```
from collections import Counter  
  
def is_anagram(w1, w2):  
    return Counter(w1) == Counter(w2)
```

---

### Código directamente desde fichero

Esta forma es muy común, ya que se usa tanto para hacer referencia al código fuente de la aplicación directamente, como a código separado en ficheros para mantener el tamaño de la memoria manejable.

Suponiendo que tenemos el fichero `sources/snippets.py`, para incluirlo en-

---

<sup>3</sup>En realidad no todos, y si no mira en estos fuentes cómo hemos metido el fin de bloque `lstlisting` dentro del propio bloque.



tero basta con usar el comando `\lstinputlisting`:

---

```
\lstinputlisting[language=python]{sources/snippets.py}
```

---

Con este comando conseguiríamos el siguiente resultado:

---

```
def all_unique(l):  
    return len(l) == len(set(l))  
  
def is_palindrome(l):  
    return l == l[::-1]
```

---

En caso de que se desease importar sólo parte del fichero, se pueden indicar las filas que delimitan el trozo de código. Por ejemplo:

---

```
\lstinputlisting[  
    language=python,  
    firstline=4,  
    lastline=5  
]{sources/snippets.py}
```

---

Esto haría que sólo se imprimiesen las filas 4 y 5, correspondientes a la segunda función:

---

```
def is_palindrome(l):  
    return l == l[::-1]
```

---

Ambas opciones son opcionales, y los valores por defecto de `firstline` y `lastline` serán el principio y el final del fichero respctivamente.

## Etiquetando bloques de código

Al igual que con el resto de bloques *float* (e.g. figuras o tablas), se pueden<sup>4</sup> (**deben**) añadir pies de texto a los bloques de código, lo cual hace más legible su cometido.

Para ello basta con añadir el argumento `caption` a las opciones del bloque. Por ejemplo:

---

```
\begin{lstlisting}[language=python, caption=Función para
    determinar cuando una palabra \texttt{w1} es anagrama de otra
    palabra \texttt{w2}]
from collections import Counter

def is_anagram(w1, w2):
    return Counter(w1) == Counter(w2)
\end{lstlisting}
```

---

Nos daría como resultado el siguiente bloque de código:

**Listado 8.2.** Función para determinar cuando una palabra `w1` es anagrama de otra palabra `w2`

---

```
from collections import Counter

def is_anagram(w1, w2):
    return Counter(w1) == Counter(w2)
```

---

### 8.3.2. Figuras

El código siguiente (listado 3.3) renderizará una imagen.

---

<sup>4</sup>Pongo *pueden* porque es opcional, pero en realidad se **deben** poner, porque si no los listados de fuentes quedan horribles, como el de esta plantilla por ejemplo (échale un vistazo si no lo has hecho antes).

**Listado 8.3.** Inserción de una figura

---

```
\begin{figure}[H]
  \centering
  \includegraphics[width=0.25\textwidth]{figures/vault-boy.
    png}
  \caption{\label{fig:img-vault-boy} Vault Boy approves that}
\end{figure}
```

---

La sintaxis es bastante autoexplicativa. El entorno `figure` es el que delimita el contenido de la figura. El comando `centering` determina que se tiene que centrar. Luego, el comando `caption` determina el pie de imagen, el cual además incluye una etiqueta (comando `label`) que sirve para referenciar. Por último, se incluye la imagen con el comando `includegraphics`.

En definitiva, la imagen (figura 3.1) se mostrará con un ancho igual al 25 % del ancho que ocupa el texto, centrada, con un pie de foto y una etiqueta para referenciar.



**Figura 8.1.** Vault Boy approves that

El comando `includegraphics` puede importar los formatos típicos de imagen, como jpeg, png o pdf. También admite una serie de opciones como rotación, alto, ancho (éste le hemos especificado con `width`), etcétera. ¡Ojo! Siempre que se pueda, hay que intentar insertar imágenes vectoriales. De esta manera, se mantiene la calidad de la imagen. Si no, puede ocurrir que se pixelee y no quede nada bien.

## Subfiguras

¿Y qué pasa cuando queremos incluir múltiples imágenes dentro de una figura? Bueno, pues aquí hay que usar el entorno `subfigure`. En el listado 3.4 vemos un ejemplo de cómo se manejan.

**Listado 8.4.** Inserción de varias subfiguras

---

```
\begin{figure}[H]
  \centering
  \begin{subfigure}{.3\textwidth}
    \includegraphics[width=\linewidth]{figures/vault-boy.png}
    \caption{\label{fig:subfigure-1}Vault Boy 1}
  \end{subfigure}%
  \begin{subfigure}{.3\textwidth}
    \includegraphics[width=\textwidth]{figures/vault-boy.png}
    \caption{Vault Boy 2}
  \end{subfigure}%
  \begin{subfigure}{.3\textwidth}
    \includegraphics[width=\textwidth]{figures/vault-boy.png}
    \caption{Vault Boy 3}
  \end{subfigure}
  \caption{\label{fig:subfigures}Todos los Vault Boy}
\end{figure}
```

---

En realidad cada subfigura se trata como una figura normal, pero en relación con el *float* contenedor. Cuando a una subfigura se le especifica un ancho, se le está diciendo al compilador de qué ancho es esa subfigura en concreto (en nuestro caso 0.3 veces el ancho de la línea). Sin embargo, a la imagen se le da un ancho total de `linewidth`, porque al estar dentro de su espacio de subfigura, el ancho ha cambiado. El resultado es el que se observa en la figura 3.2. Por cierto, también podemos referenciar a los pies de las subfiguras (e.g. Así: 3.2a).



(a) Vault Boy 1

(b) Vault Boy 2

(c) Vault Boy 3

**Figura 8.2.** Todos los Vault Boy

### 8.3.3. Tablas

Las tablas (en realidad «cuadros») son una forma muy eficaz de presentar información. En los resultados de casi cualquier trabajo existen cuadros de algún tipo para que los datos se comprendan de un único vistazo (o para que al menos sea más fácil identificarlos).



#### ¿Por qué “cuadro” en lugar de “tabla”?

Tal y como se indica en el [FAQ de CervanTex](#), *table* (inglés) y *tabla* (español) son falsos amigos; el inglés *table* tiene un sentido más general que el español *tabla*, cuyo uso es únicamente para aquellos cuadros dedicados a la disposición de números (e.g. tabla de multiplicar o tabla de logaritmos).

Sin embargo, las tablas suelen ser bastante complicadas en  $\text{\LaTeX}$ , por lo menos para la gente que empieza. Para no escribir demasiado, la respuesta para casi toda maquetación de tabla está en <https://www.tablesgenerator.com/>. En serio, no perdáis el tiempo si no es estrictamente necesario. La maquetáis visualmente, la generáis (con estilo *booktabs*) y a correr.

### 8.3.4. Pseudocódigos

En el algoritmo 1 aparece un ejemplo en pseudocódigo.

---

**Algorithm 1** *Additional Louvain* **input**=( $A, \mathcal{M}$ ) **output**= $P$ 


---

```

1:  $\forall i \in V$ , let  $i$  be an isolated community
2:  $o = \text{permutation}(V)$ 
3: for  $k \in o$  do
4:   search in  $A$  all the neighbours of  $k, j$ 
5:    $\forall j$ , calculate  $\Delta Q_k(j)$  in matrix  $\mathcal{M}$ 
6:    $j^* = \{ j \mid \Delta Q_k(j^*) = \max_j \{ Q_k(j) \} \}$ 
7:   if  $\Delta Q_k(j^*) > 0$  then
8:     Move node  $k$  to  $j^*$  's community
9:   else
10:     $k$  remains in its community
11:   end if
12: end for

```

---

## 8.4. Enlaces de hipertexto

TBD

## 8.5. Fórmulas matemáticas

TBD

## 8.6. Glosario

El glosario de una memoria es el lugar donde se encuentran los términos que se usan a lo largo del documento y que se considera que requieren una aclaración. En esta plantilla, en el momento que generemos un término, se creará un capítulo al final de la memoria con el listado de todos aquellos términos definidos.

Para gestionar el glosario se hace uso del paquete `glossaries` el cual es relativamente complejo de configurar. También su documentación es muy ex-

tenso<sup>5</sup>, así que en esta sección hablaremos únicamente de lo esencial.

### 8.6.1. ¿Cuándo y cómo especificar términos?

La regla general del «cuándo» es una vez terminada la memoria. En ese punto, seremos conscientes de qué términos son los más interesantes para incluir en el glosario. En ese punto deberemos ir término por término sustituyéndolo por la entrada del glosario para que el proceso automático se encargue de la indexación y numeración de páginas.

El «cómo» se refiere a de qué manera escribirlos. La regla general en el castellano (y hasta donde el autor de la plantilla sabe, en cualquier idioma) es de la manera en la que aparecería en medio del texto. Es decir, si la palabra se escribe generalmente en minúscula (e.g. *El jugador blandía un **hacha de batalla***) se deberá incluir dentro del glosario en minúscula, mientras que si se escribe generalmente en mayúscula (e.g. *Encontró el **Arco de la Perdición***) irá en mayúscula.

### 8.6.2. Definiendo los términos del glosario

Las entradas se escribirán dentro del fichero `frontmatter/glossary.tex`. La forma estándar de definir un término es la que se muestra en el listado 3.5.

**Listado 8.5.** Código para crear una entrada en el glosario

---

```
\newglossaryentry{hacha-batalla}{  
  name={hacha de batalla},  
  description={Herramienta antigua utilizada en combate}  
}
```

---

Luego, dentro del texto, podremos hacer referencia a dichas entradas con los comandos que se muestran en la tabla 3.2.

---

<sup>5</sup>Pero aún así es el sitio donde ir a buscar información. El paquete, junto con su documentación está disponible en la dirección <https://www.ctan.org/pkg/glossaries>.

**Tabla 8.2.** Comandos para incluir términos del glosario en el texto de la memoria

Comando	Ejemplo con la clave <b>hacha-batalla</b>
<code>\gls</code>	<i>hacha de batalla</i>
<code>\Gls</code>	<i>Hacha de batalla</i>
<code>\glspl</code>	<i>hacha de batallas</i>
<code>\Glspl</code>	<i>Hacha de batallas</i>

Como los plurales los gestiona automáticamente, puede ser que queramos, como en este caso, modificar el plural de nuestro término. Para ello debemos añadir la opción `plural` a la entrada para especificar cómo es el plural de la entrada, como se muestra en el listado 3.6.

**Listado 8.6.** Especificando el plural para un término del glosario

```
\newglossaryentry{python}{
  name={Python},
  plural={Pythonacos},
  description={El mejor lenguaje de programación}
}
```

Así, el plural de la clave `python` descrita quedaría como *Pythonacos*, en lugar del valor por defecto que sería *Pythons*.

Un caso particular de términos del glosario son las siglas y los acrónimos. No vamos a entrar en detalle aquí<sup>6</sup> sino que vamos a introducir las siglas como caso especial de entrada de glosario. Cuando tengamos una sigla, la crearemos en el glosario como se muestra en el listado 3.7.

**Listado 8.7.** Entrada genérica de una sigla o acrónimo en el glosario

```
\newacronym[
  description={Proyecto Fin de Grado. Proyecto a realizar al
    final de una titulación de Grado},
```

<sup>6</sup>Pero recomendamos visitar <https://www.fundeu.es/recomendacion/siglas-y-acronimos-claves-de-redaccion/> y darle una leída porque es interesante.



---

```
longplural={Proyectos Fin de Grado}
]{pfg}{PFG}{Proyecto Fin de Grado}
```

---

En el ejemplo se puede ver que hay dos entradas, `longplural` y `description` que son opcionales. La primera es la equivalente a `plural` de `newglossaryentry`, y no necesita más explicación.

La segunda, `description` suele utilizarse para acrónimos, cuando necesitamos describir la entrada. Cuidado en este caso porque si hace referencia a varias palabras estas se deberían incluir dentro de la descripción (como en el ejemplo, «**Proyecto Fin de Grado**»).

La regla general de los acrónimos y las siglas es que la primera vez que aparecen en el texto, deben aparecer con el nombre completo mientras que el resto de veces pueden aparecer indistintamente como sigla o forma larga. De esto se encarga automáticamente el comando `gls`. Es decir, si tenemos la sigla `special`, la primera vez que incluyamos la sigla con `\gls{special}` saldrá *Strenght, Perception, Endurance, Charisma, Intelligence, Agility & Luck (SPECIAL)* mientras que el resto de veces que la incluyamos se verá simplemente *SPECIAL*.

Con los acrónimos se incluyen comandos adicionales para controlar su presentación. Estos son los mostrados en la tabla 3.3

**Tabla 8.3.** Comandos específicos para controlar la presentación de acrónimos

Comando	Ejemplo con la clave <b>rpg</b>
<code>\acrshort</code>	<i>RPG</i>
<code>\acrshortpl</code>	<i>RPG</i>
<code>\acrlong</code>	<i>Role-Playing Game</i>
<code>\Acrlong</code>	<i>Role-Playing Game</i>
<code>\acrlongpl</code>	<i>Role-Playing Games</i>
<code>\Acrlongpl</code>	<i>Role-Playing Games</i>
<code>\acrfull</code>	<i>Role-Playing Game (RPG)</i>
<code>\Acrfull</code>	<i>Role-Playing Game (RPG)</i>
<code>\acrfullpl</code>	<i>Role-Playing Games (RPG)</i>
<code>\Acrfullpl</code>	<i>Role-Playing Games (RPG)</i>

---

Por cierto, en castellano las siglas **no incluyen la «s» al final**, así que no deberíamos usar los comandos que terminan en `pl`. Por eso la definición que se ha hecho de la sigla `rpg` es la mostrada en la figura 3.8.

**Listado 8.8.** Entrada de `rpg` en `glossaries.tex`

---

```
\newacronym[
  description={Role-Playing Game. Juego de rol},
  shortplural={RPG}
]{rpg}{RPG}{\textit{Role-Playing Game}}
```

---

## 8.7. Notas

TBD

## 8.8. Referencias cruzadas

Las etiquetas (*label*) son una herramienta muy útil en el proceso de composición tipográfica. Se puede pensar en ellas como punteros a zonas de interés del documento, de tal manera que se les pueda referenciar sin necesidad de conocer su posición final en la composición.

Por ejemplo, lo normal es que en un libro, a la hora de referenciar una figura, aparezca una frase del estilo “[...] como muestra la Figura 3 [...]”. Lo que es bastante raro son las frases del estilo “[...] como muestra la Figura de los muñecos amarillos [...]” o “[...] como muestra la siguiente Figura [...]”<sup>7</sup>.

Una de las propiedades más útiles y, en ocasiones, infravaloradas de  $\text{\LaTeX}$  es la facilidad y potencia de su sistema de etiquetado. Este sistema permite referenciar tablas, listados de código fuente, ecuaciones, capítulos, secciones, etc.,

---

<sup>7</sup>Sí, bueno, quizá la segunda frase no es tan rara, pero siempre es preferible referenciar directamente a dar posiciones relativas.

con facilidad y flexibilidad. Además,  $\text{\LaTeX}$  las numera y referencia automáticamente, cambiando la numeración en función de las adiciones y supresiones sin que el autor tenga que hacer nada.

Para referenciar un elemento, lo primero que hay que crear es una etiqueta **después** del elemento a referenciar. Esto ya lo hemos visto anteriormente, por ejemplo en el listado 3.3. Si nos fijamos, se declara una etiqueta justo después de la etiqueta `caption` con el nombre `fig:img-vault-boy`. De esta manera, podemos referenciar varios indicadores de la figura, como se muestra en el listado 3.9

#### Listado 8.9. Referenciando una figura y su página

---

```
Mira la Figura~\ref{fig:img-vault-boy} en la página~\nameref{fig:img-vault-boy}.
```

---

Dicho listado daría el siguiente resultado:

«Mira la Figura 3.1 titulada **Vault Boy approves that** en la página 18.»



**¿Por qué a mí me aparece el símbolo ?? en lugar de una referencia?** Pues lo más seguro es que sea un error a la hora de escribir la etiqueta. Menos común, pero también puede pasar, es que el documento no se haya compilado bien. Hay que tener en cuenta que  $\text{\LaTeX}$  fue creado en una época donde las máquinas tenían poca (¡poquísima!) RAM, y para funcionar lo que se hacían eran varias compilaciones sobre el documento, almacenando los valores temporales en ficheros. Y como nadie quiere perder tiempo en cambiar y *debuggear* algo que funciona estupendamente bien, no se reimplementa. De todas formas, si te animas, ahí tienes un buen proyecto que si lo sacas adelante te va a hacer muy famoso.

## 8.9. Referencias bibliográficas

Hay muchas formas diferentes de gestionar las referencias bibliográficas, así que aquí hemos decidido elegir una de ellas por considerarla la más cómoda y simple, que es mediante el paquete *biblatex*.

El fichero de referencias, `references.bib`, incluirá una entrada por cada una de las referencias que se citan durante la memoria. Luego, en el cuerpo del texto, se podrán hacer referencias a dichas entradas y será  $\text{\LaTeX}$  después quien se encargue de indexar correctamente, crear los hipervínculos y maquetar automáticamente.

El fichero `references.bib` puede tener muchas más de las referencias que se citan en el cuerpo del texto. Sin embargo, sólo aparecerán las referencias que se citen en el texto.



**No has dicho en ningún momento *bibliografía*** Sí. Las referencias bibliográficas, también conocidas como lista de referencias o simplemente referencias, son todas aquellas fuentes bibliográficas que han sido citadas a lo largo del documento. La bibliografía, también conocida como referencias externas, es simplemente una lista de recursos utilizados, citados o no. Como generalmente los no referenciados no se usan para dar soporte a un texto científico se suelen descartar.

### 8.9.1. ¿Cómo creamos nuevas referencias?

El fichero `references.bib` contará cero o más entradas con la estructura mostrada en el listado 3.10.

**Listado 8.10.** Estructura general de una referencia

```
@tipo{id,  
  author = "Autor",  
  title = "Título de la referencia (libro, artículo, enlace,
```

```
    ...)",  
    campo1 = "valor",  
    campo2 = "valor",  
    \ldots  
}
```

---

En esta entrada, `@tipo` indica el tipo de elemento (p. ej. `@article` para artículos o `@book` para libros) e `id` es un identificador **único en todo el documento** para el elemento. El resto de campos dependerán del tipo de la referencia, aunque generalmente casi todos los tipos comparten los campos de `author`, `title` o `year`.

### 8.9.2. ¿De qué manera puedo citar las referencias?

TBD [1]

## 8.10. Referencias cruzadas

TBD

## 8.11. Referencias a recursos externos

TBD

Cuando se publica la obra en el archivo digital, por defecto lo hace con la licencia de *Creative Commons* Reconocimiento - Sin obra derivada - No comercial. Aunque usar esta licencia es correcto, no es una licencia libre y a algunos nos parece algo malo.

Considero que todo el conocimiento generado en una universidad pública ha de ser público y libre. Por ello esta obra se publica con la licencia *Creative Commons* Reconocimiento - Sin obra derivada - No comercial - Compartir igual, de tal manera que se garantiza que la obra se comparte con las mismas libertades y así, todo el mundo puede hacer de esta obra el uso que quiera.

# A.

## Escuelas y títulos

A continuación se describen todas las opciones de grados y títulos disponibles en la memoria.

### A.1. Escuelas

Las escuelas disponibles se describen en el cuadro [A.1](#).

Clave	Valor
etsii	E.T.S. de Ingeniería Informática

**Tabla A.1.** Relación entre el código de la plantilla y la escuela a la que se refiere

De momento no están todas, así que si te apetece añadir la tuya puedes, o bien contactar con los autores, o bien modificarlo (mira el apéndice [B](#)) y también contactar con los autores, así lo podemos hacer público con la mayor cantidad de usuarios posible.

### A.2. Titulaciones

Cada una de las escuelas poseen ciertas titulaciones que se han de añadir a la configuración.

#### A.2.1. E.T.S. de Ingeniería Informática

La ETSISI tiene configurados como colores principal y de link el RGB (0, 177, 230). El logo es el mostrado la figura [A.1](#).



**Figura A.1.** Logo de la ETSI utilizado en la cubierta trasera de la memoria

Las titulaciones que existen la última vez que se actualizó este documento son las siguientes:

- 61CD: Grado en Ciencia de Datos e Inteligencia Artificial.
- 61CI: Grado en Ingeniería de Computadores.
- 61IW: Grado en Ingeniería del Software.
- 61SI: Grado en Sistemas de Información.
- 61TI: Grado en Tecnologías para la Sociedad de la Información.



## **B.**      ¿Cómo ampliar la plantilla?

---

TBD

## **C. Lista de paquetes incluidos**

---

TBD

# Bibliografía

---

- [1] W. S. McCulloch y W. Pitts, «A logical calculus of the ideas immanent in nervous activity,» *The bulletin of mathematical biophysics*, vol. 5, n.º 4, págs. 115-133, 1943.

# Índice de términos

---

## Glosario

**Arco de la Perdición** Arco único que posee poderes destructivos capaz de desatar grandes catástrofes y de traer desgracias a aquellos que se encuentren en su camino. [22](#)

**hacha de batalla** Herramienta antigua utilizada en combate, caracterizada por su doble función de arma y herramienta. [22](#), [23](#)

**Python** El mejor lenguaje de programación. [23](#)

## Siglas

**ONG** organización no gubernamental. [2](#)

**PFG** Proyecto a realizar al final de una titulación de Grado. [1](#), [2](#), [4](#), [6](#), [9](#), [10](#), [24](#)

**PFM** Proyecto a realizar al final de una titulación de Máster. [1](#), [4](#), [6](#), [9](#), [10](#)

**RPG** Role-Playing Game. Juego de rol. [24](#)

**SPECIAL** S.P.E.C.I.A.L es la sigla usada para los atributos de Fuerza (**S**trenght), Percepción (**P**erception), Resistencia (**E**ndurance), Carisma, (**C**harisma), Inteligencia, (**I**ntelligence), Agilidad, (**A**gility), y Suerte (**L**uck). [24](#)



Universidad  
Rey Juan Carlos

Escuela Técnica Superior  
Ingeniería Informática