

UNIVERSIDAD POLITÉCNICA DE MADRID

E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

PROYECTO FIN DE GRADO

GRADO EN INGENIERÍA DE COMPUTADORES

Plantilla de proyecto de fin de estudios para la UPM

Desarrollado por: Santa Tecla y San Blador

Dirigido por: San Bláster y Panflerto Díaz

Madrid, 6 de octubre de 2025



Plantilla de proyecto de fin de estudios para la UPM

Desarrollado por: Santa Tecla y San Blador

Dirigido por: San Bláster y Panflerto Díaz

Proyecto Fin de Grado, 6 de octubre de 2025

E.T.S. de Ingeniería de Sistemas Informáticos

Campus Sur UPM, Carretera de Valencia (A-3), km. 7

28031, Madrid, España

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

```
@mastersthesis{2025SantaTeclaySanBlador,  
  title = {Plantilla de proyecto de fin de estudios para la UPM},  
  type = {Bachelor's Thesis},  
  author = {Tecla, S. y Blador, S.},  
  school = {E.T.S. de Ingeniería de Sistemas Informáticos},  
  year = {2025},  
  month = {10},  
}
```

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/blazaid/UPM-Report-Template>.



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

Oigo y olvido, veo y recuerdo, hago y aprendo.

— Confucio

Agradecimientos

Agradecimientos que se quieran dar. Si no se quiere añadir agradecimientos, es tan sencillo como eliminar la entrada `\acknowledgements` del fichero `report.tex`.

Resumen

El resumen de un [proyecto de fin de grado](#) o de un [proyecto de fin de máster](#) condensa en tres o cuatro párrafos el contenido de la memoria. Se debería asumir que el lector tiene una cierta idea de lo que se trata y también que si no le enganchamos, probablemente pase de él (eso es malo).

Condensado no quiere decir incompleto. Debe contener la información más destacable. Lo ideal: que ocupe entre media y una cara de un folio A4. Comenzará por el propósito y principales objetivos de la memoria, seguirá con los aspectos más destacables de la metodología empleada, y luego de los resultados obtenidos. Por último se presentarán la o las conclusiones más importantes a las que se ha llegado.

Que tenga un estilo claro y conciso, sin ambigüedades de ningún tipo. Además, al ser un resumen de todo el contenido, ni que decir tiene que deberá ser lo último que escribamos, y deberá mantener una absoluta fidelidad con el contenido de la memoria.

Palabras clave: Cuatro o cinco; Expresiones clave; Que clasifiquen; Nuestro proyecto o; Investigación

Abstract

This section must contain the summary that we have written before in Spanish, but in English, as well as the keywords.

Keywords: Four or five; Key Expressions; Summarising; Our Project or; Research

Índice general

1	Introducción	1
1.1	Objetivos	1
1.2	Motivación	1
1.3	Justificación	2
1.4	Estructura de la memoria	2
2	¿Cómo estructurar la memoria?	3
3	Escribiendo la memoria	5
3.1	Primeros pasos	6
3.2	Empezando con la memoria en sí	7
3.3	El caso concreto de los PFM	8
3.4	Reglas y recomendaciones	8
4	Componentes de la plantilla	9
4.1	Columnas	9
4.2	Ecuaciones	9
4.3	Elementos flotantes	11
4.4	Enlaces de hipertexto	18
4.5	Fórmulas matemáticas	18

4.6	Glosario	18
4.7	Notas	22
4.8	Referencias cruzadas	22
4.9	Referencias bibliográficas	23
4.10	Referencias cruzadas	25
4.11	Referencias a recursos externos	25
5	Licencia	26
A	¿Cómo ampliar la plantilla?	27
B	Escuelas y títulos	28
B.1	Escuelas	28
B.2	Titulaciones	28
	Glosario	32
	Siglas	32

Índice de figuras

4.1	Vault Boy approves that	16
4.2	Todos los Vault Boy	18

Índice de tablas

4.1	Opciones para los elementos flotantes de \LaTeX	12
4.2	Comandos para incluir términos del glosario en el texto de la memoria .	20
4.3	Comandos específicos para controlar la presentación de acrónimos . .	21
B.1	Relación entre el código de la plantilla y la escuela a la que se refiere . .	28

Índice de listados

3.1	Primeras líneas del fichero <code>report.tex</code>	5
3.2	Configurando autor, título del proyecto y director	5
3.3	Declarando el resumen en español	6
3.4	La lista de palabras clave en español	6
3.5	Incluyendo ficheros externos en el documento	7
4.1	Ejemplo de inserción de fórmulas en línea	9
	<code>sources/adding-blocks.tex</code>	12
	<code>sources/adding-blocks.tex</code>	13
	<code>sources/adding-blocks.tex</code>	13
	<code>sources/adding-blocks.tex</code>	13
	<code>sources/snippets.py</code>	14
	<code>sources/snippets.py</code>	14
	<code>sources/adding-blocks.tex</code>	15
4.2	Función para determinar cuando una palabra <code>w1</code> es anagrama de otra palabra <code>w2</code>	15
4.3	Inserción de una figura	15
4.4	Inserción de varias subfiguras	17
4.5	Código para crear una entrada en el glosario	19
4.6	Especificando el plural para un término del glosario	20

4.7	Entrada genérica de una sigla o acrónimo en el glosario	20
4.8	Entrada de rpg en glossaries.tex	21
4.9	Referenciando una figura y su página	22
4.10	Estructura general de una referencia	24
4.11	Referenciando una entrada bibliográfica	25

1.

Introducción

La introducción a un trabajo de fin de estudios es el punto de entrada a todo el trabajo realizado y es considerada la más importante tras el resumen, donde se resume el trabajo entero. Hay que dejar claro **qué** se ha hecho y el **porqué** de su importancia. Intenta generar expectativa. Un gancho típico suele ser discutir sobre un dato relevante o controvertido o plantear una pregunta relevante en lo que se está trabajando.

1.1. Objetivos

Esta sección es **obligatoria**, y determinas la **finalidad** del proyecto. Suele encajar en una de las siguientes categorías:

- **Contraste** o validación de una hipótesis, típico de en [PFM](#),
- **Desarrollo** o diseño de algo (*hardware*, sistemas, edificios), común en ingenierías,
- **Estudio** de un tema que deduce o descubre nuevo conocimiento, típico en ciencias puras y humanidades.

Nos vienen muy bien para evaluar el éxito del proyecto, porque así en las conclusiones podemos indicar qué objetivos se han logrado y cuáles no, y el porqué. Para poder decir si un objetivo se ha cumplido ha de ser (i) **acotado en el tiempo** —así se puede planificar—, (ii) **medible** —podemos cuantificar su éxito—, (iii) **específico** —evitando así tareas irrelevantes—, (iv) **Alcanzable** —asegurando su finalización— y (v) **relevante** —es decir, relacionado con el campo de estudio—.

1.2. Motivación

¿Qué factores han hecho que te decantes por este tema? No se refiere a hablar de que «es algo que nos ha apasionado desde niños», o que «queremos ampliar nuestro conoci-

miento». Es más bien buscar la motivación en el impacto que este trabajo puede tener en nuestro entorno (medioambiental, social, tecnológico, ...).

Algunas fuentes donde encontrar este tipo de motivaciones pueden ser las revistas especializadas, periódicos, organismos de estandarización, [ONG](#), etcétera.

1.3. Justificación

Esta sección no es obligatoria, pero puede ser interesante. Es donde se explican las razones por las que se ha elegido el tema, así como su relevancia. A veces se junta con la sección anterior. Algunos elementos clave a abordar aquí pueden ser:

1. **Relevancia del tema:** ¿Existe alguna necesidad o problema específico que tu proyecto pueda abordar?
2. **Justificación teórica:** Mención sobre qué teorías, enfoques o modelos existentes en la literatura respalden la importancia de abordar este tema.
3. **Brecha en el conocimiento:** ¿Qué aspectos no se han explorado lo suficiente o no han sido abordados en estudios previos? ¿Cómo puede el proyecto contribuir a cerrar esa brecha en el conocimiento?
4. **Contribución práctica:** Aplicaciones del proyecto y cómo pueden beneficiar a la comunidad académica, profesional o a la sociedad en general.

Si se escribe, debe ser lo suficientemente clara y convincente para que los lectores comprendan por qué el proyecto es relevante y necesario.

1.4. Estructura de la memoria

Esta sección permite al lector comprender la secuencia lógica de cómo se ha desarrollado el proyecto o la investigación. Presentará un resumen de los diferentes capítulos que conforman la memoria (a poder ser en un único párrafo, como mucho dos).

2. ¿Cómo estructurar la memoria?

Un PFG tiene como propósito demostrar que se poseen las destrezas y los conocimientos que se esperan de un egresado de la titulación cursada. Esto quiere decir que, a diferencia de otros tipos de trabajo académico, en éste no es necesario realizar aportaciones originales al estado de la cuestión sobre el que se trabaja.

¿Y sobre la estructura? «Como buenamente se quiera/pueda». Dependerá del tipo de trabajo, pero con carácter general, los trabajos suelen seguir cierta estructura. La típica suele ser la siguiente:

1. Una **portada** que muestre lo requerido por la normativa de la escuela.
2. Un **resumen** y un **abstract** resumiendo toda la memoria (el primero en español, el segundo en inglés).
3. Índice de contenidos, de figuras y de tablas, que apuntan a los diferentes elementos de interés de la memoria.
4. La **introducción**, que explica qué se va a hacer y por qué.
5. El **estado de la cuestión**¹, donde se dice qué es lo que existe sobre lo que se sustenta nuestro proyecto.
6. La **metodología** que describe paso por paso el proceso de cómo se ha realizado el proyecto.
7. Los **resultados** obtenidos tras finalizar el proyecto junto con una **discusión** (en el mismo capítulo o en uno nuevo) donde se explican las implicaciones que el autor de la memoria considere oportuno destacar sobre los mismos.

¹Se suele decir «estado del arte», pero se desaconseja su uso por ser un anglicismo derivado de la expresión *state of the art* en favor expresiones como «estado de la cuestión», cosa que, por otro lado, suena bastante bien. Más información en <https://www.rae.es/dpd/arte>.

8. Las **conclusiones** del proyecto, donde el autor indica el nivel de consecución de los objetivos, se tira a la piscina a la hora de dar respuestas a cuestiones relacionadas con los resultados, propone trabajos futuros **de interés o utilidad** (no chorradas) y poco más.
9. Un apéndice describiendo el impacto social y medioambiental del proyecto.
10. Más apéndices que el autor considere necesarios.
11. Referencias bibliográficas.
12. Glosario, abreviaturas y demás.

La plantilla genera automáticamente la portada de acuerdo a los que dicta la normativa de la escuela, los índices de contenidos, figuras y tablas, las referencias bibliográficas y los apéndices (con ayuda, por supuesto, pero se dedican del mercado). El resto lo tiene que poner el autor; aun así no te preocupes que es más fácil de lo que parece.

3. Escribiendo la memoria

¿Cómo comenzamos a escribir la memoria? Con cuidado, y por el principio. Y el principio es el fichero `report.tex`, por lo que vamos a ver las primeras líneas. Por el principio, y con cuidado. Esto es, con el fichero `report.tex`. Veamos la primera línea del fichero (listado 3.1).

Listado 3.1. Primeras líneas del fichero `report.tex`

```
\documentclass[%  
  school=etsisi,%  
  type=pfg,%  
  degree=61CI,%  
{upm-report}
```

Estos valores configuran gran parte de cómo se comportará el renderizado de la plantilla. Los parámetros y sus opciones son los siguientes:

- **school:** La escuela a la que pertenece el estudiante. Determinará, entre otras cosas, direcciones, formato de portada y colores principales. Las opciones se describen en el [Apéndice B](#).
- **type:** El tipo de memoria. Modifica algunos textos, incluida la portada. Puede tomar los valores `pfg` ([proyecto de fin de grado](#)) y `pfm` ([proyecto de fin de máster](#)).
- **degree:** El código del grado. Al igual que con las escuelas, los códigos de los grados soportados se encuentran en el [Apéndice B](#).

Las siguientes líneas (listado 3.2) representan el título del proyecto, el autor o autores más sus entradas bibliográficas, y el director o directores más sus entradas bibliográficas.

Listado 3.2. Configurando autor, título del proyecto y director

```
\title{Plantilla de proyecto de fin de estudios para la UPM}  
\author{Santa Tecla \and San Blador}  
\bibauthor{Tecla, S. \and Blador, S.}  
\director{San Bláster \and Panflerto Díaz}
```

Eso de las entradas bibliográficas para autores y directores no son más que los nombres que queremos que aparezcan en la segunda página, donde se indica cómo citar el proyecto. El día que quien está escribiendo esto aprenda cómo hacerlo automáticamente, este párrafo desaparecerá como lágrimas en la lluvia¹.

3.1. Primeros pasos

Ahora empieza el primer contenido de verdad: el **resumen** y el **abstract**. Ambos dos son obligatorios y se añaden con la macro `\abstract`, donde se especificarán el idioma (`spanish` o `english`) y el contenido. En el [Listado 3.3](#) se muestran las primeras líneas donde se declara el resumen de este documento.

Listado 3.3. Declarando el resumen en español

```
El resumen de un \acrlong{pfg} o de un \acrlong{pfm}  
condensa en  
tres o cuatro párrafos el contenido de la memoria. Se  
debería asumir  
que el lector tiene una cierta idea de lo que se trata y  
también que  
si no le enganamos, probablemente pase de él (eso es malo  
).
```

De la misma forma, las palabras clave se añaden con la macro `\keywords`. Son obligatorios, así que no te los saltes.

Listado 3.4. La lista de palabras clave en español

¹Bueno, y si tú, queridísima lectora o lector, si sabes cómo hacerlo, haz un *pull request* al repositorio de la plantilla: <https://github.com/blazaid/UPM-Report-Template>.

```
Cuatro o cinco;  
Expresiones clave;  
Que clasifiquen;  
Nuestro proyecto o;  
Investigación  
}
```

También existe la opción de añadir agradecimientos con `\acknowledgements`. Si no se pone no se muestra 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.

3.2. Empezando con la memoria en sí

Y ahora sí, empezamos con la memoria del proyecto en sí. Habrá que ir escribiendo un capítulo tras otro. En este ejemplo está todo escrito en el mismo fichero (`report.tex`), pero es muy común escribirlo en varios ficheros diferentes e ir incluyéndolos mediante la macro `\include`, tal y como se muestra en el [Listado 3.5](#).

Listado 3.5. Incluyendo ficheros externos en el documento

```
\include{introduction.tex}  
\include{state-of-the-art.tex}  
\include{methodology.tex}  
\include{results.tex}  
\include{conclusions.tex}
```

Una vez hemos terminado de escribir los capítulos principales, la macro `\appendix` indica desde qué punto empiezan los apéndices. No son obligatorios, ni mucho menos, pero en algunos [PFG](#) y [PFM](#) los incluyen para dar información adicional que no se considera clave en el contenido de la memoria, pero que es interesante para complementarla. 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.

3.3. El caso concreto de los PFM

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.

3.4. Reglas y recomendaciones

TODO: Explicar el el PFG es trabajo del estudiante, el profesor realiza la supervisión. Es trabajo autónomo del alumno y debe llegar a un resultado.

TODO: Prevenir contra el uso de IA generativa al tuntún

4. Componentes de la plantilla

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

4.1. Columnas

TBD

4.2. Ecuaciones

La facilidad de composición de ecuaciones es una de las cosas que más atrae de \LaTeX a muchos autores. \LaTeX mantiene dos renderizadores diferentes, uno para el texto y otro para las ecuaciones, denominados modo párrafo y modo matemático¹. 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.

4.2.1. 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 4.1.

Listado 4.1. Ejemplo de inserción de fórmulas en línea

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

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

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 $a > 0$, $a^p \equiv a \pmod{p}$

4.2.2. 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:

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

$$\int_a^b f'(x) \, dx = f(b) - f(a) \quad (4.1)$$

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:

```
\begin{align}
  u &= \arctan x \\
  du &= \frac{1}{1+x^2} dx
\end{align}
```

$$u = \arctan x \quad (4.2)$$

$$du = \frac{1}{1+x^2} dx \quad (4.3)$$

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*}
```

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

En caso de querer generar un índice de ecuaciones del documento, de la misma manera que los listados de Figuras, Tablas y Listados. Para ello, debemos añadir a la lista de ecuaciones la ecuación definida con el comando `myequations`. Es importante destacar que para generar un listado de ecuaciones estas deben estar numeradas, es decir.

```

\begin{equation}
  \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1
\end{equation}
\myequations{Ecuación canónica
  de la hipérbola}

```

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad (4.4)$$

4.3. 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 \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 4.8](#) de este capítulo), y después dejar que \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 [4.1](#):

Tabla 4.1. Opciones para los elementos flotantes de \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 \LaTeX

4.3.1. Código fuente

Para la gestión de los listados de código fuente se utiliza el paquete `listings`. El estilo usado es una modificación² de [Solarized](#) desarrollado por Ethan Schoonover.

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}
```

²En realidad la modificación es cambiar el fondo de crema a blanco

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 \LaTeX ³ 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

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

tamaño de la memoria manejable.

Suponiendo que tenemos el fichero `sources/snippets.py`, para incluirlo entero 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 deseara 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 respectivamente.

Etiquetando bloques de código

Al igual que con el resto de bloques *float* (e.g. figuras o tablas), se pueden⁴ (**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 \textt{w1} es anagrama de otra
    palabra \textt{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 4.2. 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)
```

4.3.2. Figuras

El código siguiente (listado 4.3) renderizará una imagen.

Listado 4.3. Inserción de una figura

⁴Pongo *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).



Figura 4.1. Vault Boy approves that

```
\begin{figure}
  \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 4.1) 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.

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 pixele 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 4.4 vemos un ejemplo de cómo se manejan.

Listado 4.4. Inserción de varias subfiguras

```

\begin{figure}
  \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 4.2. Por cierto, también podemos referenciar a los pies de las subfiguras (e.g. Así: 4.2a).

4.3.3. Tablas

Las tablas (en realidad «cuadros», pero bueno) 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).

Sin embargo, las tablas suelen ser bastante complicadas en \LaTeX , por lo menos para la gente que empieza. Para no escribir demasiado, la respuesta para casi toda maqueta-

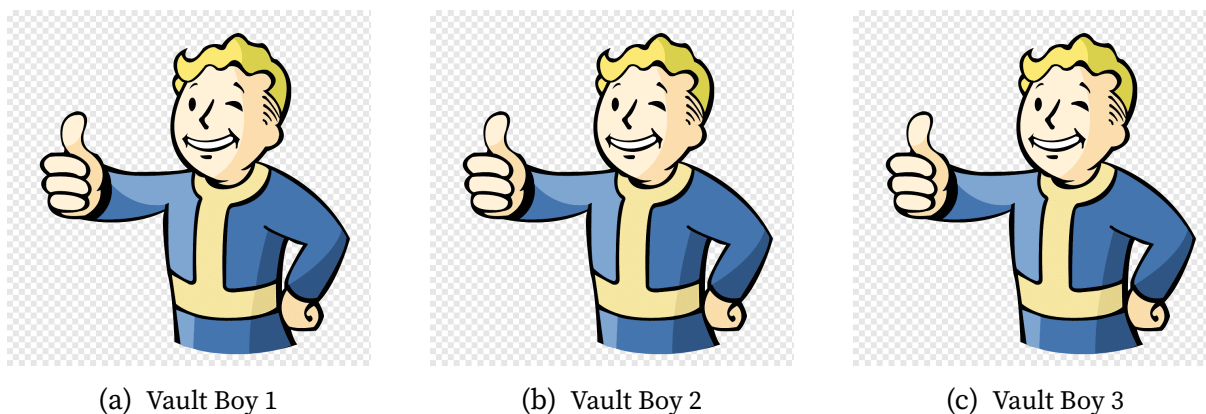


Figura 4.2. Todos los Vault Boy

ción 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.

4.4. Enlaces de hipertexto

TBD

4.5. Fórmulas matemáticas

TBD

4.6. 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 `glossar` es el cual es relativamente complejo de configurar. También su documentación es muy extensa⁵, así que en esta

⁵Pero 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>.

sección hablaremos únicamente de lo esencial.

4.6.1. ¿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.

4.6.2. 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 4.5.

Listado 4.5. 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 4.2.

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 4.6.

Tabla 4.2. Comandos para incluir términos del glosario en el texto de la memoria

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

Listado 4.6. 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 aquí⁶ 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 4.7.

Listado 4.7. 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},
  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

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

Tabla 4.3. Comandos específicos para controlar la presentación de acrónimos

Comando	Ejemplo con la clave rpg
<code>\acrshort</code>	RPG
<code>\acrshortpl</code>	RPG
<code>\acrlong</code>	juego de rol
<code>\Acrlong</code>	Juego de rol
<code>\acrlongpl</code>	juego de rol
<code>\Acrlongpl</code>	Juego de rol
<code>\acrfull</code>	juego de rol (RPG)
<code>\Acrfull</code>	Juego de rol (RPG)
<code>\acrfullpl</code>	juego de rol (RPG)
<code>\Acrfullpl</code>	Juego de rol (RPG)

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 de 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 **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 [4.3](#)

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 [4.8](#).

Listado 4.8. Entrada de `rpg` en `glossaries.tex`

```
\newacronym[
```

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

4.7. Notas

TBD

4.8. 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 [...]”⁷.

Una de las propiedades más útiles y, en ocasiones, infravaloradas de \LaTeX es la facilidad y potencia de su sistema de etiquetado. Este sistema permite referenciar tablas, listados de código fuente, ecuaciones, capítulos, secciones, etc., con facilidad y flexibilidad. Además, \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 4.3. 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 4.9

Listado 4.9. Referenciando una figura y su página

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

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 4.1 titulada [Vault Boy approves that](#) en la página 16.»



¿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 \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 re-implementa. De todas formas, si te animas, ahí tienes un buen proyecto que si lo sacas adelante te va a hacer muy famoso.

4.9. Referencias bibliográficas

Hay muchas formas diferentes hacerlo, así que en esta plantilla 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 (o no) durante la memoria. Luego, en el cuerpo del texto, se podrán hacer referencias a estas. Será \LaTeX después quien se encargue de indexar correctamente, crear los hipervínculos y maquetar automáticamente.

Las referencias que no aparezcan citadas en el texto no aparecerán en el capítulo de referencias, cosa que tiene sentido porque si no se habla de ellas, ¿para qué las vamos a nombrar?



No has dicho en ningún momento *bibliografía* Ya. 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.

4.9.1. ¿Cómo creamos nuevas referencias?

El fichero `references.bib` contará cero o más entradas con la estructura mostrada en el listado 4.10.

Listado 4.10. 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`.

4.9.2. ¿De qué manera puedo citar las referencias?

Pues básicamente hay dos estilos: sin paréntesis, donde el autor y el año aparecen para dar un estilo narrativo, y con paréntesis, donde aparecen entre paréntesis, dando a entender que no forman parte de la narrativa. Se usan tal y como se muestra en el

listado 4.11.

Listado 4.11. Referenciando una entrada bibliográfica

```
Tal y como describen \cite{mcculloch1943logical}, las neuronas  
artificiales son la caña \parencite{mcculloch1943logical}.
```

Algo que queda como sigue:

«Tal y como se describen [1], las neuronas artificiales son la caña [1].»

4.9.3. Más información

Tienes muchas más información acerca de las referencias bibliográficas en [BibTeX.eu](https://bibtex.eu)⁸.

4.10. Referencias cruzadas

TBD

4.11. Referencias a recursos externos

TBD

⁸Url: <https://bibtex.eu/es/biblatex/>. Último acceso el 18 de julio de 2024.

5.

Licencia

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. ¿Cómo ampliar la plantilla?

TBD

B. Escuelas y títulos

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

B.1. Escuelas

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

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 [A](#)) y también contactar con los autores, así lo podemos hacer público con la mayor cantidad de usuarios posible.

B.2. Titulaciones

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

B.2.1. E.T.S. de Ingeniería Agronómica, Alimentaria y de Biosistemas

La ETSIAAB tiene configurados como colores principal y de link el RGB (99, 178, 76).

Clave	Valor
etsiaab	E.T.S. de Ingeniería Agronómica, Alimentaria y de Biosistemas
etsidi	E.T.S. de Ingeniería y Diseño Industrial
etsisi	E.T.S. de Ingeniería de Sistemas Informáticos

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

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

- 20BT: Grado en Biotecnología,
- 20BI: Grado en Ciencias Agrarias y Bioeconomía,
- 20IG: Grado en Ingeniería Agrícola,
- 02IA: Grado en Ingeniería Agroambiental,
- 20IA: Grado en Ingeniería Alimentaria.

B.2.2. E.T.S. de Ingeniería y Diseño Industrial

La ETSIDI tiene configurados como colores principal y de link el RGB (223, 30, 64).

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

- 56IE: Grado en Ingeniería Eléctrica,
- 56IA: Grado en Ingeniería Electrónica Industrial y Automática,
- 56IM: Grado en Ingeniería Mecánica,
- 56IQ: Grado en Grado en Ingeniería Química,
- 56DD: Grado en Ingeniería en Diseño Industrial y Desarrollo de Producto.

B.2.3. E.T.S. de Ingeniería de Sistemas Informáticos

La ETSISI tiene configurados como colores principal y de link el RGB (0, 114, 206).

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.

Referencias

- [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 [19](#)

hacha de batalla Herramienta antigua utilizada en combate, caracterizada por su doble función de arma y herramienta [19, 20](#)

Python El mejor lenguaje de programación [20](#)

Siglas

ONG organización no gubernamental [2](#)

PFG proyecto de fin de grado [3, 5, 7, 8, 21](#)

PFM proyecto de fin de máster [1, 5, 7, 8](#)

RPG Del inglés *role-playing game* [21](#)

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) [21](#)

