

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

Autora: Autora Proyecto

Directora: Directora Proyecto

Madrid, 13 de julio de 2023



Plantilla de proyecto de fin de estudios para la UPM
Proyecto Fin de Grado, 13 de julio de 2023

Autora: Autora Proyecto

Directora: Directora Proyecto

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 BibTeX es la siguiente:

```
@mastersthesis{citekey,  
title = {Plantilla de proyecto de fin de estudios para la UPM},  
author = {Proyecto, A. \& Proyecto, D.},  
school = {E.T.S. de Ingeniería de Sistemas Informáticos},  
year = {2023},  
month = {7},  
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/blazaid/UPM-Report-Template>.



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

El resumen de un [Proyecto Fin de Grado](#) o de un [Proyecto Fin de Máster](#) condensa en tres o cuatro párrafos el contenido de la memoria. Se debe dar por sentado que el lector podrá tener una idea clara de lo que trata, y suele ser la primera barrera donde decide si continúa leyendo o no el texto.

Condensado no quiere decir incompleto. Debe contener la información más destacable. Lo ideal es que ocupe entre media y una cara de un folio A4. Comenzará por el propósito y principales objetivos de la memoria. Luego hablaremos sobre los aspectos más destacables de la metodología empleada, seguido de los resultados obtenidos. Por último se presentarán las conclusiones de forma condensada.

Debe tener 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 elaboraremos, y deberá mantener una absoluta fidelidad con el contenido de la memoria.

Palabras clave: Cuatro o cinco; Expresiones clave; Que resuman; 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	2
1.3	Justificación	3
1.4	Estructura de la memoria	3
2	Configuración de la memoria	4
2.1	¿Cómo empiezo a escribir la memoria?	4
2.2	¿Cómo estructurar la memoria?	8
3	Componentes de la plantilla	10
3.1	Columnas	10
3.2	Ecuaciones	10
3.3	Elementos flotantes	12
3.4	Enlaces de hipertexto	20
3.5	Fórmulas matemáticas	20
3.6	Glosario	20
3.7	Notas	24
3.8	Referencias cruzadas	24

3.9	Referencias bibliográficas	25
3.10	Referencias cruzadas	27
3.11	Referencias a recursos externos	27
4	Licencia	28
A	Escuelas y títulos	29
A.1	Escuelas	29
A.2	Titulaciones	29
B	¿Cómo ampliar la plantilla?	32
C	Lista de paquetes incluidos	33

Índice de figuras

3.1	Vault Boy approves that	17
3.2	Todos los Vault Boy	19
A.1	Logo de la ETSIDI utilizado en la cubierta trasera de la memoria .	30
A.2	Logo de la ETSISI utilizado en la cubierta trasera de la memoria .	30

Índice de tablas

3.1	Opciones para los elementos flotantes de \LaTeX	13
3.2	Comandos para incluir términos del glosario en el texto de la memoria	21
3.3	Comandos específicos para controlar la presentación de acrónimos	23
A.1	Relación entre el código de la plantilla y la escuela a la que se refiere	29

Índice de listados

2.1	Primeras líneas del fichero <code>report.tex</code>	5
2.2	Inclusión del fichero de referencias bibliográficas <code>references.bib</code>	5
2.3	Configurando autor, título del proyecto y director	6
2.4	Cómo se incluyen los capítulos y los apéndices	7
3.1	Ejemplo de inserción de fórmulas en línea	10
	<code>sources/adding-blocks.tex</code>	13
	<code>sources/adding-blocks.tex</code>	13
	<code>sources/adding-blocks.tex</code>	14
	<code>sources/adding-blocks.tex</code>	14
	<code>sources/snippets.py</code>	15
	<code>sources/snippets.py</code>	15
	<code>sources/adding-blocks.tex</code>	16
3.2	Función para determinar cuando una palabra w_1 es anagrama de otra palabra w_2	16
3.3	Inserción de una figura	17
3.4	Inserción de varias subfiguras	18
3.5	Código para crear una entrada en el glosario	21
3.6	Especificando el plural para un término del glosario	21
3.7	Entrada genérica de una sigla o acrónimo en el glosario	22

3.8	Entrada de <code>rpg</code> en <code>glossaries.tex</code>	23
3.9	Referenciando una figura y su página	24
3.10	Estructura general de una referencia	26

1.

Introducción

La introducción a un [Proyecto Fin de Grado \(PFG\)](#) o a un [Proyecto Fin de Máster \(PFM\)](#) 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. Aquí hay que dejar claro **qué** trabajo se ha realizado y el **porqué** de su importancia. Se deben generar expectativas. Un gancho típico en los trabajos suele ser el de aportar un dato relevante o controvertido para discutir sobre él o plantear una pregunta relevante para el contexto en el que se está trabajando.

Dentro del capítulo, tras introducir el trabajo realizado, se suelen incluir las siguientes secciones para establecer bien su alcance y limitaciones: **motivación, objetivos, suposiciones/limitaciones** y, a veces, **estructura de la memoria**. Ni que decir tiene que esta estructura planteada, tanto del capítulo como de la memoria en si es únicamente un ejemplo o propuesta. Cada proyecto es único y a veces es más cómodo escribirlo de otro modo.

1.1. Objetivos

Una de las partes más importante y complicada. Se considera **la finalidad** del proyecto en cuestión a realizar y suele encajar dentro de una de las siguientes categorías:

- **Contraste** o validación de una hipótesis. Suele usarse en [PFMs](#), no tanto en [PFGs](#).
- **Desarrollo** o diseño de algo (e.g. Software, hardware, sistema, edificio). Suele ser el más común en las ingenierías.
- **Estudio** de un tema que deduce o descubre nuevo conocimiento. Suele ser más común en las ramas de las ciencias puras y humanidades.

Sirve como primer indicador de la consecución del proyecto, ya que planteando objetivos podemos determinar en las conclusiones su grado de cumplimiento. Ahora bien, ¿cómo determinamos que un objetivo se ha cumplido? pues definiéndolo para que se pueda cumplir, es decir, intentando que sea:

- **Acotado en el tiempo**, así es más fácil establecer un marco temporal para su realización y programar temporalmente las partes de las que se compone.
- **Medible**, para saber cómo de lejos estamos de llegar a un resultado aceptable.
- **Específico**, de manera que esté bien acotado y sea difícil embarcarse en tareas que no nos acerquen a su consecución.
- **Alcanzable**, porque si no lo es, por mucha intención y esfuerzo que le pongamos no se va a terminar.
- **Relevante**, porque si, en un PFG para Ingeniería del Software, desarrollamos un producto mecánico para sexar pollos, pues por muy importante que sea, poco tiene que ver con lo que se ha estudiado durante todos estos años.

Regla mnemotécnica: *AMEAR*.

1.2. Motivación

Se refiere a los factores que han hecho que el estudiante se decante por trabajar en éste tema.

Lo suyo sería apoyarse en buscar motivaciones más allá de las expresiones tipo «ampliar mis conocimientos». Algunas fuentes donde encontrarla son las revistas especializadas, periódicos, organismos de estandarización, [organizaciones no gubernamentales \(ONG\)](#), etcétera.

1.3. Justificación

En esta sección se deben explicar y argumentar las razones por las cuales se eligió el tema del proyecto, así como su importancia y relevancia. Algunos elementos clave que se pueden abordar en esta sección son:

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.

La sección no tiene por qué ser demasiado extensa, ni tiene por qué incluir (o limitarse) a los puntos anteriores, pero 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

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.

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. 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.

2.1. ¿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 2.1. Primeras líneas del fichero `report.tex`

```
\documentclass[%  
  school=etsisi,%  
  type=pfg,%  
  degree=61CI,%  
  authorsex=f,%  
  directorsex=f,%  
]{upm-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 2.2. 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 2.3. Configurando autor, título del proyecto y director

```
\title{Plantilla de proyecto de fin de estudios para la UPM}  
\author{Autora Proyecto}  
\bibauthor{Proyecto, A.}  
\director{Directora 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 desaparezca¹.

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

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

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 2.4.

Listado 2.4. Cómo se incluyen los capítulos y los apéndices

```
\input{chapters/introduccion}
\input{chapters/configuracion}
\input{chapters/componentes}
\input{chapters/licencia}

\appendix

\input{appendices/escuelas-y-titulos}
\input{appendices/ampliar}
\input{appendices/paquetes}

\end{document}
```

La macro `\appendix` del medio indica a partir de qué punto se añaden los apéndices. No son obligatorios, ni mucho menos, pero en algunos PFGs y 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.

2.2. ¿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 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.

3. Componentes de la plantilla

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

3.1. Columnas

TBD

3.2. 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.

3.2.1. 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 3.1. Ejemplo de inserción de fórmulas en línea

¹Existe 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 $a > 0$, $a^p \equiv a \pmod{p}$

3.2.2. 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}
```

$$\int_a^b f'(x) \, dx = f(b) - f(a) \quad (3.1)$$

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}
```

$$u = \arctan x \quad (3.2)$$

$$du = \frac{1}{1 + x^2} dx \quad (3.3)$$

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*}
```

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

3.3. 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 3.8](#) 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 [3.1](#):

3.3.1. 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.

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

Tabla 3.1. 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

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 \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 tamaño de la memoria manejable.

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

³En 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 desee 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 \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 3.2. 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)
```

3.3.2. Figuras

El código siguiente (listado 3.3) renderizará una imagen.

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).

Listado 3.3. 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 3.1. 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 3.4. 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).

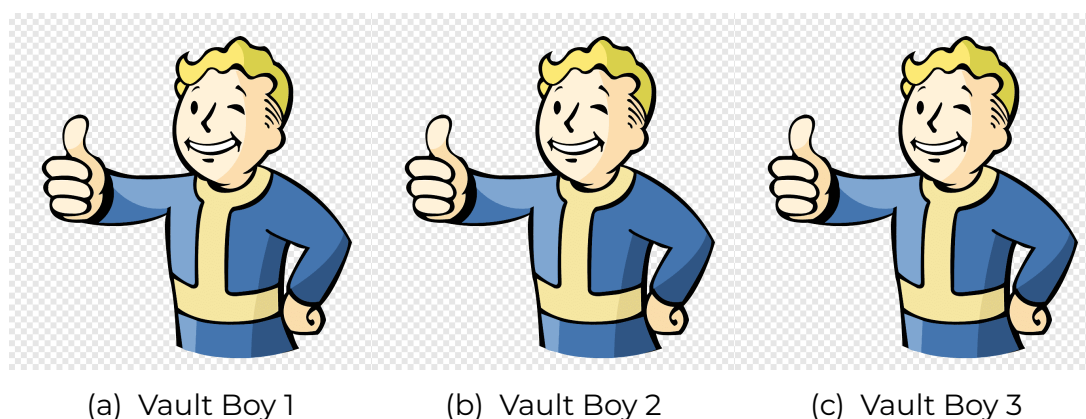


Figura 3.2. Todos los Vault Boy

3.3.3. 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 \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.

3.4. Enlaces de hipertexto

TBD

3.5. Fórmulas matemáticas

TBD

3.6. 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 extensa⁵, así que en esta sección hablaremos únicamente de lo esencial.

3.6.1. ¿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 cas-

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>.

tellano (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.

3.6.2. 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 3.5. 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.

Tabla 3.2. Comandos para incluir términos del glosario en el texto de la memoria

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

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 3.6. 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 3.7.

Listado 3.7. 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 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

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

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 *Strength, 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 3.3. Comandos específicos para controlar la presentación de acrónimos

Comando	Ejemplo con la clave <code>rpg</code>
<code>\acrshort</code>	<code>RPG</code>
<code>\acrshortpl</code>	<code>RPG</code>
<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 3.8. Entrada de `rpg` en `glossaries.tex`

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

3.7. Notas

TBD

3.8. 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 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 3.9. 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 3.1 titulada [Vault Boy approves that](#) en la página 17.»



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

3.9. 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 \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.

3.9.1. ¿Cómo creamos nuevas referencias?

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

Listado 3.10. 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`.

3.9.2. ¿De qué manera puedo citar las referencias?

TBD [1]

3.10. Referencias cruzadas

TBD

3.11. Referencias a recursos externos

TBD

4.

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.

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
etsidi	E.T.S. de Ingeniería y Diseño Industrial
etsisi	E.T.S. de Ingeniería de Sistemas Informáticos

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 y Diseño Industrial

La ETSIDI tiene configurados como colores principal y de link el RGB (223, 30, 64). El logo es el mostrado la figura A.1.



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

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.

A.2.2. E.T.S. de Ingeniería de Sistemas Informáticos

La ETSISI tiene configurados como colores principal y de link el RGB (0, 177, 230). El logo es el mostrado la figura A.2.



Figura A.2. Logo de la ETSISI 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. [21](#)

hacha de batalla Herramienta antigua utilizada en combate, caracterizada por su doble función de arma y herramienta. [21](#)

Python El mejor lenguaje de programación. [22](#)

Siglas

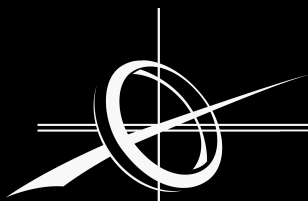
ONG organización no gubernamental. [2](#)

PFG Proyecto a realizar al final de una titulación de Grado. [1](#), [2](#), [4](#), [5](#), [7](#), [8](#), [22](#)

PFM Proyecto a realizar al final de una titulación de Máster. [1](#), [4](#), [5](#), [7](#), [8](#)

RPG Role-Playing Game. Juego de rol. [23](#)

SPECIAL S.P.E.C.I.A.L es la sigla usada para los atributos de Fuerza (**S**trengh), Percepción (**P**erception), Resistencia (**E**ndurance), Carisma, (**C**harisma), Inteligencia, (**I**ntelligence), Agilidad, (**A**gility), y Suerte (**L**uck). [23](#)



Universidad
Politécnica
de Madrid

ETSI **SISTEMAS**
INFORMÁTICOS