



TRABAJO FIN DE GRADO

# Título del trabajo

Realizado por  
**Nombre del alumno**

Para la obtención del título de  
Grado en Ingeniería Informática - Ingeniería del Software

Dirigido por  
Nombre del profesor tutor

En el departamento de  
Lenguajes y Sistemas Informáticos

**Convocatoria de junio/julio/diciembre, curso 20XX/YY**

*Aquí la dedicatoria del trabajo*

# Agradecimientos

---

Quiero agradecer a X por...

También quiero agradecer a Y por...

# Resumen

---

Incluya aquí un resumen de los aspectos generales de su trabajo, en español.

**Palabras clave:** Palabra clave 1, palabra clave 2, ..., palabra clave N

# Abstract

---

This section should contain an English version of the Spanish abstract.

**Keywords:** Keyword 1, keyword 2, ..., keyword N

# Índice general

---

|          |                                  |           |
|----------|----------------------------------|-----------|
| <b>1</b> | <b>Ejemplos de uso de LaTeX</b>  | <b>1</b>  |
| 1.1.     | Introducción                     | 1         |
| 1.2.     | Estilos                          | 1         |
| 1.3.     | Listados                         | 1         |
| 1.4.     | Subsecciones                     | 2         |
| 1.4.1.   | Primera subsección               | 2         |
| 1.4.2.   | Segunda subsección               | 2         |
| 1.5.     | Imágenes y figuras               | 2         |
| 1.6.     | Tablas                           | 3         |
| 1.7.     | Referencias                      | 4         |
| 1.8.     | Extractos de código              | 4         |
| 1.9.     | Enlaces                          | 6         |
| 1.10.    | Citas y bibliografía             | 6         |
| 1.11.    | Ecuaciones                       | 6         |
| 1.12.    | Caracteres y símbolos especiales | 7         |
| <b>2</b> | <b>Introducción</b>              | <b>8</b>  |
| <b>3</b> | <b>Estudio previo</b>            | <b>9</b>  |
| 3.1.     | Introducción                     | 9         |
| 3.2.     | Objetivos                        | 9         |
| 3.3.     | Metodología                      | 9         |
| 3.4.     | Planificación                    | 9         |
| 3.5.     | Presupuesto                      | 9         |
| 3.6.     | Conclusiones                     | 9         |
| <b>4</b> | <b>Análisis del problema</b>     | <b>10</b> |
| 4.1.     | Introducción                     | 10        |
| 4.2.     | Requisitos de información        | 10        |
| 4.3.     | Requisitos funcionales           | 10        |
| 4.4.     | Requisitos no funcionales        | 10        |
| 4.5.     | Conclusiones                     | 10        |
| <b>5</b> | <b>Diseño de la solución</b>     | <b>11</b> |
| 5.1.     | Introducción                     | 11        |
| 5.2.     | Conclusiones                     | 11        |
| <b>6</b> | <b>Implementación</b>            | <b>12</b> |
| 6.1.     | Introducción                     | 12        |
| 6.2.     | Herramientas                     | 12        |
| 6.3.     | Conclusiones                     | 12        |

|                                 |           |
|---------------------------------|-----------|
| <b>7 Pruebas</b> . . . . .      | <b>13</b> |
| 7.1. Introducción . . . . .     | 13        |
| 7.2. Conclusiones . . . . .     | 13        |
| <b>8 Conclusiones</b> . . . . . | <b>14</b> |
| <b>Bibliografía</b> . . . . .   | <b>15</b> |

# Índice de figuras

---

|                                    |   |
|------------------------------------|---|
| 1.1. Un feroz depredador . . . . . | 2 |
|------------------------------------|---|



# Índice de tablas

---

|   |   |
|---|---|
| 1.1. Tabla LaTeX de ejemplo . . . . .                 | 3 |
| 1.2. Tabla compleja introducida como figura . . . . . | 3 |

# Índice de extractos de código

---

|                                  |   |
|----------------------------------|---|
| 1.1. Código Python . . . . .     | 4 |
| 1.2. Código Java . . . . .       | 5 |
| 1.3. Código HTML . . . . .       | 5 |
| 1.4. Código JavaScript . . . . . | 5 |

# 1. Ejemplos de uso de LaTeX

---

Este capítulo se incluye únicamente como ayuda y referencia de uso de LaTeX. No debe aparecer en el documento final.

## 1.1. Introducción

En este capítulo se muestran ejemplos de uso de LaTeX para operaciones comunes.

## 1.2. Estilos

Se pueden aplicar estilos al texto como **negritas**, *cursiva*, subrayado y monoespaciado. También se **pueden aplicar colores**, y combinar estilos. Se recomienda usar sólo negritas para hacer énfasis, y no abusar de este recurso.

Por comodidad para usuarios no habituados con LaTeX, esta plantilla define algunos alias de comandos más fáciles de recordar para estilos de texto comunes: **negritas**, *cursiva* y código.

## 1.3. Listados

Con itemize se pueden crear listas no numeradas:

- Fresas
- Melocotones
- Piñas
- Nectarinas

De manera similar, enumerate permite crear listas numeradas:

1. Elaborar la memoria del TFG
2. Elaborar la presentación
3. Presentar el TFG
4. Solicitar el título de Grado

## 1.4. Subsecciones

Se pueden definir subsecciones con el comando subsection:

### 1.4.1. Primera subsección

Esto es una subsección

### 1.4.2. Segunda subsección

Esto es otra subsección.

#### Sub-sub-sección

Este es un tercer nivel de profundidad, que no aparece en el índice general. Se recomienda no utilizarlo, si es posible.

## 1.5. Imágenes y figuras

Todas las imágenes y figuras del documento se incluirán en la carpeta “figures”. Se pueden incluir de la siguiente manera:



**Figura 1.1:** Un feroz depredador

Observe que las figuras se numeran automáticamente según el capítulo y el número de figuras que hayan aparecido anteriormente en dicho capítulo. Existen muchas maneras de definir el tamaño de una figura, pero se aconseja utilizar la mostrada en este ejemplo: se define el ancho de la figura como un porcentaje del

ancho total de la página, y la altura se escala automáticamente. De esta manera, el ancho máximo de una figura sería  $1.0 * \text{textwidth}$ , lo que aseguraría que se muestra al máximo tamaño posible sin sobrepasar los márgenes del documento.

Tenga en cuenta que LaTeX intenta incluir las figuras en el mismo sitio donde se declaran, pero en ocasiones no es posible por motivos de espacio. En esos casos, LaTeX colocará la figura lo más cerca posible de su declaración, puede que en una página diferente. Esto es un comportamiento normal.

## 1.6. Tablas

Existe una gran variedad de formas de crear tablas en LaTeX puro, y todas ellas tienen cierta complejidad. A continuación se muestra un ejemplo simple de tabla nativa, en la Tabla 1.1. Se recomienda crear un archivo en la carpeta *tables* por cada tabla nativa que se desee incluir, y enlazarla mediante el comando `input`.

| Columna L        | Columna C        | Columna R        | Columna P        |
|------------------|------------------|------------------|------------------|
| Texto de ejemplo | Texto de ejemplo | Texto de ejemplo | Texto de ejemplo |
| ABC              | DEF              | HIJ              | KLM              |

Tabla 1.1: Tabla LaTeX de ejemplo

Para tablas con un formato más complejo, considere la posibilidad de diseñarla usando otro software externo (por ejemplo Excel) e incluirla de manera similar a una figura. **Observe en el código LaTeX a continuación cómo usar el comando `captionof{table}`, en lugar de simplemente `caption`, hace que se liste como una Tabla en lugar de como una Figura:**

| Group  | N   | %        |         |          | Comparisons |    |        |
|--|-----|----------|---------|----------|-------------|----|--------|
|  |     | positive | neutral | negative | $\chi^2$    | df | p      |
| Q1 (a) Programming Labs are useful for learning the subject  |     |          |         |          |             |    |        |
| CS group   | 400 | 73       | 7       | 20       | 0.9         | 4  | ns     |
| Math group   | 400 | 70       | 8       | 22       |             |    |        |
| Q1 (b) Working with data makes predictions more reliable     |     |          |         |          |             |    |        |
| CS group   | 400 | 52       | 12      | 36       | 13.7        | 4  | < 0.05 |
| Math group   | 400 | 44       | 15      | 41       |             |    |        |
| Q1 (c) Lessons in class are more active than web conferences |     |          |         |          |             |    |        |
| CS group   | 400 | 46       | 11      | 43       | 14.8        | 4  | < 0.01 |
| Math group   | 400 | 31       | 14      | 55       |             |    |        |

Tabla 1.2: Tabla compleja introducida como figura

## 1.7. Referencias

Observe cómo en el código fuente de esta sección se ha usado varias veces el comando `label`. Este comando permite marcar un elemento, ya sea capítulo, sección, figura, etc. para hacer una referencia numérica al mismo. Para referenciar una label se usa el comando `ref` incluyendo el nombre de la referencia:

Este es el capítulo [1](#).

En la sección [1.2](#) se muestran ejemplos de estilos.

La subsección [1.4.1](#) explica...

En la Figura [1.1](#) vemos que...

Esto evita que tengamos que escribir directamente los índices de las secciones y figuras que queremos mencionar, ya que LaTeX lo hace por nosotros y además se encarga de mantenerlos actualizados en caso de que cambien (pruebe a mover este capítulo al final del documento y observe cómo se actualizan automáticamente todos los índices referenciados). Además, las referencias mediante “`ref`” actúan como hipervínculos dentro del documento que llevan al elemento referenciado al pulsar en ellas.

Es habitual nombrar las “label” con un prefijo que indica el tipo de elemento para encontrarlo luego más fácilmente, pero no es obligatorio.

## 1.8. Extractos de código

Se pueden incluir extractos de código mediante `lstlisting`:

---

```
1 num = float(input("Enter a number: "))
2 if num > 0:
3     print("Positive number")
4 elif num == 0:
5     print("Zero")
6 else:
7     print("Negative number")
```

---

**Extracto de código 1.1:** Código Python

Para evitar tener que incluir el código directamente en el texto del documento, se pueden guardar en archivos separados y referenciarlos:

Los extractos de código también se pueden referenciar mediante `label/ref`: Extractos de código [1.1](#), [1.2](#), [1.3](#), [1.4](#).

---

```
1 class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello, LaTeX!");
4     }
5 }
```

---

**Extracto de código 1.2: Código Java**

---

```
1 <html>
2     <head>
3         <title>LaTeX example</title>
4     </head>
5
6     <body>
7         Hello, LaTeX!
8     </body>
9 </html>
```

---

**Extracto de código 1.3: Código HTML**

---

```
1 function main() {
2     console.log("Hello, LaTeX!");
3 }
```

---

**Extracto de código 1.4: Código JavaScript**

## 1.9. Enlaces

Puede enlazar una web externa mediante el comando `url`: <https://www.example.com>. También se puede vincular un enlace a un texto mediante el comando `href`: [dominio de ejemplo](#).

## 1.10. Citas y bibliografía

En LaTeX, los elementos de la bibliografía se almacenan en un fichero bibliográfico en un formato llamado BibTeX, en el caso de este proyecto se encuentran en “bibliografia.bib”. Para citar un elemento se usa el comando `cite`. Se pueden citar tanto artículos científicos [1] como enlaces web [2].

También se puede usar el comando `citet` para incluir una referencia junto con el nombre de su autor o autores: Borrego et al. [3]. Todas las citas se numeran automáticamente y se incluyen en la sección de bibliografía del trabajo. El orden por defecto es según su orden de aparición en el documento. Para ordenarlas por orden alfabético del autor, puede modificar el comando `bibliographystyle` del archivo principal y reemplazar su valor por el estilo `plainnat` (orden alfabético, nombres completos) o `abbrvnat` (orden alfabético, nombres abreviados).

Observe cómo los elementos bibliográficos almacenados en “bibliografia.bib” tienen una etiqueta asociada, que es la que se usa al citarlos mediante `cite`. **Añadir una referencia al fichero bibliográfico no hace que ésta aparezca automáticamente en la sección de bibliografía del trabajo, es necesario citarla en algún lugar del mismo.**

## 1.11. Ecuaciones

LaTeX tiene un potente motor para mostrar ecuaciones matemáticas y un amplio catálogo de símbolos matemáticos. El entorno matemático se puede activar de muchas maneras. Para incluir ecuaciones simples en un texto se pueden rodear de símbolos dólar:  $1 + 2 = 3$ ,  $\sqrt{81} = 3^2 = 9$ ,  $\forall x \in y \exists z : S_z < 4$ .

Las ecuaciones más complejas pueden expresarse aparte y son numeradas: ecuación 1.1.

$$\lim_{x \rightarrow 0} \frac{e^x - 1}{2x} \stackrel{\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}}{\text{H}} \lim_{x \rightarrow 0} \frac{e^x}{2} = \frac{1}{2} + 7 \int_0^2 \left( -\frac{1}{4} (e^{-4t_1} + e^{4t_1-8}) \right) dt_1 \quad (1.1)$$

Dispone [aquí](#) de un amplio listado de símbolos que pueden usarse en modo matemático.



## 1.12. Caracteres y símbolos especiales

Algunos caracteres y símbolos deben ser escapados para poder representarse en el documento, ya que tienen un significado especial en LaTeX. Algunos de ellos son:

- El símbolo dólar \$ se usa para ecuaciones.
- El tanto por ciento % se usa para comentarios en el código fuente.
- El símbolo euro € suele dar problemas si se escribe directamente.
- El guión bajo \_ se usa para subíndices en modo matemático.
- Las comillas deben expresarse ‘así’ para comillas simples y “así” para comillas dobles. Las comillas españolas pueden expresarse «así».
- La barra invertida o contrabarra \ se usa para comandos LaTeX.
- Otros símbolos que deben escaparse son las llaves { }, el ampersand &, la almohadilla # y los símbolos mayor que > y menor que <.

## 2. Introducción

---

# **3. Estudio previo**

---

## **3.1. Introducción**

Breve introducción al capítulo

## **3.2. Objetivos**

## **3.3. Metodología**

## **3.4. Planificación**

## **3.5. Presupuesto**

## **3.6. Conclusiones**

# **4. Análisis del problema**

---

## **4.1. Introducción**

En este capítulo explicaremos...

## **4.2. Requisitos de información**

Los requisitos de información son...

## **4.3. Requisitos funcionales**

Los requisitos funcionales son...

## **4.4. Requisitos no funcionales**

Los requisitos no funcionales son...

## **4.5. Conclusiones**

En este capítulo concluimos que...

# **5. Diseño de la solución**

---

## **5.1. Introducción**

En este capítulo explicaremos...

## **5.2. Conclusiones**

En este capítulo concluimos que...

# **6. Implementación**

---

## **6.1. Introducción**

En este capítulo explicaremos...

## **6.2. Herramientas**

## **6.3. Conclusiones**

En este capítulo concluimos que...

# 7. Pruebas

---

## 7.1. Introducción

En este capítulo explicaremos...

## 7.2. Conclusiones

En este capítulo concluimos que...

## 8. Conclusiones

---



# Bibliografía

---

- [1] Agustín Borrego, Daniel Ayala, Inma Hernández, Carlos R. Rivero, and David Ruiz. Generating rules to filter candidate triples for their correctness checking by knowledge graph completion techniques. In *K-CAP*, pages 115–122, 2019. doi: 10.1145/3360901.3364418.
- [2] Universidad de Sevilla. Página principal de la escuela técnica de ingeniería informática, 2020. URL <https://www.informatica.us.es>.
- [3] Agustín Borrego, Daniel Ayala, Inma Hernández, Carlos R. Rivero, and David Ruiz. CAFE: Knowledge graph completion using neighborhood-aware features. *Eng. Appl. Artif. Intell.*, 103:104302, 2021. doi: 10.1016/j.engappai.2021.104302. URL <https://doi.org/10.1016/j.engappai.2021.104302>.