

TRABAJO FIN DE GRADO

Trabajo fin de grado

Realizado por
Nombre Del Alumno

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

Dirigido por
Director 1
Director 2

En el departamento de
Lenguajes y Sistemas Informáticos

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



Escuela Técnica Superior de
Ingeniería Informática

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

1 Ejemplos de uso de Typst	1
1.1. Introducción	1
1.2. Estilos	1
1.3. Listados	2
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	5
1.9. Enlaces	6
1.10. Citas y bibliografía	6
1.11. Ecuaciones	7
1.12. Caracteres y símbolos especiales	7
2 Introducción	8
3 Estudio previo	9
3.1. Introducción	9
3.2. Objetivos	9
3.3. Metodología	9
3.4. Planificación	9
3.5. Presupuesto	9
3.6. Conclusiones	9
4 Análisis del problema	10
4.1. Introducción	10
4.2. Requisitos de información	10
4.3. Requisitos funcionales	10
4.4. Requisitos no funcionales	10
4.5. Conclusiones	11
5 Diseño de la solución	12

Índice general

5.1. Introducción	12
5.2. Conclusiones	12
6 Implementación	13
6.1. Introducción	13
6.2. Herramientas	13
6.3. Conclusiones	13
7 Pruebas	14
7.1. Introducción	14
7.2. Conclusiones	14
8 Conclusiones	15
Bibliografía	16

Índice de figuras

1.1. Un feroz depredador	3
------------------------------------	---

Índice de tablas

1.1. Tabla Typst de ejemplo	4
1.2. Tabla compleja introducida como imagen	4

Índice de extractos de código

1.1. Código Python	5
1.2. Código Java	5
1.3. Código HTML	6
1.4. Código JavaScript	6

Capítulo 1

Ejemplos de uso de Typst

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

1.1. Introducción

En este capítulo se muestran ejemplos de uso de Typst para operaciones comunes. En el código fuente se pueden observar dos modos: contenido, y código. En un archivo, el modo por defecto es el modo contenido. Para usar el modo código dentro del contenido, por ejemplo para llamar a una función, se usa el símbolo #, pero si ya se está en modo código no es necesario usarlo. De forma inversa, para indicar contenido estando en el modo código, se delimita el contenido con [...].

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.

Para cambiar el formato de grandes cantidades de contenido, deben usarse las funciones `#set funcion(parametro:valor)` (altera un parámetro siempre que se llame a la función) y `#show funcion: it => {...}` (reemplaza el resultado de una función completamente).

1.3. Listados

Con “-” se pueden crear listas no numeradas:

- Fresas
- Melocotones
- Piñas
- Nectarinas

De manera similar, “1.” 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 diferentes cantidades de “=”:

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.

Tenga en cuenta que Typst intenta incluir las figuras en el mismo sitio donde se declaran, pero en ocasiones no es posible por motivos de espacio. En estos casos, Typst 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 Typst puro. Se pueden usar dos funciones similares: `#table` y `#grid`. `#grid` es la función básica, mientras que `#table` añade básicamente presets para algunos atributos relacionados con el aspecto. Mediante `#set` y `#show` se pueden alterar estilos para una fila o columna concreta. A continuación se muestra un ejemplo simple de tabla nativa, en la [Tabla 1.1](#).

Usando un `#grid` se podría colocar manualmente dónde hay bordes con funciones indicadas en la documentación.

Columna L	Columna C	Columna R	Columna 2cm
Texto de ejemplo	Texto de ejemplo	Texto de ejemplo	Texto de ejemplo
ABC	DEF	GHI	JKL

Tabla 1.1: Tabla Typst 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 Typst a continuación cómo usar el atributo kind para indicar manualmente el tipo de figura (figura en Typst no son las imágenes sino un elemento genérico con leyenda). Se puede indicar cualquier kind personalizado como se indica en la documentación:**

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

Tabla 1.2: Tabla compleja introducida como imagen

1.7. Referencias

Observe cómo en el código fuente de esta sección se ha usado varias veces el siguiente formato: `<etiqueta>`. Esto permite marcar un elemento, ya sea capítulo, sección, figura, etc. para hacer una referencia numérica al mismo. Para referenciar una etiqueta se usa el formato de `@referencias` incluyendo el nombre de la referencia:

Esta es la [Sección 1](#).

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

La [Sección 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 Typst lo hace por nosotros y además

1. Ejemplos de uso de Typst

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 actúan como hipervínculos dentro del documento que llevan al elemento referenciado al pulsar en ellas.

Es habitual nombrar las etiquetas 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, existiendo paquetes que los hacen más bonitos como [codly](#):

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

 Python

Listado 1.1: Código Python

Se puede usar la función proporcionada para cambiar desde el momento que se usa el estilo de los fragmentos. Por ejemplo, al comienzo de esta plantilla se llama a Typst `#codly(zebra-fill:none)` para eliminar el sombreado alterno de las líneas.

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

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

 Java

Listado 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>

```

 HTML**Listado 1.3:** Código HTML

```

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

```

 JavaScript**Listado 1.4:** Código JavaScript

1.9. Enlaces

Puede enlazar una web externa mediante la función `#link` o directamente escribiendo el enlace: <https://example.com>. También se puede vincular un enlace a un texto: [dominio de ejemplo](#).

1.10. Citas y bibliografía

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

También se puede usar la función `#cite` para incluir opciones adicionales o personalizar el resultado, como incluir una referencia junto con el nombre de su autor o autores: A. Borrego, D. Ayala, I. Hernández, C. R. Rivero, y D. Ruiz [3]¹. 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 la opción `style` de

¹El estilo bibliográfico, determinado por un archivo CSL que puede personalizarse, determina cuando la lista de autores se reemplaza por "et al". Ver <https://github.com/typst/hayagriva/issues/164>

1. Ejemplos de uso de Typst

la función `#bibliography("/bibliografia.bib", style:...)` del archivo principal y reemplazar su valor por otro estilo.

Observe cómo los elementos bibliográficos almacenados en `bibliografia.bib` tienen una etiqueta asociada, que es la que se usa al citarlos. **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

Typst 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 \(1.1\)](#)

$$\lim_{x \rightarrow 0} \frac{e^x - 1}{2x} \stackrel{[0]}{\equiv} \lim_{x \rightarrow 0} \frac{e^2}{2} = \frac{1}{2} + 7 \int_0^2 \left(-\frac{1}{4}(e^{-4t_1} + e^{4t_1-8}) \right) dt_1 \quad (1.1)$$

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 Typst. Algunos de ellos son:

- El símbolo dólar \$ se usa para ecuaciones.
- Los símbolos usados para formato como *, _, `.
- Las comillas deben expresarse 'así' para comillas simples y "así" para comillas dobles. Las comillas españolas pueden expresarse «así».
- La almohadilla # se usa para llamar a funciones dentro del modo contenido.

Capítulo 2

Introducción

Capítulo 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

Capítulo 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...

Capítulo 5

Diseño de la solución

5.1. Introducción

En este capítulo explicaremos...

5.2. Conclusiones

En esta capítulo concluimos que...

Capítulo 6

Implementación

6.1. Introducción

En este capítulo explicaremos...

6.2. Herramientas

6.3. Conclusiones

En este capítulo concluimos que...

Capítulo 7

Pruebas

7.1. Introducción

En este capítulo explicaremos...

7.2. Conclusiones

En este capítulo concluimos que...

Capítulo 8

Conclusiones

Bibliografía

- [1] A. Borrego, D. Ayala, I. Hernández, C. R. Rivero, y D. Ruiz, «Generating Rules to Filter Candidate Triples for their Correctness Checking by Knowledge Graph Completion Techniques», en *K-CAP*, 2019, pp. 115-122. doi: [10.1145/3360901.3364418](https://doi.org/10.1145/3360901.3364418).
- [2] U. de Sevilla, «Página principal de la Escuela Técnica de Ingeniería Informática». [En línea]. Disponible en: <https://www.informatica.us.es/>
- [3] A. Borrego, D. Ayala, I. Hernández, C. R. Rivero, y D. Ruiz, «CAFE: Knowledge graph completion using neighborhood-aware features», *Eng. Appl. Artif. Intell.*, vol. 103, p. 104302, 2021, doi: [10.1016/j.engappai.2021.104302](https://doi.org/10.1016/j.engappai.2021.104302).