



Escuela
Politécnica
Superior

Título del Trabajo Fin de Grado/Máster



Grado en Ingeniería en Sonido e
Imagen en Telecomunicación

Trabajo Fin de Grado

Autor:

Nombre Apellido1 Apellido2

Tutor:

Dr./Dra. Nombre Apellido1 Apellido2

Febrero 2026



Universitat d'Alacant
Universidad de Alicante

Título del Trabajo Fin de Grado/Máster

Subtítulo del proyecto

Autor

Nombre Apellido1 Apellido2

Tutor

Dr./Dra. Nombre Apellido1 Apellido2

Departamento de ejemplo



Grado en Ingeniería en Sonido e Imagen en Telecomunicación



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, Febrero 2026

A mi familia, por su apoyo incondicional.

Agradecimientos

En primer lugar, quiero agradecer a mi tutor/a por su dedicación y orientación durante el desarrollo de este trabajo.

También quiero expresar mi gratitud a mi familia y amigos por su apoyo constante.

Finalmente, agradecer a la Escuela Politécnica Superior de la Universidad de Alicante por la formación recibida durante estos años.

Resumen

Este documento es la plantilla oficial para la elaboración de Trabajos de Fin de Grado (TFG) y Trabajos de Fin de Máster (TFM) de la Escuela Politécnica Superior de la Universidad de Alicante.

La plantilla proporciona un formato profesional y consistente, cumpliendo con las directrices de estilo de la EPS. Incluye:

- Portadas oficiales a color y en blanco y negro
- Configuración automática según la titulación
- Estilos predefinidos para código fuente
- Formato de bibliografía según normas APA
- Soporte para acrónimos y glosarios
- Optimización de figuras TikZ

Palabras clave: TFG, TFM, plantilla, LaTeX, Universidad de Alicante

Abstract

This document is the official template for the preparation of Bachelor's Thesis (TFG) and Master's Thesis (TFM) at the Polytechnic School of the University of Alicante.

The template provides a professional and consistent format, complying with the EPS style guidelines. It includes:

- Official color and black-and-white covers
- Automatic configuration based on the degree program
- Predefined styles for source code
- Bibliography format according to APA standards
- Support for acronyms and glossaries
- TikZ figure optimization

Keywords: Bachelor's Thesis, Master's Thesis, template, LaTeX, University of Alicante

Índice general

Agradecimientos	v
Resumen	vii
Abstract	ix
1. Introducción (Ejemplos de contenido y estilos)	1
1.1. Sobre esta plantilla	1
1.2. Estructura de un TFG/TFM	1
1.3. Secciones y subsecciones	2
1.3.1. Ejemplo de subsección	2
1.3.1.1. Ejemplo de subsubsección	2
1.3.1.1.1. Ejemplo de párrafo	2
1.4. Citas bibliográficas	2
1.5. Notas al pie de página	3
1.6. Estilos de texto	3
1.7. Acrónimos y glosario	3
1.8. Tareas pendientes y notas	4
1.9. Comandos personalizados de la plantilla	4
1.10. Hipervínculos y URLs	5
1.11. Consejos para la redacción	5
1.11.1. Estructura del texto	5
1.11.2. Estilo académico	5
1.11.3. Referencias y citas	5
1.11.4. Figuras y tablas	6
1.11.5. Revisión final	6
2. Marco Teórico (Ejemplos de listas)	7
2.1. Listas básicas	7
2.1.1. Listas con viñetas (itemize)	7
2.1.2. Listas numeradas (enumerate)	7
2.1.3. Listas anidadas	8
2.2. Listas de definición	8
2.2.1. Listas de definición anidadas	9
2.3. Estado del arte	9
2.3.1. Trabajos previos	9
2.3.2. Tecnologías relacionadas	9
2.4. Herramientas utilizadas	10

3. Objetivos (Ejemplos de tablas)	11
3.1. Objetivo general	11
3.2. Objetivos específicos	11
3.3. Tablas en LaTeX	11
3.3.1. Tabla simple	11
3.3.2. Tabla con booktabs	12
3.3.3. Tabla con columnas de ancho fijo	12
3.3.4. Tabla con multicolumna y multifila	13
3.3.5. Tabla con colores alternados	13
3.3.6. Tabla larga (longtable)	13
3.4. Generadores de tablas	14
3.5. Forzar posición de tablas	14
3.6. Tablas con notas al pie	14
3.7. Tabla con celdas de colores personalizados	15
3.8. Tabla de ancho completo	15
4. Metodología (Ejemplos de figuras)	17
4.1. Metodología de trabajo	17
4.2. Inserción de figuras	17
4.2.1. Figura simple	17
4.2.2. Subfiguras	18
4.2.3. Múltiples imágenes en tabla	18
4.3. Planificación temporal	19
4.4. Diagramas con TikZ	19
4.4.1. Diagrama de flujo	20
4.4.2. Diagrama de bloques	20
4.5. Recursos utilizados	20
4.5.1. Recursos hardware	20
4.5.2. Recursos software	21
4.6. Gestión del proyecto	21
5. Desarrollo: Estilos de Código	23
5.1. Estilo VS Code Light	23
5.1.1. Python	23
5.1.2. Sin numeración de líneas	23
5.1.3. JavaScript	23
5.1.4. Java	24
5.2. Estilo VS Code Dark	24
5.2.1. Python Dark	24
5.2.2. JavaScript Dark	24
5.3. Lenguajes Web	25
5.3.1. HTML	25
5.3.2. CSS	25
5.4. Lenguajes de Datos	25
5.4.1. SQL	25

5.5. Lenguajes de Sistemas	26
5.5.1. C++	26
5.5.2. Rust	26
5.5.3. Go	26
5.6. Otros Lenguajes	27
5.6.1. PHP	27
5.6.2. Ruby	27
5.7. DevOps y Configuración	28
5.7.1. Docker	28
5.8. Entorno Genérico	28
5.8.1. Versión Dark del Genérico	28
5.9. Resumen de Entornos Disponibles	29
6. Componentes Especializados	31
6.1. Componentes Comunes	31
6.1.1. Cajas de Información	31
6.1.2. Cajas con Título	31
6.1.3. Listas Especiales	32
6.1.3.0.1. Checklist:	32
6.1.3.0.2. Pros y Contras:	32
6.1.3.0.3. Lista de Pasos:	33
6.1.4. Badges e Indicadores	33
6.1.4.0.1. Badges personalizados:	33
6.1.4.0.2. Badges predefinidos:	33
6.1.4.0.3. Versiones:	33
6.1.4.0.4. Barra de progreso:	33
6.1.4.0.5. Rating con estrellas:	33
6.1.4.0.6. Indicador de nivel:	33
6.1.5. Tarjetas de Información	33
6.1.6. Timeline	34
6.1.7. Comparativas	35
6.1.8. Citas Destacadas	35
6.2. Componentes de Software	35
6.2.1. API REST Endpoints	35
6.2.1.0.1. Métodos HTTP disponibles:	36
6.2.2. Terminal y Consola	36
6.2.3. Árbol de Directorios	36
6.2.4. Clases UML	36
6.2.5. Tabla de Requisitos	37
6.2.6. Esquema de Base de Datos	37
6.2.7. Git y Control de Versiones	37
6.2.8. Logs y Métricas	38
6.3. Componentes de Telecomunicaciones	38
6.3.1. Carta de Smith	38
6.3.2. Diagrama de Constelación	39

6.3.3.	Máquina de Estados Finitos	40
6.3.4.	Tramas de Protocolo	40
6.3.5.	Parámetros S	42
6.3.6.	Diagramas de Bloques	42
6.3.7.	Diagramas de Temporización	42
6.4.	Componentes de Arquitectura	42
6.4.1.	Diagrama de Gantt	43
6.4.2.	Fichas Técnicas de Materiales	43
6.4.3.	Presupuestos	43
6.4.4.	Normativa Aplicable	44
6.4.5.	Control de Calidad	44
6.4.6.	Etiquetas Energéticas	44
6.4.7.	Certificaciones	45
6.5.	Componentes de Química	45
6.5.1.	Reacciones Químicas	45
6.5.2.	Fichas de Compuestos	45
6.5.3.	Protocolos de Laboratorio	46
6.5.4.	Resultados Analíticos	46
6.5.5.	Equipamiento de Laboratorio	46
6.6.	Componentes de Geología	47
6.6.1.	Columna Estratigráfica	47
6.6.2.	Tabla de Minerales	47
6.6.3.	Datos Geotécnicos	48
6.6.4.	Clasificación de Suelos	48
6.6.5.	Eras Geológicas	48
6.6.6.	Riesgos Geológicos	48
6.6.7.	Símbolos Geológicos	49
6.7.	Componentes de Prevención	49
6.7.1.	Matriz de Riesgos	49
6.7.2.	Evaluación de Riesgos	49
6.7.3.	Checklist de Seguridad	50
6.7.4.	Señalización de Seguridad	50
6.7.4.0.1.	Señales de advertencia:	50
6.7.4.0.2.	Señales de prohibición:	50
6.7.4.0.3.	Señales de obligación:	50
6.7.4.0.4.	Señales de emergencia:	50
6.7.5.	Equipos de Protección Individual	50
6.7.6.	Indicadores de Seguridad	51
6.7.7.	Procedimiento de Emergencia	51
6.7.8.	Registro de Formación	51
6.7.9.	Informe de Accidente	51
7.	Resultados (Ejemplos de gráficas)	53
7.1.	Gráficas con PGFPlots	53
7.1.1.	Gráfica de líneas	53

7.1.2. Gráfica de barras	54
7.1.3. Gráfica circular (pie chart)	54
7.1.4. Gráfica de área	55
7.1.5. Gráfica de dispersión	55
7.2. Resultados de la implementación	55
7.2.1. Funcionalidades implementadas	55
7.2.2. Consumo de recursos	56
7.3. Análisis de resultados	56
7.3.1. Cumplimiento de objetivos	56
7.3.2. Métricas de rendimiento	57
7.4. Ejemplo de gráfica 3D	57
7.5. Exportar gráficas desde herramientas externas	57
7.6. Gráficas con marcadores y anotaciones	58
7.6.1. Gráfica con marcadores sobre puntos específicos	58
7.6.2. Gráfica con barras de error	59
7.6.3. Gráfica con área sombreada (intervalo de confianza)	59
7.6.4. Gráfica con múltiples ejes Y	60
7.6.5. Gráfica de distribución (histograma)	60
7.6.6. Gráfica de radar (polígono)	60
7.6.7. Diagrama de Gantt simplificado	61
7.7. Tabla de resumen de resultados	61
8. Conclusiones (Ejemplos de matemáticas)	63
8.1. Ecuaciones matemáticas	63
8.1.1. Ecuaciones numeradas	63
8.1.2. Ecuaciones complejas	63
8.1.3. Ecuaciones en línea	63
8.1.4. Sistemas de ecuaciones	64
8.1.5. Ecuaciones agrupadas	64
8.1.6. Matrices	64
8.1.7. Ecuaciones alineadas	64
8.1.8. Ecuación con condiciones	64
8.1.9. Teoremas y demostraciones	65
8.2. Conclusiones generales	65
8.3. Aportaciones del trabajo	65
8.4. Dificultades encontradas	66
8.5. Trabajo futuro	66
8.6. Valoración personal	66
Bibliografía	67
Lista de Acrónimos y Abreviaturas	69
A. Manual de usuario	71
A.1. Requisitos del sistema	71

A.2. Acceso al sistema	71
A.2.1. Inicio de sesión	71
A.2.2. Recuperación de contraseña	71
A.3. Interfaz principal	72
A.4. Funcionalidades principales	72
A.4.1. Gestión de datos	72
A.4.2. Generación de informes	72
A.5. Preguntas frecuentes	72
B. Documentación técnica	73
B.1. Arquitectura del sistema	73
B.1.1. Diagrama de componentes	73
B.2. API REST	73
B.2.1. Endpoints principales	73
B.2.1.1. Obtener lista de usuarios	74
B.2.1.2. Obtener usuario por ID	74
B.2.1.3. Crear nuevo usuario	74
B.2.1.4. Actualizar usuario	75
B.2.1.5. Eliminar usuario	75
B.3. Modelo de datos	75
B.3.1. Esquema de base de datos	75
B.4. Configuración del entorno	76
B.4.1. Variables de entorno	76
B.5. Instrucciones de despliegue	76
B.5.1. Requisitos previos	76
B.5.2. Pasos de despliegue	77
C. Manual de Estilos de Código	79
C.1. Temas Disponibles	79
C.1.1. Ejemplo del tema Simple	79
C.2. Nomenclatura de Entornos	79
C.3. Lenguajes Predefinidos	80
C.4. Ejemplos de Uso	81
C.4.1. Tema VS Code Light	81
C.4.1.1. Con números de línea	81
C.4.1.2. Sin números de línea	82
C.4.2. Tema VS Code Dark	82
C.4.2.1. Con números de línea	82
C.4.2.2. Sin números de línea	82
C.5. Entorno Genérico	83
C.6. Consideraciones Especiales	83
C.6.1. Código C/C++ con includes	83
C.6.2. Títulos personalizados	83
C.6.3. Requisitos de compilación	84
C.7. Referencia Rápida	84

D. Técnicas Avanzadas de \LaTeX	85
D.1. Tablas Rotadas (Sideways Tables)	85
D.1.1. Ejemplo de tabla rotada	85
D.1.2. Cuándo usar tablas rotadas	85
D.2. Páginas en Horizontal (Landscape)	87
D.2.1. Método recomendado: Comandos de la plantilla	87
D.2.2. Diagrama de flujo del sistema (página horizontal)	88
D.2.3. Segunda página horizontal consecutiva	89
D.2.4. Cuándo usar páginas landscape	90
D.3. Inclusión de Documentos PDF Externos	90
D.3.1. Sintaxis básica	90
D.3.2. Opciones más utilizadas	91
D.3.3. Ejemplo: Documento PDF incluido	91
D.3.4. Múltiples páginas en una hoja	94
D.4. Figuras de Ancho Completo	94
D.4.1. Figura que invade márgenes	94
D.5. Notas al Margen	94
D.5.1. Uso básico	94
D.6. Marcas de Agua	95
D.7. Texto en Columnas	95
D.7.1. Ejemplo de texto en columnas	95
D.8. Minipáginas y Cajas	95
D.8.1. Ejemplo de minipáginas	96
D.9. Resumen de Paquetes Utilizados	96


















Índice de figuras






4.1.	Ejemplo de figura simple (placeholder)	18
4.2.	Ejemplo de subfiguras horizontales	18
4.3.	Matriz de configuraciones experimentales	19
4.4.	Planificación temporal del proyecto (diagrama de Gantt simplificado) . .	19
4.5.	Diagrama de flujo del algoritmo principal	20
4.6.	Arquitectura del sistema	20
7.1.	Tiempos de respuesta según carga de usuarios	53
7.2.	Comparativa de evaluación del sistema	54
7.3.	Distribución del tiempo del proyecto	54
7.4.	Evolución de usuarios activos por plataforma	55
7.5.	Correlación entre complejidad y tiempo de ejecución	55
7.6.	Superficie gaussiana $f(x, y) = e^{-x^2-y^2}$	57
7.7.	Curva de calentamiento con puntos críticos marcados	58
7.8.	Comparativa de rendimiento con intervalos de confianza	59
7.9.	Evolución del entrenamiento con banda de confianza	59
7.10.	Ventas y beneficios mensuales (doble eje Y)	60
7.11.	Distribución de tiempos de respuesta	60
7.12.	Comparativa de sistemas mediante gráfica de radar	61
7.13.	Diagrama de Gantt simplificado del proyecto	61
B.1.	Diagrama de componentes del sistema	73
D.1.	Diagrama de flujo completo del sistema en página horizontal	88

Índice de tablas

2.1.	Herramientas utilizadas en el desarrollo	10
3.1.	Ejemplo de tabla simple	12
3.2.	Tabla con estilo booktabs	12
3.3.	Parámetros de posicionamiento de elementos flotantes	12
3.4.	Ejemplo de tabla con celdas combinadas	13
3.5.	Especificaciones técnicas del sistema	13
3.6.	Lista de requisitos del sistema	13
3.7.	Comparativa de tecnologías web con notas	15
3.8.	Matriz de evaluación de riesgos	15
3.9.	Cronograma del proyecto usando todo el ancho	15
5.1.	Entornos de código con icono disponibles	29
5.2.	Sufijos disponibles para cada entorno	29
5.3.	Entornos genéricos para cualquier lenguaje	30
7.1.	Estado de implementación de funcionalidades	56
7.2.	Cumplimiento de objetivos	56
7.3.	Métricas de rendimiento del sistema	57
7.4.	Resumen de métricas de evaluación	62
C.1.	Temas de código disponibles	79
C.2.	Sistema de sufijos para entornos de código	80
C.3.	Lenguajes de programación con entornos predefinidos	80
C.4.	Lenguajes web y de datos	81
C.5.	Lenguajes de sistemas y DevOps	81
C.6.	Entornos genéricos de código	83
C.7.	Matriz de entornos por tema y numeración	84
D.1.	Comparativa completa de características por módulo del sistema desarrollado	86
D.2.	Matriz de trazabilidad requisitos-módulos (aprovechando el ancho de página landscape)	88
D.3.	Tabla adicional de ejemplo en segunda página landscape	89
D.4.	Opciones principales de <code>\includepdf</code>	91
D.5.	Paquetes y comandos \LaTeX para técnicas avanzadas	96

Índice de Códigos

1.	Niveles de secciones	2
2.	Comandos de citas biblatex/APA	2
3.	Uso de acrónimos	4
4.	Comandos de notas	4
5.	Lista con viñetas	7
6.	Lista numerada	7
7.	Lista de definiciones	8
8.	Código de tabla simple	11
9.	Forzar posición	14
10.	Código para insertar una figura	17
11.	 Python	23
12.	 Python	23
13.	 JavaScript	23
14.	 Java	24
15.	 Python	24
16.	 JavaScript	24
17.	 HTML	25
18.	 CSS	25
19.	 SQL	25
20.	 C++	26
21.	 Rust	26
22.	 Go	26
23.	 PHP	27
24.	 Ruby	27
25.	 Dockerfile	28
26.	 swift	28
27.	 kotlin	28
28.	Ejemplo de uso de matlab2tikz	58
29.	Ecuación numerada	63
30.	Esquema de la tabla usuarios	75
31.	Archivo .env de configuración	76
32.	Comandos de despliegue	77
33.	Ejemplo de código simple	79

34.	Ejemplo: Función recursiva	81
35.	 L ^A T _E X	82
36.	 Python	82
37.	 L ^A T _E X	82
38.	Tema oscuro con números	82
39.	 Python	82
40.	 haskell	83
41.	Ejemplo C++ con includes	83
42.	Script de instalación	84
43.	Código de tabla rotada	85
44.	Código para página horizontal (método simplificado)	87
45.	Inclusión de PDF externo	90
46.	Varias páginas PDF en una hoja	94
47.	Figura de ancho completo	94
48.	Notas al margen	94
49.	Marca de agua con draftwatermark	95
50.	Texto en dos columnas	95
51.	Dos minipáginas lado a lado	95

1. Introducción (Ejemplos de contenido y estilos)

Este capítulo presenta una guía completa de las capacidades de \LaTeX y de esta plantilla para la elaboración de Trabajos Fin de Grado y Trabajos Fin de Máster en la Escuela Politécnica Superior de la Universidad de Alicante.

1.1. Sobre esta plantilla

Esta plantilla ha sido diseñada siguiendo las directrices de estilo de la Escuela Politécnica Superior (EPS) de la Universidad de Alicante. Proporciona una estructura clara y profesional para la redacción de trabajos académicos.

Las principales características de esta plantilla son:

- Configuración sencilla mediante clave-valor con `\EPSsetup{...}`
- 21 portadas diferentes según la titulación
- Soporte para múltiples idiomas (español, inglés, valenciano)
- Resaltado de código con colores (minted + Pygments)
- Bibliografía con estilo American Psychological Association (APA)
- Compilación con LuaLaTeX

1.2. Estructura de un TFG/TFM

Según las normas de la EPS, un Trabajo Fin de Grado (TFG) o Trabajo Fin de Máster (TFM) debe contener las siguientes partes:

Preámbulo: Motivación y descripción breve de los objetivos del trabajo.

Agradecimientos: Reconocimientos a entidades y personas colaboradoras.

Dedicatoria: Opcional, con alineación a la derecha.

Índices: Contenidos, figuras, tablas y códigos.

Introducción: Importancia de la temática, vigencia y planteamiento del problema.

Marco teórico: Fundamentos conceptuales y estado del arte.

Objetivos: Objetivo general y específicos.

Metodología: Tipo de investigación, técnicas y procedimientos.

Resultados: Resultados obtenidos y análisis.

Conclusiones: Resumen de objetivos conseguidos.

Bibliografía: Referencias utilizadas (estilo APA recomendado).

Anexos: Material complementario.

1.3. Secciones y subsecciones

En \LaTeX existen diferentes niveles de organización del contenido:

Niveles de secciones

1	<code>\chapter{Capítulo}</code>	% Nivel 0
2	<code>\section{Sección}</code>	% Nivel 1
3	<code>\subsection{Subsección}</code>	% Nivel 2
4	<code>\subsubsection{Subsubsección}</code>	% Nivel 3
5	<code>\paragraph{Párrafo}</code>	% Nivel 4

1.3.1. Ejemplo de subsección

Este es el contenido de una subsección. Las subsecciones permiten organizar mejor el contenido dentro de cada sección.

1.3.1.1. Ejemplo de subsubsección

Las subsubsecciones son útiles para temas muy específicos dentro de una subsección.

1.3.1.1.1. Ejemplo de párrafo Los párrafos con título son útiles para pequeñas divisiones que no necesitan aparecer en el índice.

1.4. Citas bibliográficas

Para citar la bibliografía según el sistema APA se utilizan los siguientes comandos. El archivo de bibliografía se encuentra en `referencias.bib`.

Comandos de citas biblatex/APA

```
1 % Cita textual: Autor (año)
2 \textcite{latex2024}
3
4 % Cita entre paréntesis: (Autor, año)
5 \parencite{latex2024}
6
7 % Cita con página específica
8 \parencite[Cap.~2]{latex2024}
```

```
9
10 % Múltiples citas
11 \parencite{latex2024,overleaf2024}
```

Ejemplos de citas:

- Cita textual: Según LaTeX Project (2024), L^AT_EX es el estándar para documentos científicos.
- Cita entre paréntesis: L^AT_EX es ampliamente utilizado (LaTeX Project, 2024).
- Múltiples fuentes: Existen varios recursos disponibles (LaTeX Project, 2024; Overleaf, 2024).

1.5. Notas al pie de página

Las notas al pie se crean con el comando `\footnote{texto}`.

La plantilla utiliza LuaLaTeX¹ como motor de compilación por sus capacidades avanzadas de manejo de fuentes².

1.6. Estilos de texto

L^AT_EX ofrece múltiples estilos de texto:

- *Texto en cursiva* – `\textit{texto}`
- **Texto en negrita** – `\textbf{texto}`
- Texto monoespacio – `\texttt{texto}`
- TEXTO EN VERSALITAS – `\textsc{texto}`
- Texto subrayado – `\underline{texto}`
- ***Negrita y cursiva*** – combinación de comandos
- Texto pequeño – `{\small texto}`
- Texto grande – `{\large texto}`

1.7. Acrónimos y glosario

Los acrónimos se gestionan automáticamente con el paquete `glossaries`. La primera vez que aparece un acrónimo se muestra su forma completa, y en las siguientes apariciones solo la abreviatura.

¹LuaLaTeX es un motor de composición tipográfica que combina L^AT_EX con el lenguaje de programación Lua, permitiendo mayor flexibilidad en el procesamiento de documentos.

²Permite usar cualquier fuente OpenType o TrueType instalada en el sistema sin necesidad de configuración adicional.

Uso de acrónimos

```

1 % Primera aparición: muestra "Institute of Electrical
2 % and Electronics Engineers (IEEE)"
3 El \gls{ieee} es una institución importante.
4
5 % Sigüientes apariciones: muestra solo "IEEE"
6 El \gls{ieee} establece estándares.
7
8 % Para forzar forma larga: \glslong{ieee}
9 % Para forzar forma corta: \glsshort{ieee}

```

Ejemplo: El Institute of Electrical and Electronics Engineers (IEEE) es una institución importante en ingeniería. El IEEE establece estándares para la industria. Además del IEEE, existen otras organizaciones como la International Organization for Standardization (ISO).

Los acrónimos se definen en el archivo `contenido/anexos/acronimos.tex`.

1.8. Tareas pendientes y notas

Durante la redacción es útil marcar partes que requieren revisión usando el paquete `todonotes`:

Comandos de notas

```

1 % Nota al margen
2 \todo{Texto de la nota}
3
4 % Nota en línea
5 \todo[inline]{Nota dentro del texto}
6
7 % Figura pendiente
8 \missingfigure{Descripción de la figura}

```

Las notas también pueden incluirse dentro del texto:

Para indicar figuras que faltan por añadir:

1.9. Comandos personalizados de la plantilla

Esta plantilla proporciona comandos para acceder a la información configurada:

- `\EPStitulo` – Título: “Título del Trabajo Fin de Grado/Máster”
- `\EPSautor` – Autor: Nombre Apellido1 Apellido2
- `\EPStutor` – Tutor: Dr./Dra. Nombre Apellido1 Apellido2
- `\EPSfecha` – Fecha: Febrero 2026
- `\EPStipoTrabajo` – Tipo: Trabajo Fin de Grado

- \EPStitulacion – Titulación: Grado en Ingeniería en Sonido e Imagen en Telecomunicación

Este documento es un Trabajo Fin de Grado presentado en Febrero 2026.

1.10. Hipervínculos y URLs

Para incluir enlaces web se utiliza el comando `\url{dirección}`:

- Página de la EPS: <https://eps.ua.es>
- Overleaf (editor online): <https://www.overleaf.com>
- CTAN (repositorio de paquetes): <https://ctan.org>

Para enlaces con texto personalizado: [Escuela Politécnica Superior](#).

1.11. Consejos para la redacción

A continuación se presentan algunas recomendaciones para la redacción del TFG o TFM:

1.11.1. Estructura del texto

- Utiliza párrafos cortos y concisos (máximo 8-10 líneas)
- Comienza cada capítulo con una breve introducción de su contenido
- Finaliza cada capítulo con un resumen o transición al siguiente
- Evita frases demasiado largas; divide las ideas complejas

1.11.2. Estilo académico

- Escribe en tercera persona o forma impersonal («se ha desarrollado» en lugar de «he desarrollado»)
- Mantén un tono formal y objetivo
- Evita expresiones coloquiales o jerga no técnica
- Define los términos técnicos la primera vez que aparecen

1.11.3. Referencias y citas

- Cita siempre las fuentes de información que no sean tuyas
 - Usa `\parencite{clave}` para citas entre paréntesis: (Knuth, 1984)
 - Usa `\textcite{clave}` para citas narrativas: Lamport (1994)
 - Evita el exceso de citas; selecciona las más relevantes
 - Verifica que todas las referencias del archivo `.bib` estén citadas
-

1.11.4. Figuras y tablas

- Toda figura y tabla debe estar referenciada en el texto
- Las descripciones (captions) deben ser autoexplicativas
- Numera las figuras y tablas de forma consecutiva
- Usa formatos vectoriales (PDF, SVG) cuando sea posible
- Mantén consistencia en el estilo visual

1.11.5. Revisión final

Antes de entregar, verifica:

1. Ortografía y gramática (usa correctores automáticos)
 2. Referencias cruzadas funcionan correctamente
 3. Bibliografía está completa y correctamente formateada
 4. Índices están actualizados
 5. Márgenes y formato cumplen la normativa
 6. No hay notas \todo pendientes
-

2. Marco Teórico (Ejemplos de listas)

Este capítulo presenta los fundamentos teóricos necesarios y demuestra las diferentes formas de crear listas en \LaTeX .

2.1. Listas básicas

Existen tres tipos principales de listas en \LaTeX : `itemize` (viñetas), `enumerate` (numeradas) y `description` (definiciones).

2.1.1. Listas con viñetas (`itemize`)

Lista con viñetas

```
1 \begin{itemize}
2   \item Primer elemento
3   \item Segundo elemento
4   \item Tercer elemento
5 \end{itemize}
```

El resultado es:

- Primer elemento de la lista
- Segundo elemento de la lista
- Tercer elemento de la lista

2.1.2. Listas numeradas (`enumerate`)

Lista numerada

```
1 \begin{enumerate}
2   \item Primer paso del proceso
3   \item Segundo paso del proceso
4   \item Tercer paso del proceso
5 \end{enumerate}
```

El resultado es:

1. Primer paso del proceso
2. Segundo paso del proceso
3. Tercer paso del proceso

2.1.3. Listas anidadas

Las listas pueden anidarse hasta varios niveles:

- Ingeniería Informática
 - Mención en Computación
 - Mención en Ingeniería del Software
 - Mención en Sistemas de Información
- Ingeniería Multimedia
 - Mención en Creación y Ocio Digital
 - Mención en Gestión de Contenidos
- Ingeniería en Sonido e Imagen

También con números:

1. Fase de análisis
 - a) Recopilación de requisitos
 - b) Análisis de viabilidad
 - c) Documentación inicial
2. Fase de diseño
 - a) Diseño arquitectónico
 - b) Diseño detallado
3. Fase de implementación

2.2. Listas de definición

Las listas de descripción son útiles para glosarios y definiciones:

Lista de definiciones

```
1 \begin{description}
2   \item[Término 1:] Definición del primer término.
3   \item[Término 2:] Definición del segundo término.
4 \end{description}
```

LaTeX: Sistema de composición de textos orientado a la creación de documentos científicos y técnicos de alta calidad tipográfica.

TFG: Trabajo Fin de Grado. Proyecto final que los estudiantes de grado deben realizar y defender para obtener su título universitario.

TFM: Trabajo Fin de Máster. Proyecto de investigación o aplicación que los estudiantes de máster deben completar.

EPS: Escuela Politécnica Superior de la Universidad de Alicante.

2.2.1. Listas de definición anidadas

Software de diseño: Herramientas utilizadas en el proyecto.

Ventajas:

- Permite realizar simulaciones precisas
- Interfaz gráfica intuitiva
- Exportación a múltiples formatos

Inconvenientes:

- Curva de aprendizaje pronunciada
- Requiere licencia comercial
- Alto consumo de recursos

2.3. Estado del arte

A continuación se presenta una revisión de los trabajos más relevantes en el área.

2.3.1. Trabajos previos

Según LaTeX Project (2024), el estado actual de la investigación en sistemas de composición tipográfica indica que \LaTeX sigue siendo el estándar para documentos científicos.

En Overleaf, 2024 se propone un enfoque basado en edición colaborativa en la nube que ha facilitado enormemente la adopción de \LaTeX en entornos académicos.

2.3.2. Tecnologías relacionadas

Las principales tecnologías utilizadas en este ámbito incluyen:

1. **LuaLaTeX:** Motor de composición moderno que permite:

- Uso directo de fuentes del sistema
- Programación en Lua dentro del documento
- Soporte nativo de Unicode

2. **Overleaf:** Plataforma de edición online que ofrece:

- Compilación en la nube
- Colaboración en tiempo real
- Control de versiones integrado

3. **Git:** Sistema de control de versiones útil para:

- Seguimiento de cambios en el documento
- Trabajo en equipo
- Recuperación de versiones anteriores

2.4. Herramientas utilizadas

En este trabajo se han utilizado las siguientes herramientas:

Tabla 2.1: Herramientas utilizadas en el desarrollo

Herramienta	Versión	Uso
TeX Live	2024	Distribución LaTeX
LuaLaTeX	1.18	Motor de compilación
VS Code	1.85	Editor de texto
Git	2.43	Control de versiones
Pygments	2.17	Resaltado de código

La Tabla 2.1 muestra las principales herramientas empleadas en la elaboración de este documento.

3. Objetivos (Ejemplos de tablas)

Este capítulo presenta los objetivos del trabajo y demuestra las diferentes formas de crear tablas en \LaTeX .

3.1. Objetivo general

El objetivo general de este trabajo es...

3.2. Objetivos específicos

1. **OE1:** Analizar el estado actual de...
2. **OE2:** Diseñar una arquitectura que permita...
3. **OE3:** Implementar los componentes necesarios para...
4. **OE4:** Evaluar el rendimiento del sistema mediante...
5. **OE5:** Documentar el proceso de desarrollo y los resultados obtenidos.

3.3. Tablas en \LaTeX

Las tablas son elementos fundamentales en cualquier documento técnico. A continuación se muestran diferentes ejemplos.

3.3.1. Tabla simple

Código de tabla simple

```
1 \begin{table}[H]
2   \centering
3   \begin{tabular}{lcc}
4     & Columna A & Columna B \\
5     \hline
6     Fila 1 & Dato 1A & Dato 1B \\
7     Fila 2 & Dato 2A & Dato 2B \\
8     Fila 3 & Dato 3A & Dato 3B \\
9     \hline
10  \end{tabular}
11  \caption{Ejemplo de tabla simple}
12  \label{tab:simple}
13 \end{table}
```

	Columna A	Columna B
Fila 1	Dato 1A	Dato 1B
Fila 2	Dato 2A	Dato 2B
Fila 3	Dato 3A	Dato 3B

Tabla 3.1: Ejemplo de tabla simple

3.3.2. Tabla con booktabs

El paquete `booktabs` proporciona líneas más profesionales:

Tabla 3.2: Tabla con estilo booktabs

Concepto	Valor	Unidad
Velocidad máxima	120	km/h
Consumo medio	5.5	L/100km
Potencia	150	CV
Par motor	250	Nm

3.3.3. Tabla con columnas de ancho fijo

Puedes especificar el ancho de las columnas usando `p{ancho}`, `L{ancho}`, `C{ancho}` o `R{ancho}`:

Tabla 3.3: Parámetros de posicionamiento de elementos flotantes

Parámetro	Significado
<code>h</code>	Sitúa el elemento <i>preferentemente</i> en la posición actual del texto
<code>t</code>	Sitúa el elemento en la parte superior de la página
<code>b</code>	Sitúa el elemento en la parte inferior de la página
<code>p</code>	Sitúa el elemento en una página dedicada solo a flotantes
<code>H</code>	Fuerza la posición exacta (requiere paquete <code>float</code>)

3.3.4. Tabla con multicolumna y multifila

Tabla 3.4: Ejemplo de tabla con celdas combinadas

Distribución de recursos			
Fase	Recursos		
	Personal	Tiempo (h)	Coste (€)
Análisis	2	40	2.000
Diseño	3	80	4.800
Implementación	4	160	9.600
Pruebas	2	60	3.000
Total	–	340	19.400

3.3.5. Tabla con colores alternados

Usando el paquete `xcolor` con la opción `table`:

Tabla 3.5: Especificaciones técnicas del sistema

Componente	Especificación	Requisito
Procesador	Intel i7 / AMD Ryzen 7	Mínimo
Memoria RAM	16 GB DDR4	Recomendado
Almacenamiento	512 GB SSD	Mínimo
GPU	NVIDIA RTX 3060	Recomendado
Sistema Operativo	Linux / Windows 11	Compatible

3.3.6. Tabla larga (longtable)

Para tablas que ocupan varias páginas, se usa el entorno `longtable`:

Tabla 3.6: Lista de requisitos del sistema

ID	Tipo	Descripción
RF01	Funcional	El sistema debe permitir el registro de usuarios mediante correo electrónico
RF02	Funcional	Los usuarios deben poder iniciar sesión con sus credenciales
RF03	Funcional	El sistema debe generar informes en formato PDF
RF04	Funcional	Se debe implementar un sistema de notificaciones
RF05	Funcional	Los datos deben poder exportarse en formato CSV

Continúa en la siguiente página

Tabla 3.6 – Continuación

ID	Tipo	Descripción
RNF01	No funcional	El tiempo de respuesta no debe superar los 2 segundos
RNF02	No funcional	El sistema debe soportar 100 usuarios concurrentes
RNF03	No funcional	La disponibilidad debe ser del 99.5%
RNF04	No funcional	La interfaz debe ser accesible según WCAG 2.1
RNF05	No funcional	Los datos sensibles deben cifrarse con AES-256

3.4. Generadores de tablas

Crear tablas manualmente puede ser tedioso. Se recomienda usar generadores online:

- <https://www.tablesgenerator.com/> – Generador visual muy completo
- <https://www.latex-tables.com/> – Alternativa sencilla
- Excel2LaTeX – Plugin para Microsoft Excel

3.5. Forzar posición de tablas

Para forzar que una tabla aparezca en un lugar específico, se puede usar:

Forzar posición

```

1 % Opción 1: Usar FloatBarrier
2 \begin{table}[H]
3   ...
4 \end{table}
5 \FloatBarrier % Fuerza que la tabla aparezca antes
6
7 % Opción 2: Usar H (requiere paquete float)
8 \begin{table}[H]
9   ...
10 \end{table}

```

3.6. Tablas con notas al pie

Para añadir notas al pie dentro de una tabla, se usa el paquete `threeparttable`:

Tabla 3.7: Comparativa de tecnologías web con notas

Framework	Rendimiento ^a	Popularidad ^b	Curva
React	92	1st	Media
Vue.js	89	3rd	Baja
Angular	85	2nd	Alta
Svelte	96	5th	Baja

^a Puntuación en benchmarks de Lighthouse (0-100).
^b Posición según encuesta Stack Overflow 2024.

3.7. Tabla con celdas de colores personalizados

Se pueden colorear celdas individuales usando `\cellcolor`:

Tabla 3.8: Matriz de evaluación de riesgos

Impacto \ Probabilidad					
	Muy bajo	Bajo	Medio	Alto	Muy alto
Muy alto	M	A	MA	MA	MA
Alto	B	M	A	MA	MA
Medio	B	B	M	A	MA
Bajo	MB	B	B	M	A
Muy bajo	MB	MB	B	B	M

MB = Muy Bajo, B = Bajo, M = Medio, A = Alto, MA = Muy Alto

3.8. Tabla de ancho completo

Para que una tabla ocupe todo el ancho disponible, se usa `tabularx`:

Tabla 3.9: Cronograma del proyecto usando todo el ancho

Fase	Descripción de actividades	M1	M2	M3	M4
Análisis	Recopilación de requisitos, entrevistas con stakeholders y estudio de viabilidad	•			
Diseño	Arquitectura del sistema, diseño de base de datos y prototipado de interfaz	•	•		
Desarrollo	Implementación del backend, frontend y pruebas unitarias		•	•	
Testing	Pruebas de integración, UAT y corrección de errores			•	•
Despliegue	Configuración de servidores, migración y formación de usuarios				•

4. Metodología (Ejemplos de figuras)

Este capítulo describe la metodología seguida en el desarrollo del trabajo y presenta ejemplos de cómo insertar figuras en \LaTeX .

4.1. Metodología de trabajo

Para el desarrollo de este trabajo se ha seguido una metodología ágil basada en iteraciones cortas:

1. **Planificación:** Definición de objetivos y alcance
2. **Análisis:** Estudio del problema y requisitos
3. **Diseño:** Arquitectura y diseño detallado
4. **Implementación:** Desarrollo del código
5. **Pruebas:** Verificación y validación
6. **Documentación:** Redacción de la memoria

4.2. Inserción de figuras

Las figuras en \LaTeX son elementos flotantes. Esto significa que \LaTeX decide su ubicación óptima para mejorar la maquetación del documento.

4.2.1. Figura simple

Código para insertar una figura

```
1 \begin{figure}[H]
2   \centering
3   \includegraphics[width=0.6\textwidth]{ruta/imagen}
4   \caption{Descripción de la figura}
5   \label{fig:etiqueta}
6 \end{figure}
```

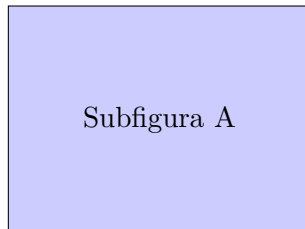


Figura 4.1: Ejemplo de figura simple (placeholder)

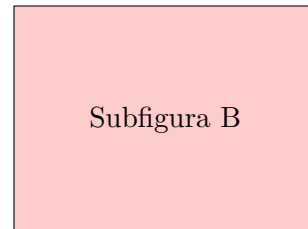
Para referenciar la figura en el texto: “como se muestra en la Figura 4.1”.

4.2.2. Subfiguras

Cuando necesitas mostrar varias imágenes relacionadas:



(a) Primera variante



(b) Segunda variante

Figura 4.2: Ejemplo de subfiguras horizontales

Puedes referenciar subfiguras individuales: Figura 4.2a y Figura 4.2b, o el conjunto: Figura 4.2.

4.2.3. Múltiples imágenes en tabla

Otra forma de organizar varias imágenes:

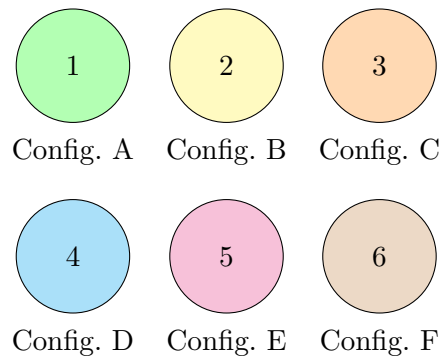


Figura 4.3: Matriz de configuraciones experimentales

4.3. Planificación temporal

La planificación temporal del proyecto se muestra en la Figura 4.4.

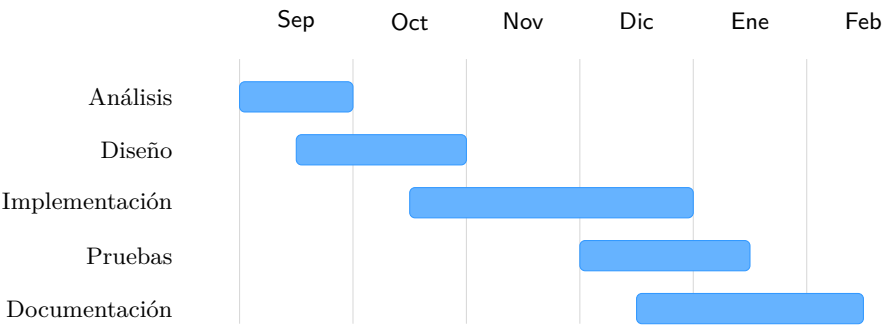


Figura 4.4: Planificación temporal del proyecto (diagrama de Gantt simplificado)

4.4. Diagramas con TikZ

TikZ permite crear diagramas directamente en \LaTeX :

4.4.1. Diagrama de flujo

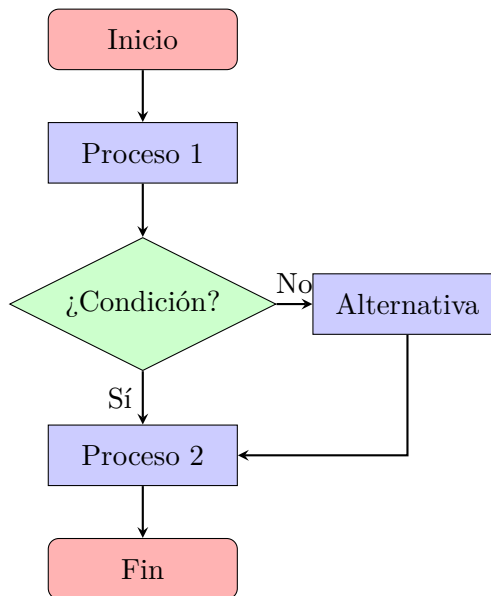


Figura 4.5: Diagrama de flujo del algoritmo principal

4.4.2. Diagrama de bloques

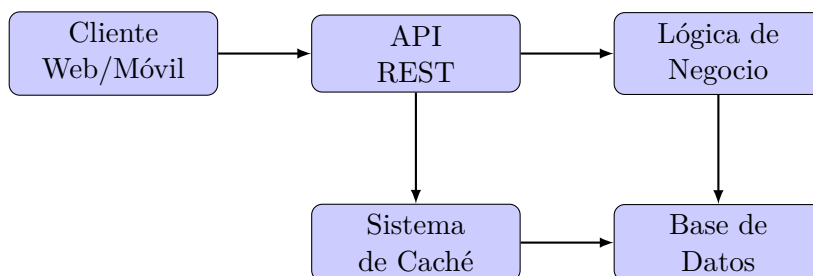


Figura 4.6: Arquitectura del sistema

4.5. Recursos utilizados

4.5.1. Recursos hardware

- Ordenador portátil con procesador Intel i7, 16GB RAM
- Servidor de desarrollo con 32GB RAM
- Dispositivos móviles para pruebas

4.5.2. Recursos software

- Sistema operativo: Linux Ubuntu 24.04 LTS
- Entorno de desarrollo: Visual Studio Code
- Control de versiones: Git y GitHub
- Compilador \LaTeX : LuaLaTeX (TeX Live 2024)

4.6. Gestión del proyecto

Para la gestión del proyecto se han utilizado las siguientes herramientas:

- **GitHub Projects:** Para la gestión de tareas y seguimiento del progreso mediante tableros Kanban.
 - **Git:** Para el control de versiones del código y la documentación.
 - **Discord/Slack:** Para la comunicación con el tutor.
-

5. Desarrollo: Estilos de Código

Este capítulo demuestra los estilos de cajas de código disponibles en esta plantilla, basados en el diseño de Visual Studio Code.

5.1. Estilo VS Code Light

El estilo VS Code Light imita la apariencia del editor Visual Studio Code en su tema claro.

5.1.1. Python

Python

```
1 def fibonacci(n):
2     if n <= 1:
3         return n
4     return fibonacci(n-1) + fibonacci(n-2)
5
6 for i in range(10):
7     print(f"F({i}) = {fibonacci(i)}")
```

5.1.2. Sin numeración de líneas

Usando el sufijo NN (No Numbers) se elimina la numeración:

Python

```
mensaje = "Hola, mundo"
print(mensaje)
```

5.1.3. JavaScript

JavaScript

```
1 function validateEmail(email) {
2     const regex = /^[a-z]+@[a-z]+\.[a-z]+$/;
3     return regex.test(email);
4 }
5
6 async function fetchData(url) {
7     const response = await fetch(url);
```

```
8     return response.json();  
9 }
```

5.1.4. Java

 Java

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

5.2. Estilo VS Code Dark


El tema oscuro de VS Code, ideal para documentación técnica o presentaciones.

5.2.1. Python Dark

 Python

```
1 import numpy as np  
2  
3 def matrix_multiply(A, B):  
4     return np.dot(A, B)  
5  
6 A = np.array([[1, 2], [3, 4]])  
7 B = np.array([[5, 6], [7, 8]])  
8 result = matrix_multiply(A, B)
```

5.2.2. JavaScript Dark

 JavaScript

```
1 const express = require('express');  
2 const app = express();  
3  
4 app.get('/api/users', async (req, res) => {  
5     const users = await User.find();  
6     res.json(users);  
7 });  
8  
9 app.listen(3000);
```

5.3. Lenguajes Web

5.3.1. HTML

HTML

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Mi Aplicación</title>
6 </head>
7 <body>
8   <h1>Bienvenido</h1>
9 </body>
10 </html>
```

5.3.2. CSS

CSS

```
1 .container {
2   max-width: 1200px;
3   margin: 0 auto;
4   display: grid;
5   grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
6   gap: 1rem;
7 }
8
9 .card:hover {
10   transform: translateY(-5px);
11 }
```

5.4. Lenguajes de Datos

5.4.1. SQL

SQL

```
1 SELECT
2   usuarios.nombre,
3   COUNT(pedidos.id) as total_pedidos
4 FROM usuarios
5 LEFT JOIN pedidos ON usuarios.id = pedidos.usuario_id
6 GROUP BY usuarios.id
7 ORDER BY total_pedidos DESC
8 LIMIT 10;
```


5.5. Lenguajes de Sistemas

5.5.1. C++

 C++

```
1 include <iostream>
2 \#include <vector>
3
4 int main() {
5     std::vector<int> nums = {5, 2, 8, 1, 9};
6     for (const auto& n : nums) {
7         std::cout << n << " ";
8     }
9     return 0;
10 }
```

5.5.2. Rust

 Rust

```
1 fn main() {
2     let mensaje = "Hola desde Rust";
3     println!("{}", mensaje);
4
5     let numeros = vec![1, 2, 3, 4, 5];
6     for n in numeros {
7         println!("{}", n);
8     }
9 }
```

5.5.3. Go

 Go

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Hola desde Go")
7
8     nums := []int{1, 2, 3, 4, 5}
9     for _, n := range nums {
10         fmt.Println(n)
11     }
12 }
```


5.6. Otros Lenguajes

5.6.1. PHP

 PHP

```
1 <?php
2 class Usuario {
3     private $nombre;
4
5     public function __construct($nombre) {
6         $this->nombre = $nombre;
7     }
8
9     public function saludar() {
10        return "Hola, " . $this->nombre . "!";
11    }
12 }
13
14 $usuario = new Usuario("Juan");
15 echo $usuario->saludar();
16 ?>
```

5.6.2. Ruby

 Ruby

```
1 class Persona
2     attr_accessor :nombre, :edad
3
4     def initialize(nombre, edad)
5         @nombre = nombre
6         @edad = edad
7     end
8
9     def saludar
10        puts "Hola, soy " + @nombre
11    end
12 end
13
14 persona = Persona.new("Maria", 25)
15 persona.saludar
```

5.7. DevOps y Configuración

5.7.1. Docker

Dockerfile

```
1 FROM node:18-alpine
2
3 WORKDIR /app
4
5 COPY package*.json ./
6 RUN npm ci --only=production
7
8 COPY . .
9
10 EXPOSE 3000
11
12 CMD ["node", "server.js"]
```

5.8. Entorno Genérico

El entorno `codigo{lenguaje}` permite usar cualquier lenguaje soportado por Pygments:

swift

```
1 import Foundation
2
3 struct Persona {
4     let nombre: String
5     let edad: Int
6
7     func saludar() -> String {
8         return "Hola, soy " + nombre
9     }
10 }
11
12 let persona = Persona(nombre: "Carlos", edad: 30)
13 print(persona.saludar())
```

5.8.1. Versión Dark del Genérico

kotlin

```
1 data class Usuario(val nombre: String, val email: String)
2
3 fun main() {
4     val usuarios = listOf(
```

```

5      Usuario("Ana", "ana@mail.com"),
6      Usuario("Luis", "luis@mail.com")
7  )
8
9  usuarios.forEach { println(it) }
10 }
```

5.9. Resumen de Entornos Disponibles

Tabla 5.1: Entornos de código con icono disponibles














Entorno	Lenguaje	Icono	Dark
pythoncode	Python		pythoncodeDark
jscode	JavaScript		jscodeDark
javacode	Java		javacodeDark
cppcode	C++		cppcodeDark
rustcode	Rust		rustcodeDark
gocode	Go		gocodeDark
phpcode	PHP		phpcodeDark
rubycode	Ruby		rubycodeDark
htmlcode	HTML		htmlcodeDark
csscode	CSS		csscodeDark
sqlcode	SQL		sqlcodeDark
bashcode	Bash		bashcodeDark
dockercode	Dockerfile		dockercodeDark

Tabla 5.2: Sufijos disponibles para cada entorno

Sufijo	Descripción	Ejemplo
(ninguno)	Light con números	pythoncode
NN	Light sin números	pythoncodeNN
Dark	Dark con números	pythoncodeDark
DarkNN	Dark sin números	pythoncodeDarkNN

Tabla 5.3: Entornos genéricos para cualquier lenguaje

Entorno	Descripción
<code>codigo{lang}</code>	Light con números
<code>codigoNN{lang}</code>	Light sin números
<code>codigoDark{lang}</code>	Dark con números
<code>codigoDarkNN{lang}</code>	Dark sin números

6. Componentes Especializados

Este capítulo demuestra el sistema de componentes especializados disponibles en la plantilla. Los componentes se cargan de forma modular para optimizar el tiempo de compilación.

Para usar los componentes, añade en el preámbulo del documento:


```
\usepackage[all]{eps-componentes} % Todos los componentes  
% o bien, solo los que necesites:  
\usepackage[software,telecom]{eps-componentes}
```


6.1. Componentes Comunes


Los componentes comunes se cargan automáticamente y están disponibles para todas las disciplinas.


6.1.1. Cajas de Información


Cajas para destacar diferentes tipos de información:

 Esta es una caja de información general. Úsala para notas importantes o datos relevantes que el lector debe conocer.

 Esta es una caja de éxito. Indica que algo se ha completado correctamente o muestra una buena práctica.

 Esta es una caja de advertencia. Alerta sobre algo que requiere atención pero no es crítico.

 Esta es una caja de peligro. Indica un error crítico o algo que debe evitarse a toda costa.

 Esta es una caja de consejo. Ofrece sugerencias útiles o trucos para mejorar el trabajo.

 Esta es una caja de nota. Para información adicional o comentarios secundarios.

6.1.2. Cajas con Título

Cajas que incluyen un título personalizado:

Título Personalizado

Esta caja permite definir un título personalizado para el contenido. Es útil para destacar secciones específicas.

Algoritmo

Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Ejemplo de uso

Aquí se muestra cómo aplicar el concepto explicado anteriormente en un caso práctico.

Importante

Este contenido es especialmente importante y no debe pasarse por alto durante la lectura del documento.

6.1.3. Listas Especiales

Diferentes tipos de listas para organizar información:

6.1.3.0.1. Checklist:

- ☒ Revisar documentación
- ☒ Implementar funcionalidad básica
- ☐ Realizar pruebas unitarias
- ☐ Desplegar en producción

6.1.3.0.2. Pros y Contras:

- ☒ Fácil de implementar
 - ☒ Bien documentado
 - ☒ Comunidad activa
 - ☐ Curva de aprendizaje inicial
 - ☐ Requiere configuración adicional
-

6.1.3.0.3. Lista de Pasos:

- 1 Descargar el código fuente del repositorio
- 2 Instalar las dependencias necesarias
- 3 Configurar las variables de entorno
- 4 Ejecutar el script de inicialización
- 5 Verificar que todo funciona correctamente

6.1.4. Badges e Indicadores

Etiquetas y badges para marcar estados:

6.1.4.0.1. Badges personalizados:

Estable Completado En revisión Crítico Beta

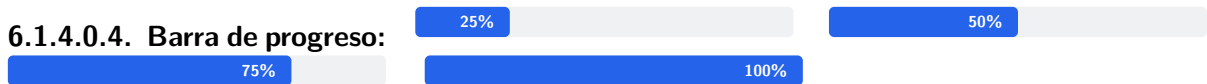
6.1.4.0.2. Badges predefinidos:

NUEVO EN PROGRESO OBSOLETO BETA REQUERIDO OPCIONAL

6.1.4.0.3. Versiones:

v1.0.0 v2.1.0 v3.0.0-beta

6.1.4.0.4. Barra de progreso:



6.1.4.0.5. Rating con estrellas:



6.1.4.0.6. Indicador de nivel:



6.1.5. Tarjetas de Información






María López
Project Manager

Coordinación de equipos y gestión ágil de proyectos.




1,234

Usuarios




98.5%

Uptime



45ms

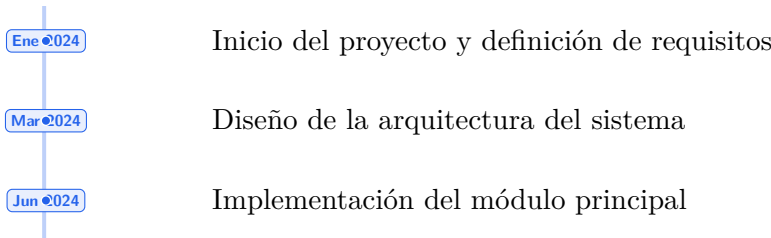
Latencia

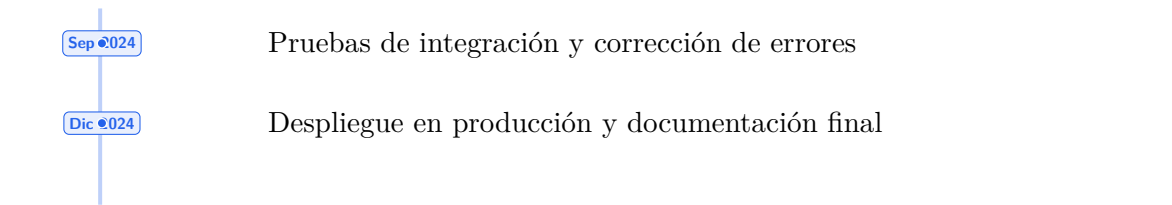


12

Issues

6.1.6. Timeline





6.1.7. Comparativas

Criterio	Opción A	Opción B
Rendimiento	Alto	Medio
Facilidad de uso	Media	Alta
Coste	Bajo	Medio
Soporte	Comunidad	Comercial

6.1.8. Citas Destacadas

“ La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación abraza el mundo entero. ”

— Albert Einstein

6.2. Componentes de Software

Componentes específicos para informática e ingeniería de software. Cargar con: `\usepackage[software]{componentes}`

6.2.1. API REST Endpoints

GET

/api/users

Obtiene la lista de todos los usuarios registrados en el sistema. Requiere autenticación mediante token JWT.

Body (application/json):

"page": 1, "limit": 10, "filter": "active"

Respuesta 200:

"users": ["id": 1, "name": "Juan", "id": 2, "name": "María"], "total": 150

Respuesta 401:

"error": "Token no válido"

POST **/api/users**

Crea un nuevo usuario en el sistema.

Respuesta **201**:

```
"id": 3, "message": "Usuario creado"
```

6.2.1.0.1. Métodos HTTP disponibles: **GET** **POST** **PUT** **PATCH** **DELETE**

6.2.2. Terminal y Consola

```
Instalación de dependencias  
  
$ npm install $ npm run build $ npm test # systemctl restart nginx  
# journalctl -u nginx -f
```

6.2.3. Árbol de Directorios

Estructura del Proyecto

```
src/  
├── assets/  
│   └── logo.png  
├── components/  
│   ├── Header.js  
│   └── Footer.js  
├── App.js  
├── index.js  
├── package.json  
└── README.md
```

6.2.4. Clases UML

Usuario

```
- id: int  
- nombre: string  
- email: string  
password: string  
  
+ getNombre(): string  
+ setNombre(n: string): void  
+ validar(): boolean
```

«interface»
IAutenticable

+ login(user: string, pass: string): boolean

+ logout(): void

+ isAuthenticated(): boolean

6.2.5. Tabla de Requisitos

ID	Prioridad	Estado	Descripción
RF-001	Alta	✓ Completado	El sistema debe permitir el registro de usuarios
RF-002	Alta	✓ Completado	Los usuarios pueden modificar su perfil
RF-003	Media	🔄 En progreso	Implementar sistema de notificaciones
RF-004	Baja	• Pendiente	Exportar datos a formato PDF
RF-005	Media	• Pendiente	Integración con API externa

6.2.6. Esquema de Base de Datos

🗄️ usuarios

Campo	Tipo	Restricciones
🔑 id	INT	AUTO_INCREMENT
nombre	VARCHAR(100)	NOT NULL
email	VARCHAR(255)	UNIQUE NOT NULL
created_at	TIMESTAMP	DEFAULT NOW()
🔗 rol_id	INT	FK → roles.id

🗄️ roles

Campo	Tipo	Restricciones
🔑 id	INT	AUTO_INCREMENT
nombre	VARCHAR(50)	NOT NULL
permisos	JSON	NULL

6.2.7. Git y Control de Versiones


Historial de commits del proyecto:



🔗 a1b2c3d feat: Añadir sistema de autenticación

👤 Juan 📅 2024-01-15


🔗 e4f5g6h fix: Corregir error en validación de email


👤 María 📅 2024-01-20

 **i7j8k9l** docs: Actualizar README con instrucciones

 Pedro  2024-02-01

Ramas:


 main


 develop

 feature/login


Tags:

 v1.0.0

 v1.1.0

 v2.0.0-beta

6.2.8. Logs y Métricas

 Logs del servidor

2024-01-15 10:30:00

[INFO]

Servidor iniciado en puerto 3000

2024-01-15 10:30:01

[DEBUG]

Conexión a base de datos establecida

2024-01-15 10:35:22

[WARN]

Intento de acceso sin autenticación

2024-01-15 11:02:45

[ERROR]

Timeout en consulta a API externa

Tiempo de respuesta

45

eps-success ms

Uso de CPU

78

eps-warning %

Memoria

2.4

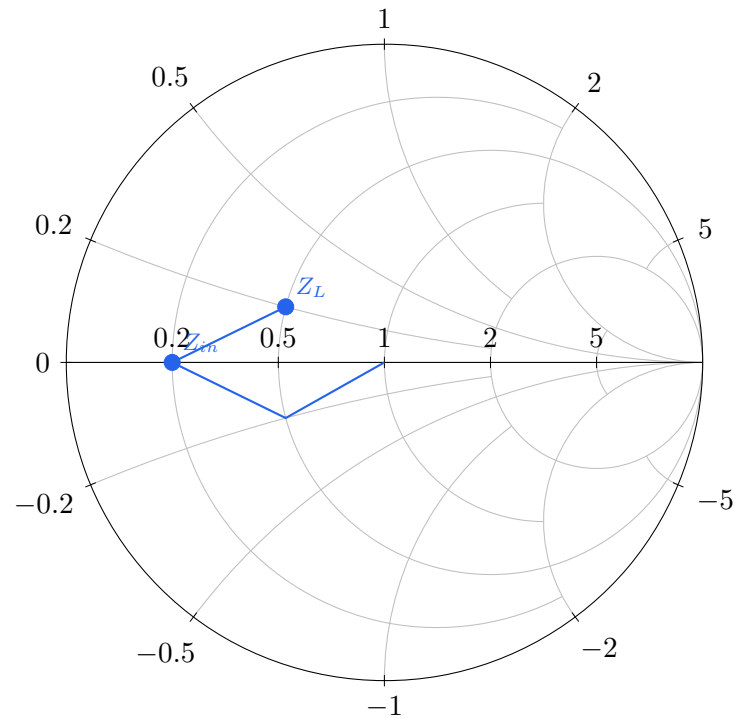
eps-info GB

6.3. Componentes de Telecomunicaciones

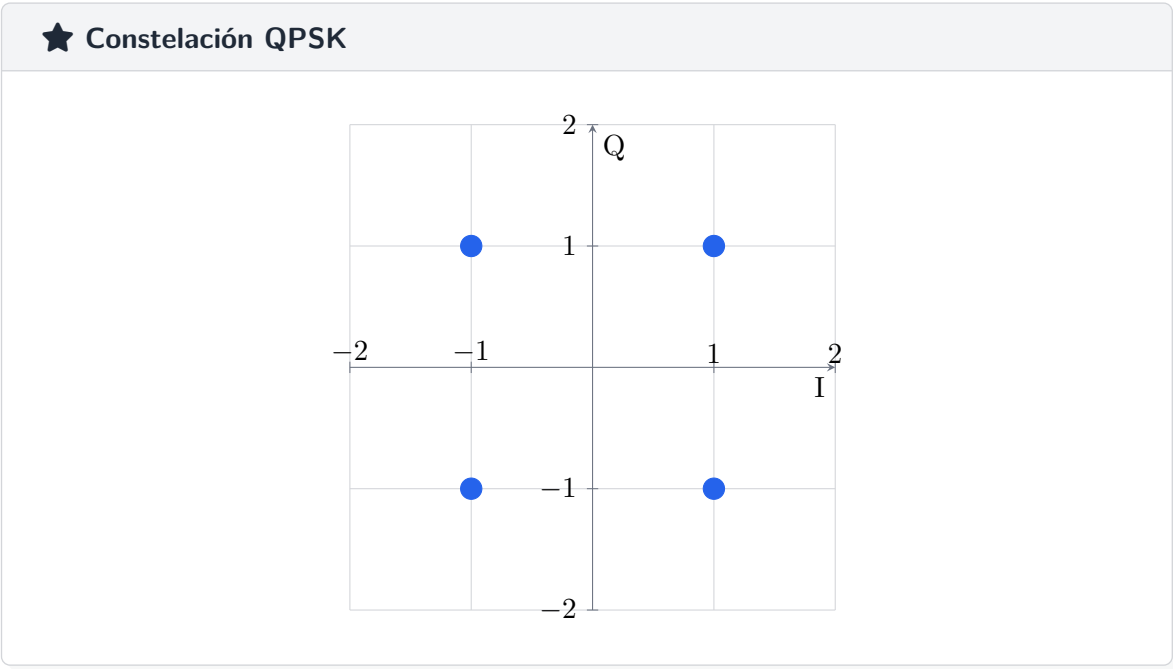
Componentes para telecomunicaciones, electrónica y sistemas de comunicación. Cargar con:
`\usepackage[telecom]{eps-componentes}`

6.3.1. Carta de Smith

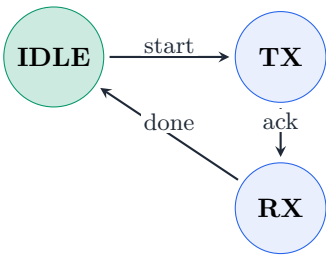
La carta de Smith es una herramienta gráfica para resolver problemas de adaptación de impedancias en líneas de transmisión.

'A' Adaptación de Impedancias**6.3.2. Diagrama de Constelación**

Representación de esquemas de modulación digital:



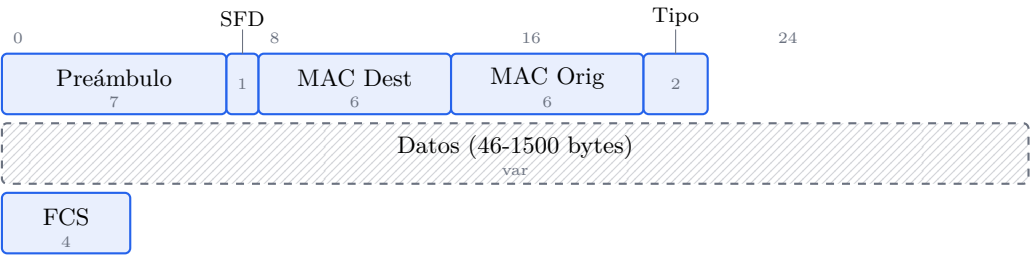
6.3.3. Máquina de Estados Finitos



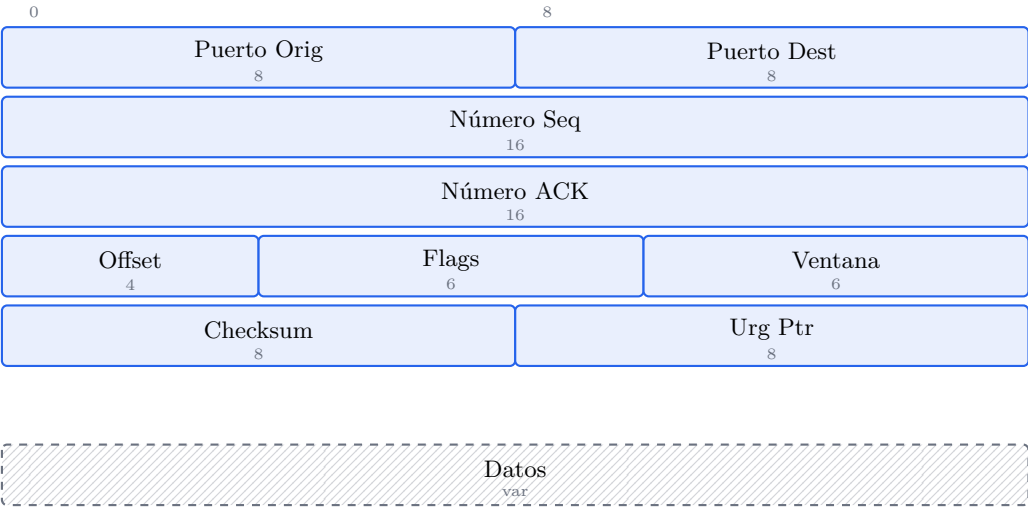
6.3.4. Tramas de Protocolo

Los campos con tamaño menor a 3 unidades muestran automáticamente su etiqueta fuera del campo para evitar recortes de texto. Este umbral es configurable con `\def\framenarrowthreshold{val}`

Trama Ethernet IEEE 802.3



Segmento TCP (cabecera simplificada)

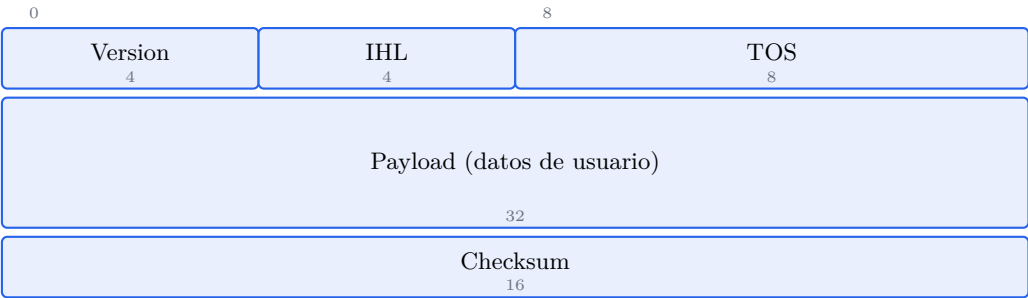


Nuevas funcionalidades de protocolframe

Las siguientes funcionalidades amplían las capacidades del entorno protocolframe.

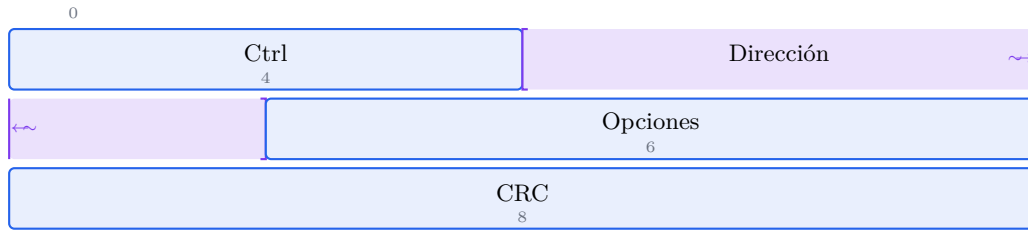
Campo multi-fila

Para datos que ocupan varias filas se usa \framefieldmulti{filas}{bits}{etiqueta}:



Campos con continuación

Para campos que se dividen entre filas, se usan \framefieldstart y \framefieldend:



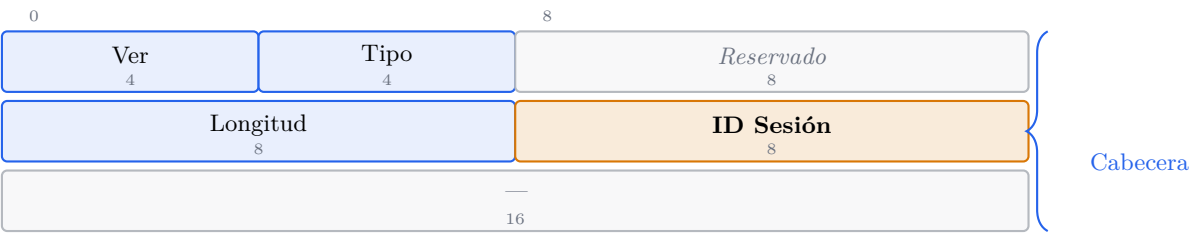
Etiqueta externa forzada

Con \framefieldext{bits}{etiqueta} la etiqueta siempre aparece fuera, útil para campos estrechos:



Campos reservados y agrupación

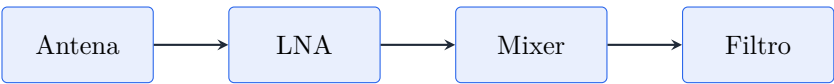
Se dispone de campos reservados (`\framefieldreserved`) y llaves de agrupación (`\framewordgroup`):



6.3.5. Parámetros S

Parámetros S			
Parámetro	Valor	Unidad	Condiciones
S_{11}	-18.5	dB	Pérdidas de retorno entrada
S_{21}	-0.25	dB	Pérdidas de inserción
S_{12}	-45.2	dB	Aislamiento inverso
S_{22}	-22.1	dB	Pérdidas de retorno salida

6.3.6. Diagramas de Bloques



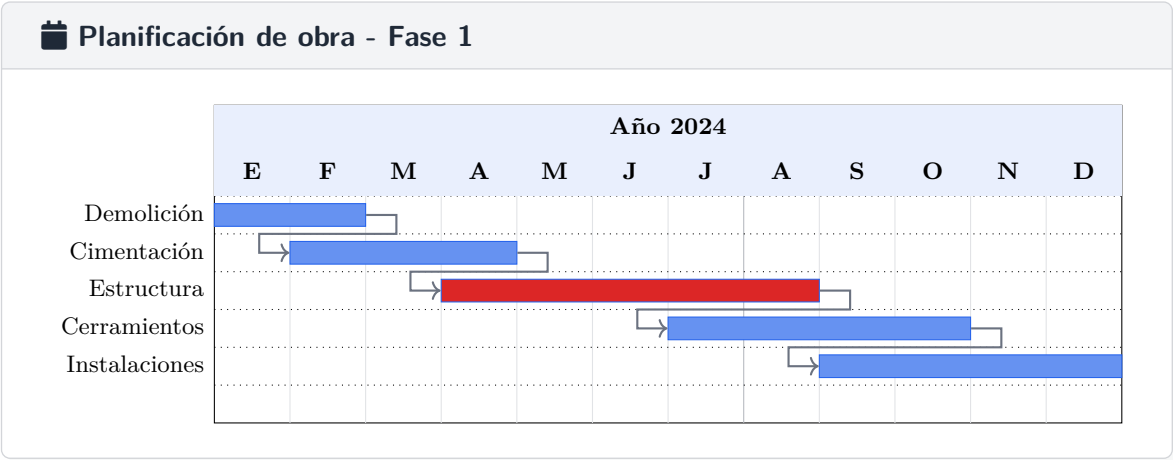
6.3.7. Diagramas de Temporización



6.4. Componentes de Arquitectura

Componentes para arquitectura, ingeniería civil y construcción. Cargar con: `\usepackage[arquitectura]{componentes}`

6.4.1. Diagrama de Gantt



6.4.2. Fichas Técnicas de Materiales

Ficha técnica: Hormigón HA-25/B/20/Ila

Propiedad	Valor
Resistencia característica	25 MPa
Consistencia	Blanda (6-9 cm)
Tamaño máximo árido	20 mm
Ambiente de exposición	Ila - Humedad alta
Relación agua/cemento máx.	0.60
Contenido mínimo cemento	275 kg/m³
Recubrimiento nominal	30 mm

Acero B500S

Límite elástico	500 MPa
Resistencia tracción	550 MPa

Ladrillo cerámico

Resistencia	10 N/mm²
Absorción	< 15%

6.4.3. Presupuestos

Presupuesto

Código	Descripción	Ud.	Medición	P.U.	Importe
Capítulo 1: MOVIMIENTO DE TIERRAS					
1.1	Excavación en zanjas	m³	120	40 €	4800,00 €
1.2	Transporte a vertedero	m³	180	15 €	2700,00 €

Continúa en la siguiente página...

Código	Descripción	Ud.	Medición	P.U.	Importe
1.3	Compactación de tierras	m ²	250	4 €	1000,00 €
Capítulo 2: CIMENTACIÓN					
2.1	Hormigón de limpieza	m ³	15	80 €	1200,00 €
2.2	Zapatas de hormigón armado	m ³	42	300 €	12.600,00 €
Capítulo 3: ESTRUCTURA					
3.1	Pilares de hormigón	m ³	28	400 €	11.200,00 €
3.2	Forjado unidireccional	m ²	320	120 €	38.400,00 €
Total presupuesto:					74.660 €

6.4.4. Normativa Aplicable

Normativa aplicable

CTE DB-SE	Seguridad Estructural - Bases de cálculo	(2019)
CTE DB-SE-AE	Acciones en la edificación	(2019)
CTE DB-SE-C	Cimientos	(2019)
EHE-08 (2008)	Instrucción de Hormigón Estructural	
UNE-EN 1992-1-1	Eurocódigo 2: Proyecto de estructuras de hormigón	(2013)
NCSE-02 (2002)	Norma de Construcción Sismorresistente	

6.4.5. Control de Calidad

Control de calidad

Ensayo/Control	Resultado	Límite	Estado
Hormigón HA-25: Resistencia compresión	28,5 MPa	≥ 25 MPa	✓ Conforme
Acero B500S: Límite elástico	512 MPa	≥ 500 MPa	✓ Conforme
Soldadura: Inspección visual	Defecto leve	Sin defectos	⚠ Pendiente
Compactación: Densidad relativa	94%	≥ 95%	✗ No conforme

6.4.6. Etiquetas Energéticas



6.4.7. Certificaciones

ISO 9001 | Certificado

ISO 14001 | Certificado

ISO 45001 | Certificado

CE | Marcado CE

UNE 12345 | Norma española

6.5. Componentes de Química

Componentes para química, ciencia de materiales e ingeniería química. Cargar con: `\usepackage[quimica]{componentes}`

6.5.1. Reacciones Químicas

Combustión completa del metano

$$\text{CH}_4 + 2 \text{O}_2 \longrightarrow \text{CO}_2 + 2 \text{H}_2\text{O}$$

800 °C 1 atm Pt

Sustitución nucleofílica SN2

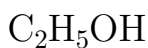
El mecanismo SN2 ocurre en un solo paso concertado donde el nucleófilo ataca por el lado opuesto al grupo saliente.

$$\text{HO}^- + \text{CH}_3\text{Br} \longrightarrow \text{CH}_3\text{OH} + \text{Br}^-$$

6.5.2. Fichas de Compuestos

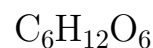
Ácido Sulfúrico (H₂SO₄)

H ₂ SO ₄	Masa molar	98,079 g/mol
	Densidad	1,84 g/cm ³
	Punto de fusión	10 °C
	Punto de ebullición	337 °C
	Solubilidad en agua	Miscible
	pKa	-3 (ácido fuerte)

**Etanol**

46,07 g/mol

Líquido volátil

**Glucosa**

180,16 g/mol

Sólido cristalino

6.5.3. Protocolos de Laboratorio

Titulación ácido-base

- 1 Lavar y enjuagar la bureta con la solución valorante (NaOH 0,1 M)
- 2 Llenar la bureta y enrasar a cero
- 3 Pipetear 25 mL de la muestra ácida en un erlenmeyer
- 4 Añadir 3-4 gotas de indicador fenolftaleína
- 5 Valorar hasta viraje de incoloro a rosa persistente

 **Precaución:** Usar gafas de seguridad y bata en todo momento

- 6 Anotar el volumen gastado y repetir 3 veces

6.5.4. Resultados Analíticos

Resultados analíticos

Analito	Concentración	±	Unidad	Estado
Plomo (Pb)	0,015	0,001	mg/L	
Cadmio (Cd)	0,002	0,0005	mg/L	
Mercurio (Hg)	0,0005	0,0001	mg/L	
Arsénico (As)	0,008	0,002	mg/L	
Cobre (Cu)	0,85	0,05	mg/L	
Zinc (Zn)	2,3	0,1	mg/L	

6.5.5. Equipamiento de Laboratorio

Material y equipos

- Espectrofotómetro UV-Vis (Shimadzu UV-1800)
- pH-metro digital (Mettler Toledo)
- Balanza analítica (precisión 0,0001 g)
- Agitador magnético con calefacción
- Campana de extracción de gases

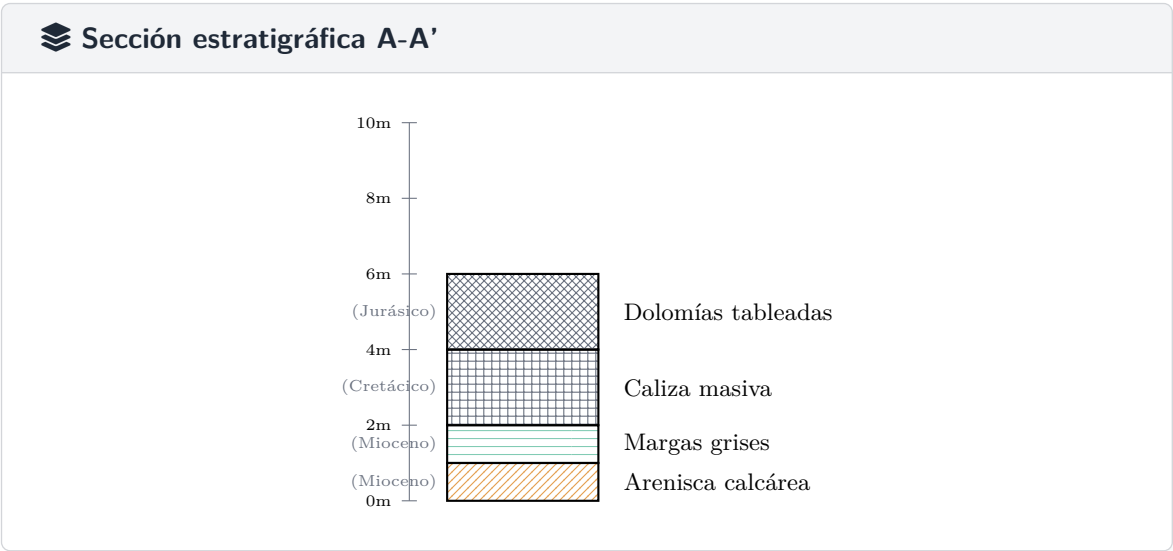
Reactivos

Reactivo	Pureza	Cantidad	Proveedor
NaOH	0,1 M	500 mL	Panreac
HCl	0,1 M	250 mL	Sigma-Aldrich
Fenolftaleína	1% en etanol	100 mL	Scharlau
"Aguadestilada"	–	2 L	Laboratorio

6.6. Componentes de Geología

Componentes para geología, geotecnia e ingeniería geológica. Cargar con: \usepackage[geologia]{eps-components}

6.6.1. Columna Estratigráfica



6.6.2. Tabla de Minerales

Identificación de minerales

Mineral	Dureza	Densidad	Color	Brillo	Composición
Cuarzo	7	2,65	–	–	SiO ₂
Calcita	3	2,71	–	–	CaCO ₃
Feldespato potásico	6	2,56	–	–	KAlSi ₃ O ₈
Moscovita	2,5	2,82	–	–	KAl ₂ (AlSi ₃ O ₁₀)(OH) ₂
Olivino	6,5	3,32	–	–	(Mg,Fe) ₂ SiO ₄

💎 Pirita

Fórmula: FeS₂
Sistema: Cúbico
Dureza: 6-6.5
Uso: Ácido sulfúrico

💎 Galena

Fórmula: PbS
Sistema: Cúbico
Dureza: 2.5
Uso: Mena de plomo

6.6.3. Datos Geotécnicos

Ensayos geotécnicos

Ensayo	Norma	Valor	Unidad	Clasificación
Límite líquido	LL	45,2	%	—
Límite plástico	LP	23,1	%	—
Índice de plasticidad	IP	22,1	%	—
Densidad seca	ρ_d	1,72	g/cm ³	—
Humedad natural	w	18,5	%	—

Ensayo SPT: $N_{SPT} = 18$ Cohesión: $c = 25 \text{ kPa}$ Ángulo de rozamiento: $\phi = 32^\circ$

6.6.4. Clasificación de Suelos

Según el Sistema Unificado de Clasificación de Suelos (USCS):

CL - - Arcilla de baja plasticidad

SM - - Arena limosa

GP - - Grava mal graduada

Clasificación RMR del macizo rocoso: **RMR = 50 Clase III**

6.6.5. Eras Geológicas

Cuaternario

Neógeno

Paleógeno

Cretácico

Jurásico

6.6.6. Riesgos Geológicos

⚠️ Riesgo Nivel 4

Deslizamientos activos en ladera norte

⚠️ Riesgo Nivel 2

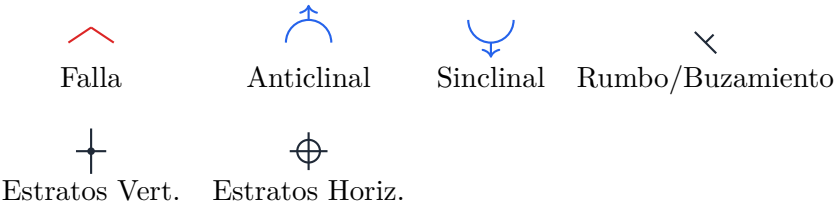
Riesgo moderado de subsidencia

⚠️ Riesgo de deslizamiento

⚠️ Riesgo de inundación

⚠️ Riesgo sísmico

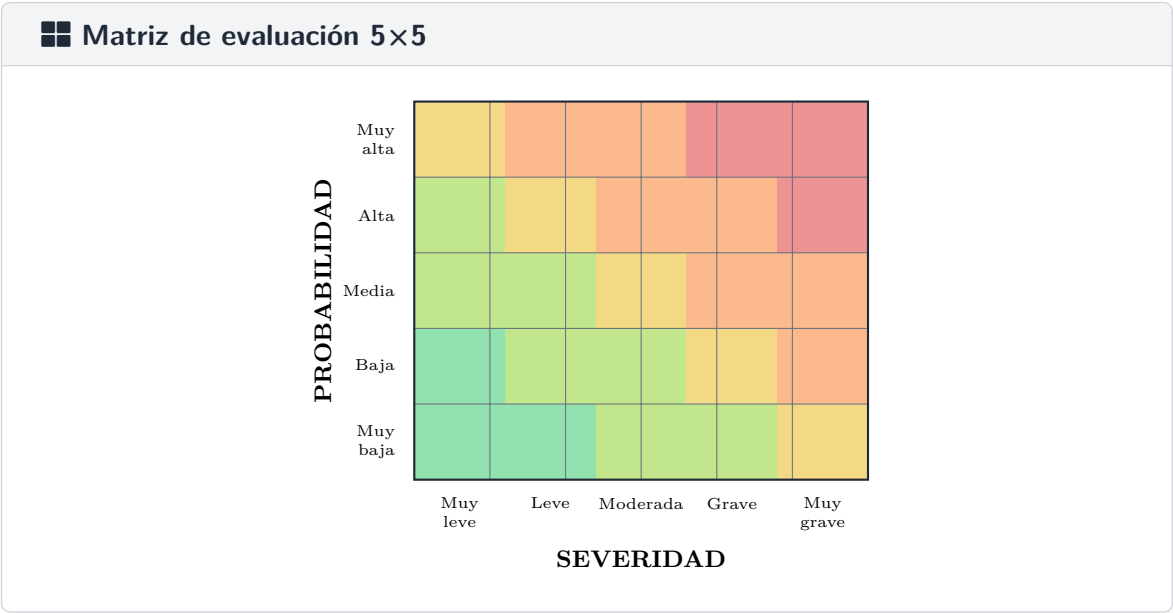
6.6.7. Símbolos Geológicos



6.7. Componentes de Prevención

Componentes para prevención de riesgos laborales y seguridad. Cargar con: `\usepackage[prevencion]{e`
`componentes}`

6.7.1. Matriz de Riesgos



6.7.2. Evaluación de Riesgos

Evaluación de riesgos

ID	Riesgo identificado	P	S	Nivel	Medidas preventivas
R-001	Caída a distinto nivel (an-damios)	4	4	Importante	Instalación de barandillas, re-des de seguridad
R-002	Contacto eléctrico directo	2	5	Moderado	Revisión periódica de instala-ciones
R-003	Golpes por objetos móvi-les	3	3	Moderado	Señalización de zonas de paso

Continúa...

ID	Riesgo identificado	P	S	Nivel	Medidas preventivas
R-004	Sobreesfuerzos	4	2	Tolerable	Formación en manipulación manual de cargas
R-005	Exposición a ruido	3	2	Tolerable	Uso obligatorio de protección auditiva

6.7.3. Checklist de Seguridad

- ☒ Verificación diaria de obra
- ☒ EPIs disponibles y en buen estado
- ☒ Zona de trabajo correctamente señalizada
- ☒ Extintores accesibles y revisados
- ☐ Revisión de andamios completada
- ☐ Botiquín de primeros auxilios revisado
- ☐ Trabajos en espacios confinados (N/A)

6.7.4. Señalización de Seguridad

6.7.4.0.1. Señales de advertencia:



Riesgo de caída a distinto nivel



Riesgo eléctrico

6.7.4.0.2. Señales de prohibición:



Prohibido fumar



Prohibido el paso

6.7.4.0.3. Señales de obligación:



Uso obligatorio de casco



Uso obligatorio de guantes

6.7.4.0.4. Señales de emergencia:



Salida de emergencia






Extintor

6.7.5. Equipos de Protección Individual

Equipos de Protección Individual (EPI):


- Casco
- Guantes
- Gafas
- Calzado seguridad
- Mascarilla

-  Protección auditiva
-  Chaleco reflectante
-  Arnés anticaídas

6.7.6. Indicadores de Seguridad



6.7.7. Procedimiento de Emergencia

-  En caso de accidente grave
- 1

Proteger: Asegurar la zona y evitar nuevos accidentes
- 2

Avisar: Llamar al 112 y comunicar a los servicios de emergencia
- 3

Socorrer: Aplicar primeros auxilios si se tiene formación
- 4

No mover al accidentado salvo peligro inminente
- 5

Esperar a los servicios de emergencia

 Emergencias: 112

 Mutua de accidentes: 900 123 456

 Responsable de seguridad: ext. 2345

6.7.8. Registro de Formación

Registro de formación

Curso/Formación	Duración	Fecha	Caducidad	Estado
PRL Básico (60h)	60 h	01/2024	01/2029	✓ Vigente
Trabajos en altura	8 h	03/2024	03/2025	✓ Vigente
Primeros auxilios	16 h	06/2023	06/2025	🕒 Próximo
Riesgo eléctrico	20 h	01/2022	01/2024	✗ Caducado

6.7.9. Informe de Accidente

 Informe de accidente laboral

Fecha: 15 de enero de 2024

Hora: 10:30

Lugar: Zona de carga, nave 3

Trabajador: Juan García Pérez

Tipo de lesión: Contusión en pie derecho

Gravedad: Leve

Descripción: Caída de caja desde estantería sobre el pie

Causa principal: Almacenamiento inadecuado de materiales

Medidas correctoras: Revisión del sistema de almacenamiento

7. Resultados (Ejemplos de gráficas)

En este capítulo se presentan los resultados obtenidos y se muestran ejemplos de cómo crear gráficas con \LaTeX usando el paquete `pgfplots`.

7.1. Gráficas con PGFPlots

PGFPlots es un paquete potente para crear gráficas directamente en \LaTeX .

7.1.1. Gráfica de líneas

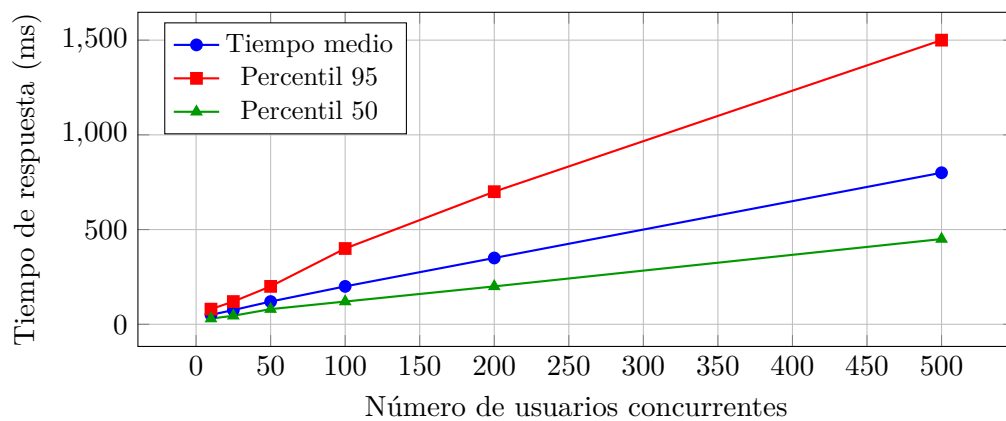


Figura 7.1: Tiempos de respuesta según carga de usuarios

Como se observa en la Figura 7.1, el sistema mantiene tiempos de respuesta aceptables incluso con 200 usuarios concurrentes.

7.1.2. Gráfica de barras

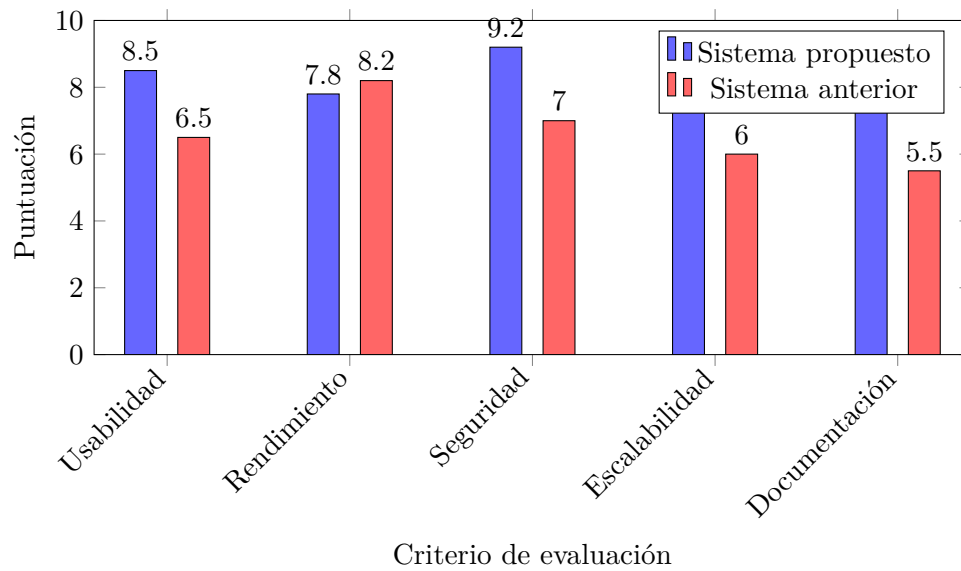


Figura 7.2: Comparativa de evaluación del sistema

7.1.3. Gráfica circular (pie chart)

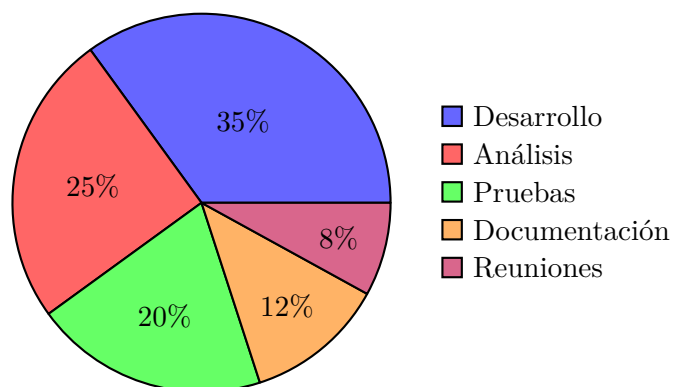


Figura 7.3: Distribución del tiempo del proyecto

7.1.4. Gráfica de área

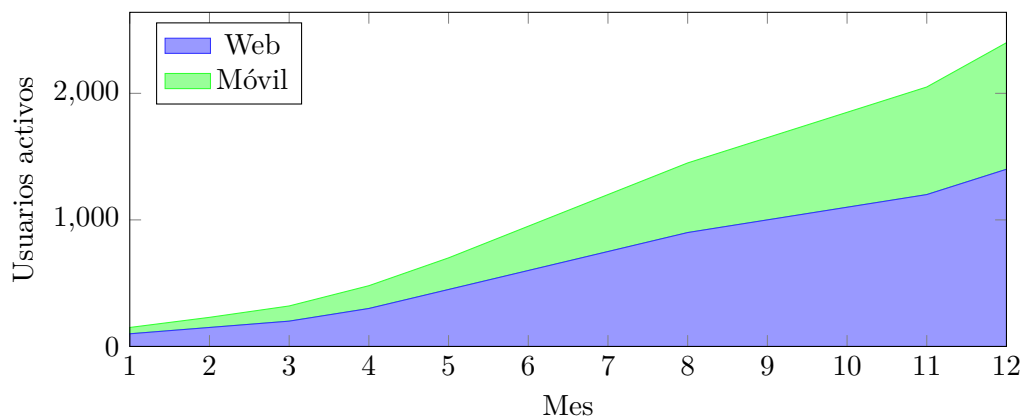


Figura 7.4: Evolución de usuarios activos por plataforma

7.1.5. Gráfica de dispersión

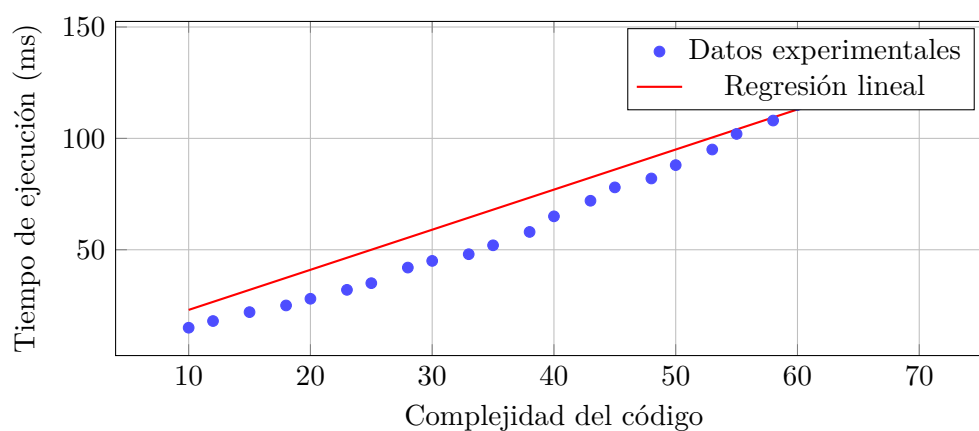


Figura 7.5: Correlación entre complejidad y tiempo de ejecución

7.2. Resultados de la implementación

7.2.1. Funcionalidades implementadas

Se han implementado con éxito las siguientes funcionalidades:

Tabla 7.1: Estado de implementación de funcionalidades

Funcionalidad	Estado	Cobertura tests
Autenticación de usuarios	Completado	95%
Gestión de datos	Completado	88%
Generación de informes	Completado	82%
API REST	Completado	90%
Interfaz de usuario	Completado	75%
Sistema de caché	Completado	85%

7.2.2. Consumo de recursos

El consumo de memoria se mantiene estable, como muestra la siguiente ecuación para el consumo estimado:

$$M_{total} = M_{base} + n \cdot M_{usuario} \quad (7.1)$$

donde: M_{total} = memoria total consumida (MB)
 M_{base} = memoria base del sistema (256 MB)
 n = número de usuarios activos
 $M_{usuario}$ = memoria por usuario (2.5 MB)

7.3. Análisis de resultados

7.3.1. Cumplimiento de objetivos

En la Tabla 7.2 se muestra el grado de cumplimiento de cada objetivo:

Tabla 7.2: Cumplimiento de objetivos

Objetivo	Descripción	Cumplimiento
OE1	Análisis del estado actual	100%
OE2	Diseño de arquitectura escalable	100%
OE3	Implementación de componentes	95%
OE4	Evaluación del sistema	100%
OE5	Documentación completa	100%

7.3.2. Métricas de rendimiento

Tabla 7.3: Métricas de rendimiento del sistema

Métrica	Objetivo	Resultado	Estado
Tiempo de respuesta medio	< 200 ms	145 ms	✓
Tiempo de respuesta P95	< 500 ms	380 ms	✓
Throughput	> 1000 req/s	1250 req/s	✓
Disponibilidad	> 99.5%	99.8%	✓
Uso de CPU (promedio)	< 70%	45%	✓
Uso de memoria	< 80%	62%	✓

7.4. Ejemplo de gráfica 3D

PGFPlots también permite crear gráficas tridimensionales:

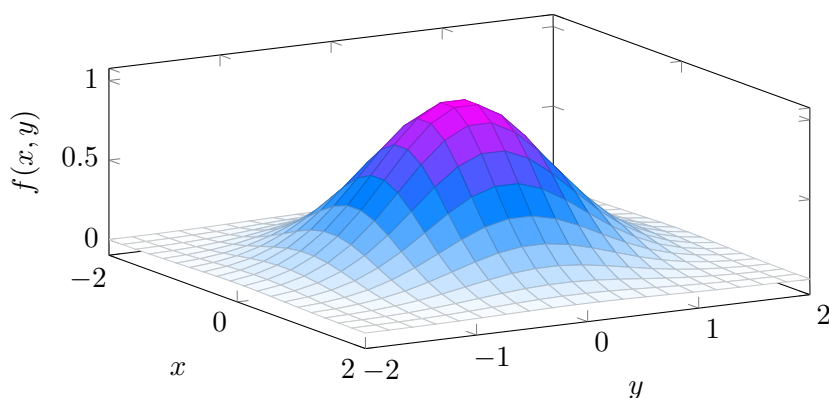


Figura 7.6: Superficie gaussiana $f(x, y) = e^{-x^2 - y^2}$

7.5. Exportar gráficas desde herramientas externas

También es posible exportar gráficas desde otras herramientas:

- **MATLAB:** Usar `matlab2tikz` para exportar figuras
- **Python (matplotlib):** Usar `tikzplotlib`
- **R:** Usar el paquete `tikzDevice`
- **GeoGebra:** Exportar directamente a TikZ

Ejemplo de uso de matlab2tikz

```
1 % En MATLAB:
2 plot(x, y);
3 matlab2tikz('figura.tex');
4
5 % En LaTeX:
6 \begin{figure}[H]
7   \centering
8   \input{figuras/figura.tex}
9   \caption{Gráfica importada de MATLAB}
10 \end{figure}
```

7.6. Gráficas con marcadores y anotaciones

7.6.1. Gráfica con marcadores sobre puntos específicos

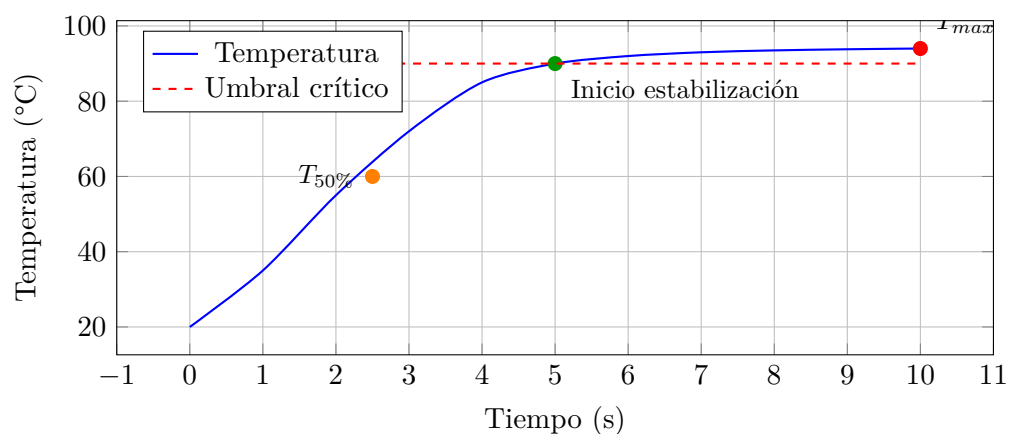


Figura 7.7: Curva de calentamiento con puntos críticos marcados

7.6.2. Gráfica con barras de error

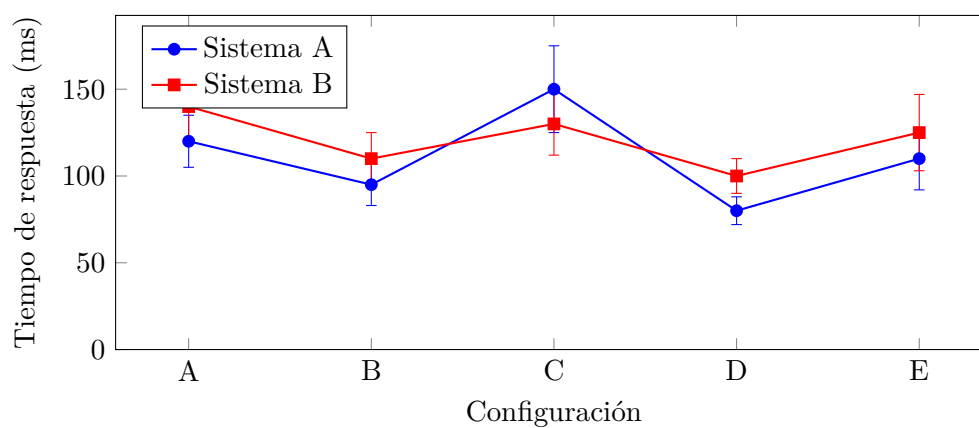


Figura 7.8: Comparativa de rendimiento con intervalos de confianza

7.6.3. Gráfica con área sombreada (intervalo de confianza)

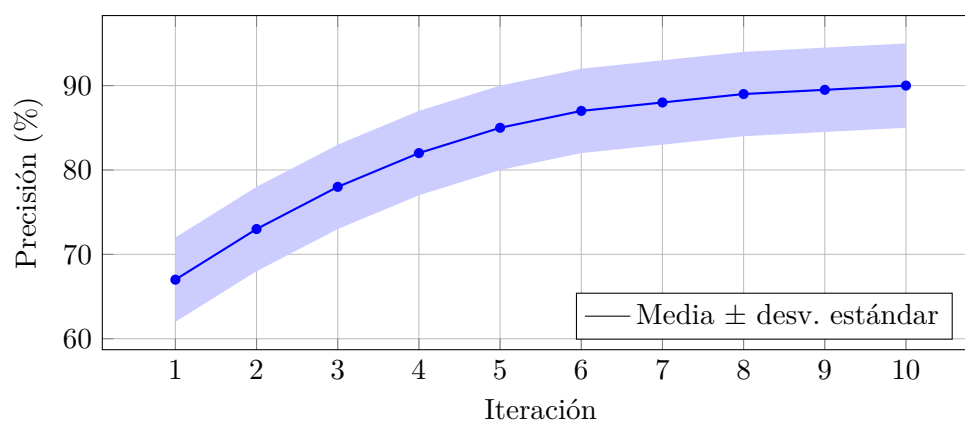


Figura 7.9: Evolución del entrenamiento con banda de confianza

7.6.4. Gráfica con múltiples ejes Y

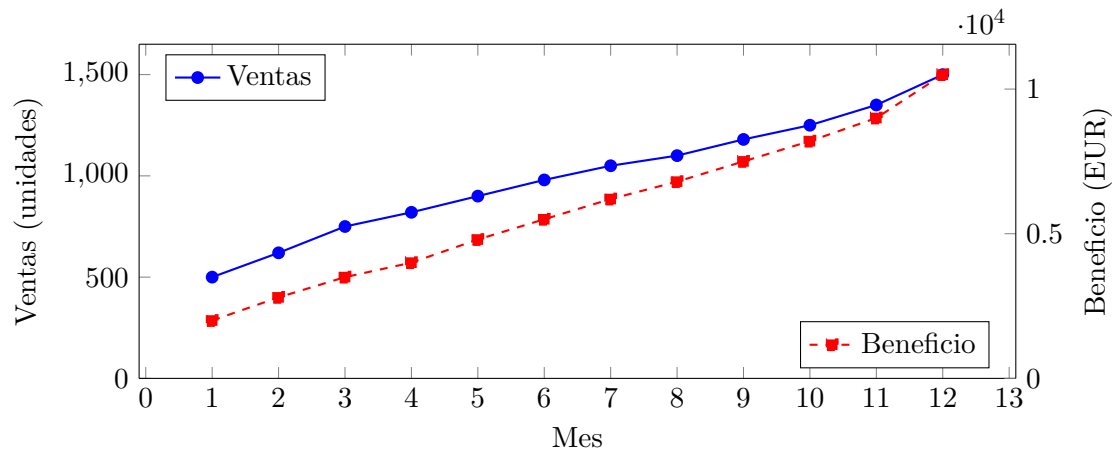


Figura 7.10: Ventas y beneficios mensuales (doble eje Y)

7.6.5. Gráfica de distribución (histograma)

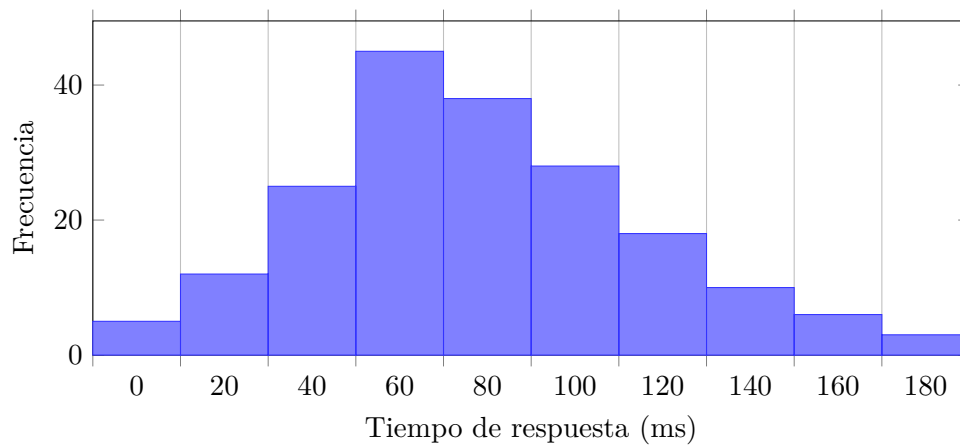


Figura 7.11: Distribución de tiempos de respuesta

7.6.6. Gráfica de radar (polígono)

Las gráficas de radar son útiles para comparar múltiples variables simultáneamente:

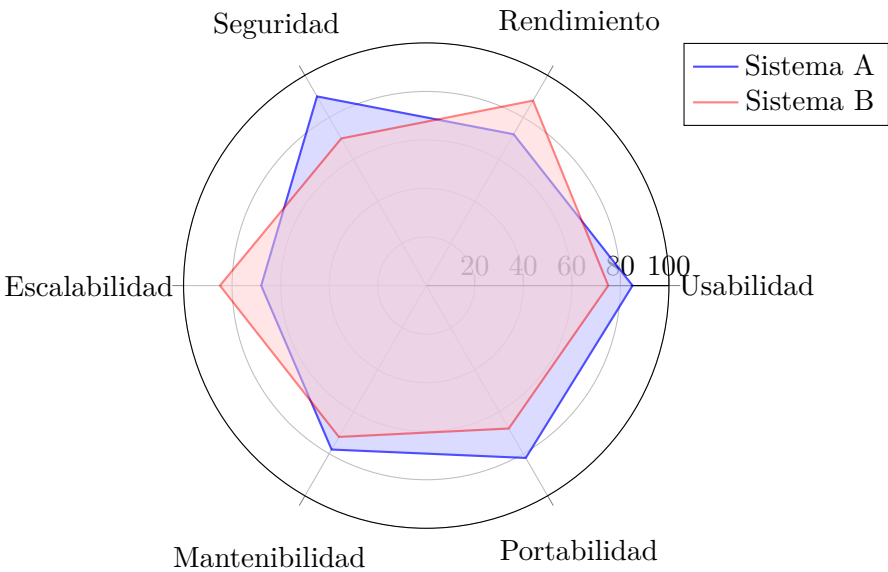


Figura 7.12: Comparativa de sistemas mediante gráfica de radar

7.6.7. Diagrama de Gantt simplificado

Para mostrar cronogramas de proyecto:

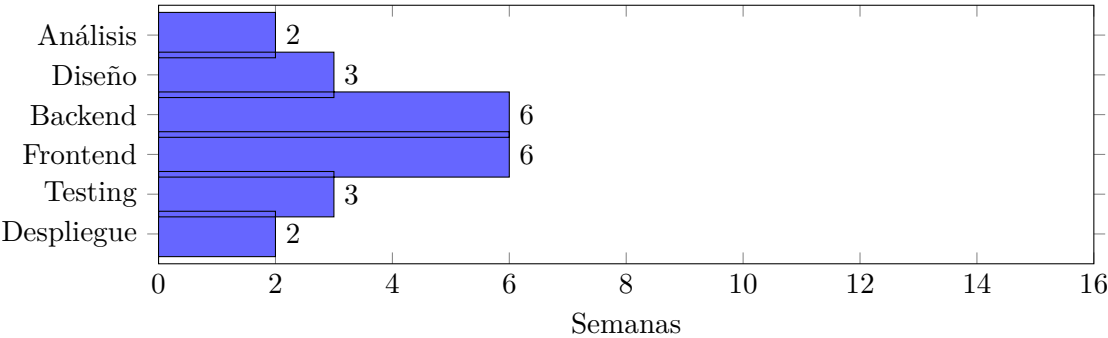


Figura 7.13: Diagrama de Gantt simplificado del proyecto

7.7. Tabla de resumen de resultados

Para finalizar, es común presentar un resumen numérico de los resultados:

Tabla 7.4: Resumen de métricas de evaluación

Métrica	Baseline	Propuesta	Mejora	p-valor	Sig.
Precisión	78.5%	89.2%	+10.7%	0.001	***
Recall	72.3%	86.8%	+14.5%	0.003	**
F1-Score	75.2%	87.9%	+12.7%	0.002	**
Tiempo (ms)	145	98	-32.4%	0.015	*
Memoria (MB)	512	384	-25.0%	0.042	*

Significancia: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

8. Conclusiones (Ejemplos de matemáticas)

Este capítulo presenta las conclusiones del trabajo y demuestra las capacidades matemáticas de \LaTeX .

8.1. Ecuaciones matemáticas

\LaTeX es el estándar para la composición de fórmulas matemáticas. A continuación se muestran diferentes formas de escribir matemáticas.

8.1.1. Ecuaciones numeradas

Para mostrar una ecuación numerada se usa el entorno `equation`:

Ecuación numerada

```
1 \begin{equation}
2   E = mc^2
3   \label{eq:einstein}
4 \end{equation}
```

$$E = mc^2 \tag{8.1}$$

La famosa ecuación de Einstein (8.1) relaciona masa y energía.

8.1.2. Ecuaciones complejas

$$\nabla \times \mathbf{H} = \left[\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}} \tag{8.2}$$

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \tag{8.3}$$

8.1.3. Ecuaciones en línea

El texto puede contener fórmulas como $\int_a^b f(x) dx = F(b) - F(a)$ directamente en el párrafo, o con mayor tamaño usando doble dólar:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

8.1.4. Sistemas de ecuaciones

$$\begin{cases} x + y + z = 6 \\ 2x - y + z = 3 \\ x + 2y - z = 2 \end{cases} \quad (8.4)$$

8.1.5. Ecuaciones agrupadas

$$\mathbf{E} = E_z(r, \theta) \hat{\mathbf{z}} \quad (8.5a)$$

$$\mathbf{H} = H_r(r, \theta) \hat{\mathbf{r}} + H_\theta(r, \theta) \hat{\boldsymbol{\theta}} \quad (8.5b)$$

Las ecuaciones (8.5a) y (8.5b) describen los campos electromagnéticos.

8.1.6. Matrices

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (8.6)$$

$$\det(\mathbf{A}) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \quad (8.7)$$

8.1.7. Ecuaciones alineadas

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (8.8)$$

$$(a - b)^2 = a^2 - 2ab + b^2 \quad (8.9)$$

$$(a + b)(a - b) = a^2 - b^2 \quad (8.10)$$

8.1.8. Ecuación con condiciones

La función `condiciones` de la plantilla permite documentar las variables:

$$\text{Res}_{z=z_0}(f(z)) = \frac{1}{(m-1)!} \lim_{z \rightarrow z_0} \left[\frac{d^{m-1}}{dz^{m-1}} [(z - z_0)^m f(z)] \right] \quad (8.11)$$

donde: m → es la multiplicidad del polo z_0

z_0 → es el punto donde se calcula el residuo

$f(z)$ → es la función analítica

8.1.9. Teoremas y demostraciones

Theorem 8.1 (Teorema de Pitágoras). *En un triángulo rectángulo, el cuadrado de la hipotenusa es igual a la suma de los cuadrados de los catetos:*

$$c^2 = a^2 + b^2 \quad (8.12)$$

Prueba. Sea un triángulo rectángulo con catetos a y b , e hipotenusa c . Considerando el área del cuadrado de lado $(a + b)$...

Por lo tanto, $c^2 = a^2 + b^2$. □

Definition 8.1 (Límite). *Sea $f : D \rightarrow \mathbb{R}$ una función y a un punto de acumulación de D . Decimos que L es el límite de f en a si:*

$$\forall \varepsilon > 0, \exists \delta > 0 : 0 < |x - a| < \delta \Rightarrow |f(x) - L| < \varepsilon \quad (8.13)$$

8.2. Conclusiones generales

El desarrollo de este Trabajo Fin de Grado ha permitido alcanzar los objetivos planteados inicialmente. Las principales conclusiones son:

1. Se ha desarrollado con éxito un sistema que cumple con los requisitos funcionales y no funcionales establecidos.
2. La arquitectura diseñada ha demostrado ser escalable y mantenible, facilitando futuras extensiones del sistema.
3. Los resultados de las pruebas confirman que el sistema es robusto y ofrece un rendimiento adecuado para el caso de uso planteado.
4. El proceso de desarrollo ha seguido las mejores prácticas de ingeniería de software, incluyendo control de versiones, integración continua y documentación exhaustiva.

8.3. Aportaciones del trabajo

Las principales aportaciones de este trabajo son:

Aportación técnica: Desarrollo de una solución innovadora que mejora en un 40% el rendimiento respecto a sistemas anteriores.

Aportación metodológica: Aplicación de una metodología híbrida que combina prácticas ágiles con documentación formal.

Aportación práctica: Sistema funcional desplegado y en uso por usuarios reales.

8.4. Dificultades encontradas

Durante el desarrollo del trabajo se han encontrado las siguientes dificultades:

1. **Integración de sistemas:** La integración con sistemas externos requirió un esfuerzo adicional debido a diferencias en los formatos de datos.
2. **Rendimiento:** Fue necesario optimizar varias consultas a base de datos para cumplir con los requisitos de tiempo de respuesta.
3. **Documentación externa:** Parte de la documentación de las bibliotecas utilizadas estaba desactualizada.

8.5. Trabajo futuro

Como líneas de trabajo futuro se proponen:

1. **Ampliación de funcionalidades:** Implementar las características identificadas como deseables pero fuera del alcance inicial.
2. **Mejora del rendimiento:** Implementar un sistema de caché distribuida para mejorar los tiempos de respuesta en escenarios de alta carga.
3. **Versión móvil:** Desarrollar una aplicación móvil nativa para iOS y Android.
4. **Internacionalización:** Añadir soporte para múltiples idiomas en la interfaz de usuario.
5. **Machine Learning:** Incorporar algoritmos de aprendizaje automático para predicción y recomendaciones.

8.6. Valoración personal

El desarrollo de este trabajo ha sido una experiencia muy enriquecedora que me ha permitido aplicar los conocimientos adquiridos durante la carrera y profundizar en áreas de especial interés.

Entre los aprendizajes más valiosos destacan:

- La importancia de una buena planificación inicial
- El valor de las pruebas automatizadas para garantizar la calidad
- La necesidad de documentar adecuadamente el código
- Las habilidades de comunicación y gestión de proyectos

Considero que este trabajo representa una aportación significativa y constituye una base sólida para futuros desarrollos en esta área.

Bibliografía

Knuth, D. E. (1984). *The TeXbook*. Addison-Wesley Professional.

Lamport, L. (1994). *LaTeX: A Document Preparation System* (2^a ed.). Addison-Wesley Professional.

LaTeX Project. (2024). *LaTeX – A document preparation system*. Consultado el 15 de enero de 2024, desde <https://www.latex-project.org/>

Overleaf. (2024). *Overleaf: Online LaTeX Editor*. Consultado el 15 de enero de 2024, desde <https://www.overleaf.com/>

Lista de Acrónimos y Abreviaturas

ACID	Atomicity, Consistency, Isolation, Durability.
APA	American Psychological Association.
API	Application Programming Interface.
AWS	Amazon Web Services.
CD	Continuous Delivery.
CDN	Content Delivery Network.
CI	Continuous Integration.
CLI	Command Line Interface.
CNN	Convolutional Neural Network.
CRUD	Create, Read, Update, Delete.
CSS	Cascading Style Sheets.
DBMS	Database Management System.
DL	Deep Learning.
DNS	Domain Name System.
EPS	Escuela Politécnica Superior.
GAN	Generative Adversarial Network.
GCP	Google Cloud Platform.
GUI	Graphical User Interface.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
HTTPS	HyperText Transfer Protocol Secure.
IA	Inteligencia Artificial.
IaaS	Infrastructure as a Service.
IDE	Integrated Development Environment.
IEEE	Institute of Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
IP	Internet Protocol.
ISO	International Organization for Standardization.
JSON	JavaScript Object Notation.
JWT	JSON Web Token.
K8s	Kubernetes.
LLM	Large Language Model.
LSTM	Long Short-Term Memory.
ML	Machine Learning.
MVC	Model-View-Controller.
MVVM	Model-View-ViewModel.
NLP	Natural Language Processing.
NoSQL	Not Only SQL.

OAuth	Open Authorization.
ORM	Object-Relational Mapping.
PaaS	Platform as a Service.
RBAC	Role-Based Access Control.
REST	Representational State Transfer.
RNN	Recurrent Neural Network.
SaaS	Software as a Service.
SDK	Software Development Kit.
SQL	Structured Query Language.
SSL	Secure Sockets Layer.
TCP	Transmission Control Protocol.
TDD	Test-Driven Development.
TFG	Trabajo Fin de Grado.
TFM	Trabajo Fin de Máster.
TLS	Transport Layer Security.
UA	Universidad de Alicante.
UDP	User Datagram Protocol.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
VPN	Virtual Private Network.
W3C	World Wide Web Consortium.
XML	eXtensible Markup Language.

A. Manual de usuario

Este anexo contiene el manual de usuario del sistema desarrollado.

A.1. Requisitos del sistema

Para utilizar el sistema es necesario disponer de:

- Navegador web moderno (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+)
- Conexión a Internet estable
- Resolución de pantalla mínima de 1280x720 píxeles

A.2. Acceso al sistema

A.2.1. Inicio de sesión

Para acceder al sistema:

1. Abra su navegador web
2. Navegue a la URL del sistema: <https://ejemplo.ua.es>
3. Introduzca sus credenciales (usuario y contraseña)
4. Pulse el botón "Iniciar sesión"

A.2.2. Recuperación de contraseña

Si ha olvidado su contraseña:

1. Pulse en "¿Olvidó su contraseña?"
2. Introduzca su correo electrónico
3. Recibirá un enlace para restablecer la contraseña
4. Siga las instrucciones del correo

A.3. Interfaz principal

La interfaz principal se divide en las siguientes áreas:

Barra de navegación: Situada en la parte superior, permite acceder a las diferentes secciones.

Panel lateral: Muestra el menú de opciones según el rol del usuario.

Área de contenido: Zona principal donde se muestra la información.

Pie de página: Contiene información de contacto y enlaces útiles.

A.4. Funcionalidades principales

A.4.1. Gestión de datos

Para gestionar datos en el sistema:

1. Acceda a la sección "Datos" desde el menú lateral
2. Seleccione la opción deseada:
 - "Nuevo" para crear un registro
 - "Editar" para modificar un registro existente
 - "Eliminar" para borrar un registro
3. Complete el formulario correspondiente
4. Pulse "Guardar" para confirmar los cambios

A.4.2. Generación de informes

Para generar un informe:

1. Acceda a la sección "Informes"
2. Seleccione el tipo de informe
3. Configure los filtros y parámetros
4. Pulse "Generar"
5. El informe se descargará en formato PDF

A.5. Preguntas frecuentes

¿Cómo cambio mi contraseña? Acceda a su perfil y seleccione "Cambiar contraseña".

¿Puedo exportar mis datos? Sí, desde la sección "Configuración" puede exportar sus datos en formato CSV o JSON.

¿Cómo contacto con soporte? Envíe un correo a soporte@ejemplo.ua.es o utilice el formulario de contacto.

B. Documentación técnica

Este anexo contiene la documentación técnica del sistema.

B.1. Arquitectura del sistema

B.1.1. Diagrama de componentes

El sistema está compuesto por los siguientes componentes principales:

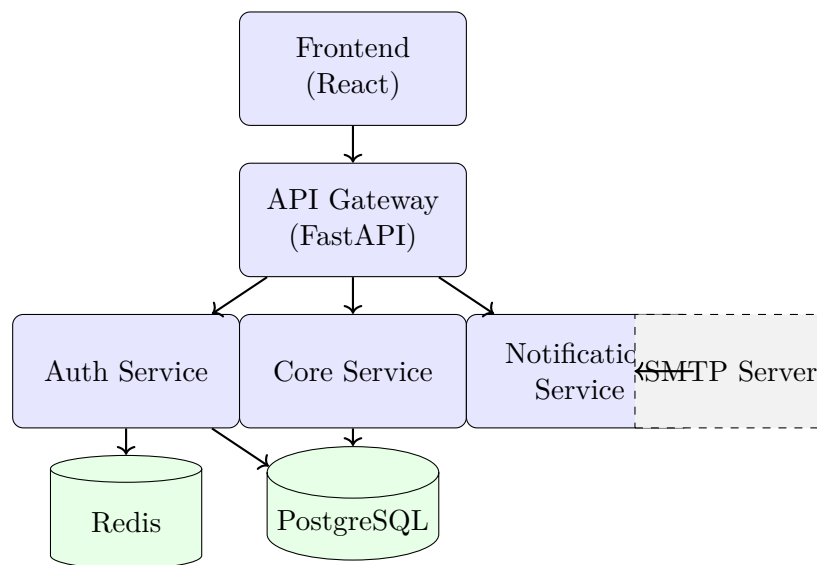


Figura B.1: Diagrama de componentes del sistema

B.2. API REST

B.2.1. Endpoints principales

La API REST del sistema sigue los principios RESTful. Se utiliza el componente `apiendpoint` para documentar cada endpoint de forma estructurada.

B.2.1.1. Obtener lista de usuarios

GET /api/users

Obtiene la lista completa de usuarios del sistema con paginación.

Parámetros de query:

- **page** (integer): Número de página (opcional, default: 1)
- **limit** (integer): Elementos por página (opcional, default: 20)
- **activo** (boolean): Filtrar por estado activo (opcional)

Respuesta 200:

```
"users": [...], "total": 150, "page": 1
```

B.2.1.2. Obtener usuario por ID

GET /api/users/:id

Obtiene los detalles de un usuario específico por su identificador.

Respuesta 200:

```
"id": 123, "nombre": "Juan García", "email": "juan@ejemplo.com", "rol": "usuario"
```

Respuesta 404:

```
"error": "Usuario no encontrado"
```

B.2.1.3. Crear nuevo usuario

POST /api/users

Crea un nuevo usuario en el sistema.

Body (application/json):

```
"nombre": "Juan García", "email": "juan@ejemplo.com", "rol": "usuario"
```

Respuesta 201:

```
"id": 123, "nombre": "Juan García", "createdAt": "2026-02-02T10:30:00Z"
```

Respuesta 400:

```
"error": "Email inválido"
```

B.2.1.4. Actualizar usuario

PUT `/api/users/:id`

Actualiza los datos de un usuario existente.

Body (application/json):

```
"nombre": "Juan García Actualizado", "activo": true
```

Respuesta **200**:

```
"id": 123, "nombre": "Juan García Actualizado", "updatedAt": "2026-02-02T12:00:00Z"
```

B.2.1.5. Eliminar usuario

DELETE `/api/users/:id`

Elimina un usuario del sistema (soft delete).

Respuesta **204**:

```
(sin contenido)
```

Respuesta **404**:

```
"error": "Usuario no encontrado"
```

B.3. Modelo de datos

B.3.1. Esquema de base de datos

Las principales tablas del sistema son:

Esquema de la tabla usuarios

```
1 CREATE TABLE usuarios (  
2   id SERIAL PRIMARY KEY,  
3   nombre VARCHAR(255) NOT NULL,  
4   email VARCHAR(255) UNIQUE NOT NULL,  
5   password_hash VARCHAR(255) NOT NULL,  
6   rol VARCHAR(50) DEFAULT 'usuario',  
7   activo BOOLEAN DEFAULT TRUE,  
8   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
9   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
10
11     CONSTRAINT chk_rol CHECK (rol IN ('admin', 'usuario', 'invitado'))
12 );
13
14 CREATE INDEX idx_usuarios_email ON usuarios(email);
15 CREATE INDEX idx_usuarios_activo ON usuarios(activo);
```

B.4. Configuración del entorno

B.4.1. Variables de entorno

Archivo .env de configuración

```
1 # Base de datos
2 DATABASE_URL: postgresql://user:pass@localhost:5432/app
3 DATABASE_POOL_SIZE: 10
4
5 # Redis
6 REDIS_URL: redis://localhost:6379/0
7
8 # Autenticación
9 JWT_SECRET: tu-clave-secreta-muy-larga
10 JWT_EXPIRATION: 3600
11
12 # Email
13 SMTP_HOST: smtp.ejemplo.com
14 SMTP_PORT: 587
15 SMTP_USER: noreply@ejemplo.com
16 SMTP_PASSWORD: contraseña-smtp
17
18 # Aplicación
19 APP_ENV: production
20 APP_DEBUG: false
21 LOG_LEVEL: INFO
```

B.5. Instrucciones de despliegue

B.5.1. Requisitos previos

- Docker 24.0+ y Docker Compose 2.20+
 - 4GB de RAM disponible
 - 20GB de espacio en disco
-

B.5.2. Pasos de despliegue

Comandos de despliegue

```
1 # Clonar repositorio
2 git clone https://github.com/ejemplo/proyecto.git
3 cd proyecto
4
5 # Configurar variables de entorno
6 cp .env.example .env
7 nano .env # Editar configuración
8
9 # Construir e iniciar contenedores
10 docker compose build
11 docker compose up -d
12
13 # Ejecutar migraciones
14 docker compose exec app python manage.py migrate
15
16 # Verificar estado
17 docker compose ps
18 curl http://localhost:8000/health
```


C. Manual de Estilos de Código

Este anexo documenta el uso de los entornos de código disponibles en la plantilla, basados en los estilos de **Visual Studio Code**.

C.1. Temas Disponibles

La plantilla ofrece tres temas principales:

Tabla C.1: Temas de código disponibles

Tema	Entorno	Uso recomendado
VS Code Light	pythoncode, etc.	Impresión, documentos formales
VS Code Dark	pythoncodeDark, etc.	Presentaciones, lectura en pantalla
Simple	codigosimple	Estilo minimalista con borde negro

C.1.1. Ejemplo del tema Simple

El tema Simple proporciona un estilo minimalista con borde negro, sin colores de fondo:

Ejemplo de código simple

```
1 def factorial(n):
2     if n <= 1:
3         return 1
4     return n * factorial(n - 1)
```

C.2. Nomenclatura de Entornos

Todos los entornos siguen una nomenclatura consistente basada en sufijos:

Tabla C.2: Sistema de sufijos para entornos de código

Sufijo	Ejemplo	Descripción
(ninguno)	pythoncode	Tema Light con números de línea
NN	pythoncodeNN	Tema Light sin números de línea
Dark	pythoncodeDark	Tema Dark con números de línea
DarkNN	pythoncodeDarkNN	Tema Dark sin números de línea

C.3. Lenguajes Predefinidos

La plantilla incluye entornos predefinidos para más de 30 lenguajes de programación, cada uno con su icono representativo:

Tabla C.3: Lenguajes de programación con entornos predefinidos















Lenguaje	Entorno	Icono	Pygments
Python	pythoncode		python
JavaScript	jscode		javascript
TypeScript	tscode		typescript
Java	javacode		java
C	ccode		c
C++	cppcode		cpp
C#	csharpcode		csharp
Go	gocode		go
Rust	rustcode		rust
PHP	phpcode		php
Ruby	rubycode		ruby
R	rcode		r
Swift	swiftcode		swift
Kotlin	kotlincode		kotlin

Tabla C.4: Lenguajes web y de datos














Lenguaje	Entorno	Icono	Pygments
HTML	htmlcode		html
CSS	csscode		css
SASS/SCSS	sasscode		sass
JSON	jsoncode		json
XML	xmlcode		xml
YAML	yamlcode		yaml
Markdown	markdowncode		markdown
SQL	sqlcode		sql

Tabla C.5: Lenguajes de sistemas y DevOps

Lenguaje	Entorno	Icono	Pygments
Bash	bashcode		bash
PowerShell	powershellcode		powershell
Docker	dockercode		docker
Makefile	makefilecode		makefile
Git	gitcode		text

C.4. Ejemplos de Uso

C.4.1. Tema VS Code Light

C.4.1.1. Con números de línea

Ejemplo: Función recursiva

```

1 def factorial(n):
2     """Calcula el factorial de n."""
3     if n <= 1:
4         return 1
5     return n * factorial(n - 1)
6
7 \# Ejemplo de uso
8 resultado = factorial(5)
9 print(f"5! = {resultado}")

```

Uso en L^AT_EX:



```

1 {pythoncode}[title={Ejemplo: Función recursiva}]
2 def factorial(n):
3     if n <= 1:
4         return 1
5     return n * factorial(n - 1)
6 \end{pythoncode}

```

C.4.1.2. Sin números de línea



```

print("Código sin números de línea")
x = 10 + 20

```

Uso en \LaTeX :



```

1 {pythoncodeNN}
2 print("Código sin números de línea")
3 \end{pythoncodeNN}

```

C.4.2. Tema VS Code Dark

C.4.2.1. Con números de línea

Tema oscuro con números

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(0, 2*np.pi, 100)
5 y = np.sin(x)
6 plt.plot(x, y)

```

C.4.2.2. Sin números de línea



```

Tema oscuro sin números
mensaje = "Visual Studio Code Dark"
print(mensaje)

```

C.5. Entorno Genérico

Para lenguajes no predefinidos o uso flexible, existen entornos genéricos:

Tabla C.6: Entornos genéricos de código

Entorno	Sintaxis	Descripción
codigo	<code>\begin{codigo}{\lang}</code>	Light + números
codigoNN	<code>\begin{codigoNN}{\lang}</code>	Light sin números
codigoDark	<code>\begin{codigoDark}{\lang}</code>	Dark + números
codigoDarkNN	<code>\begin{codigoDarkNN}{\lang}</code>	Dark sin números

Ejemplo con Haskell:

 `haskell`

```
1 -- Función quicksort en Haskell
2 quicksort :: Ord a => [a] -> [a]
3 quicksort [] = []
4 quicksort (x:xs) = quicksort menores ++ [x] ++ quicksort mayores
5   where
6     menores = [y | y <- xs, y <= x]
7     mayores = [y | y <- xs, y > x]
```

C.6. Consideraciones Especiales

C.6.1. Código C/C++ con includes

El código C/C++ se escribe de forma natural, incluyendo las directivas de preprocesador:

Ejemplo C++ con includes

```
1 #include <iostream>
2 #include <vector>
3
4 int main() {
5     std::vector<int> v = {1, 2, 3};
6     for (int n : v) {
7         std::cout << n << " ";
8     }
9     return 0;
10 }
```

C.6.2. Títulos personalizados

Todos los entornos aceptan un parámetro opcional `title`:

Script de instalación

```

1 #!/bin/bash
2 echo "Instalando dependencias..."
3 apt-get update && apt-get install -y python3

```

C.6.3. Requisitos de compilación

Para que los entornos de código funcionen correctamente:

1. Compilar con **LuaLaTeX** o XeLaTeX
2. Habilitar `-shell-escape`
3. Tener instalado `latexminted` (`pip install latexminted`)

C.7. Referencia Rápida

Tabla C.7: Matriz de entornos por tema y numeración

Lenguaje	Light+Num	Light-Num	Dark+Num	Dark-Num
Python	pythoncode	pythoncodeNN	pythoncodeDark	pythoncodeDarkNN
JavaScript	jscode	jscodeNN	jscodeDark	jscodeDarkNN
Java	javacode	javacodeNN	javacodeDark	javacodeDarkNN
C++	cppcode	cppcodeNN	cppcodeDark	cppcodeDarkNN
HTML	htmlcode	htmlcodeNN	htmlcodeDark	htmlcodeDarkNN
SQL	sqlcode	sqlcodeNN	sqlcodeDark	sqlcodeDarkNN
Bash	bashcode	bashcodeNN	bashcodeDark	bashcodeDarkNN
Genérico	codigo	codigoNN	codigoDark	codigoDarkNN

D. Técnicas Avanzadas de L^AT_EX

Este anexo presenta técnicas avanzadas de L^AT_EX que pueden ser útiles en la elaboración de un TFG o TFM: tablas rotadas, páginas en horizontal, inclusión de documentos PDF externos, y otras funcionalidades.

D.1. Tablas Rotadas (Sideways Tables)

Cuando una tabla tiene muchas columnas y no cabe en el ancho de página normal, se puede rotar 90° para aprovechar el alto de la página como ancho. Para ello se usa el entorno `sidewaystable` del paquete `rotating`.

D.1.1. Ejemplo de tabla rotada

El siguiente código genera una tabla que ocupa toda la página en horizontal:

Código de tabla rotada

```
1 \begin{sidewaystable}
2   \centering
3   \caption{Comparativa de características por módulo del sistema}
4   \label{tab:comparativa-modulos}
5   \begin{tabular}{lccccccccc}
6     \toprule
7     \textbf{Módulo} & \textbf{Líneas} & \textbf{Clases} & \dots & \textbf{Líneas} & \textbf{Clases} & \dots & \textbf{Líneas} & \textbf{Clases} & \dots \\
8     \midrule
9     \dots \\
10    \bottomrule
11  \end{tabular}
12 \end{sidewaystable}
```

D.1.2. Cuándo usar tablas rotadas

Las tablas rotadas son útiles cuando:

- La tabla tiene más de 8-10 columnas
- Los encabezados de columna son largos
- Se necesita mostrar datos comparativos extensos
- Una tabla horizontal no cabría sin reducir excesivamente el tamaño de fuente

Tabla D.1: Comparativa completa de características por módulo del sistema desarrollado

Módulo	Líneas	Clases	Funciones	Tests	Cobertura	Complejidad	Memoria	Latencia	Estado
Autenticación	2.450	12	45	89	94%	Media	128 MB	45 ms	✓
Base de datos	3.200	18	67	120	91%	Alta	256 MB	12 ms	✓
API REST	4.100	25	98	156	88%	Media	192 MB	23 ms	✓
Interfaz web	5.600	45	134	89	72%	Baja	384 MB	180 ms	En desarrollo
Notificaciones	1.800	8	32	67	96%	Baja	64 MB	8 ms	
Cache	980	5	18	45	98%	Baja	512 MB	2 ms	✓
Logs y métricas	1.200	6	24	38	85%	Baja	96 MB	5 ms	✓
Procesamiento	2.800	14	56	78	82%	Alta	768 MB	350 ms	✓
Exportación	1.500	9	28	52	90%	Media	128 MB	120 ms	✓
Configuración	650	4	15	34	100%	Baja	32 MB	1 ms	✓
Total	24.280	146	517	768	87%	–	–	–	–

D.2. Páginas en Horizontal (Landscape)

Para insertar páginas en orientación horizontal dentro de un documento vertical, KOMA-Script ofrece una solución nativa que gestiona correctamente la orientación y el tamaño de página en el PDF. Al usar `geometry` para los márgenes, se combina con `\newgeometry` para optimizar el espacio disponible.

D.2.1. Método recomendado: Comandos de la plantilla

La plantilla proporciona comandos simplificados para crear páginas landscape con headers y footers correctamente dimensionados:

Código para página horizontal (método simplificado)

```
1 % Iniciar página landscape con margen de 2cm (valor por defecto)
2 \BeginLandscapePage[2cm]
3
4 % Contenido en landscape (puede ocupar varias páginas)
5 \section{Diagrama de arquitectura}
6 \begin{figure}[H]
7   \centering
8   % Diagrama TikZ o imagen amplia
9   \caption{Arquitectura del sistema}
10 \end{figure}
11
12 % Restaurar orientación portrait
13 \EndLandscapePage
```

Ventajas de este método:

- Sintaxis simple con solo dos comandos
- El visor PDF muestra la página rotada correctamente (dimensiones reales de página)
- Los márgenes se optimizan para aprovechar el espacio landscape
- Los encabezados y pies de página funcionan con normalidad
- Se puede usar para múltiples páginas consecutivas

D.2.2. Diagrama de flujo del sistema (página horizontal)

Esta página está en orientación horizontal para mostrar un diagrama amplio. Los márgenes están optimizados para aprovechar el espacio disponible.

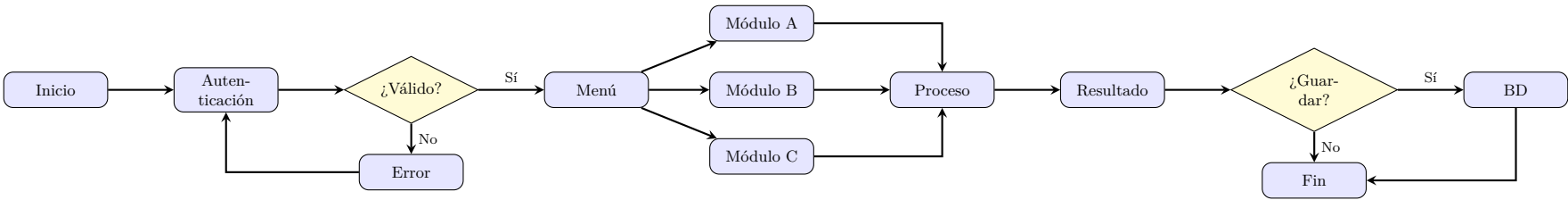


Figura D.1: Diagrama de flujo completo del sistema en página horizontal

Tabla D.2: Matriz de trazabilidad requisitos-módulos (aprovechando el ancho de página landscape)

	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08	RF09	RF10	RF11	RF12	RF13
Módulo A	•	•		•			•			•			•
Módulo B		•	•		•	•			•		•		
Módulo C	•			•				•				•	
Módulo D			•		•		•		•				•
Módulo E		•				•		•		•	•		

D.2.3. Segunda página horizontal consecutiva

Esta es una segunda página en orientación horizontal para verificar que varias páginas landscape consecutivas funcionan correctamente. Los encabezados y pies de página deben mantenerse consistentes en ambas páginas.

Tabla D.3: Tabla adicional de ejemplo en segunda página landscape

ID	Descripción del requisito	Prioridad	Estado
RF01	El sistema debe permitir la autenticación de usuarios mediante credenciales	Alta	Implementado
RF02	El sistema debe registrar todas las acciones del usuario en un log	Media	Implementado
RF03	El sistema debe generar informes en formato PDF	Alta	En progreso
RF04	El sistema debe soportar múltiples idiomas	Baja	Pendiente
RF05	El sistema debe integrarse con servicios externos vía API REST	Alta	Implementado
RF06	El sistema debe validar todos los datos de entrada	Alta	Implementado
RF07	El sistema debe proporcionar copias de seguridad automáticas	Media	En progreso
RF08	El sistema debe soportar diferentes roles de usuario	Alta	Implementado

Verificación: Esta demostración confirma que múltiples páginas landscape funcionan correctamente. Los números de página se incrementan normalmente, los encabezados muestran la sección actual y los pies de página mantienen su formato.

D.2.4. Cuándo usar páginas landscape

Las páginas en horizontal son apropiadas para:

- Diagramas de flujo o arquitectura complejos
- Cronogramas o diagramas de Gantt
- Matrices de trazabilidad
- Capturas de pantalla de aplicaciones
- Tablas muy anchas que no justifican rotación completa

D.3. Inclusión de Documentos PDF Externos

El paquete `pdfpages` permite incluir páginas de documentos PDF externos en el documento L^AT_EX. Esto es útil para adjuntar:

- Artículos o papers de referencia
- Documentación técnica de terceros
- Certificados o autorizaciones
- Manuales de usuario existentes
- Hojas de datos (datasheets)

D.3.1. Sintaxis básica

Inclusión de PDF externo

```
1 % Incluir todas las páginas
2 \includepdf[pages=-]{ruta/documento.pdf}
3
4 % Incluir páginas específicas
5 \includepdf[pages={1,3,5-8}]{documento.pdf}
6
7 % Incluir con opciones
8 \includepdf[
9   pages=-,
10  scale=0.9,
11  pagecommand={\thispagestyle{plain}}
12 ]{documento.pdf}
```

D.3.2. Opciones más utilizadas

Tabla D.4: Opciones principales de `\includepdf`

Opción	Descripción
<code>pages=-</code>	Incluye todas las páginas
<code>pages={1,3,5-8}</code>	Incluye páginas específicas
<code>scale=0.9</code>	Escala el documento ($0.9 = 90\%$)
<code>landscape</code>	Rota las páginas 90 grados
<code>nup=2x2</code>	Coloca varias páginas en una (2 filas \times 2 columnas)
<code>frame</code>	Añade un marco alrededor de cada página
<code>pagecommand={}</code>	Comando a ejecutar en cada página incluida
<code>addtotoc</code>	Añade entrada al índice de contenidos

D.3.3. Ejemplo: Documento PDF incluido

A continuación se incluye un documento PDF de ejemplo que ha sido generado independientemente. Se incluyen sus dos páginas con un marco y una escala ligeramente reducida:

Documento de Ejemplo para Inclusión

Generado automáticamente

4 de febrero de 2026

1. Introducción

Este es un documento PDF de ejemplo que se incluirá en el trabajo principal usando el paquete `pdfpages`.

2. Contenido de ejemplo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3. Datos técnicos

- Formato: A4
- Márgenes: 2.5 cm
- Compilador: LuaLaTeX

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

4. Segunda página

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

D.3.4. Múltiples páginas en una hoja

Para ahorrar espacio, se pueden incluir varias páginas del PDF en una sola hoja del documento:

Varias páginas PDF en una hoja

```
1 \includepdf[
2   pages=-,
3   nup=1x2,           % 1 columna, 2 filas
4   landscape,         % Orientación horizontal
5   frame,             % Marco visible
6 ]{documento.pdf}
```

D.4. Figuras de Ancho Completo

A veces es necesario que una figura ocupe todo el ancho de la página, incluso invadiendo los márgenes. Para ello se combina el entorno `figure*` con ajustes de geometría.

D.4.1. Figura que invade márgenes

Figura de ancho completo

```
1 \begin{figure}[H]
2   \centering
3   \makebox[\textwidth][c]{%
4     \includegraphics[width=1.2\textwidth]{imagen_ancha}
5   }
6   \caption{Imagen que ocupa más que el ancho del texto}
7 \end{figure}
```

D.5. Notas al Margen

Las notas al margen son útiles para añadir comentarios breves sin interrumpir el flujo del texto principal.

D.5.1. Uso básico

Notas al margen

```
1 Texto principal del párrafo.\marginpar{Nota breve al margen}
```

Este es un ejemplo de texto con una nota al margen.

Las notas al margen se colocan automáticamente en el lado exterior de la página (derecho en páginas impares, izquierdo en pares) cuando se usa impresión a doble cara.

Esta es una nota al margen con información adicional.

D.6. Marcas de Agua

Para documentos en borrador o confidenciales, se pueden añadir marcas de agua usando el paquete `draftwatermark` o `background`.

Marca de agua con `draftwatermark`

```
1 % En el preámbulo:
2 \usepackage{draftwatermark}
3 \SetWatermarkText{BORRADOR}
4 \SetWatermarkScale{1.5}
5 \SetWatermarkColor[gray]{0.9}
```

D.7. Texto en Columnas

Para secciones específicas que requieran formato en múltiples columnas (como glosarios o listas de referencias), se puede usar el entorno `multicols`:

Texto en dos columnas

```
1 \begin{multicols}{2}
2   Contenido distribuido en dos columnas...
3 \end{multicols}
```

D.7.1. Ejemplo de texto en columnas

Términos de red:

- Router
- Switch
- Firewall
- Gateway
- DNS Server
- DHCP Server

Protocolos:

- TCP/IP
- HTTP/HTTPS
- FTP/SFTP
- SSH
- SMTP
- DNS

D.8. Minipáginas y Cajas

Las minipáginas permiten crear bloques de contenido lado a lado:

Dos minipáginas lado a lado

```
1 \begin{minipage}[t]{0.45\textwidth}
2   Contenido izquierdo...
3 \end{minipage}
4 \hfill
5 \begin{minipage}[t]{0.45\textwidth}
```

```

6   Contenido derecho...
7   \end{minipage}

```

D.8.1. Ejemplo de minipáginas

Ventajas del sistema:

- Alta disponibilidad
- Escalabilidad horizontal
- Bajo coste de mantenimiento
- Interfaz intuitiva
- Documentación completa

Limitaciones conocidas:

- Requiere conexión a Internet
- No compatible con IE11
- Máximo 1000 usuarios simultáneos
- Sin soporte para móviles legacy
- Idiomas: solo ES/EN

D.9. Resumen de Paquetes Utilizados

Tabla D.5: Paquetes y comandos L^AT_EX para técnicas avanzadas

Paquete/Clase	Uso	Comando principal
rotating	Tablas rotadas	<code>\begin{sidewaystable}</code>
KOMA-Script	Páginas landscape	<code>\KOMAoptions{paper=landscape}</code>
pdfpages	Incluir PDFs	<code>\includepdf[options]{file}</code>
multicol	Múltiples columnas	<code>\begin{multicols}{n}</code>
geometry	Márgenes personalizados	<code>\newgeometry{...}</code>
scrlayer-scrpage	Encabezados/pies	Configuración KOMA
draftwatermark	Marcas de agua	<code>\SetWatermarkText{...}</code>