

Trabajo Fin de Máster

Máster en Ingeniería de Telecomunicación

Entrenamiento y despliegue de un modelo de clasificación de audio

Autor: Antonio José Aragón Molina

Tutor: María del Mar Elena Pérez

Dpto. Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 5 de noviembre de 2023



Trabajo Fin de Máster
Máster en Ingeniería de Telecomunicación

Entrenamiento y despliegue de un modelo de clasificación de audio

Autor:

Antonio José Aragón Molina

Tutor:

María del Mar Elena Pérez

Profesor Titular

Dpto. Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 5 de noviembre de 2023

Trabajo Fin de Máster: Entrenamiento y despliegue de un modelo
de clasificación de audio

Autor: Antonio José Aragón Molina
Tutor: María del Mar Elena Pérez

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

El diseño de una hoja de estilo en \LaTeX para un texto no es en absoluto trivial. Por un lado hay que conocer bien los usos, costumbres y reglas que se emplean a la hora de establecer márgenes, tipos de letras, tamaños de las mismas, títulos, estilos de tablas, y un sinfín de otros aspectos. Por otro, la programación en \LaTeX de esta hoja de estilo es muy tediosa, incluida la selección de los mejores paquetes para ello. La hoja de estilo adoptada por nuestra Escuela y utilizada en este texto es una versión de la que el profesor Payán realizó para un libro que desde hace tiempo viene escribiendo para su asignatura. Además, el prof. Payán ha participado de forma decisiva en la adaptación de dicha plantilla a los tres tipos de documentos que se han tenido en cuenta: libro, tesis y proyectos final de carrera, grado o máster. Y también en la redacción de este texto, que sirve de manual para la utilización de estos estilos. Por todo ello, y por hacerlo de forma totalmente desinteresada, la Escuela le está enormemente agradecida.

A esta hoja de estilos se le incluyó unos nuevos diseños de portada. El diseño gráfico de las portadas para proyectos fin de grado, carrera y máster, está basado en el que el prof. Fernando García García, de la Facultad de Bellas Artes de nuestra Universidad, hiciera para los libros, o tesis, de la sección de publicación de nuestra Escuela. Nuestra Escuela le agradece que pusiera su arte y su trabajo, de forma gratuita, a nuestra disposición.

Juan José Murillo Fuentes
Subdirección de Comunicaciones y Recursos Comunes

Sevilla, 2013

Resumen

En nuestra Escuela se producen un número considerable de documentos, tantos docentes como investigadores. Nuestros alumnos también contribuyen a esta producción a través de sus trabajos de fin de grado, máster y tesis. El objetivo de este material es facilitar la edición de todos estos documentos y a la vez fomentar nuestra imagen corporativa, facilitando la visibilidad y el reconocimiento de nuestro Centro.

Abstract

In our school there are a considerable number of documents, many teachers and researchers. Our students also contribute to this production through its work in order of degree, master's theses. The aim of this material is easier to edit these documents at the same time promote our corporate image, providing visibility and recognition of our Center.

... -translation by google-

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1 Introducción	1
1.1 Motivación	1
1.2 Planificación	3
1.3 Diseño de la solución	4
2 Diseño del modelo	7
2.1 Introducción	7
3 Despliegue del modelo	9
3.1 Problemática y solución	9
3.2 Enfoque escogido	9
3.3 Aplicación web	10
3.4 Aplicación Flask en producción	11
3.5 Despliegue	12
4 Notas sobre la Redacción del Texto	15
4.1 Preliminares	15
4.2 Texto principal	16
4.3 Bibliografía	16
4.4 Anexos	16
5 Ejemplo de Capítulo	17
5.1 Ejemplo de sección	17
5.2 Elementos del texto	18
5.3 Una nueva sección después del resumen	23
Problemas Propuestos	25
Anexo	27
5.4 Señales: definición y clasificación	27
6 Ejemplo de Capítulo de Problemas	29
6.1 Ruido y sensibilidad	29
6.2 No linealidad	30
P. 6.1 Radioenlace del servicio fijo a 13 GHz	30
P. 6.2 Radioenlace del servicio fijo a 13 GHz	31

6.3	...Ruido y sensibilidad	32
7	Estilo tipográfico LibroETSI	35
7.1	Bloque 0	35
7.2	Bloque 1: Aspectos generales	35
7.3	Bloque 2: Idioma, Codificación y Fuentes	36
7.4	Bloque 3: Tabla de contenidos	39
7.5	Bloque 4: Estilos de Páginas y Títulos	41
7.6	Bloque 5: Gestión general del documento	44
7.7	A modo de conclusión	50
	<i>Índice de Figuras</i>	51
	<i>Índice de Tablas</i>	53
	<i>Índice de Códigos</i>	55
	<i>Bibliografía</i>	57
	<i>Índice alfabético</i>	59
	<i>Glosario</i>	61

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1 Introducción	1
1.1 Motivación	1
Dataset	2
Enfoque	2
Hardware	2
Entorno de despliegue	3
1.2 Planificación	3
1.2.1 Objetivo y alcance	3
1.2.2 Requisitos	3
Requisitos funcionales	3
Requisitos operacionales	3
Requisitos de diseño	3
Requisitos de seguridad	4
1.3 Diseño de la solución	4
1.3.1 Creación del modelo	4
1.3.2 Despliegue del modelo	4
2 Diseño del modelo	7
2.1 Introducción	7
3 Despliegue del modelo	9
3.1 Problemática y solución	9
3.2 Enfoque escogido	9
3.3 Aplicación web	10
3.3.1 Flask	10
3.3.2 Estructura de la aplicación	10
3.3.3 Interfaz web	10
3.4 Aplicación Flask en producción	11
3.4.1 Unicorn	11
3.4.2 Traefik	11
3.4.3 Docker	11
3.4.4 Docker Compose	12
3.5 Despliegue	12
3.5.1 Amazon Web Services	13
3.5.2 Google Cloud Platform	13
3.5.3 VPS	13

4	Notas sobre la Redacción del Texto	15
4.1	Preliminares	15
4.1.1	Cubierta	15
4.1.2	Portada	15
4.1.3	Resumen	15
4.1.4	Prefacio o Introducción	15
4.1.5	Índice y/o Índice general	16
4.1.6	Lista de ilustraciones y tablas	16
4.1.7	Lista de abreviaturas y símbolos	16
4.1.8	Glosario	16
4.2	Texto principal	16
4.3	Bibliografía	16
4.4	Anexos	16
5	Ejemplo de Capítulo	17
5.1	Ejemplo de sección	17
5.1.1	Ejemplo de subsección	18
5.2	Elementos del texto	18
5.2.1	Figuras	18
5.2.2	Tablas	18
5.2.3	Listados de programas	18
5.2.4	Ecuaciones	20
5.2.5	Ejemplos	21
5.2.6	Lemas, teoremas y similares	21
5.2.7	Resúmenes	21
	Resumen de Teoría de Información	21
5.3	Una nueva sección después del resumen	23
	Problemas Propuestos	25
	Anexo	27
5.4	Señales: definición y clasificación	27
5.4.1	Clasificación de señales	27
6	Ejemplo de Capítulo de Problemas	29
6.1	Ruido y sensibilidad	29
6.1.1	Temperatura y figura de ruido	29
6.1.2	Sensibilidad	29
6.2	No linealidad	30
P. 6.1	Radioenlace del servicio fijo a 13 GHz	30
P. 6.2	Radioenlace del servicio fijo a 13 GHz	31
6.3	...Ruido y sensibilidad	32
6.3.1	...Temperatura y figura de ruido	32
6.3.2	...Temperatura y figura de ruido	33
6.3.3	...Temperatura y figura de ruido	33
7	Estilo tipográfico LibroETSI	35
7.1	Bloque 0	35
7.2	Bloque 1: Aspectos generales	35
7.3	Bloque 2: Idioma, Codificación y Fuentes	36
7.3.1	Idioma	36
7.3.2	Codificación	37
7.3.3	Fuentes del texto y comandos asociados	37
7.3.4	Fuentes matemáticas y símbolos	39
7.4	Bloque 3: Tabla de contenidos	39
7.5	Bloque 4: Estilos de Páginas y Títulos	41

7.6	Bloque 5: Gestión general del documento	44
7.6.1	Apéndices	44
7.6.2	Colores	44
7.6.3	Aspectos genéricos de tratamiento del texto	45
7.6.4	Elementos flotantes	46
7.6.5	Gestión de índices alfabéticos y glosario	47
7.6.6	Ajustes en fórmulas	47
7.6.7	Hiperenlaces	48
7.6.8	Listado de códigos	48
7.6.9	Entornos, teoremas y similares	49
7.6.10	Otros comandos	50
7.7	A modo de conclusión	50
	<i>Índice de Figuras</i>	51
	<i>Índice de Tablas</i>	53
	<i>Índice de Códigos</i>	55
	<i>Bibliografía</i>	57
	<i>Índice alfabético</i>	59
	<i>Glosario</i>	61

1 Introducción

I am going into an unknown future, but I'm still all here, and still while there's life, there's hope.

JOHN LENNON, '70s

Durante los últimos años, la sociedad ha experimentado un auge en el empleo de la Inteligencia Artificial (IA) en diferentes ámbitos. La gran capacidad de especialización en un problema que concreto que presentan estos sistemas, junto con la gran cantidad de datos que se generan en la actualidad, han hecho que la IA se haya convertido en una herramienta muy útil en la resolución de problemas complejos.

Estas características han fomentado el empleo de soluciones basadas en IA, obteniendo resultados aceptables para problemas que de otro modo serían irresolubles.

De forma paralela, se observa un crecimiento en el interés por el uso de estas tecnologías. Muchos sistemas empiezan a aparecer en el día a día de usuario promedio, como por ejemplo: [1]

- Sistemas de búsqueda y recomendación
- IA generativa de texto, imágenes o audio
- Sistemas de predicción de eventos
- Asistentes de voz
- Vehículos autónomos
- Compras personalizadas

Sin embargo, la investigación sobre la aceptación de tecnologías que incluyen IA está aún en curso. Algunos estudios sugieren que en ciertos escenarios culturales, la necesidad de contacto humano no puede ser replicada o reemplazada. [3]

Este trabajo busca adentrarse en los límites sobre qué puede hacer un modelo de IA, en concreto en el campo de clasificación de audios. El objetivo es crear un modelo que sea capaz de clasificar audios en diferentes categorías, identificadas por emociones humanas.

Una vez creado el modelo, se aborda el problema de despliegue del modelo en un entorno de producción, para que pueda ser utilizado por usuarios finales. En un caso real de uso, podría diseñarse una interfaz o una API por ejemplo, de modo que la interacción con el modelo se adapte a las requerimientos del proyecto.

Debido a esta clara diferenciación, la memoria ha sido dividida en dos partes, que darán nombre a los capítulos principales: **Modelado** y **Despliegue**. En el capítulo 2 se aborda el problema de modelado, mientras que en el capítulo 4 se aborda el problema de despliegue.

1.1 Motivación

La idea de realizar un modelo capaz de diferenciar emociones humanas a partir de audios surge de la participación en un proyecto anterior. Este proyecto, a grandes rasgos, buscaba crear un modelo que sirviese de ayuda a personas a detectar emociones muy concretas para el caso de aplicación.

Por diversas circunstancias, el proyecto no pudo avanzar correctamente y acabó siendo abandonado. Sin embargo, un año más tarde y tras pensar en abordar de nuevo el problema, con un enfoque mucho más abierto, las sensaciones fueron muy positivas

En primer lugar, el avance de la tecnología a lo largo del tiempo ha permitido resolver problemas que antes parecían imposibles. En concreto, ha sido notable la diferencia en cuanto a todo lo relacionado con el modelo, dataset, plataforma de desarrollo, hosting, etc.

En segundo lugar, durante este tiempo, gracias al aprendizaje adquirido en el Máster y a la experiencia laboral, la problemática ha podido ser abordada con más madurez y conocimiento de las tecnologías existentes. Contar con algo de experiencia junto con una visión más amplia de las herramientas disponibles, permite dislumbrar opciones que antes eran invisibles. Además, enfocar un problema conociendo la pila completa ayuda a dividir en bloques la solución y a buscar alternativas para cada uno de ellos en caso de que sea necesario.

Por otro lado, al ser un trabajo con fines de aprendizaje, se eliminaron varias restricciones impuestas durante la realización del proyecto. Para ilustrar cómo limitaban estas restricciones y el resultado obtenido al haberlas modificado, se dedica un apartado para cada una de ellas.

Dataset

En el anterior proyecto, la sensibilidad de los datos era especialmente elevada. Esto generaba diversos problemas, ya que en ningún caso podrían salir de las instalaciones de la empresa.

Por otro lado, otro obstáculo fue la creación del dataset, ya que los datos debían ser recogidos y etiquetados por la propia empresa. Esto puede parecer una liberación de carga de trabajo en un principio. Sin embargo, al no disponer de datos de entrada, es difícil avanzar en el desarrollo del modelo.

En este proyecto, al no estar diseñando la solución para un dataset concreto, se puede realizar una búsqueda más general en bancos de datos públicos y elegir alguno similar al objetivo. De este modo se podría haber avanzado en el desarrollo del modelo, mientras se recogían los datos por parte de la empresa. El aprendizaje obtenido en este asunto es que es posible avanzar en el desarrollo del modelo, aprendiendo las técnicas y requisitos necesarios para el problema, con un dataset semejante, sin necesidad de esperar a tener los datos finales.

Enfoque

Durante el proyecto, fue difícil encontrar un método para abordar el problema. La búsqueda de proyectos que resolviesen problemas similares no fue muy fructífera, debido a la particular naturaleza del problema.

De forma similar a lo comentado en el párrafo anterior, un mejor acercamiento hubiese sido encontrar un problema similar y adaptarlo poco a poco al objetivo. La mejora en el estado del arte y una mayor madurez en el conocimiento de la materia han permitido realizar búsquedas de soluciones que pueden ser adaptadas al problema.

Además, la aparición de herramientas de mayor alto nivel permiten un acercamiento más paulatino al problema, pudiendo introducirnos en detalles más concretos más adelante, también con los datos finales. El enfoque de lo general a lo particular, junto con la división por bloques, permite avanzar en el desarrollo de cualquier proyecto, pero de nuevo, la experiencia ayuda en estas tareas.

Hardware

Uno de los principales problemas que presenta el desarrollo de modelos de IA es la necesidad de una alta capacidad de cómputo. Otros inconvenientes como la necesidad de una gran cantidad de datos pueden ser apaciguados empleando técnicas de Data Augmentation, pero para entrenar un modelo complejo es necesaria una gran capacidad de cómputo.

No todos los problemas requieren de soluciones extremadamente complejas, de hecho en el mundo del Machine Learning (ML) a veces las soluciones más simples ofrecen mejores resultados. Sin embargo, como se verá más adelante, la complejidad del problema requiere de un modelo complejo, lo que nos obliga a disponer de hardware con gran capacidad de computación en paralelo si queremos obtener resultados en un tiempo razonable.

Hoy en día existen diversas alternativas para entrenar grandes modelos aunque no dispongas de un gran equipo (el crecimiento en el número de alternativas viene motivado a su vez por el crecimiento en el interés en la IA). Continuando con las reflexiones anteriores, en este caso la solución ha sido emplear una plataforma de entrenamiento en la nube, brindando la posibilidad de entrenar modelos complejos sin necesidad de disponer de un equipo dedicado o desgastar en exceso un equipo personal, todo ello por un coste reducido.

Volviendo al proyecto original, es necesario recordar que esta solución no podría ser aplicada debido a que los datos no podrían salir de las instalaciones de la empresa. De todos modos, puede servir para iniciar en el proceso de investigación y creación de modelos alternativos, con datasets alternativos.

Entorno de despliegue

Otro punto problemático residía en torno al entorno de despliegue. Ante un gran problema, difícil de dividir a simple vista, el no disponer de una visualización del resultado final genera malestar y dudas que no favorecen al desarrollo del proyecto. Para que no ocurriera esto, se ha optado por buscar como resultado gráfico final lo mínimo necesario por una persona para poder utilizar el modelo.

Más allá de la interfaz, el paradigma de despliegue de cualquier software requiere de un entorno robusto que asegure su funcionamiento. El aprendizaje recibido durante el máster sobre contenedores, ha permitido utilizarlos para encapsular el modelo como una aplicación web y desplegarlo en una plataforma de hosting, asegurando su funcionamiento.

La conclusión en cuanto a este apartado es que, para poder pensar en un producto final, con una interfaz gráfica perfilada y plena funcionalidad, es necesario primero probar una versión más simple, con funcionalidad reducida, pero que permita validar el modelo y el despliegue. Teniendo esto en mente, es posible avanzar en el desarrollo del modelo, a la vez que se imaginan las distintas posibilidades para el caso de uso concreto.

1.2 Planificación

Una vez definido el contexto en el que surge esta idea, vamos a definir cómo se va a desarrollar el proyecto. Debemos definir el objetivo y el alcance del proyecto, para no perder la visión del mismo y poder evaluar el resultado final.

Al mismo tiempo, se incluye un listado de requisitos, que ayudarán a dar soporte a la definición del objetivo.

Además, se incluye la planificación temporal inicial del proyecto, en la que destacan varios hitos importantes.

1.2.1 Objetivo y alcance

El **objetivo** principal de este proyecto consiste en la creación y el despliegue de un modelo clasificador de audios según emociones humanas.

Al no ser un proyecto destinado para un caso de uso concreto (además de muchas otras limitaciones que presenta), el alcance del proyecto es un poco más abierto. Definimos el **alcance** del proyecto como la creación de una primera versión de una aplicación que implemente la mínima funcionalidad. Debe poder permitir utilizar el modelo de un modo sencillo, sin necesidad de tener conocimientos técnicos, debido a que en un caso real de uso, el usuario final no tendría por qué tener conocimientos técnicos.

1.2.2 Requisitos

Para acompañar al objetivo y el alcance, se ha diseñado una tabla de requisitos, separados por categorías:

Requisitos funcionales

- **F.1:** El sistema debe ser capaz de clasificar audios en diferentes categorías.
- **F.2:** El sistema debe ser capaz de recibir audios en formato WAV.
- **F.3:** Debe ser accesible desde múltiples dispositivos.
- **F.4:** El modelo permitirá ser actualizado periódicamente con nuevos datos de entrenamiento.

Requisitos operacionales

- **O.1:** La respuesta del sistema debe ser inferior a 5 segundos.
- **O.2:** El sistema debe ser robusto en entornos de grabación de alto ruido.
- **O.3:** El sistema debe ser accesible desde cualquier lugar y en cualquier instante.
- **O.4:** El coste computacional de la inferencia del sistema debe ser reducido.

Requisitos de diseño

- **D.1:** La implementación del sistema se realizará en Python.
- **D.2:** El despliegue del sistema será desplegado mediante contenedores.
- **D.3:** El equipo de desarrollo será un portátil personal.
- **D.4:** El sistema se ejecutará en un equipo con 2vcpu y 4GB de RAM.

Requisitos de seguridad

- **S.1:** Los audios no podrán abandonar nunca las instalaciones.
- **S.2:** La implementación web debe realizarse sobre un servidor seguro.
- **S.3:** El sistema debe ser robusto ante ataques de denegación de servicio.

Muchos de estos requisitos son utilizados a modo de muestra, pero no tienen mayor relevancia en el desarrollo del proyecto. Sin embargo, han servido para imaginar cómo podría ser el sistema en un caso de uso real, además de problemas que podrían surgir en caso de que el sistema se desplegara en un entorno real. Estos requisitos nos ayudan a dar forma a la solución final, entendiendo cómo podríamos mejorar su funcionalidad, además de su robustez.

Es interesante comentar que, aunque no es alcance para este proyecto, los requisitos de seguridad pueden llegar a ser los más críticos en un caso de uso real. En este caso, se ha optado por no profundizar en este aspecto, pero desde luego no es un detalle que deba pasarse por alto.

1.3 Diseño de la solución

Previamente, se ha explicado de qué trata el problema y de dónde surge la idea. En esta sección, intentamos responder a la pregunta de cómo se ha abordado el problema.

Siguiendo el famoso dicho, divide y vencerás, se ha optado por dividir el problema en dos bloques principales: **Creación del modelo** y **Despliegue del modelo**. De este modo, podemos centrarnos en cada uno de los bloques por separado, sin perder la visión del proyecto.

Los dos bloques son lo suficientemente importantes y están separados entre sí lo suficiente como para que puedan ser abordados por separado y no interfieran entre sí.

En un proyecto real, esta división sería apropiada para separar dos grupos de trabajo. Se podría designar cada bloque a un grupo de trabajo o departamento interno, de modo que cada uno de ellos se encargue únicamente de su parte. Conseguiríamos así una mayor especialización en cada uno de los bloques, además de una mayor eficiencia en el desarrollo del proyecto.

Como en este caso solo hay una persona trabajando en el desarrollo de la solución, la paralelización no influye directamente en el tiempo de desarrollo. Sin embargo, pensarlo de este modo ayuda a:

- Dividir la complejidad del problema a la mitad en primera instancia.
- Focalizar el objetivo de cada bloque.
- Independizar los bloques en caso de que uno de ellos no pueda ser completado.

1.3.1 Creación del modelo

El primer bloque, **Creación del modelo**, se centra en la creación de un modelo capaz de clasificar audios según emociones humanas. El objetivo de este bloque será crear un modelo que pueda clasificar los audios con cierta precisión entre un conjunto de emociones previamente definidas.

Algunos aspectos que se tendrán en cuenta en este bloque son:

- Elección del dataset de entrenamiento
- Estudio del estado del arte
- Elección de la arquitectura del modelo
- Entrenamiento del modelo
- Evaluación del modelo
- Exportación del modelo

1.3.2 Despliegue del modelo

El segundo bloque, **Despliegue del modelo**, se centra en la creación de una aplicación que permita utilizar el modelo creado en el bloque anterior. El objetivo de este bloque será utilizar el modelo de un modo determinado, ponerlo a disposición del usuario final, y que pueda ser desplegado en cualquier entorno.

Algunos aspectos relevantes en cuanto a este bloque son:

- Diseño de la interfaz
- Despliegue de la aplicación
- Hosting de la aplicación
- Pruebas de funcionamiento

2 Diseño del modelo

I don't know where I'm going from here, but I promise it won't be boring.

DAVID BOWIE

2.1 Introducción

Aquí voy a explicar algo muy bonito y divertido adsf

3 Despliegue del modelo

Less is more.

LUDWIG MIES VAN DER ROHE

3.1 Problemática y solución

Nos encontramos en esta situación: el modelo ha sido entrenado o está siendo desarrollado por otro equipo. Nuestro trabajo consiste en implementarlo en un entorno de producción para que pueda ser utilizado por los usuarios finales.

Este problema puede ser abordado de varias formas, y la solución estará altamente condicionada por los requisitos del proyecto. Al no tener ningún requisito impuesto para este trabajo, surgen muchos interrogantes cuya respuesta no es trivial:

- **Aspecto final de la solución:** ¿Cómo se va a utilizar el modelo? ¿Qué tipo de interfaz se va a utilizar? ¿Qué tipo de dispositivo se va a utilizar?
- **Requisitos de la solución:** ¿Qué requisitos de rendimiento tiene la solución? ¿Qué requisitos de seguridad tiene la solución? ¿Qué requisitos de escalabilidad tiene la solución?
- **Arquitectura de la solución:** ¿En qué lenguaje se va a implementar la solución? ¿Qué tipo de arquitectura se va a utilizar? ¿Qué tipo de servidores se van a utilizar?

En este caso, se ha optado por simplificar el aspecto final de la solución para centrarnos en la implementación del modelo en un entorno de producción. Se ha decidido crear una interfaz web que permita a los usuarios finales interactuar con el modelo, y una implementación mediante contenedores Docker para facilitar el despliegue en cualquier entorno.

3.2 Enfoque escogido

En un principio se pensó en realizar inferencia en tiempo real, es decir, que la aplicación, una vez iniciada, estuviera grabando continuamente y realizando predicciones. Esta opción sin embargo, no ha podido ser llevada a cabo de forma exitosa debido al tiempo de respuesta que ofrece el servidor en el que se ha desplegado la aplicación.

Además, para implementarlo de forma correcta, habría que añadir diversos mecanismos que ayuden a filtrar los audios y nos permitieran extraer muestras que pudieran ser utilizadas para realizar predicciones. Esto implicaría detección de inicio y fin de actividad vocal, filtrado de silencios, etc. La complicación de este proceso, unido al tiempo de respuesta del servidor, ha hecho que se descarte esta opción.

Finalmente, contamos con una aplicación web que permite a los usuarios finales grabar audios y obtener una predicción de la clase a la que pertenece el audio. La aplicación está pensada para ser utilizada en una sola dirección: el usuario debe iniciar la grabación, grabar un mensaje, parar la grabación y esperar el resultado. Una vez obtenido el resultado, puede volver a grabar otro mensaje.

La aplicación es accesible desde cualquier dispositivo que tenga un navegador web a través de la dirección <https://www.classifier-web.com/>.

3.3 Aplicación web

Una forma sencilla de crear una interfaz que sea accesible desde cualquier dispositivo es crear una aplicación web.

Aunque el modelo creado puede ser integrado utilizando cualquier lenguaje de programación, se ha optado por utilizar Python para la implementación de la aplicación web. Python cuenta con una gran cantidad de librerías que facilitan la implementación de aplicaciones web, como Flask o Django, quizás las más populares.

Se ha optado por utilizar Flask, debido a que es una librería más ligera que Django y a que es más sencilla de utilizar. Además, contamos con cierta experiencia previa en el uso de Flask, lo que nos permite acelerar el desarrollo de la aplicación.

3.3.1 Flask

Flask es un microframework para Python que permite crear aplicaciones web de forma sencilla.

Está diseñado para ser extensible, por lo que es posible añadirle funcionalidades mediante extensiones, aunque en este caso no vamos a utilizar ninguna. Sin embargo, estas extensiones de alto nivel nos abren las puertas a posibles líneas futuras, como lecturas de bases de datos, autenticación de usuarios, etc.

Utilizar Python para la creación web no siempre es la solución idónea, ya que existen otros lenguajes de programación que están más orientados a la creación de aplicaciones web. Sin embargo, si no se tiene experiencia previa en estos lenguajes, o el objetivo es lanzar una aplicación web de forma rápida, Flask es idóneo. No estamos exentos de tener que crear plantillas en otros lenguajes propios de la web, como HTML, CSS o JavaScript, pero Flask nos permite crear una aplicación web funcional en muy poco tiempo.

3.3.2 Estructura de la aplicación

La estructura de la aplicación es muy sencilla, y se puede ver en la figura `"""poner figura"""`

Primero el modelo es cargado en memoria, y se crea una instancia de Flask.

El modelo es cargado en memoria para evitar tener que cargarlo cada vez que se realiza una predicción, lo que se traduciría en un aumento del tiempo de respuesta de la aplicación. Esto puede ralentizar sin embargo el arranque de la aplicación, pero es una operación que se realiza una única vez, por lo que no es un problema.

Posteriormente se crean las rutas de la aplicación, que son las direcciones a las que se puede acceder desde un navegador web.

Contamos con la ruta principal, que es la que se utiliza para cargar la página principal de la aplicación, y la ruta de predicción, que es la que se utiliza para realizar las predicciones.

La ruta principal simplemente carga un fichero HTML que contiene el código de la página principal.

La ruta de predicción es llamada internamente mediante una petición POST cuando un usuario termina una grabación. La grabación se guarda localmente en el servidor momentáneamente para que el modelo pueda realizar la predicción sobre ella, y posteriormente se borra, por cuestiones de espacio y privacidad.

3.3.3 Interfaz web

El desarrollo de interfaces web es un mundo aparte, y no es el objetivo de este trabajo crear una interfaz especialmente atractiva, sino más bien que nos proporcione la funcionalidad básica. Existen desarrolladores especializados únicamente en el desarrollo de interfaces web, y es un campo que requiere de un conocimiento muy amplio, además de experiencia.

Debido a tratar esta parte como algo secundario, sumado a la falta de conocimiento acerca del manejo de audios en la web, se ha optado por basar la interfaz en trabajo previo realizado por otros desarrolladores. En concreto, este proyecto ha utilizado como base `"""insertar referencia a la interfaz web"""`.

La interfaz web es muy sencilla, y se puede ver en la figura `"""poner figura"""`. Contiene lo básico para que un usuario pueda realizar la grabación de un audio y obtener una predicción de la clase a la que pertenece el audio.

3.4 Aplicación Flask en producción

Flask integra un servidor web de desarrollo, que es el que se utiliza por defecto cuando se lanza la aplicación, llamado Werkzeug. Este servidor es muy sencillo de utilizar, pero no está pensado para ser utilizado en producción, ya que no está optimizado para ello.

3.4.1 Gunicorn

Para lanzar la aplicación en producción, se ha optado por utilizar Gunicorn, un servidor web HTTP WSGI para Python. Es uno de los servidores más utilizados para lanzar aplicaciones Flask en producción, y es el que se recomienda en la documentación oficial de Flask.????????????

Gunicorn es un servidor web que se encarga de gestionar las peticiones HTTP que llegan a la aplicación, y de lanzar procesos de la aplicación para atender estas peticiones. Esto permite que la aplicación pueda atender varias peticiones simultáneamente, lo que se traduce en un aumento del rendimiento de la aplicación.

Para lanzar un servicio Flask con Gunicorn, simplemente hay que ejecutar el siguiente comando: ""
Insertar comando ""

Este comando lanzará un servidor web en el puerto 8000, que es el puerto por defecto de Gunicorn.

El siguiente paso es configurar un servidor web que actúe como proxy inverso, para que las peticiones HTTP que lleguen al servidor web sean redirigidas al servidor Gunicorn.

3.4.2 Traefik

Para configurar el servidor web que actúe como proxy inverso, se ha optado por utilizar Traefik, un servidor web que permite realizar balanceo de carga y que actúa como proxy inverso.

Aunque Nginx es quizás el servidor web más utilizado para realizar esta tarea, se ha optado por utilizar Traefik principalmente por su facilidad de configuración. Es comentado que Nginx es más rápido que Traefik, a la vez que ofrece más funcionalidades, pero para este caso con una configuración básica es suficiente.

La mayor ventaja que nos ha brindado Traefik es la facilidad de generar certificados SSL para la aplicación, lo que nos permite utilizar HTTPS. Esto es importante, ya que si no se utiliza HTTPS, los navegadores web no permiten acceder al micrófono del dispositivo, lo que hace imposible la grabación de audios.

No es una tarea difícil de realizar correctamente para un desarrollador experimentado mediante un servidor web como Nginx, pero es mucho más sencillo de realizar con Traefik, y además, al contar con poca experiencia en este campo, nos ha permitido solventar este problema de forma rápida y sencilla. Además, Traefik aún está dando sus primeros pasos, y está ganando popularidad entre desarrolladores, por lo que quizás en un futuro sea una alternativa a Nginx también en entornos reales de producción.

"" insertar foto de <https://monitor.classifier-web.com> indicando admin:admin""

3.4.3 Docker

Para facilitar el despliegue de la aplicación en cualquier entorno, se ha optado por utilizar contenedores Docker. Esta tecnología permite encapsular una aplicación y sus dependencias en un contenedor, que puede ser ejecutado en cualquier entorno que tenga instalado Docker. De este modo nos aseguramos que únicamente tenemos que preocuparnos de que el entorno tenga instalado Docker.

Esta tecnología ayuda a eliminar muchos problemas a la hora de desplegar servicios, pero incorpora otros de los que hay que ser conscientes. En particular, Docker presenta un problema de seguridad, ya que los contenedores son ejecutados por defecto con privilegios de root. Esto implicaría que si un atacante consigue acceder al contenedor, puede tener acceso a todo el sistema.

Este problema se ha solventado creando un usuario no privilegiado dentro del contenedor al construir la imagen de la aplicación, y ejecutando la aplicación con este usuario. Sin embargo, las implicaciones de seguridad de Docker son un tema muy amplio y precisamente pueden llegar a ser determinantes para no utilizar esta tecnología en entornos de producción con requisitos de seguridad muy estrictos. No es el caso de este trabajo, pero es un tema que hay que tener en cuenta y debería ser estudiado en profundidad antes de utilizar Docker en entornos de producción.

A pesar de ello, las ventajas que ofrece Docker son muy interesantes, y es una tecnología que ha ganado mucha popularidad en los últimos años. Las principales ventajas que ofrece son las siguientes:

- **Portabilidad:** Docker permite encapsular una aplicación y sus dependencias en un contenedor, que puede ser ejecutado en cualquier entorno que tenga instalado Docker.

- **Escalabilidad:** Docker permite crear múltiples contenedores de una misma aplicación, lo que permite escalar la aplicación de forma horizontal.
- **Aislamiento:** Docker permite aislar una aplicación y sus dependencias en un contenedor, lo que permite que la aplicación no se vea afectada por otras aplicaciones que se estén ejecutando en el mismo entorno.
- **Rapidez:** Docker permite crear imágenes de aplicaciones de forma rápida, lo que permite desplegar aplicaciones en muy poco tiempo.

3.4.4 Docker Compose

Docker Compose es una herramienta que permite definir y ejecutar aplicaciones Docker de forma sencilla. Permite definir las imágenes de los contenedores, las redes, los volúmenes, etc., en un fichero YAML, y ejecutarlos con un único comando.

Es especialmente útil cuando se tienen varias aplicaciones que dependen unas de otras, ya que permite definir todas las aplicaciones en un único fichero. En nuestro caso contamos solo con dos contenedores, pero crear un fichero Docker Compose nos permite definirlos de forma sencilla, construir las imágenes con las dependencias que nosotros definamos y levantar el despliegue con un único comando.

La sintaxis general de un fichero Docker Compose consiste en definir los servicios que se van a utilizar, las imágenes que se van a utilizar para cada servicio, los volúmenes que deben ser creados, las redes, variables de entorno, etc.

En este caso, algunos aspectos a destacar de la definición del fichero son los siguientes:

- **Servicios:** Han sido definidos dos servicios, uno para la aplicación Flask y otro para el servidor Traefik.
- **Imagen:** Se ha utilizado la imagen oficial para el servidor Traefik, y una imagen de Python personalizada con las dependencias necesarias para la aplicación Flask.
- **Volúmenes:** Para el servicio de Traefik se han definido varios volúmenes para almacenar los certificados SSL y la configuración de Traefik.
- **Redes:** No ha sido necesario definir ninguna red, ya que por defecto Docker Compose crea una red interna que es suficiente para que los contenedores se comuniquen entre sí.
- **Variables de entorno:** Han sido definidas varias variables principalmente para el servicio de Traefik, de modo que pueda servir la aplicación Flask con HTTPS.

En el apéndice ?? se puede ver el fichero Docker Compose completo. Además, el proyecto completo está disponible en GitHub, en la siguiente dirección: <https://github.com/antaramol/classifier-web.git>.

3.5 Despliegue

Una vez que la aplicación está lista para ser desplegada, es necesario elegir un entorno de producción.

Varias opciones han sido probadas para este trabajo, ya que no se tenía ninguna restricción en cuanto al entorno de producción. Al no tener un servidor propio, se ha optado por utilizar servicios de terceros, que ofrecen servidores virtuales a un precio muy asequible.

De nuevo, igual que comentábamos en el apartado anterior, no es posible utilizar un ordenador personal para este servicio, ya que debería estar conectado todo el tiempo. Además, existen cada vez más opciones de hosting y es muy interesante utilizarlas, ya que nos permiten centrarnos en el desarrollo de la aplicación y no en la gestión del servidor.

Esta es la principal ventaja, nos olvidamos de gestionar la infraestructura. Además muchos servicios permiten gran facilidad para escalar horizontalmente, lo que nos permite aumentar la capacidad de la aplicación de forma sencilla.

Sin embargo, el principal problema es el coste, ya que estos servicios no son gratuitos. Es un inconveniente que, sobre todo para iniciados en este mundo, puede ser determinante para no utilizar estos servicios.

Varias opciones han sido probadas para este trabajo, ya que cada una de ellas ofrece diferentes opciones y precios.

3.5.1 Amazon Web Services

Amazon Web Services (AWS) es una plataforma de servicios en la nube que ofrece servicios de computación, almacenamiento, bases de datos, etc. Esta ha sido la primera opción probada debido a su gran popularidad y que se contaba con cierto conocimiento previo.

"" Insertar gráfico de popularidad ""

AWS ofrece una gran cantidad de servicios, y es una de las plataformas más completas que existen. Este ha sido el principal problema, ya que está pensado para ser utilizado por empresas que necesitan una gran cantidad de servicios por su fácil integración entre ellos. Como este trabajo solo necesita un servidor virtual, quizás AWS no sea la mejor opción.

En concreto, ha sido probado el servicio EC2, que ofrece servidores virtuales en la nube. Cuenta con una capa gratuita, que permite utilizar un servidor virtual de forma gratuita durante un año, pero con ciertas limitaciones.

La capacidad de cómputo en la capa gratuita es muy limitada, pero es suficiente para probar la aplicación. Sin embargo, el número de horas de uso es limitado, y una vez que se agotan, hay que pagar por cada hora de uso. Esto hace que no sea una opción viable para este trabajo, ya que el coste sería muy elevado. Este segmento está quizás pensado para pruebas puntuales de aplicaciones, pero no para desplegar aplicaciones en producción.

Sin embargo, sirvió para realizar diversas pruebas de funcionamiento, pruebas de rendimiento, etc., y para familiarizarse con este tipo de servicios.

3.5.2 Google Cloud Platform

El resultado es similar al de AWS, ya que Google Cloud Platform (GCP) está enfocado igualmente a grandes proyectos.

Además, la documentación de estos servicios es tan extensa que puede llegar a ser abrumadora para un desarrollador que no esté familiarizado con este tipo de servicios. Una mejor opción para este trabajo es utilizar servicios más sencillos, que estén pensados para pequeños proyectos y que sean más fáciles de utilizar, suavizando así la curva de aprendizaje.

La atracción hacia Google Cloud Platform es que ofrece una gran cantidad de crédito gratuito para utilizar sus servicios, lo que permite utilizarlos de forma gratuita durante un tiempo. El resultado es muy similar a AWS, esta vez se ha utilizado el servicio Compute Engine, que ofrece servidores virtuales en la nube.

3.5.3 VPS

Una opción más sencilla y económica es utilizar un VPS (Virtual Private Server), que es un servidor virtual que se encuentra alojado en un servidor físico. Este tipo de servidores cuentan con una ventaja con respecto a los anteriores, y es que el precio es fijo, y no depende del uso que se haga del servidor.

En concreto, se ha utilizado el servicio de VPS de Contabo, recomendado por un compañero de la universidad. Este servicio ofrece servidores virtuales a un precio muy asequible, y con una gran cantidad de recursos.

Es muy sencillo desplegar una instancia de un servidor virtual, a la que podemos conectarnos mediante SSH. Si a esto le sumamos la facilidad que nos ofrece Docker Compose para desplegar la aplicación, junto con herramientas de edición como VSCode que nos permiten conectarnos y editar los ficheros de la aplicación de forma remota, el resultado es muy satisfactorio.

Contabo además nos ofrece muchas opciones que nos ayudan a perfilar el resultado final. En nuestro caso, se ha utilizado el área de manejo de DNS para configurar el dominio de la aplicación, de modo que podemos otorgarle un nombre más amigable a la aplicación. Contabo permite crear registros DNS de forma sencilla, y además ofrece un servicio de DNS dinámico, que permite asignar un nombre de dominio a una IP dinámica, que es la que nos ofrece el servidor virtual.

"" insertar imagen de dns zone de contabo ""

Para esta aplicación, se ha elegido un servidor de 4 núcleos, 8 GB de RAM y 50 GB de almacenamiento, suficiente para albergar el modelo. Si en un futuro se necesitara más capacidad, se podría escalar horizontalmente, creando más instancias de la aplicación y balanceando la carga entre ellas. Para esta implementación, quizás sería más conveniente estudiar en profundidad los servicios de AWS o GCP, ya que ofrecen más facilidades para escalar horizontalmente, pero para el propósito de este trabajo, es más que suficiente.

4 Notas sobre la Redacción del Texto

Texto elaborado por Marisa Balsa, de la Biblioteca de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla

Se recomienda consultar la norma UNE 50136:1997: Documentación, presentación de tesis y documentos similares. Por otro lado, existen recursos en línea que ayudan a organizar todo el proceso de elaboración del Trabajo. El personal de la Biblioteca imparte formación y asesoramiento sobre su uso. Para establecer la estructura y orden de los datos y partes del Trabajo Fin de Grado nos basamos en la Normativa de los Trabajos Fin de Grado de la Escuela Técnica Superior de Ingeniería de Sevilla (Aprobada en la sesión de la Junta de Escuela de 12 de julio de 2013, modificada en Junta de Escuela de 05 de febrero de 2014) y en la norma UNE 50136:1997: Documentación, presentación de tesis y documentos similares.

4.1 Preliminares

4.1.1 Cubierta

Deberá utilizarse la plantilla aprobada por la Junta de Escuela del 25/4/2014 que se describe en este texto y cuyo modelo puede tomarse de este documento. Debe contener todos los datos que se especifican y en el orden y modelo que se establece: nombre completo del autor, título del Trabajo, nombre completo del tutor, nombre del Departamento, fecha, Grado al que se opta, etc.

4.1.2 Portada

Igualmente deberá utilizarse la plantilla... y contener todos los datos que figuran en ésta, en el orden especificado: el nombre de la Escuela, la titulación y, en su caso, la intensificación, el título del TFG, los nombres del autor, del tutor(es) y, en su caso, del ponente, el Área de Conocimiento, el Departamento y el año de ejecución del proyecto.

4.1.3 Resumen

Texto que muestra de forma abreviada y precisa el contenido del trabajo. Se podrá también incorporar una versión en inglés que se colocará a continuación precedida del término Abstract.

4.1.4 Prefacio o Introducción

Debe incluir una breve explicación de las razones que han llevado a la realización del Trabajo, el propósito y los objetivos que se pretenden, el ámbito, alcance y límites de la investigación, así como la metodología empleada y, si se considera oportuno, un avance de las conclusiones alcanzadas.

4.1.5 Índice y/o Índice general

Tabla de contenidos donde se reflejan todas las partes del Trabajo y sus anexos, si los hubiera. Deben aparecer los títulos, en su orden y con indicación de la página en la que se pueden encontrar. Si el Trabajo consta de varios volúmenes, cada uno deberá llevar su propio índice, pero se debe incluir también un índice general.

4.1.6 Lista de ilustraciones y tablas

Si el Trabajo incluye ilustraciones y tablas se puede añadir un listado que incluya el número identificativo que figura dentro del texto, la leyenda y el número de la página en la que se encuentra. También es conveniente mencionar los datos sobre las fuentes de donde se han obtenido dichas ilustraciones si no se han incluido en el propio texto de la memoria.

4.1.7 Lista de abreviaturas y símbolos

El Trabajo debe contener las abreviaturas y símbolos internacionalmente reconocidos. Si se incorporan unidades, abreviaturas o acrónimos que puedan ser poco conocidos se deberán explicar brevemente en estas listas.

4.1.8 Glosario

Los términos que requieran definición o explicación se deberán incorporar en un glosario.

4.2 Texto principal

Para su elaboración es importante tener en cuenta su distribución en capítulos y secciones numeradas, siguiendo la plantilla elaborada por... El texto debe comenzar con una introducción que muestre las investigaciones previas existentes sobre el tema y destacar los objetivos y métodos seguidos para llevar a cabo la investigación o análisis del tema tratado. Para finalizar se deben escribir las conclusiones que deben estar en relación con los objetivos marcados previamente.

4.3 Bibliografía

Debe contener las referencias bibliográficas de los documentos consultados para demostrar las bases del trabajo realizado y avalar los datos incorporados y citados en el texto. Se elaborará de forma normalizada, para lo que se aconseja utilizar la norma UNE vigente (actualmente la “UNE ISO 690:2013. Información y documentación. Directrices para la redacción de referencias bibliográficas y de citas de recursos de información”). Para la elaboración de esta parte del Trabajo se recomienda consultar la Web de la Biblioteca de Ingeniería que contiene recursos, guías y ayudas para la elaboración de las referencias bibliográficas.

4.4 Anexos

Se puede incluir de esta forma material extenso utilizado en el trabajo, importante para justificar los resultados y las conclusiones obtenidas, pero que no es esencial para la comprensión del texto principal. Pueden ser datos estadísticos, legislación, etc. La paginación debe ser correlativa y continuar la del texto principal. Cada uno de los anexos debe identificarse con una letra mayúscula del alfabeto, comenzando por la letra A, precedida de la palabra Anexo

5 Ejemplo de Capítulo

Una de las virtudes del ingeniero es la eficiencia.

GUANG TSE

El formato de capítulo abarca diversos factores. Un capítulo puede incluir, además de texto, los siguientes elementos:

- Figuras
- Tablas
- Ecuaciones
- Ejemplos
- Resúmenes, con recuadros en gris, por ejemplo
- Lemas, corolarios, teoremas,... y sus demostraciones
- Cuestiones
- Problemas propuestos
- ...

En este capítulo se propone incluir ejemplos de todos estos elementos, para que el usuario pueda modificarlos fácilmente para su uso. Consulte el código suministrado, para ver cómo se escriben en L^AT_EX.

5.1 Ejemplo de sección

En la Figura 5.1 se incluye a modo de ejemplo la imagen del logo de la ETSI ¹. El código para que aparezca dicha imagen se muestra en el cuadro siguiente:

Si nos detenemos en los comandos que hemos utilizado, con `width` se controla el ancho, y se escala así el tamaño de la imagen. En L^AT_EX existen diversas opciones para situar la figura en la página: con `t` o `b` se le indica que las incluya arriba o abajo (top/bottom) y con `!` se le pide que la deje dónde está, tras el texto anterior.

Código 5.1 Código para incluir una figura.

```
\begin{figure}[htbp]
\centering
\includegraphics[width=3 cm]{capituloLibroETSI/figuras/logoESI.pdf}
\caption{Logo de la ETSI}
\label{fig:figura1}
\end{figure}
```

Para dar énfasis a algún texto, usamos `\emph`. Así, por ejemplo,

¹ Se usa aquí el package de acrónimos, que la primera vez define el acrónimo y ya luego sólo incluye el mismo. Esto facilita luego generar de forma automática la lista de acrónimos.

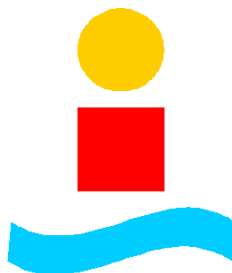


Figura 5.1 Logo de la ETSI.

No olvide intentar utilizar este formato en sus publicaciones de la ETSI

hace aparecer el anterior texto en itálica. Pero si escribiésemos, por ejemplo,

No olvide intentar utilizar este formato siempre en sus publicaciones de la ETSI

vemos cómo hemos destacado la palabra “siempre” en torno a su contexto. Para ello, hemos escrito, realmente, `\emph{siempre}` dentro de la frase original.

5.1.1 Ejemplo de subsección

Si se usaba `\section` para indicar una sección, se utiliza `\subsection` para una subsección.

5.2 Elementos del texto

5.2.1 Figuras

Además del tipo de figura que vimos anteriormente, el normal, podemos desear incluir una figura en modo apaisado ocupando toda la página. Para ello utilizamos el entorno de figura siguiente `\begin{sidewaysfigure}`, cuyo resultado se puede observar en la Figura 5.2.

Aunque puede optar por la forma que desee, en el fichero `notacion.sty` se incluyen definiciones para que pueda usar `\LABFIG{etiqueta}` y `\FIG{etiqueta}` para poner una etiqueta y hacer referencia a la misma luego. Además, está definido para que `\FIG{etiqueta}` incluya por delante el término Figura.

5.2.2 Tablas

A modo de ejemplo, Tabla 5.1 incluye un ejemplo de tabla. Al igual que con figura, si usa `notacion.sty` puede usar `\LABTAB{etiqueta}` y `\TAB{etiqueta}` para poner una etiqueta y una referencia, y el `\TAB{etiqueta}` ya incluye el nombre Tabla por delante.

Una alternativa al uso de estos comandos está representado por el uso del comando `\autoref{etiqueta}` que, en conjunción con el paquete `babel` genera automáticamente los nombres de Figura o Tabla, en función de la etiqueta correspondiente.

Tabla 5.1 Valores de parámetros.

Definición	notación	valor
Potencia transmitida (entregada a antena)	P_{et}	-5 a 20 dBm
Ganancia antenas	G	40.5 dBi

5.2.3 Listados de programas

Es muy habitual en nuestros documentos que tengamos que incluir listados de programas. Para ello, se propone la utilización de un paquete denominado `listings`. Se obtiene con él un listado como el mostrado en el Código 5.2 de MATLAB® siguiente:

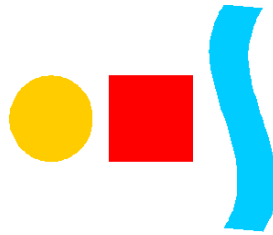


Figura 5.2 Logo de la ETSI.

Código 5.2 Representación de la función $\text{rect}(t - T/2)$.

```

clear all
close all
T = 1;
A = 1;
L = 100;
tstep = T/L;
t = 0:tstep:T-tstep;
g_t = A*ones(1,L);
figure(1);
subplot(211);
h=plot(t,g_t); axis( [0 T -A-0.1 A+0.1]);
set(h,'linewidth', 1.0);
ylabel('g(t)'), xlabel('t[s]'); grid on;

g_n = g_t;
subplot(212);
h=stem(g_n, '.', 'filled'); axis( [1 L -0.1 A+0.1]);
set(h,'linewidth', 1.0);
ylabel('g(n)'), xlabel('n');

```

También se puede generar en este caso una relación de los códigos usados en nuestro documento, de manera equivalente a la relación de figuras o tablas. Para ello, observar la correspondiente codificación en el fichero principal.

5.2.4 Ecuaciones

Para escribir expresiones matemáticas, como por ejemplo $2 + 2 = 4$, sólo hace falta que meta la expresión entre símbolos $\$$. En el fichero notacion.sty se incluyen muchas definiciones para facilitar la escritura de estas expresiones y de ecuaciones. Para escribir una ecuación, con una o más líneas, se aconseja utilizar `align`, como en el siguiente ejemplo, en las ecuaciones (5.1)-(5.3),

$$T = kT_b, \quad (5.1)$$

$$R_b = \frac{1}{T_b}, \quad (5.2)$$

$$D = \frac{1}{T} = \frac{R_b}{k} = \frac{R_b}{\log_2 M}. \quad (5.3)$$

Si no quiere numerar una línea, utilice la instrucción `\nonumber` antes de poner `\\` para escribir la siguiente línea. Y con `&` puede alinear las ecuaciones.

Un ejemplo más complejo de ecuaciones sería el siguiente: decimos que el vector aleatorio \mathbf{Z} es gaussiano si su función densidad de probabilidad conjunta viene dada por:

$$f_{\mathbf{Z}}(\mathbf{z}) = \frac{1}{(2\pi)^N |\mathbf{C}_{\mathbf{Z}}|^{1/2}} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m}_{\mathbf{Z}})^{\top} \mathbf{C}_{\mathbf{Z}}^{-1}(\mathbf{z}-\mathbf{m}_{\mathbf{Z}})} \quad (5.4)$$

con el vector media la matriz $\mathbf{m}_{\mathbf{Z}}$ y la matriz de covarianza real $\mathbf{C}_{\mathbf{Z}}$ ($2N \times 2N$) simétrica definida positiva dado por:

$$\mathbf{m}_{\mathbf{Z}} = \begin{bmatrix} \mathbf{m}_{\mathbf{X}} \\ \mathbf{m}_{\mathbf{Y}} \end{bmatrix}, \quad \mathbf{C}_{\mathbf{Z}} = \begin{bmatrix} \mathbf{C}_{\mathbf{X}} & \mathbf{C}_{\mathbf{XY}} \\ \mathbf{C}_{\mathbf{YX}} & \mathbf{C}_{\mathbf{Y}} \end{bmatrix} \quad (5.5)$$

con el vector ω dado por:

$$\omega = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_N \\ \omega_{N+1} \\ \vdots \\ \omega_{2N} \end{bmatrix} \quad (5.6)$$

Si no desea que se numere una ecuación puede poner asterisco, tanto en el entorno `equation` como `align`.

5.2.5 Ejemplos

Para incluir un ejemplo, utilice el entorno `\ejmp`, usando el entorno `\begin{ejmp}` y `\end{ejmp}`, y para la solución el entorno `\begin{sol}`.

Ejemplo 5.2.1 Calcule $2 + 2$.

Solución. Para resolver esto se puede utilizar que $1 + 1 = 2$, de la siguiente forma

$$2 + 2 = (1 + 1) + (1 + 1) = 4,$$

donde se ha contado, pruebe a utilizar los dedos de su mano, a cuatro.

Observad que antes de comenzar el ejemplo y tras su finalización se han incluido unos *filetes* a modo de resalte en el texto. En el caso de una serie de ejemplos, los entornos `\begin{ejmpn}` y `\begin{soln}`, junto con los entornos de cierre correspondientes, permiten que no existan estos filetes entre los ejemplos y soluciones intermedias de la serie.

5.2.6 Lemas, teoremas y similares

Se incluyen ejemplos de estos elementos de texto. Empezamos con la Definición 5.2.1 y la Propiedad 5.2.1:

Definición 5.2.1 (Suma) La suma es la operación que permite contar sobre un número, otro.

Propiedad 5.2.1 (Suma) *Los números enteros se pueden sumar.*

Lema 5.2.1 (Suma de 1 y 1) *La suma $1 + 1$ es igual a 2.*

Demostración. Ponga un dedo a la vista, junto a otro, y cuéntelos. ■

Teorema 5.2.1 (Suma) *La suma de cualquier número y dos es igual a la suma del mismo número más uno más uno.*

Demostración. Por inducción y el Lema 5.2.1. ■

Corolario 5.2.1.1 (Contables) *Los números enteros son contables.*

Demostración. Por el Teorema 5.2.1. ■

5.2.7 Resúmenes

Para incluir un resumen de una sección o un conjunto de secciones o en cualquier otro punto que consideremos interesante, se utiliza el entorno `\begin{Resumen}`, que admite como parámetro opcional un nombre que queramos asignarle al resumen. Por defecto, se denomina “Resumen”. Observar que se ha modificado la cabecera de las páginas impares. Una vez finalizado el resumen, con el comando `\end{Resumen}`, se recupera la anterior cabecera automáticamente. Los resúmenes que se deseen incluir aparecen en la tabla de contenidos como una sección sin numeración, con el nombre elegido o el nombre por defecto de Resumen. En el siguiente ejemplo hemos utilizado este parámetro opcional de nombre.

Resumen de Teoría de Información

Debido al considerable número de definiciones, teoremas y propiedades que hemos descrito en los apartados anteriores, vamos a presentar un resumen de los principales resultados, no necesariamente en el mismo orden que el expuesto anteriormente. Supondremos en este resumen que las variables aleatorias X , Y y Z son discretas, definidas en el alfabeto \mathcal{X} , \mathcal{Y} y \mathcal{Z} respectivamente.

Entropía de una variable aleatoria discreta

Se define la entropía $H(X)$ de una variable aleatoria discreta X , con función masa de probabilidad $p(x)$, en la forma:

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = \mathbb{E}[-\log p(X)]$$

1. Se cumple:

$$0 \leq H(X) \leq \log |\mathcal{X}|$$

con la igualdad en la izquierda si y sólo si $p_i = 1$ para algún $x_i \in \mathcal{X}$ y con la igualdad a la derecha si y sólo si la variable aleatoria está uniformemente distribuida; esto es, $p_i = 1/|\mathcal{X}|$ para todo i .

2. $H(X) = 0$ si y sólo si X es determinista.
3. $H(X) = H(p(x))$ es una función cóncava en $p(x)$.
4. Se define la *Función de Entropía Binaria* en la forma:

$$h_b(p) \stackrel{\text{def}}{=} -p \log p - (1-p) \log (1-p)$$

5. La función entropía binaria $h_b(p)$ es una función cóncava en p .
6. Si X y \hat{X} son dos variables aleatorias estadísticamente independientes igualmente distribuidas,

$$\Pr(X = \hat{X}) \geq 2^{-H(X)}$$

con la igualdad si y sólo si X tiene una distribución uniforme.

Entropía conjunta y entropía condicional

Definimos la *entropía conjunta* de las variables aleatorias X e Y , $H(X, Y)$ en la forma:

$$H(X, Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{1}{p(x, y)} = \mathbb{E}[-\log p(X, Y)]$$

Definimos la *entropía condicional* $H(X | Y)$ en la forma:

$$\begin{aligned} H(X | Y) &= \sum_{y \in \mathcal{Y}} p(y) H(X | Y = y) = \\ &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x | y) = \\ &= \mathbb{E}[-\log p(X | Y)] \end{aligned}$$

1. $H(X, Y) \leq H(X) + H(Y)$

con la igualdad si y sólo si X e Y son estadísticamente independientes.

- 2.

$$\begin{aligned} H(X | Y) &\leq H(X) \\ H(Y | X) &\leq H(Y) \end{aligned}$$

con la igualdad si y sólo si X e Y son estadísticamente independientes.

3. $H(X|Y) = 0$ si y sólo si X es una función de Y .

4. $H(X|X) = 0$

5. $H(X,Y) = H(Y) + H(X|Y)$

6. $H(X,Y) = H(X) + H(Y|X)$

7. $H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$

8. *Desigualdad de Fano* Sean X y \hat{X} dos variables aleatorias que toman valores en el mismo alfabeto \mathcal{X} . Se verifica:

$$H(X|\hat{X}) \leq h_b(p_e) + p_e \log(|\mathcal{X}| - 1)$$

Reglas de las cadenas

Sea \mathbf{X} un vector formado por las N variables aleatorias $X_i, i = 1, 2, \dots, N$.

1. *Regla de la cadena para la entropía*

$$\begin{aligned} H(X_1, X_2, \dots, X_N) &= \sum_{i=1}^N H(X_i | X_1, \dots, X_{i-1}) = \\ &= H(X_1) + H(X_2 | X_1) + \dots + H(X_N | X_1, \dots, X_{N-1}) \end{aligned}$$

2. *Regla de la cadena para la entropía condicional*

$$H(X_1, X_2, \dots, X_N | Y) = \sum_{i=1}^N H(X_i | X_1, \dots, X_{i-1}, Y)$$

3. *Regla de la cadena para la información mutua*

$$I(X_1, X_2, \dots, X_N; Y) = \sum_{i=1}^N I(X_i; Y | X_1, \dots, X_{i-1})$$

4. *Regla de la cadena para la Información Mutua Condicional*

$$I(X_1, X_2, \dots, X_N; Z | Y) = \sum_{i=1}^N I(X_i; Z | X_1, \dots, X_{i-1}, Y)$$

5.3 Una nueva sección después del resumen

Problemas Propuestos

Esto es un ejemplo de cómo incluir cuestiones y/o problemas al final de un capítulo, con o sin solución. Para poner un problema o cuestión, usar `\begin{prob}` y `\end{prob}`. Para incluir la solución, a continuación usar `\begin{soln}` seguido del texto terminado en `\end{soln}`.

5.1 Sean A , B y C sucesos de un cierto experimento con probabilidades dadas por:

$$\Pr(A) = \frac{1}{3}$$

$$\Pr(B) = \frac{1}{4}$$

$$\Pr(C) = \frac{1}{5}$$

$$\Pr(A \cap B) = \frac{1}{12}$$

$$\Pr(A \cap C) = \frac{1}{15}$$

$$\Pr(B \cap C) = \frac{1}{20}$$

$$\Pr(A \cap B \cap C) = \frac{1}{30}$$

- a) ¿Son los sucesos independientes dos a dos? ¿Son estadísticamente independientes?
- b) Encontrar $\Pr(A \cup B)$.
- c) Encontrar $\Pr(A \cup B \cup C)$.

5.2 Suponer que tenemos una moneda con las siguientes características: cuando se lanza, la probabilidad de que salga cara es $\Pr(c) = p$ y la probabilidad de que salga cruz $\Pr(+) = q$. Lanzamos dos veces la moneda y queremos conocer acerca de la independencia estadística de los siguientes sucesos:

A = Sale c en la primera tirada

B = En las dos tiradas sale lo mismo

C = Sale c en la segunda tirada

5.3 Determinar la media, la autocorrelación y el espectro densidad de potencia de la salida de un sistema con respuesta impulsiva dada por:

$$h(n) = \begin{cases} 1 & n = 0, 2 \\ -2 & n = 1 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

cuando la señal de entrada es un ruido blanco $X(n)$ con varianza σ_X^2 .

Solución. Si el ruido es blanco, su valor esperado será cero y su espectro densidad de potencia será una constante: $S_X(\Omega) = C$. Calculemos su autocorrelación. Se tiene:

$$R_X(k) = \mathbb{E}[F]^{-1} [S_X(\Omega)] = \mathbb{E}[F]^{-1} [C] = \frac{1}{2\pi} \int_{-\pi}^{\pi} C e^{jk\Omega} d\Omega = \begin{cases} k \neq 0 & \frac{C}{2\pi} \frac{e^{jk\Omega}}{jk} \Big|_{-\pi}^{\pi} \Rightarrow \\ k = 0 & C \end{cases}$$

$$R_X(k) = C\delta(k) = \begin{cases} C & k = 0 \\ 0 & k \neq 0 \end{cases}$$

Ahora bien: (...)

□

Si quiere introducir un separador dentro de un capítulo puede utilizar la instrucción subchapter. Esto le puede interesar, por ejemplo, para introducir alguna información adicional al final de un capítulo como un anexo al mismo.

En las secciones que siguen vamos a repasar algunas materias que utilizamos ampliamente a lo largo del texto y que sostienen de forma rigurosa el estudio de las comunicaciones digitales.

5.4 Señales: definición y clasificación

Una puede definirse como una función que transmite información generalmente sobre el estado o el comportamiento de un sistema físico, [5]. Aunque las señales puedan representarse de muchas maneras, en todos los casos la información está contenida en la variación de alguna magnitud física. Matemáticamente se representan como una función de una o más variables independientes. Por ejemplo, una señal de voz se puede representar como una función del tiempo y una imagen fotográfica puede representarse como una variación de la luminosidad respecto a dos parámetros espaciales. En cualquier caso, es una práctica común denotar como tiempo, t , a la variable independiente, en el caso de una variación continua de la variable independiente, y n en caso contrario.

5.4.1 Clasificación de señales

Establezcamos a continuación una clasificación de las señales atendiendo a diversos puntos de vista.

Señales deterministas y aleatorias

Una señal se clasifica como determinista cuando no hay incertidumbre alguna acerca del valor que tiene en cualquier instante de tiempo. Estas señales pueden modelarse como una función matemática, por ejemplo, $g(t) = 10\cos(4\pi t^2)$.

Una señal aleatoria es aquella para la que existe cierta incertidumbre respecto a su valor. Matemáticamente vamos a modelarla como una función muestra de un proceso aleatorio.

Para que una señal transmita información debe tener un carácter aleatorio, [6].

Señales periódicas y no periódicas.

Una señal $g(t)$ es periódica, con periodo T_0 , si existe una cantidad $T_0 > 0$ tal que:

$$g(t) = g(t + T_0) \quad \forall t \quad (5.7)$$

siendo T_0 el valor más pequeño que cumple esta relación. Una señal que no cumpla (5.7) se denomina no periódica.

Señales analógicas, discretas, muestreadas y digitales

Una señal analógica $g(t)$ es aquella que está definida para todo t . Una señal discreta sólo está definida en un conjunto numerable² de valores del tiempo. Una señal muestreada está definida para todo instante de tiempo, aunque sólo puede tomar valores en un conjunto numerable y una señal digital es aquella que sólo está definida en un conjunto numerable de valores del tiempo y toma valores en un conjunto numerable.

² Un conjunto es numerable o contable cuando sus elementos pueden ponerse en correspondencia uno a uno con el conjunto de los números naturales. Con posterioridad veremos que el concepto de conjunto numerable o contable juega un papel importante en el desarrollo de numerosos aspectos de la teoría.

6 Ejemplo de Capítulo de Problemas

Este es un ejemplo de capítulo de libro de problemas, donde cada sección es un problema distinto, con enunciado y solución. Se incluye un problema cualquiera, para que el lector pueda aprender cómo utilizarlo. El enunciado de cada problema comienza con \problema cómo si se tratara de una sección más. Como tal aparecerá en el índice del libro y en las cabeceras de páginas correspondientes.

Bl bla,..., los conceptos más relevantes necesarios para la resolución de los problemas de este capítulo se refiere al lector al texto [2]. La notación es la incluida al comienzo de este documento. Los conceptos necesarios para resolver los problemas planteados se pueden consultar en [4], excepto los relacionados con el cálculo de propagación, para los que se puede recurrir a [2]. A continuación se incluye una descripción de los cálculos más relevantes utilizados en estos problemas, utilizando la notación introducida al comienzo del texto.

6.1 Ruido y sensibilidad

Vamos a estudiar el ruido y la sensibilidad en ...

6.1.1 Temperatura y figura de ruido

En este texto, ver comienzo del mismo, se denota por f_r y T_r la figura y la temperatura equivalente de ruido, respectivamente, del receptor, formado éste por los elementos que van desde el conector de antena a la entrada al demodulador. Por otro lado, f_a y T_a se utilizan para denotar la figura y la temperatura equivalente de ruido de la antena. Y f_s y T_s denotan la figura de ruido y la temperatura equivalente del sistema completo, antena más receptor. Aquí las figuras de ruido están en unidades naturales, y las denotamos por ello en minúsculas. La misma notación para las figuras de ruido en mayúsculas se utilizará para denotar dB. (...)

6.1.2 Sensibilidad

En radiocomunicaciones digitales es habitual utilizar [2]

$$\frac{c}{n} = \frac{e_b/T_b}{n_0 \cdot B} = \frac{e_b \cdot R_b}{n_0 \cdot B}, \quad (6.1)$$

donde c es la potencia de portadora en u.n., B es el ancho de banda equivalente de ruido y $R_b = 1/T_b$ es el régimen binario.

Por otro lado se define $w = e_b/n_0$ como (...)

6.2 No linealidad

Si la sensibilidad nos impone un mínimo a la potencia mínima recibida, el cálculo de la distorsión por no linealidad permite determinar la potencia máxima que el receptor, o transmisor, puede manejar. La distorsión por no linealidad de un dispositivo se mide de diferentes formas. Una de las más utilizadas es calcular la relación esperada entre la potencia útil a la salida, P_o , y la potencia de intermodulación de tercer orden, I_3 . Esta relación, una diferencia si se escribe en decibelios, se denota por relación de protección frente a intermodulación de tercer orden, $RP = P_o - I_3$.

Para calcular este valor se (...)

P. 6.1 Radioenlace del servicio fijo a 13 GHz. Título muy largo que en realidad no cabría en la cabecera

Una compañía de telefonía móvil desea instalar un radioenlace digital del servicio fijo a 13 GHz, para unir dos estaciones base a través de un vano de 15 km de longitud. El enlace tiene un obstáculo agudo de incidencia rasante, esto es, con despejamiento $h = 0$ y que introduce unas pérdidas de 6 dB a la par que evita cualquier reflexión en el suelo.

Se propone utilizar para ello equipos de la serie Flexi-Hopper de Nokia, cuyos datos se incluyen en la Tabla 6.2 para una transmisión dúplex¹ de $2 \times 2E1$, resultando 4.2 Mbps en cada sentido modulados en un radiocanal. Tanto el transmisor como el receptor constan de una unidad interior (IU), al pie de la torre, y una unidad exterior (OU) con la etapa de RF junto a la antena, en el extremo superior de las torres, de 15 m. Se sabe, además, que hay una probabilidad $\eta = 25.73\%$ de que exista actividad multitrayecto en el vano para esa zona climática. Además, para esa zona climática se conoce la atenuación, A_p , excedida en el $p\%$ del tiempo, ver Figura 6.2.

El operador ha fijado como criterios de calidad el $\%$ del tiempo en el que el sistema sufre interrupciones largas y cortas, y como objetivos que las primeras ($> 10s$, $BER > 10^{-3}$) no superen el 0.036 $\%$ y las cortas ($\leq 10s$, $BER > 10^{-3}$) no superen el 0.006 $\%$ anual. El departamento de radio tiene que calcular la mínima potencia transmitida para la que el enlace es viable, evitando así interferencias a otros sistemas.

Se pide

- Calcular el valor de esta potencia. Indique si es necesario mantener en el diseño el HSB (Hot Stand By).

Tabla 6.1 Datos de las estaciones, con equipos Flexi-Hopper de Nokia.

Potencia transmitida (entregada a antena) regulable	-5 a 20 dBm
Ganancia antenas	40.5 dBi
Modulación	$\pi/4$ -DQPSK
Potencia recibida necesaria para $BER = 10^{-3}$ y $2E1$ (4.2Mbps)	$P_{dr} = -89$ dBm
Figura de ruido del sistema	6 dB
HSB 1+1, tanto en OU e IU	
Tiempo medio entre fallos para OU	$MTBF=35$ años
Tiempo medio entre fallos para IU	$MTBF=110$ años
Tiempo medio en reparar para OU como IU	$MTTR=20$ horas
Signatura normalizada del receptor	$k = \log_2 M$
M el número de puntos de la constelación de la modulación	

Solución.

- En principio la frecuencia de trabajo no es muy alta, lo que hace pensar que el enlace no estará limitado por lluvia. Los tiempos medios entre fallos son grandes, lo que parece indicar que la indisponibilidad, o interrupciones largas, no va a ser un problema. Mientras que sí lo será la pérdida de fidelidad, o

¹ el $2 \times$ indica que se utilizan las dos polarizaciones para transmitir.

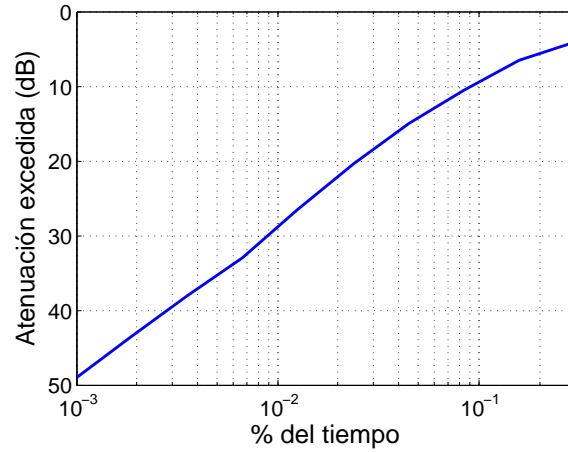


Figura 6.1 Atenuación por lluvia excedida en el $p\%$ del tiempo para 13 GHz y 15 km de distancia.

interrupciones breves. En todo caso, como el régimen binario es bajo, ≤ 34 Mbps, el desvanecimiento selectivo no es importante.

El margen bruto es el que domina en ambos criterios de calidad. De forma que podemos calcular el margen necesario para que se cumplan los objetivos de interrupciones largas (indisponibilidad) y cortas (fidelidad) y quedarnos con el más restrictivo, el mayor.

Calculamos primero el margen necesario para que se cumpla la indisponibilidad. La indisponibilidad de equipos sería la resultante de la suma de la indisponibilidad en cada extremo. En cada extremo tenemos dos equipos funcionando, la IU y la OU, el $MTBF$ equivalente es

$$\begin{aligned} MTBF_e &= (MTBF_{IU}^{-1} + MTBF_{OU}^{-1})^{-1} \\ &= ((110 \cdot 24 \cdot 365)^{-1} + (35 \cdot 24 \cdot 365)^{-1})^{-1} = 2.3259 \cdot 10^5. \end{aligned} \quad (6.2)$$

Y calculamos

$$q = MTTR / (MTTR + MTBF_e) = 20 / (20 + 2.3259 \cdot 10^5) = 8.5980 \cdot 10^{-5}. \quad (6.3)$$

para finalmente estimar la indisponibilidad

$$\begin{aligned} U_e &= 2 \cdot 100 \cdot \binom{M+N}{N+1} (mq)^{N+1} \\ &= 2 \cdot 100 \cdot \binom{1+1}{1+1} (1 \cdot 8.5980 \cdot 10^{-5})^{1+1} = 1.4785 \cdot 10^{-6} \%, \end{aligned} \quad (6.4)$$

donde $m = 1$ porque tenemos un vano y $M = N = 1$ porque el HSB (Hot Stand By) es $M + N = 1 + 1$. Dado que el máximo valor permitido es 0.036 % y que el valor anterior de indisponibilidad por equipos es muy muy pequeño, el máximo valor permitido para indisponibilidad por propagación es, aproximadamente este mismo valor (...)

P. 6.2 Radioenlace del servicio fijo a 13 GHz

Una compañía de telefonía móvil desea instalar un radioenlace digital del servicio fijo a 13 GHz, para unir dos estaciones base a través de un vano de 15 km de longitud. El enlace tiene un obstáculo agudo de incidencia rasante, esto es, con despejamiento $h = 0$ y que introduce unas pérdidas de 6 dB a la par que evita cualquier reflexión en el suelo.

Se propone utilizar para ello equipos de la serie Flexi-Hopper de Nokia, cuyos datos se incluyen en la Tabla 6.2 para una transmisión dúplex² de $2 \times 2E1$, resultando 4.2 Mbps en cada sentido modulados en un radiocanal. Tanto el transmisor como el receptor constan de una unidad interior (IU), al pie de la torre, y

² el $2 \times$ indica que se utilizan las dos polarizaciones para transmitir.

una unidad exterior (OU) con la etapa de RF junto a la antena, en el extremo superior de las torres, de 15 m. Se sabe, además, que hay una probabilidad $\eta = 25.73\%$ de que exista actividad multitrayecto en el vano para esa zona climática. Además, para esa zona climática se conoce la atenuación, A_p , excedida en el $p\%$ del tiempo, ver Figura 6.2.

El operador ha fijado como criterios de calidad el % del tiempo en el que el sistema sufre interrupciones largas y cortas, y como objetivos que las primeras ($> 10s$, $BER > 10^{-3}$) no superen el 0.036 % y las cortas ($\leq 10s$, $BER > 10^{-3}$) no superen el 0.006 % anual. El departamento de radio tiene que calcular la mínima potencia transmitida para la que el enlace es viable, evitando así interferencias a otros sistemas.

Se pide

- a) Calcular el valor de esta potencia. Indique si es necesario mantener en el diseño el HSB (Hot Stand By).

Tabla 6.2 Datos de las estaciones, con equipos Flexi-Hopper de Nokia.

Potencia transmitida (entregada a antena) regulable	-5 a 20 dBm
Ganancia antenas	40.5 dBi
Modulación	$\pi/4$ -DQPSK
Potencia recibida necesaria para $BER = 10^{-3}$ y 2E1 (4.2Mbps)	$P_{dr} = -89$ dBm
Figura de ruido del sistema	6 dB
HSB 1+1, tanto en OU e IU	
Tiempo medio entre fallos para OU	$MTBF=35$ años
Tiempo medio entre fallos para IU $MTBF=110$ años	
Tiempo medio en reparar para OU como IU	$MTTR=20$ horas
Signatura normalizada del receptor	$k = \log_2 M$
M el número de puntos de la constelación de la modulación	

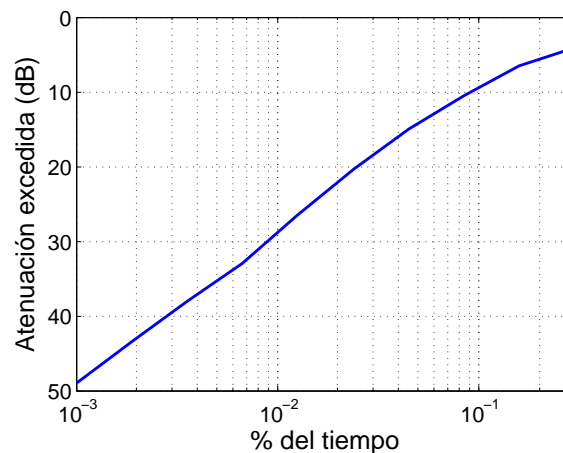


Figura 6.2 Atenuación por lluvia excedida en el $p\%$ del tiempo para 13 GHz y 15 km de distancia.

6.3 ...Ruido y sensibilidad

Vamos a estudiar el ruido y la sensibilidad en ...

6.3.1 ...Temperatura y figura de ruido

En este texto, ver comienzo del mismo, se denota por f_r y T_r la figura y la temperatura equivalente de ruido, respectivamente, del receptor, formado éste por los elementos que van desde el conector de antena a la entrada al demodulador. Por otro lado, f_a y T_a se utilizan para denotar la figura y la temperatura equivalente de ruido de la antena. Y f_s y T_s denotan la figura de ruido y la temperatura equivalente del sistema completo, antena

más receptor. Aquí las figuras de ruido están en unidades naturales, y las denotamos por ello en minúsculas. La misma notación para las figuras de ruido en mayúsculas se utilizará para denotar dB. (...)

6.3.2 ...Temperatura y figura de ruido

... En este texto, ver comienzo del mismo, se denota por f_r y T_r la figura y la temperatura equivalente de ruido, respectivamente, del receptor, formado éste por los elementos que van desde el conector de antena a la entrada al demodulador. Por otro lado, f_a y T_a se utilizan para denotar la figura y la temperatura equivalente de ruido de la antena. Y f_s y T_s denotan la figura de ruido y la temperatura equivalente del sistema completo, antena más receptor. Aquí las figuras de ruido están en unidades naturales, y las denotamos por ello en minúsculas. La misma notación para las figuras de ruido en mayúsculas se utilizará para denotar dB. (...)

6.3.3 ...Temperatura y figura de ruido

... En este texto, ver comienzo del mismo, se denota por f_r y T_r la figura y la temperatura equivalente de ruido, respectivamente, del receptor, formado éste por los elementos que van desde el conector de antena a la entrada al demodulador. Por otro lado, f_a y T_a se utilizan para denotar la figura y la temperatura equivalente de ruido de la antena. Y f_s y T_s denotan la figura de ruido y la temperatura equivalente del sistema completo, antena más receptor. Aquí las figuras de ruido están en unidades naturales, y las denotamos por ello en minúsculas. La misma notación para las figuras de ruido en mayúsculas se utilizará para denotar dB. (...)

7 Estilo tipográfico LibroETSI

En este Capítulo se analiza el fichero de estilo `libroETSI.sty` en el que se han definido los diferentes elementos que constituyen el estilo tipográfico propuesto en la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla para la redacción y publicación de libros y otros tipos de documentos.

El objetivo de este análisis es explicar detalladamente la relación entre los paquetes que se han utilizado y su reflejo en la confección del documento, para permitir, si se desea, modificar, corregir o mejorar cualquiera de los mismos por los usuarios.

Por supuesto, como ocurre en cualquier diseño, existen propuestas alternativas a las que aquí se recogen y creemos que con la exposición realizada se facilitará una mayor extensión de la utilización de \LaTeX por parte de todos los miembros de nuestra Escuela.

Es importante señalar que muchas de las instrucciones del estilo *tienen que estar en el orden que se proponen*. Al ser \LaTeX un lenguaje en el que numerosos bloques de código (paquetes) se cargan consecutivamente, es importante el orden en el que se realiza esta carga, para evitar posibles incompatibilidades.

7.1 Bloque 0

Está formado por un conjunto de sentencias genéricas en cualquier hoja de estilo de \LaTeX y en él se establece el formato que va a utilizarse (LaTeX2e) además de definir las posibles opciones que se han incorporado al estilo y procesarlas. Debemos observar que estas opciones pueden venir definidas o bien en la declaración del documento, como se propone en el fichero `libroTipoETSI.tex` o bien en la propia llamada del paquete. Así, se podrían haber escrito las primeras instrucciones del fichero `libroTipoETSI.tex` en la forma:

```
\documentclass[paper=a4,10pt, twoside]{scrbook}
\usepackage[Myfinal=true, Minion=false]{libroETSI}
```

Para poder utilizar estas opciones y alguna otra que veremos interesante introducir, como el idioma, debemos saber que, por defecto, son falsas por lo que, en realidad, no hubiera sido necesaria colocar la opción `Minion=false`. Sin embargo, a veces se prefiere su declaración para hacer más explícita que se está utilizando. Observar la forma que se definen las opciones mediante las instrucciones

```
\DeclareBoolOption{Myfinal}
\DeclareBoolOption{Minion}
\DeclareBoolOption{English}
```

7.2 Bloque 1: Aspectos generales

El primer paquete que se carga se denomina `etoolbox`. Este paquete general permite importantes correcciones en el resto de la hoja de estilo, modificando de manera sustancial el comportamiento de algunos de los elementos utilizados. Así, por ejemplo, hemos utilizado un comando definido en este paquete, `appto` en la sentencia

```
\appto{\appendices}{\def\Hy@chapapp{Appendix}}
```

que ha resuelto un grave problema que aparece al generar el índice de un documento que tenga apéndices que esté escrito en español o cualquier otro idioma que tenga caracteres no comprendidos entre los primeros 128 del código ASCII.

A continuación se cargan un conjunto de paquetes que resuelven incompatibilidades con el tipo de documento que estamos utilizando, resuelven problemas internos o nos permiten realizar comparaciones booleanas.

Seguidamente se carga el paquete `microtype`, si el parámetro `Myfinal` es `true`. Este paquete es responsable de finísimos microajustes en los textos generados por \LaTeX permitiendo la expansión o comprensión de caracteres y espacios en blanco de un documento para mejorar su apariencia. Con diferencia a lo que se realiza en Word, \TeX gestiona estas correcciones a nivel de párrafo, lo que confiere un aspecto altamente profesional a los documentos generados. Debido a la complejidad de los algoritmos que se utilizan, al cargar `microtype` se entelenece considerablemente la compilación del documento, por lo que resulta conveniente definir `Myfinal` como `true` únicamente en las etapas últimas de redacción. ¡Pero no sólo en la última!, puesto que el ajuste que introduce hace que a veces cambie la situación de las líneas huérfanas y viudas, de considerable importancia desde un punto de vista tipográfico.

Siguiendo los consejos de los editores de IEEE, se han modificado un conjunto de valores por defecto lo que hace el documento menos restrictivo en lo referente a la colocación de los elementos flotantes dentro de una página. Además, se ha optado por permitir que un conjunto de fórmulas que constituyan un bloque (definidas dentro del entorno `\begin{align}... \end{align}`, por ejemplo) puedan escribirse en diferentes páginas. Si se desea cambiar este comportamiento, simplemente debemos modificar el valor de `\interdisplaylinepenalty` a un valor más elevado, por ejemplo 10000.

Aunque el tipo de documento utilizado, `scrbook`, incorpora por defecto un conjunto de instrucciones que permiten una gestión muy eficiente del tamaño de los diferentes elementos que constituyen una página (cabecera, ancho del texto, altura, pie de página, offset de encuadernación, etc,) se ha optado por utilizar el paquete `geometry` para ese fin. En el fichero de estilo simplemente se carga el paquete y los parámetros concretos se definen posteriormente en el fichero principal que estemos utilizando. Por ejemplo, este documento se ha generado con los siguientes:

```
paperheight=240mm,% Altura de la página
paperwidth=170mm,% Anchura
top=25mm,%
headsep=7.5mm,%
footskip=10mm,%
textheight=190mm,% Altura del texto
textwidth=124mm,% Ancho del texto
bindingoffset=15mm,% Offset de encuadernación
twoside
```

Finalizamos este bloque definiendo un alias para el tamaño básico de la fuente que hemos declarado en la sentencia inicial para poder realizar un escalado de los diferentes tamaños que se van a usar en el documento. Observad que se crean dos dimensiones, para la fuente y para el interlineado. `\lnormal` y `\lbnormal`, respectivamente.

7.3 Bloque 2: Idioma, Codificación y Fuentes

7.3.1 Idioma

La posibilidad de utilizar nuestra plantilla para escribir textos en inglés (es el único idioma aparte del español que se ha considerado) nos obliga a establecer una posible opción para cargar el paquete `babel` de una u otra manera. Este paquete es el responsable de establecer las reglas de partición silábica, entre otras cosas, por lo que resulta imprescindible su incorporación a una hoja de estilo. Cabe decir que en estos momentos el responsable de su mantenimiento es el español Javier Bezos, lo que nos garantiza una excelente adaptación del paquete a nuestro idioma.

Además de la partición silábica, un determinado conjunto de macros quedan automáticamente traducidos como por ejemplo, la definición de límite, máximos y mínimos que se utilizan en la escritura matemática. Observemos cómo aparecen unos y otros en función del idioma que utilicemos:

<code>\begin{align*}</code>	
<code>\lim_{n\to\infty}a_n&=10\\</code>	$\lim_{n\rightarrow\infty}a_n = 10$
<code>\max\left[{a,b}\right]&=a</code>	$\max[a,b] = a$
<code>\end{align*}</code>	
<code>\figurename, \tablename</code>	
	Figura, Tabla
<code>\selectlanguage{english}</code>	
<code>\begin{align*}</code>	
<code>\lim_{n\to\infty}a_n&=10\\</code>	$\lim_{n\rightarrow\infty}a_n = 10$
<code>\max\left[{a,b}\right]&=a</code>	$\max[a,b] = a$
<code>\end{align*}</code>	
<code>\figurename, \tablename</code>	
<code>\selectlanguage{spanish}</code>	Figure, Table

El grupo de macros que gestionan los nombres que cambian en uno y otro caso, se encuentra a continuación en el fichero de estilo.

7.3.2 Codificación

Para gestionar la forma en la que se controla la codificación en la que está escrita los ficheros fuentes de \LaTeX se utiliza el paquete `inputenc` junto con el paquete `fontenc`. Estos dos paquetes sólo son necesarios si no utilizamos \LaTeX ya que en caso de utilizarse este motor, la gestión se realiza por un mecanismo completamente diferente, interno al propio motor y del que podemos despreocuparnos.

7.3.3 Fuentes del texto y comandos asociados

La selección de la fuente a utilizar depende fundamentalmente de qué motor estemos utilizando para generar nuestro texto. Si optamos por \LaTeX (en realidad, \pdfLaTeX), la selección del texto normal se realiza en el fichero principal mediante el comando `\usepackage{tgtermes}`, que selecciona para el texto un clon de la fuente Times. En general, las fuentes que se pueden utilizar con \LaTeX no son demasiadas y una excelente recopilación de las disponibles se encuentra en la dirección <http://www.tug.dk/FontCatalogue/>.

Sin embargo, si nuestro motor es \LuaTeX , podemos utilizar como fuente de texto cualquier fuente .otf ó .ttf presente en nuestro ordenador. Cómo se cargan dichas fuentes y cómo se utilizan se encuentra recogido en la documentación del paquete `fontspec` y en nuestro estilo se muestra en las líneas de código siguientes:

```
\setmainfont[Renderer=Basic, Ligatures=TeX,%
  Scale=1.0,%
]{Times New Roman}
```

La fuente que hemos seleccionado de esta manera es la que utilizaremos en el texto normal. Es decir, en todo el texto que no sea cabeceras de página, enunciados de secciones, captions de figuras, etc. Para todos estos casos, como una elección de diseño, hemos utilizado una fuente perteneciente al tipo denominado *sin serif*. En el caso de \LaTeX hemos optado por una fuente Helvética Narrow, mientras que en el caso de \LuaTeX hemos utilizado una fuente Arial Narrow. Los comandos iniciales de esta decisión se muestran en las siguientes líneas de código.

```
\ifluatex
  \setsansfont[Ligatures=TeX, % Se puede obtener un texto sin serif con \
    ssfamily
    Scale=0.95,
  ]{Arial Narrow}
\else
  \renewcommand{\sfdefault}{phv}
\fi
```

Podemos observar la sintaxis completamente diferente que se utiliza en un caso y otro. En realidad, para facilitar la utilización de diversas variantes de ambas fuentes, en cualquiera de los dos motores de composición elegidos, hemos definido un conjunto de comandos que están recogidos en las siguientes líneas.

```
\ifluatex

\newfontfamily\helveticam{Arial}          % Helvética
\newfontfamily\helveticab{Arial Bold}     % Helvética Bold
\newfontfamily\helveticai{Arial Italic}   % Helvética itálica
\newfontfamily\helveticax{Arial Bold Italic} % Helvética Bold Itálica

\newfontfamily\titular{ArialNarrow-Bold}  % Para los titulares
\newfontfamily\titulart{ArialNarrow-Bold} % Para los títulos
\newfontfamily\titulari{ArialNarrow-BoldItalic} % Para los titulares oblicua

\newfontfamily\titulartoc{ArialNarrow}    % Para el TOC
\newfontfamily\titulartocb{ArialNarrow-Bold} % Para el TOC bold
\newfontfamily\titulartoci{ArialNarrow-BoldItalic} % Para el TOC oblicua

\else

\newcommand{\helveticam}{\usefont{T1}{phv}{m}{n}\selectfont }
\newcommand{\helveticab}{\usefont{T1}{phv}{b}{n}\selectfont }
\newcommand{\helveticai}{\usefont{T1}{phv}{m}{it}\selectfont }
\newcommand{\helveticax}{\usefont{T1}{phv}{b}{it}\selectfont }

\newcommand{\titular}{\usefont{T1}{phv}{bc}{n}\selectfont }
\newcommand{\titulari}{\usefont{T1}{phv}{bc}{it}\selectfont }
\newcommand{\titulart}{\usefont{T1}{phv}{bc}{n}\selectfont }

\newcommand{\titulartoc}{\usefont{T1}{phv}{mc}{n}\selectfont }
\newcommand{\titulartocb}{\usefont{T1}{phv}{bc}{n}\selectfont }
\newcommand{\titulartoci}{\usefont{T1}{phv}{mc}{it}\selectfont }

\fi
```

De esta manera para el usuario del paquete es transparente el uso de \LaTeX o \LuaTeX . La sintaxis en este último caso se debe a la utilización del paquete `\usepackage[no-math]{fontspec}` que hemos cargado con anterioridad y en el caso de \LaTeX al uso de instrucciones básicas del esquema de selección de fuentes.

En cualquier caso, hasta este momento sólo hemos definido los comandos que nos permitirán hacer uso de las tipografías seleccionadas. Su aplicación concreta, con el tamaño que hemos elegido debe realizarse a continuación. En primer lugar, para facilitar un escalado adecuado de las distintas fuentes, hemos definido un conjunto de dimensiones relativas como, por ejemplo,

```
\newdimen\lcatorce
\newdimen\lbcatorce
```

y posteriormente le hemos asignado valores,

```
\lcatorce=\dimexpr1.4\dimexpr\lnormal
\lbcatorce=\dimexpr1.4\dimexpr\lbnormal
```

Estos valores se corresponden en este caso a fracciones (1,4 ó 140%) del tamaño de la fuente normal que, recordemos, habíamos definido con anterioridad.

Por último, un conjunto de comandos facilita la utilización de las fuentes en sus diversas variaciones a lo largo de todo el texto. Así, por ejemplo, se ha definido en la siguiente línea de código un comando para definir la fuente y el tamaño que vamos a usar en los títulos de las secciones:


```
\newcommand{\aheadsecc}{\fontsize{\ltrece}{\lbtrece} \titular}
```

Por supuesto, una vez que hemos definido este comando, podemos utilizarlo siempre que deseemos. Así, la utilización y el resultado del comando `\aheadsecc` se muestra a continuación:

```
\aheadsecc{Como si fuera una seccin}
```

Como si fuera una seccin

7.3.4 Fuentes matemáticas y símbolos

En una Escuela de Ingeniería la gestión de las fuentes matemáticas en nuestros documentos juega un papel fundamental. Por ello, si utilizamos \LaTeX es necesario cargar los paquetes `lualatex-math` y `luatextra`. Observar además que aparte de estos dos paquetes, orientados básicamente a la utilización de aspectos concretos relacionados con símbolos matemáticos, hemos incluido el comando `\usepackage[no-math]{fontspec}`. Aunque `fontspec` se utilizará en la gestión del tipo de fuentes y comandos relacionados con las mismas, es necesario incluirlo aquí por el típico problema que hemos mencionado anteriormente del orden en el que se encargan los paquetes.

En general, para mejorar el escalado de determinados símbolos, utilizaremos el paquete `exscale`. Advertimos nuevamente que *se tiene que cargar exactamente antes que el paquete `amsmath`*, que se carga a continuación, junto con una adición recomendable, el paquete `mathtools`.

Existen numerosos paquetes que definen los símbolos matemáticos más habituales. Uno de los más completos es el paquete `MnSymbol`, especialmente indicado cuando la fuente del texto es la Minion, licenciada por Adobe. Sin embargo, la utilización tanto de la fuente de símbolos como la fuente de texto dentro de \LaTeX es un asunto no trivial. Requiere la disponibilidad de la fuente Minion, y la creación del paquete `MinionPro`. El lector interesado puede encontrar el procedimiento para crear dicho paquete en <https://github.com/sebschub/FontPro/>.

Si se han seguido las instrucciones que se encuentran en la dirección anterior y se desea utilizar la fuente, se indicará mediante uno de las opciones del programa, escribiendo `Minion=true`. Con ello, se cargarían los correspondientes paquetes mediante las primeras líneas del siguiente código:

```
\makeatletter
\ifdtsc@Minion
  \usepackage{MnSymbol}
  \usepackage[mathlf, minionint, mnsy]{MinionPro}
  \newcommand{\bm}{\ensuremath{\boldsymbol{}}}
\else
  \usepackage{amssymb}
  \usepackage{mathptmx}
  \usepackage{amsfonts}
  \usepackage{bm}
\fi
\makeatother
```

En el caso que no se tenga instalado el paquete, o no se desee utilizar, basta ignorar la opción y se ejecutarán las restantes líneas del código anterior.

7.4 Bloque 3: Tabla de contenidos

Uno de los aspectos sobresalientes de \LaTeX es la facilidad para generar la *Tabla de contenidos* (TOC) de un documento. No sólo para generarla sino también para controlar cómo se presenta y cómo se gestiona el contenido de la misma. En esta sección explicaremos los comandos que hemos utilizado con este fin, teniendo en cuenta que hemos actuado directamente renombrado y creando nuevos elementos, sin utilizar paquetes específicos para esta tarea. Si alguno desea hacer uso de los mismos, unos de los más ampliamente utilizados es el paquete `\tocloft`.

Para controlar dónde queremos que aparezca la tabla de contenido, debemos utilizar el comando `\tableofcontents`. Si analizamos el fichero principal de este documento, veremos que precediendo a este comando aparecen las líneas

```
\cleardoublepage
```

```
\phantomsection
\addcontentsline{toc}{listasf}{Índice}
\pagestyle{especial}
```

Con ellas indicamos, en primer lugar, que la tabla de contenidos siempre va a aparecer en una página impar, que haremos uso de un estilo de página¹ que denominamos **especial** y que queremos que la propia tabla de contenido (o índice del documento) aparezca referenciada en el índice, aunque pueda parecer complejo entender lo que esto significa. L^AT_EX permite controlar el nombre que le hemos asignado a la tabla de contenido mediante la sentencia `\def\contentsname{Índice}`, tal como se define en otro lugar del estilo, puesto que este nombre dependerá del idioma que utilicemos.

A la hora de construir esta tabla de contenidos, nuestra primera decisión fue establecer que iban a aparecer hasta los apartados que hemos denominados **subsubsecciones**, lo que se logra mediante el `{3}` del comando `\setcounter{tocdepth}` en **libroETSI.sty**. Cambiar este número haría aparecer menos apartados o más, dependiendo de la elección.

Para establecer la manera en la que cada nivel de segmentación aparece en el TOC hacemos uso de una secuencia de comandos similares a las mostradas a continuación para el caso del capítulo:

```
1 \makeatletter
2 \renewcommand*\l@chapter[2]
3 {
4   \addpenalty{-\@highpenalty}%
5   \vskip 0.5em \@plus\p@
6   \@dottedtocline{0}{0em}{1em}{\tocchap #1}{\tocchap #2}
7 }
8 \makeatother
```

El elemento clave está recogido en la línea 6. El significado de cada uno de los parámetros es el siguiente: el **0** señala que estamos formateando el nivel de capítulo. El siguiente parámetro, **0em** establece la indentación de la línea, **1em** define la anchura que reservamos para el número que mostramos, si es que la parte tiene números y los otros dos establecen las características tipográficas que vamos a utilizar para el título (incluyendo el número si lo hubiera) y el número de la página en el que se encuentra la correspondiente parte. Este ajuste se ha realizado sin tener en cuenta ninguno de los paquetes, únicamente utilizando sentencias del propio núcleo de L^AT_EX. De forma similar hemos procedido para el resto de los elementos de segmentación del texto.

Aunque algo redundante, nuestra siguiente decisión afecta a la manera en la que hemos querido que aparezcan en el índice los índices del texto (valga la redundancia), tales como el *Índice de Figuras*, *Índice Alfabético*, etc y otros elementos como el *Resumen*, *Prefacio*, etc o la *Bibliografía*. No es trivial pero, básicamente, hemos definido dos listas, una para los elementos que aparecen antes del Índice y otra para los que aparecen después, al final del texto, que se corresponden aproximadamente a lo que hemos denominado `\frontmatter` y `\backmatter`, respectivamente. Si nos fijamos por ejemplo en la lista que hemos denominado `\listasf`, cómo se representan los elementos que están incluidos en la misma se realiza con las instrucciones

```
\makeatletter
\newcommand*\l@listasf[2]
{
  \addpenalty{-\@highpenalty}%
  \vskip 0.05em \@plus\p@
  \@dottedtocline{0}{0em}{2.5em}{\fontsize{\lnueve}{\lbneue}\selectfont \
    helveticai #1}{\fontsize{\lnueve}{\lbneue}\titulartoc #2}
}
\makeatother
```

en las que vemos, por ejemplo, que el tamaño del texto es el tamaño relativo `\lbneue`, junto con una fuente `\helveticai` represente esta fuente lo que represente (en un apartado anterior ha quedado definido).

Una cuestión distinta es las instrucciones que debemos ejecutar para incluir un determinado elemento en la lista correspondiente. Por ejemplo, para incluir el *Prefacio* en la `\listasf`, observemos que hemos escrito el siguiente comando:

¹ Más adelante veremos cómo se definen estos estilos.

```
\addcontentsline{toc}{listasf}{Prefacio}
```

Para finalizar este apartado, también hemos propuesto que no aparezcan los habituales puntos que existen entre el texto y el número de página correspondiente de muchos índices, ajustando a 10000 el parámetro `\@dotsep`. Unas referencias interesantes para manejar todos estos elementos las encontramos en las siguientes direcciones <http://tex.stackexchange.com/questions/110253/what-the-first-argument-for-lsubsection-actually-is> y <http://tex.stackexchange.com/questions/33841/how-to-modify-the-indentation-before-sectioning-titles-in-the-table-of-contents>.

7.5 Bloque 4: Estilos de Páginas y Títulos

Este apartado ha sido uno de los más complejos en cuanto a diseño se refiere, atendiendo a la gran cantidad de elementos que componen un texto y sus interrelaciones.

En general, se recomienda que los tipos de páginas diferentes que se utilicen en un documento sea un número muy limitado. Por ello, \LaTeX no establece un mecanismo elemental para definir distintos tipos de páginas (en realidad, únicamente un tipo `plain` y un tipo `empty`) y debemos acudir a paquetes específicos que nos permitan una mayor libertad de diseño. Lo mismo podríamos decir a la hora de definir el formato de los distintos elementos, lo que hemos denominado genéricamente *Títulos*, haciendo con ello referencia a los *Títulos* de los capítulos, secciones, subsecciones, etc.

Hemos de tener en cuenta que el aspecto de un libro está básicamente determinado por el formato que se ha elegido para los diferentes títulos de las partes que lo constituyen, el formato de las páginas y qué queremos que aparezca en las cabeceras y pies de páginas del mismo. Todo esto se ha conseguido utilizando un paquete desarrollado por el español Bezos denominado `titlesec`, que se carga en nuestro fichero mediante la instrucción `\usepackage[noindentafter, pagestyles,...]{titlesec}`. En el listado siguiente se recogen algunos de los elementos de los formatos elegidos para que mediante su análisis podamos realizar nuestro propio diseño.

```
1 \newpagestyle{esitscCD}
2 {
3   \esirulehead
4   \sethead[\numpagpar \rhfont\chaptername\ \thechapter. \; \chaptertitle
5     ] [] [] %
6   {} {} {\rhfont\thesection\ \; \sectiontitle \numpagodd}
7 }
8 \newpagestyle{primera}
9 {
10  \esirulehead \footrule
11  \sethead[] [] [] {\includegraphics[width=2 cm]{logoUS.pdf}} %
12    {\raisebox{0.8cm}{\begin{minipage}{0.515\textwidth}
13      \centering
14      {\aheadsubsecc \fromtitulacion}\\
15      \vspace*{.5ex}
16      {\normalfont \fromasignatura}\\
17      \vspace*{1ex}
18      \normalfont \fromconvocatoria \quad \fromfecha
19      \end{minipage}}}
20  } %
21  {\includegraphics[width=4 cm]{logoTSC.pdf}}
22  }
23  \setfoot{} {\rhpagefont Pág. \thepage\, de\, \zpageref{LastPage}} {}
24  }
25
26 \newpagestyle{examen}
27 {
28  \esirulehead
```

```

29 \sethead[\rhpagfont Pág. \thepage\, de\, \zpageref{LastPage}][\
    aheadsubsecc\fromasignatura][[\rhpagfont\fromfecha]%
30 {[\rhpagfont\fromfecha]{\aheadsubsecc\fromasignatura}{\rhpagfont Pág. \
    thepage\, de\, \zpageref{LastPage}}
31 }
32
33 \newpagestyle{problema}
34 {
35 \esirulehead
36 \sethead[\numpagpar][[\rhfont\chaptername \; \thechapter. \
    chaptertitle]% even
37 {\rhfont\theproblema \; \problematitle}{\numpagodd}% odd
38 }
39 \newpagestyle{especial}
40 {
41 \esirulehead
42 \sethead[\numpagpar][\rhfont\chaptertitle][\ {\rhfont\chaptertitle}{\
    numpagodd}
43 }
44
45 \newpagestyle{paginablanc}
46 {
47 \sethead[] [] [] {} {} {}
48 \vspace*{0.2\textheight}
49 \begin{center}
50 {\emph{Página en blanco} }
51 \end{center}
52 }
53
54 \renewpagestyle{plain}
55 {
56 \setfoot[] [\rhpagfont\thepage] [] {} {} {}
57 }
58
59 %:Estilo de capítulos con numeración
60 \titleformat{\chapter}{\vspace{75pt}\achapnum} {\makebox[25 pt]{\
    raggedright\thechapter}}{8pt}{\hspace*{-6pt}\raggedright\achaptext
    #1}{\vspace{0.3pc} {\color{gray!75}\titlerule[3.5pt]} \vspace{50pt}}
61 \titlespacing{\chapter}{0 pt}{0 pt}{0 pt}{0 pt}
62
63 %:Estilo de capítulos sin numeración
64 \titleformat{name=\chapter,numberless}{\vspace{75pt}}{}{0pc}{\filleft\
    achaptext #1}{\vspace{0.3pc} {\color{gray!75}\titlerule[3.5pt]} \vspace
    {50pt}}
65 \titlespacing{name=\chapter,numberless}{0 pt}{0 pt}{0 pt}{0 pt}
66
67 %:Estilo de sección
68 \titleformat{\section}{\aheadsecc}{\thesection}{10 pt}{#1}
69 \titlespacing{\section}{0 pt}{3ex plus .1ex minus .2ex}{3ex plus .1ex minus
    .2ex}
70
71 %:Estilo de sección sin numeración
72 \titleformat{name=\section,numberless}{\aheadsecc}{}{0 pt}{#1}
73 \titlespacing{name=\section,numberless}{0 pt}{3ex plus .1ex minus .2ex}{3ex
    plus .1ex minus .2ex}
74
75 %:Estilo de sección sin numeración personal

```

```

76 \titleclass{\misection}{straight}[\chapter]
77 \titleformat{name=\misection,numberless}{\aheadsecc}{}{0 pt}{#1}
78 \titlespacing{name=\misection,numberless}{0 pt}{3ex plus .1ex minus .2ex}{3
    ex plus .1ex minus .2ex}
79
80 %:Estilo de problema en un libro con capítulos de problemas
81 \newcounter{problema}[chapter]
82 \makeatletter
83 \renewcommand{\theproblema}{P. \thechapter.\@arabic{c@problema}}
84 \makeatother
85 \newcommand{\problemtitle}{}
86
87 \titleclass{\problema}{straight}[\chapter]
88 \titleformat{name=\problema}{\aheadsecc}{\theproblema}{10 pt}{#1}[]
89 \titlespacing{name=\problema}{0 pt}{3ex plus .1ex minus .2ex}{3ex plus .1ex
    minus .2ex}
90
91 \makeatletter
92 \let\problemaint\problema
93 \def\problema{\@ifnextchar[\problemaii\problemai}
94 \def\problemai#1{\gdef\problematitle{#1}\problemaint{#1}}
95 \def\problemaii[#1]#2{\gdef\problematitle{#1}\problemaint[#1]{#2}}
96 \makeatother
97
98 %:Para que el cambio del tipo de página se gestione con los comandos
99 \makeatletter
100 \let\problema@without@pagestyle\problema
101 \def\problema{\pagestyle{problema}\problema@without@pagestyle}
102 \makeatother
103
104 \makeatletter
105 \let\section@without@pagestyle\section
106 \def\section{\pagestyle{esitscCD}\section@without@pagestyle}
107 \makeatother

```

La definición de los estilos de páginas se realiza siguiendo el modelo que podemos ver entre las líneas 1 y 6 para el estilo de página por defecto, que hemos denominado **esitscCD**. En la línea 3 se define el comando **\esirulehead** que es el responsable de la línea de cabecera o bien, algún comando más para controlar el pie de página, como se puede observar en la línea 31 de otro de los estilos de página o cualquier comando genérico.

A continuación, mediante la sección **\sethead[] [] [] {} {} {}** definimos los elementos que vamos a incorporar en las secciones izquierda, centro y derecha de las páginas pares ó izquierda centro y derecha de las páginas impares. Las líneas 4 y 5 son las responsables de las cabeceras de este documento y creemos que es fácil entender su funcionamiento.

Un ejemplo diferente lo encontramos en el estilo de página **especial**. Podemos observar que en este caso las páginas pares e impares son las mismas diferenciándose únicamente en la posición del número de página.

El paquete permite definir con gran libertad estilos de páginas mucho más complejos, como podemos apreciar en los estilos de página que hemos denominado **primera** (primera página de un examen) y **examen**. Así, se han incluido en las cabeceras logos, varias líneas o un contador del número de páginas de las que consta el examen, como podemos ver en un ejemplo concreto en la Figura 7.1.

Como últimos ejemplos de definición de estilos de páginas, se muestran las definiciones del estilo de página **paginablanc** que generaría una página en blanco, con el texto *Página en blanco* en su centro, o el estilo **plain**, que es una modificación del estilo definido por defecto en \LaTeX .

A continuación, en el anterior código, se muestran varios ejemplos de definiciones de los elementos de segmentación del texto. El paquete **titlesec** es la referencia para estudiar cómo se han definido y solamente destacar que se ha creado un nuevo elemento, **\problema** para poder gestionar los problemas como si fueran secciones de un libro permitiendo, por ejemplo, su inclusión en la tabla de contenidos.



Figura 7.1 Ejemplos de cabeceras de la página primera de un examen y de la página segunda.

Es importante prestar especial atención a las líneas de código 99-107. En ella establecemos que siempre que definamos un problema o una sección, nos garantizamos que las páginas que vamos a utilizar se correspondan respectivamente con la página especial de problemas o con la página general de nuestro texto. Con ello logramos que en las cabeceras correspondientes queden reflejados el problema o la sección en la que nos encontramos.

7.6 Bloque 5: Gestión general del documento

En este Bloque 5 se recogen un gran número de comandos, cargas de paquetes y ajustes que, como su nombre indica, permite una mejor gestión del documento. La gran extensión del mismo, más de 700 líneas de código, hace inviable su descripción pormenorizada y nos limitaremos a señalar los elementos más importantes. Debemos insistir que el orden en el que se ejecuta la carga de los paquetes no es arbitrario y hay restricciones sobre este orden que pueden dar lugar a inconsistencias y errores.

7.6.1 Apéndices

Para la gestión de los apéndices se ha elegido el paquete `appendix`. Para ajustar la manera en la que se muestran los apéndices en el texto, se modifica dentro del entorno el formato de `\chapter` de manera que también se incluya la palabra *Apéndice* dentro de la definición del mismo. También se ha utilizado dentro del entorno una página de estilo especial y para evitar un problema con la tabla de contenidos cuando se utiliza el idioma español, ha sido necesario incluir las siguiente líneas de código:

```
\makeatletter
\appto{\appendices}{\def\Hy@chapapp{Appendix}}
\makeatother
```

que se encuentran situadas tras la carga del paquete `hyperref` para garantizar un correcto funcionamiento.

7.6.2 Colores

Entre los diversos paquetes que permiten la utilización de colores dentro de \LaTeX hemos elegido `xcolor` con las opciones `svgnames`, `x11names`. En la documentación del mismo vemos que existe una gran variedad de colores y esquemas que pueden utilizarse con mucha facilidad. Creemos que es una buena idea definir los colores en un lugar concreto y un ejemplo de cómo puede hacerse se muestra en las siguientes líneas.

```
\usepackage[svgnames, x11names]{xcolor}
\definecolor{light-gray}{gray}{0.90}
\definecolor{shadecolor}{gray}{0.90}
\definecolor{refcolor}{named}{Black}
\definecolor{Matlabcolor}{RGB}{252,251,220}
```

También hemos definido en el fichero correspondiente a la edición colores propios de la ETSI:

```
\definecolor{etsi}{RGB}{83,16,12}
\definecolor{fondo}{RGB}{136,18,1}
\definecolor{texto}{RGB}{253,181,138}
\definecolor{logoetsi}{RGB}{176,124,96}
```

7.6.3 Aspectos genéricos de tratamiento del texto

A continuación hemos cargado un conjunto de paquetes que mejoran aspectos genéricos y facilitan la utilización de logos y acentos especiales. Por ejemplo, el paquete `icomma` corrige un pequeño error en \LaTeX y establece una separación correcta de la coma decimal.

En este bloque también está incluido la asignación de un valor al contador `secnumdepth`. Con la instrucción

```
\setcounter{secnumdepth}{4}
```

le asignamos el valor de 4, que significa que hasta las subsecciones deben aparecer numeradas. No debemos confundir este parámetro con `tocdepth`, que establece hasta qué nivel debe aparecer en la tabla de contenidos.

La gestión de los subíndices y superíndices en las ecuaciones matemáticas se realiza de forma diferente según estemos compilando con \LaTeX o \LuaTeX . En este último caso, el control es muy preciso diferenciándose incluso si la ecuación está en línea con el texto o bien no lo está.

Adicionalmente hemos definido el comando `\xb` que permite modificar la posición de los subíndices cuando se utilizan con letras descendientes como la f o g . En el siguiente ejemplo se muestra su utilización (o no) y el resultado que se obtiene.

<code>\begin{align*}</code>	
<code>f_{1}(t)&=\text{\sen}(\text{\omega}_{0}t)\text{\}</code>	$f_1(t) = \text{sen}(\omega_0 t)$
<code>f\text{\xb}{1}(t)&=\text{\sen}(\text{\omega}_{0}t)\text{\}</code>	$f_1(t) = \text{sen}(\omega_0 t)$
<code>g_{i}(t)&=\text{\sen}(\text{\omega}_{0}t)\text{\}</code>	$g_i(t) = \text{sen}(\omega_0 t)$
<code>g\text{\xb}{i}(t)&=\text{\sen}(\text{\omega}_{0}t)\text{\}</code>	$g_i(t) = \text{sen}(\omega_0 t)$
<code>\end{align*}</code>	

También hemos creado un conjunto de comandos que facilitan la escritura de los subíndices y superíndices en modo texto, `\tsp` y `\tsb`.

Asimismo se incluyen en este apartado los paquetes ya mencionados `epigraph`, que permite escribir un epígrafe en el lugar que se desee y el paquete `lettrine`, con el que obtenemos el efecto de la primera letra del texto del capítulo en mayúscula y negrita y ocupando más de un renglón. Ambos paquetes han sufrido ajustes y en las siguientes líneas se recogen los realizados en el texto y autor del epígrafe.

```
\makeatletter    %% El texto
\renewcommand{\@epitext}[1]
{
  \begin{minipage}{\epigraphwidth}
    \begin{textflush} \itshape #1\text{\}
    \ifdim\epigraphrule>\z@ \@epirule \else \vspace*{1ex} \fi
    \end{textflush}
  \end{minipage}
}
\makeatother

\makeatletter    %% El autor
\renewcommand{\@episource}[1]
{
  \begin{minipage}{\epigraphwidth}
    \begin{sourceflush} \scshape #1\end{sourceflush}
  \end{minipage}
}
\makeatother
```

\LaTeX ofrece la posibilidad de crear tablas de contenidos abreviadas de muy diversos tipos. Aunque en este texto no se han confeccionado, el paquete `shorttoc` permite su gestión. Por último, destacar el uso del paquete `enumitem` para facilitar la creación de listas enumeradas.

7.6.4 Elementos flotantes

En \LaTeX , los elementos flotantes son las tablas y figuras y tienen un gran número de parámetros que se necesitan ajustar mediante un conjunto de paquetes seleccionados. Uno de los más importantes es el paquete `caption` que permite ajustar de manera muy preciso el texto que acompaña a estos elementos flotantes.

De acuerdo con la documentación de `caption`, su carga y ajuste se ha realizado con las líneas de código que siguen:

```
\usepackage{caption}[2013/02/03]
\DeclareCaptionLabelFormat{esisf}{\aheadteoremas{#1} \aheadteoremas{#2} }
\DeclareCaptionLabelSeparator{cuadratin}{\hspace*{3pt}}
\captionsetup{labelformat=esisf, textfont=normalfont, textformat=period,
  labelsep=cuadratin, format=hang, indentation=0 cm, skip=10pt, labelformat=
  esisf}
```

Las declaraciones de formatos son diferentes para cada uno de los elementos flotantes. Así, por ejemplo, uno de los posibles recursos de \LaTeX es la inclusión en el texto de subfiguras, lo que se gestiona mediante el paquete `subfig` y la declaración de formato correspondiente, como podemos ver a continuación:

```
\DeclareCaptionLabelFormat{subfig}{\aheadteoremas{#1} \aheadteoremas{(#2)} }
\captionsetup[subfigure]{labelformat=subfig }
```

Un ejemplo de ambos paquetes podemos apreciarlo en la Figura 7.2. La referencia a las subfiguras de la figura anterior se realiza con el código siguiente, de resultado mostrado a continuación.

en la \autoref{fig04-02}\sfx{fig04-02a} se muestra la funcin de distribucin de una variable aleatoria continua y en la \autoref{fig04-02}\sfx{fig04-02b} su funcin densidad de probabilidad.

en la Figura 7.2-(a) se muestra la funcin de distribucin de una variable aleatoria continua y en la Figura 7.2-(b) su funcin densidad de probabilidad.

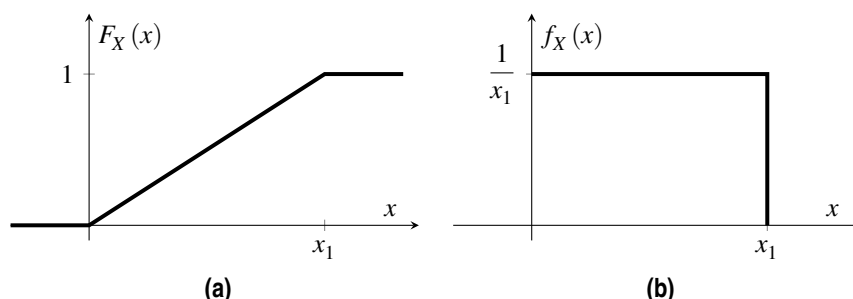


Figura 7.2 Función de distribución y Función densidad de probabilidad de una variable aleatoria continua...

Un ejemplo de cómo se gestionan las tablas que puedan segmentarse a través de más de una página se muestra en la Tabla 7.1, obtenida con el siguiente código:

```
\begin{longtable}{p{3.5cm}p{8cm}}
\caption{Funciones de manipulación de gráficos} \label{tab01-08}\\
\hline
{\rule[-8pt]{0pt}{22pt}\bfseries{Función}} & Significado\\
\hline %\rule{0pt}{1pt}
\endfirsthead
\caption[{}]{..continuación} \\
\hline
{\rule[-8pt]{0pt}{22pt}\bfseries{Función}} & Significado\\
\hline %\rule{0pt}{1pt}
\endhead
```



```

\hline
\endfoot %\rule{0pt}{14pt}
\texttt{xlabel('texto')} & Etiqueta el eje \texttt{x} de la gráfica actual\\
\texttt{ylabel('texto')} & Etiqueta el eje \texttt{y} de la gráfica actual\\
\texttt{title('texto')} & Título de la gráfica actual\\
\texttt{text(x,y, 'texto')} & Introduce “texto” en la posición \texttt{(x,y)}
de la gráfica actual\\
\texttt{legend()} & Permite definir rótulos para las distintas líneas o curvas
de la gráfica\\
\texttt{grid} & Dibuja una rejilla. Con \texttt{grid off} desaparece la cuadrí
cula\\
\texttt{axis([xmin xmax ymin ymax])} & Fija los valores máximos y mínimos de
los ejes\\
\texttt{axis equal} & Establece que la escala de los ejes sea la misma\\
\texttt{axis square} & Fija que la gráfica sea un cuadrado\\
\end{longtable}

```

Tabla 7.1 Funciones de manipulación de gráficos.

Función	Significado
<code>xlabel('texto')</code>	Etiqueta el eje x de la gráfica actual
<code>ylabel('texto')</code>	Etiqueta el eje y de la gráfica actual
<code>title('texto')</code>	Título de la gráfica actual
<code>text(x,y, 'texto')</code>	Introduce “texto” en la posición (x,y) de la gráfica actual
<code>legend()</code>	Permite definir rótulos para las distintas líneas o curvas de la gráfica
<code>grid</code>	Dibuja una rejilla. Con <code>grid off</code> desaparece la cuadrícula
<code>axis([xmin xmax ymin ymax])</code>	Fija los valores máximos y mínimos de los ejes
<code>axis equal</code>	Establece que la escala de los ejes sea la misma
<code>axis square</code>	Fija que la gráfica sea un cuadrado

7.6.5 Gestión de índices alfabéticos y glosario

Para gestionar el índice alfabético se ha utilizado el paquete `\imakeidx`. Se ha optado por generar un índice alfabético con tres columnas y con una letra antes de cada grupo de palabras del índice. Para ello se ha redefinido el entorno `theindex`.

Para incluir algo en el índice el comando standard es `\index` y además se han creado un par de comandos: el comando `\indexit` que pone la palabra correspondiente en cursiva y además se incluye en el índice y `\ind` que es idéntica a `\index`.

Para los glosarios se utiliza el paquete `glossaries` cuyo uso ya ha sido explicado.

7.6.6 Ajustes en fórmulas

La escritura de fórmulas en \LaTeX requiere el conocimiento de un conjunto de comandos que se encuentran muy bien explicados en el documento `mathmode.pdf`. Uno de los comandos que facilitan esta escritura y algunos efectos especiales se basan en el uso de los paquetes `cancel` y `kbordermatrix`. Para permitir la escritura de matrices con diferentes limitadores, hemos utilizado el paquete `array`.

Debido a un bug de \LaTeX es necesario escribir fórmulas demasiado anchas mediante el entorno `\begin{equationw}`. Un ejemplo se muestra a continuación y el resultado del mismo posteriormente.

```

\begin{equationw}
\E\left[ {Y} \right]=\E\left[ {\left( {X-{m_X}} \right)^n} \right]=\begin{cases}
1\times 3\times \cdots \times \left( {2k-1} \right)\sigma_X^{2k}=\frac{(2k)!\sigma_X^{2k}}{2^k k!} & \text{para } n=2k \\
0 & \text{para } n=2k+1
\end{cases}
\end{equationw}

```

$$\mathbb{E}[Y] = \mathbb{E}[(X - m_X)^n] = \begin{cases} 1 \times 3 \times \cdots \times (2k-1) \sigma_X^{2k} = \frac{(2k)! \sigma_X^{2k}}{2^k k!} & \text{para } n = 2k \\ 0 & \text{para } n = 2k + 1 \end{cases} \quad (7.1)$$

Podemos observar la posición del número de ecuación que se ha desplazado debajo de la fórmula. Para crear el entorno que ha permitido resolver este error (el número de ecuación se desplazaba a la derecha), ha sido necesario cargar el paquete `environ`.

También para corregir otro bug de \LaTeX hemos introducido el comando `\sqrtrlua` que evita el desplazamiento del contenido bajo el signo de la raíz cuando dicho contenido tiene una altura superior a un renglón. Esta definición hay que encontrarla en el bloque de gestión de fuentes.

7.6.7 Hiperenlaces

La creación de los hiperenlaces en los documentos escritos en \LaTeX se gestiona mediante el paquete `hyperref`. La documentación de este paquete es extensa y muy detallada y permite, por ejemplo, manejar con facilidad los colores de los hiperenlaces, la utilización de comandos inteligentes como `\autoref`, etc.

El ajuste más genérico de `hyperref` se ha realizado en esta hoja de estilo pero adicionalmente se han definido otros ajustes en el fichero principal de manera que, por ejemplo, se generen ficheros pdf con palabras claves que les permitan ser localizados con los buscadores, como podemos ver en la líneas de código siguientes:

```

\hypersetup
{
  linkcolor=black, %Tocar para poner color en enlaces
  pdfauthor={\elautor},
  pdftitle={\eltitulo},
  citecolor=black, %Tocar para poner color en enlaces, eg blue
  pdfkeywords={Formato de Libro para la ETSI, Universidad de Sevilla}
}

```

Dentro de este apartado, mencionar la utilización de los paquetes `zref-lastpage` y `zref-user` para detectar el número total de páginas de un documento, que viene dado por `\zpageref{Lastpage}`, o bien el ajuste que se realiza sobre las direcciones url, del paquete `url`, que ha sido cargado automáticamente.

7.6.8 Listado de códigos

Es habitual en textos escritos en la Escuela el uso de listados de códigos de muy diversa índole. Nosotros hemos elegido el paquete `listings` para gestionarlos en nuestros documentos. Este paquete se encuentra perfectamente documentado, lo que nos ha permitido realizar algunos ajustes que hemos considerado necesarios.

En el caso de utilizar \LaTeX es necesario que los símbolos no habituales se declaren, como hemos hecho en el siguiente bloque:

```

\ifluatex
\else
\lstset{literature=
  {á}{\'}{a}}1
  {é}{\'}{e}}1
  {í}{\'}{i}}1

```

```

{ó}{\'}{o}}1
{ú}{\'}{u}}1
{Á}{\'}{A}}1
{É}{\'}{E}}1
{Í}{\'}{I}}1
{Ó}{\'}{O}}1
{Ú}{\'}{U}}1
{ñ}{\'}{~{n}}1
{^}{\textsuperscript{\b{o}}}}1
{^}{\textsuperscript{\b{a}}}}1
{,}{\'}{,}}1
}
\fi

```

Y también hemos tenido que ajustar determinados problemas relacionados con el uso de los idiomas español e inglés. Por último, hemos ajustado el “caption” de los códigos mediante la declaración de un formato y modificado ligeramente la manera en la que se realiza el listado de los mismos, mediante el siguiente conjunto de instrucciones.

```

\makeatletter
\renewcommand*{\l@lstlisting}[2]{\@dottedtocline{1}{.1em}{2.8em}{\tocsecc
#1}{\tocsecc #2}}
\makeatother

```

7.6.9 Entornos, teoremas y similares

En este apartado hemos agrupado un conjunto de paquetes relacionados con la creación de entornos específicos y estructuras como los teoremas y elementos similares.

En primer lugar, utilizamos el paquete `mdframed` para poder crear cajas sombreadas que se continúan a través de más de una página. Aunque no se ha utilizado en este estilo, merece señalarse la existencia del paquete `tcolorbox` que permite una gestión probablemente más flexible con este mismo fin.

A continuación, señalamos la creación del entorno `Resumen` que permite realizar resúmenes en el lugar del texto que nos interese, como ya hemos mencionado y mostrado en un apartado anterior.

Para la gestión de los teoremas y elementos similares hemos escogido el paquete `ntheorem` junto con una serie de opciones. Si analizamos las siguientes líneas de código, en la que se define el entorno `teor`:

```

1 \theoremnumbering{arabic}
2 \theoremheaderfont{\aheadteoremas}
3 \theoremseparator{\hspace{.2em}}
4 \theorembodyfont{\itshape}
5 \newtheorem{teor}{\theoremname}[section]

```

Podemos observar que en la línea 1 el esquema de numeración será arábico, esto es, mediante números. Para su declaración utilizaremos el comando `\aheadteoremas` que habíamos definido anteriormente y nos indica que utilizaremos una determinada fuente con las características que habíamos señalado. El enunciado del teorema irá en *cursiva* y el nombre que usaremos viene marcado por el macro `\theoremname`. Por último, al incluir el término `section` en la línea 5 estamos señalando que la numeración de los teoremas se realizará al nivel de sección; esto es, las siguientes líneas de código, en las que también hemos utilizado el entorno `demo`, generan el teorema debajo de las mismas.

```

\begin{teor}\label{th04-02} Si una función  $f\left( {x} \right)$  tiene una
segunda derivada que es no negativa (positiva) en un intervalo dado, la
función es convexa (estrictamente convexa) en ese intervalo.
\end{teor}
\begin{demo} Para probar el teorema, desarrollemos la función en serie de
Taylor alrededor del punto  $x_0$ :
\begin{equation}\label{eq04-321}

```

```
f\left( {x} \right)=f\left( {x_{0}} \right)+f^{\prime}\left( {x_{0}} \right)\left( {x-x_{0}} \right)+\frac{f^{\prime\prime}\left( {x^{\ast}} \right)}{2}\left( {x-x_{0}} \right)^2
\end{equation}
donde  $x^{\ast}$  está entre  $x_0$  y  $x$ . Por hipótesis,  $f^{\prime\prime}\left( {x^{\ast}} \right) \geq 0$ , así que el último término es no negativo para todo  $x$ .

...
\end{demo}
```

Teorema 7.6.1 Si una función $f(x)$ tiene una segunda derivada que es no negativa (positiva) en un intervalo dado, la función es convexa (estrictamente convexa) en ese intervalo.

Demostración. Para probar el teorema, desarrollemos la función en serie de Taylor alrededor del punto x_0 :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x^*)}{2}(x - x_0)^2 \quad (7.2)$$

donde x^* está entre x_0 y x . Por hipótesis, $f''(x^*) \geq 0$, así que el último término es no negativo para todo x .
...

Existen un conjunto de entornos definidos de manera análoga al anterior y su uso puede ser fácilmente deducible del ejemplo anterior.

7.6.10 Otros comandos

Finalizamos la descripción del fichero señalando la existencia de un grupo de comandos muchos de ellos únicamente utilizables en la escritura de un texto como el presente. Otros, como las definiciones de las cabeceras de los exámenes o el que nos permite crear una dedicatoria de nuestro texto. Ambos se gestionan en el fichero principal con líneas de código similares a las siguientes:

```
\dedicatoria{A nuestras familias\\A nuestros maestros}
\titulacion{ Grado en Ingeniería de Tecnología de Telecomunicación}
\asignatura{Comunicaciones Digitales}
\convocatoria{Primera convocatoria. Curso 2013-14}
\fecha{3/02/2014}
```

con el resultado que ya hemos mostrado.

7.7 A modo de conclusión

Hemos tratado de mostrar en este capítulo los elementos más destacados del diseño de la hoja de estilo que hemos realizado para nuestra Escuela. Como hemos dicho con anterioridad, hay mucho de gusto personal en el diseño pero también existe un considerable compromiso con tendencias actuales en la maquetación de textos científicos.

Como todo diseño, es manifiestamente mejorable y nuestro deseo es que este capítulo permita a quien lo desee adaptarla a sus criterios. Tenemos intención de crear una página de preguntas y respuestas que permitan, con la aportación de todos, mejorar esta primera aportación a la creación de una imagen corporativa de los documentos generados en nuestra Escuela.

Índice de Figuras

5.1	Logo de la ETSI	18
5.2	Logo de la ETSI	19
6.1	Atenuación por lluvia excedida en el $p\%$ del tiempo para 13 GHz y 15 km de distancia	31
6.2	Atenuación por lluvia excedida en el $p\%$ del tiempo para 13 GHz y 15 km de distancia	32
7.1	Ejemplos de cabeceras de la página primera de un examen y de la página segunda	44
7.2	Función de distribución y Función densidad de probabilidad de una variable aleatoria continua	46

Índice de Tablas

5.1	Valores de parámetros	18
6.1	Datos de las estaciones, con equipos Flexi-Hopper de Nokia	30
6.2	Datos de las estaciones, con equipos Flexi-Hopper de Nokia	32
7.1	Funciones de manipulación de gráficos	47

Índice de Códigos

5.1	Código para incluir una figura	17
5.2	Representación de la función $\text{rect}(t - T/2)$	20

Bibliografía

- [1] Universidad Internacional de Valencia, *¿cuáles son las aplicaciones de la ia actuales y futuras?*, April 2023.
- [2] José María Hernando-Rábanos, *Transmisión por radio*, 6ª ed., Ed. Ramón Areces, 2008.
- [3] Sage Kelly, Sherrie-Anne Kaye, and Oscar Oviedo-Trespalacios, *What factors contribute to the acceptance of artificial intelligence? a systematic review*, *Telematics and Informatics* **77** (2023), 101925.
- [4] Juan José Murillo-Fuentes, *Fundamentos de radiación y radiocomunicación*, 1ª ed., Ed. ETSI. Universidad de Sevilla, 2007.
- [5] A.V. Oppenheim and R.W. Schafer, *Discrete-time signal processing*, 1 ed., Prentice Hall, Inc., 1989.
- [6] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications.*, reimpresión del trabajo original publicado en el mit en 1942 ed., New York: Wiley, 1949.

Índice alfabético

amsmath, 39
appto, 35
array, 47
autores, 48

babel, 36
backmatter, 40

cancel, 47
caption, 46
corolarios, 17
Cuestiones, 17

Ecuaciones, 17
Ejemplos, 17
enumitem, 45
epigraph, 45
equationw, 47
etoolbox, 35
exscale, 39

Figuras, 17
fontenc, 37
fontspec, 37
formato
 de capítulo, 17
frontmatter, 40

geometry, 36

glossaries, 47

hyperref, 48

icomma, 45
imakeidx, 47
inputenc, 37

kbordermatrix, 47

Lemas, 17
lettrine, 45
listings, 48
lualatex-math, 39
luatextra, 39

mathtools, 39
mdframed, 49
microtype, 36
MinionPro, 39
MnSymbol, 39

ntheorem, 49

Problemas, 17

Resúmenes, 17

scrbook, 36

secnumdepth, 45
señal, 27
 aleatoria, 27
 analógica, 28
 determinista, 27
 digital, 28
 discreta, 28
 muestreada, 28
 no periódica, 27
 periódica, 27
shorttoc, 45
sqrtlua, 48

Tablas, 17
tableofcontents, 39
tcolorbox, 49
teoremas, 17
theindex, 47
titlesec, 41
tocdepth, 40, 45
tocloft, 39

url, 48

xcolor, 44

zref-lastpage, 48
zref-user, 48

