

Trabajo fin de grado

Plataforma de diseño y potenciación de atletas de alto rendimiento



Alejandro Monterrubio Navarro

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente nº 11

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Plataforma de diseño y potenciación de atletas de
alto rendimiento**

Con implementación de inteligencia artificial

Autor: Alejandro Monterrubio Navarro

Tutor: Pablo Cerro Cañizares

mayo 2024

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 3 de Noviembre de 2017 por UNIVERSIDAD AUTÓNOMA DE MADRID
Francisco Tomás y Valiente, n.º 1
Madrid, 28049
Spain

Alejandro Monterrubio Navarro

Plataforma de diseño y potenciación de atletas de alto rendimiento

Alejandro Monterrubio Navarro

C\ Francisco Tomás y Valiente N.º 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mis padres, mi hermano, mi novia y mis amigos por el amor y apoyo incondicional.

A mi tutor por su paciencia y motivación.

Y a todos ellos, por no dejar de confiar en mí.

Gracias

El poder viene en respuesta a una necesidad, no a un deseo. Tienes que crear esa necesidad.

Akira Toriyama

PREFACIO

Este estilo de $\text{\LaTeX}2_{\epsilon}$ ha sido diseñado con dos propósitos. El primer propósito es el de facilitar en lo posible la escritura de trabajos de fin de grado y de máster y de tesis doctorales. En ese sentido se han diseñado un conjunto de comandos que simplifican la escritura y diseño de estos trabajos pero que reducen en cierta forma las capacidades de los paquetes de \LaTeX utilizados. Sin embargo, dado que los paquetes están incluidos en esta clase, pueden utilizarse directamente y hacer diseños más complejos pero si se hace esto se recomienda mantener una estética coherente con el resto del documento.

El segundo de los propósitos es que estos documentos mantengan una estética uniforme en la Universidad Autónoma de Madrid y fomentar una imagen corporativa en documentos tan relevantes como los trabajos de fin de grado o de máster y las tesis doctorales. Por ese motivo se recomienda mantener una coherencia estética en todo momento. El diseño facilita esa coherencia pero es posible salirse del diseño si se mantiene dicha coherencia.

Como creador de este estilo espero fervientemente que al usar este estilo te sientas cómodo y te facilite la escritura de un documento que es muy relevante en esta etapa de tu vida. Para facilitártela aún más, el código fuente de este documento también está disponible en tu ordenador o en overleaf para que te sirva a modo de ejemplo.

Eloy Anguiano Rey

AGRADECIMIENTOS

En primer lugar me gustaría agradecer a la Escuela Politécnica Superior por su apoyo para la creación de esta clase y que sea el formato básico para la creación de tesis, trabajos fin de grado y trabajos fin de master.

En particular quiero destacar el trabajo realizado por Fernando López-Colino por su apoyo en la comisión de imagen institucional y por sus comentarios para mejorar este estilo.

También quiero tener un recuerdo para Carmen Navarrete Navarrete dado que este estilo comencé a crearlo a partir de sus necesidades a la hora de escribir la tesis. Y por supuesto a no quiero olvidarme de mi esposa e hijos que han servido de conejillos de indias en sus correspondientes trabajos fin de master y de grado. No quiero olvidar a todos los estudiantes que me pidieron este estilo y lo han usado para presentar sus trabajos pero son muchos y podría olvidarme de alguno, por tanto, mi agradecimiento en general a todos ellos.

RESUMEN

La intersección entre el deporte y la nutrición es fundamental para mantener un estilo de vida saludable. Sin embargo, muchas personas enfrentan dificultades para integrar prácticas saludables en su rutina diaria debido a la falta de tiempo y conocimientos específicos. Este desafío a menudo conduce a la adopción de dietas genéricas o rutinas de ejercicio inadecuadas, lo que incrementa el riesgo de resultados negativos y lesiones. La principal barrera para adoptar un estilo de vida saludable radica en la falta de información relevante y accesible, así como en la limitación de tiempo para adquirir dichos conocimientos.

Reconociendo esta necesidad, mi Trabajo Final de Grado (TFG) se centra en simplificar el acceso a planes de nutrición y ejercicio personalizados. El objetivo es proporcionar a los usuarios, independientemente de su experiencia previa en deporte, las herramientas necesarias para mejorar su rendimiento o iniciar un camino hacia el bienestar físico, sin requerir una inversión significativa de tiempo en aprendizaje autodidacta o prácticas erróneas.

Como solución, he desarrollado FitFuelBalance, una plataforma integrada que ofrece servicios tanto en formato de aplicación web como móvil. Esta aplicación permite a los usuarios solicitar servicios personalizados de entrenadores y nutricionistas, quienes pueden utilizar un repositorio de ejercicios y dietas predefinidas para crear planes a medida. Los usuarios tienen la capacidad de revisar detalles y solicitar ajustes en sus rutinas a los profesionales a través de la aplicación, garantizando así una experiencia personalizada y eficiente. Además, se incorpora el uso de inteligencia artificial para facilitar recomendaciones instantáneas y adaptaciones, reforzando la personalización de los planes ofrecidos.

El proyecto ha despertado un gran interés entre distintos perfiles de usuarios, desde expertos deportivos hasta individuos sin experiencia previa, así como personas dedicadas al entrenamiento o la nutrición, demostrando el potencial de la tecnología para transformar la manera en que las personas acceden y gestionan su salud y bienestar. Este documento detallará el análisis, diseño, codificación, pruebas e implementación de la plataforma FitFuelBalance, dirigida a empresas, instituciones y particulares que buscan mejorar su salud y rendimiento físico de manera eficiente y personalizada.

PALABRAS CLAVE

Aplicación Web, Django, React, Nutrición, Deporte, Usuario, Ordenador, Móvil, Inteligencia Artificial

ABSTRACT

The intersection between sports and nutrition is fundamental to maintaining a healthy lifestyle. However, many individuals struggle to integrate healthy practices into their daily routines due to a lack of time and specific knowledge. This challenge often leads to the adoption of generic diets or inadequate exercise routines, increasing the risk of negative outcomes and injuries. The main barrier to adopting a healthy lifestyle lies in the lack of relevant and accessible information, as well as the limited time to acquire such knowledge.

Recognizing this need, my Final Degree Project (TFG) focuses on simplifying access to personalized nutrition and exercise plans. The aim is to provide users, regardless of their prior sports experience, with the necessary tools to improve their performance or embark on a journey towards physical well-being, without requiring significant time investment in self-learning or incorrect practices.

As a solution, I have developed FitFuelBalance, an integrated platform offering services both as a web and mobile application. This application allows users to request personalized services from trainers and nutritionists, who can use a repository of predefined exercises and diets to create tailored plans. Users have the ability to review details and request adjustments to their routines from professionals through the application, ensuring a personalized and efficient experience. Additionally, the use of artificial intelligence is incorporated to facilitate instant recommendations and adaptations, enhancing the personalization of the plans offered.

The project has garnered significant interest among various user profiles, from sports experts to individuals with no prior experience, as well as those dedicated to training or nutrition, demonstrating the potential of technology to transform how people access and manage their health and well-being. This document will detail the analysis, design, coding, testing, and implementation of the FitFuelBalance platform, aimed at businesses, institutions, and individuals seeking to improve their health and physical performance efficiently and personally.

KEYWORDS

Document design, $\text{\LaTeX}2_{\epsilon}$, thesis, final degree project, final master project

ÍNDICE

LISTAS

Lista de algoritmos

Lista de códigos

Lista de cuadros

Lista de ecuaciones

Lista de figuras

Lista de tablas

Lista de cuadros

ESTÉTICA

La estética de este documento no debe ser modificada, ni el tipo de letra, ni los colores, ni los márgenes ni ningún otro elemento predefinido en las funciones y entornos diseñados. Si por algún motivo se tiene la necesidad de utilizar elementos externos a este estilo se recomienda hacerlo utilizando unos colores y estética similares al resto del documento. A pesar de todo hay algunos elementos estéticos que el usuario puede seleccionar.

1.1. Tipo de documento

Este estilo está diseñado para escribir tres tipos de documentos distintos que tienen pequeñas variaciones entre ellos: trabajo fin de grado, trabajo fin de máster y una doctoral. Para poder seleccionar estas variaciones es necesario indicarlo en las opciones de la clase. Estas opciones son: **tfg**, **tfm** o **thesis**. Dado que el documento que más veces va a ser generado son los trabajos fin de grado la opción por defecto es **tfg** y por tanto no es necesario indicarla. La forma de seleccionar esta opción es de la siguiente forma: `\documentclass[thesis]{tfgtfmthesisuam}`.


1.2. Gama de colores

Se dispone de una gran cantidad de gamas de colores que está diseñada utilizando los colores corporativos de cada facultad y escuela universitaria de la Universidad Autónoma de Madrid, por ello la gama de color que se debe utilizar en cada caso está relacionada con la facultad o escuela correspondiente. Así mismo se dispone de una gama propia de la universidad para casos en los que no se corresponda con la imagen corporativa asociada a una determinada facultad o escuela que es la gama asociada a la universidad.

Al igual que para el tipo de documento el color corporativo debe seleccionarse como una opción del documento. Para seleccionar la imagen corporativa genérica debe utilizarse la opción **uambased** y para seleccionar la imagen corporativa de una facultad o escuela debe utilizarse alguna de las siguientes: **cienciasbased**, **derechobased**, **economicasbased**, **enfermeriabased**, **epsbased**, **filosofiabased**,

fisioterapiabased, **medicinabased**, **profesoradobased** o **psicologiabased**. A modo de ejemplo y mezclado con la selección realizada en la sección anterior: `\documentclass[thesis, epsbased]{tfgtthesisuam}`.

1.3. Colores

El uso de los colores es muy simple, basta con usar el comando `\color{nombrecolor}`. Hay muchos colores predefinidos dependiendo de la gama de colores elegida. En la tabla ?? se muestran todos estos colores donde el nombre del color se construye utilizando el prefijo correspondiente a su columna, la gama de colores representado en la fila y siempre acabado con el sufijo 'based'. Una de las columnas no tiene prefijo y se corresponde con el color institucional. Por ejemplo, el color **depsbased** sería .

	ud	vd	d		l	vl	ul	bg
uam								
ciencias								
derecho								
economicas								
enfermeria								
eps								
filosofia								
fisioterapia								
medicina								
profesorado								
psicologia								

Tabla 1.1: Gammas de colores y nombres de los colores correspondientes. El nombre del color se construye utilizando el prefijo correspondiente a su columna, la gama de colores representado en la fila y siempre acabado con el sufijo 'based'.

Así mismo existen un conjunto de colores para cada perfil de color que pueden verse en la tabla ?? para los colores básicos, para los colores 'vl' (verylight) se pueden ver en la tabla ?? y para los colores 'vd' (verydark) se pueden ver en la tabla ??.

1.4. Uso de los colores

Se puede usar cualquier color usando su nombre o con el comando `\color{nombrecolor}` según corresponda. Sin embargo no se recomienda el uso de estos colores de forma aleatoria. Según la gama de colores elegida se puede usar un conjunto de colores mucho más simple y que se corresponden

























































































	one	two	three	four	five	six	seven	eight
uam								
ciencias								
derecho								
economicas								
enfermeria								
eps								
filosofia								
fisioterapia								
medicina								
profesorado								
psicologia								

Tabla 1.2: Colores complementarios. El nombre del color se construye concatenando el nombre de la fila con el sufijo 'based' y se añade como segundo sufijo el nombre de la columna.

























































































	one	two	three	four	five	six	seven	eight
uam								
ciencias								
derecho								
economicas								
enfermeria								
eps								
filosofia								
fisioterapia								
medicina								
profesorado								
psicologia								

Tabla 1.3: Colores complementarios claros. El nombre del color se construye concatenando el nombre de la fila con el prefijo 'vi', el sufijo 'based' y se añade como segundo sufijo el nombre de la columna.























































































	one	two	three	four	five	six	seven	eight
uam								
ciencias								
derecho								
economicas								
enfermeria								
eps								
filosofia								
fisioterapia								
medicina								
profesorado								
psicologia								

Tabla 1.4: Colores complementarios oscuros. El nombre del color se construye concatenando el nombre de la fila con el prefijo 'vd', el sufijo 'based' y se añade como segundo sufijo el nombre de la columna.

con los de las tablas de color de la gama elegida. Los colores tienen el nombre adecuado al entorno en el que deben ser usados. La lista de estos colores es la siguiente:

maincolor Color principal, usado en casi todos los elementos del documento que necesiten color en el texto.

dmaincolor Color principal en su versión oscura (no se usa por defecto en ningún elemento).

descriptioncolor Color que se utiliza en elemento descrito en las listas de descripción como esta, por defecto idéntico a 'maincolor'.

headfootcolor Color utilizado en las cabeceras y pies de página, por defecto idéntico a 'maincolor'.

textcolor Siempre es negro.

textboxfgcolor Color 'foreground' de los cuadros de texto.

textboxbgcolor Color 'background' de los cuadros de texto.

codefgcolor Color 'foreground' de los códigos presentados.

codebgcolor Color 'background' de los códigos presentados.

equationfgcolor Color 'foreground' de las ecuaciones.

equationbgcolor Color 'background' de las ecuaciones.

commentcolor Color de los comentarios en los códigos.

complementaryone ... complementaryeight Son los ocho colores complementarios usables en gráficos.

complementarylightone ... complementarylighteight Son los ocho colores complementarios suaves.

complementarydarkone ... complementarydarkeight Son los ocho colores complementarios oscuros.

ESTADO DEL ARTE

En esta sección, se presenta un análisis detallado de las soluciones actuales en el ámbito de la nutrición y el ejercicio, con el objetivo de identificar las limitaciones y oportunidades para mejorar la integración de estas prácticas en la vida cotidiana de las personas. La revisión de las tecnologías y plataformas existentes proporcionará una base sólida para entender cómo la plataforma FitFuelBalance puede ofrecer soluciones innovadoras y personalizadas que aborden las necesidades de diversos perfiles de usuarios, desde deportistas profesionales hasta individuos sin experiencia previa en salud y fitness.

2.1. Contexto y Necesidad

2.1.1. Importancia de la Nutrición y el Ejercicio

La intersección entre la nutrición y el ejercicio es esencial para mantener un estilo de vida saludable. La nutrición adecuada proporciona los nutrientes necesarios para el funcionamiento óptimo del cuerpo, mejora la capacidad de recuperación y reduce el riesgo de enfermedades crónicas. Por otro lado, el ejercicio regular fortalece el sistema cardiovascular, mejora la salud mental y ayuda en el control del peso corporal.

Numerosos estudios han demostrado que una combinación equilibrada de dieta y ejercicio puede prolongar la vida útil, mejorar la calidad de vida y aumentar la capacidad física y mental. Sin embargo, la implementación de estas prácticas de manera efectiva requiere un conocimiento adecuado y la habilidad para personalizar rutinas y dietas según las necesidades individuales.

2.1.2. Desafíos Comunes

A pesar de la clara importancia de la nutrición y el ejercicio, muchas personas enfrentan desafíos significativos para integrar estas prácticas en su vida diaria. Los principales desafíos incluyen:

- **Falta de tiempo:** La vida moderna y las responsabilidades laborales y familiares a menudo dejan

poco tiempo para la planificación y la implementación de rutinas de salud y fitness. Muchas personas encuentran difícil dedicar tiempo suficiente para hacer ejercicio y preparar comidas saludables.

- **Falta de conocimientos específicos:** Sin una orientación adecuada, las personas pueden adoptar dietas genéricas o rutinas de ejercicio que no están alineadas con sus objetivos o necesidades específicas. Esto puede resultar en falta de progreso, desmotivación e incluso lesiones.
- **Adopción de prácticas inadecuadas:** La abundancia de información contradictoria y poco confiable en internet puede llevar a la adopción de dietas extremas o regímenes de ejercicio poco saludables. Sin la guía adecuada, es fácil que las personas caigan en trampas de marketing que prometen resultados rápidos sin considerar los riesgos asociados.

2.1.3. Necesidad de Información Relevante y Accesible

La principal barrera para adoptar un estilo de vida saludable radica en la falta de información relevante y accesible. Muchas personas no tienen el tiempo ni los recursos para educarse adecuadamente sobre nutrición y ejercicio. Además, la personalización de estos planes es crucial, ya que cada individuo tiene necesidades, capacidades y objetivos únicos.

Reconociendo esta necesidad, el desarrollo de herramientas y plataformas que proporcionen información precisa, relevante y personalizada es fundamental. Estas herramientas deben ser accesibles, fáciles de usar y capaces de adaptarse a las circunstancias individuales de cada usuario.

Este contexto subraya la importancia de crear soluciones como FitFuelBalance, que integren nutrición y ejercicio en una plataforma única y personalizada, abordando los desafíos mencionados y facilitando el acceso a planes de salud efectivos y adaptados.

2.2. Soluciones Existentes

En esta sección, se examinan las soluciones y plataformas actuales en el mercado que abordan la integración de la nutrición y el ejercicio. Se identifican sus características, ventajas y limitaciones.

2.2.1. Aplicaciones de Nutrición

Existen numerosas aplicaciones que ayudan a los usuarios a gestionar su nutrición. Entre las más populares se encuentran:

- **MyFitnessPal:** Esta aplicación permite a los usuarios registrar su ingesta de alimentos y monitorear su consumo calórico diario. Ofrece una base de datos extensa de alimentos, seguimiento de macronutrientes y la posibilidad de escanear códigos de barras para facilitar el registro de alimentos.

Ventajas:

- Amplia base de datos de alimentos.
- Fácil registro de alimentos mediante escaneo de códigos de barras.

- Seguimiento detallado de macronutrientes.

Limitaciones:

- Planes de nutrición genéricos.
- Necesidad de una versión premium para acceder a funciones avanzadas.
- **Yazio:** Similar a MyFitnessPal, Yazio ofrece planes dietéticos personalizados basados en los objetivos del usuario, ya sea perder peso, ganar masa muscular o mantener una dieta saludable. Incluye funciones como recetas saludables, planes de comidas y seguimiento de nutrientes.

Ventajas:

- Planes de nutrición personalizados.
- Recetas y planes de comidas saludables.
- Interfaz amigable y fácil de usar.

Limitaciones:

- Algunas funciones requieren suscripción premium.
- Menor base de datos de alimentos en comparación con MyFitnessPal.

2.2.2. Aplicaciones de Ejercicio

Las aplicaciones de ejercicio proporcionan rutinas de entrenamiento y seguimiento del progreso físico. Entre las más destacadas están:

- **Nike Training Club:** Ofrece una amplia variedad de entrenamientos dirigidos por entrenadores profesionales, adaptados a diferentes niveles de fitness y objetivos. La aplicación incluye videos instructivos y planes de entrenamiento que se pueden seguir en casa o en el gimnasio.

Ventajas:

- Variedad de entrenamientos para diferentes niveles y objetivos.
- Instrucciones detalladas y videos de alta calidad.
- Gratis para la mayoría de las funciones.

Limitaciones:

- Falta de personalización avanzada basada en datos del usuario.
- Necesidad de conexión a internet para acceder a los videos.

- **Fitbod:** Utiliza algoritmos para personalizar los entrenamientos basándose en el progreso del usuario, el equipo disponible y las metas personales. Proporciona un seguimiento detallado del rendimiento y adapta los ejercicios según los resultados obtenidos.

Ventajas:

- Personalización avanzada de los entrenamientos.
- Adaptación de ejercicios según el progreso y el equipo disponible.
- Seguimiento detallado del rendimiento.

Limitaciones:

- Suscripción necesaria para acceder a todas las funciones.
- Puede ser complejo para principiantes.

- **Calisteniapp:** Esta aplicación se centra en el entrenamiento de calistenia, utilizando el peso corporal del usuario para mejorar la fuerza y la flexibilidad. Calisteniapp ofrece una variedad de rutinas adaptadas a diferentes niveles de habilidad y permite a los usuarios personalizar sus entrenamientos según sus objetivos específicos.

Ventajas:

- Enfoque especializado en calistenia.
- Rutinas personalizables basadas en el nivel y los objetivos del usuario.
- Comunidad activa y funciones de seguimiento del progreso.

Limitaciones:

- Puede no ser adecuada para usuarios que prefieren entrenamientos con pesas o equipos de gimnasio.
- Algunas funciones avanzadas requieren suscripción premium.

2.2.3. Plataformas Combinadas

Algunas plataformas intentan integrar tanto la nutrición como el ejercicio, proporcionando una solución más holística para la gestión de la salud.

- **Fitbit:** Además de sus dispositivos wearables, Fitbit ofrece una plataforma que integra el seguimiento de la actividad física con la monitorización de la ingesta de alimentos, permitiendo a los usuarios obtener una visión completa de su salud y estado físico.

Ventajas:

- Integración de datos de actividad física y nutrición.
- Amplia gama de dispositivos wearables.
- Comunidad activa y funciones sociales.

Limitaciones:

- Personalización limitada en los planes de ejercicio y nutrición.
- Dependencia de dispositivos adicionales para obtener el máximo beneficio.

- **Samsung Health:** Esta aplicación proporciona seguimiento de la actividad física, monitoreo del sueño y registro de la ingesta de alimentos. Ofrece recomendaciones personalizadas basadas en los datos recogidos, aunque su enfoque en la personalización aún es limitado.

Ventajas:

- Amplia gama de funciones de salud y bienestar.
- Integración con dispositivos Samsung.
- Funciones de monitoreo del sueño y la salud en general.

Limitaciones:

- Personalización limitada en comparación con plataformas especializadas.
- Menor integración con dispositivos de otras marcas.

2.2.4. Limitaciones de las Soluciones Actuales

A pesar de los avances, las soluciones actuales enfrentan varias limitaciones:

- **Falta de Personalización:** Muchas aplicaciones ofrecen planes genéricos que no consideran completamente las necesidades individuales de cada usuario.
- **Acceso a Asesoramiento Profesional:** La mayoría de las plataformas no proporcionan acceso directo a profesionales de la salud y el fitness, lo que limita la capacidad de obtener asesoramiento personalizado.
- **Costo:** Algunas aplicaciones y servicios personalizados pueden ser costosos, lo que restringe el acceso para muchos usuarios.
- **Integración de Datos:** La falta de integración entre diferentes dispositivos y aplicaciones puede dificultar una visión holística de la salud del usuario.

2.3. Plataformas Personalizadas para la Nutrición y el Ejercicio

A medida que la demanda de soluciones de salud personalizadas crece, diversas plataformas han surgido para ofrecer planes de nutrición y ejercicio adaptados a las necesidades individuales de los usuarios. Estas plataformas utilizan datos personales, inteligencia artificial y el asesoramiento de expertos para proporcionar recomendaciones específicas.

2.3.1. Plataformas de Nutrición Personalizada

- **Nutrifix:** Nutrifix crea planes de comidas personalizados basados en las metas de salud del usuario, restricciones dietéticas y preferencias alimenticias. Utiliza datos de aplicaciones de seguimiento de actividad física para ajustar las recomendaciones nutricionales.

Ventajas:

- Personalización basada en datos de actividad física.
- Consideración de restricciones dietéticas y preferencias.

Limitaciones:

- Necesidad de sincronizar con otras aplicaciones para obtener datos precisos.
- Algunas funciones avanzadas requieren una suscripción.

- **PlateJoy:** Ofrece planes de comidas personalizados y listas de compras adaptadas a las necesidades dietéticas y preferencias del usuario. PlateJoy también proporciona recetas y sugerencias de preparación de comidas.

Ventajas:

- Listas de compras y recetas personalizadas.
- Adaptación a diversas necesidades dietéticas (vegetariano, sin gluten, etc.).

Limitaciones:

- Suscripción necesaria para acceder a todas las funciones.
- Dependencia de la precisión de los datos ingresados por el usuario.

2.3.2. Plataformas de Ejercicio Personalizado

- **Freeletics:** Utiliza inteligencia artificial para crear planes de entrenamiento personalizados basados en los niveles de fitness, objetivos y progreso del usuario. Freeletics también ofrece videos instructivos y seguimiento de rendimiento.

Ventajas:

- Planes de entrenamiento personalizados y adaptativos.
- Videos instructivos y seguimiento del rendimiento.

Limitaciones:

- Suscripción necesaria para acceder a funciones premium.
- Puede no ser adecuado para usuarios que prefieren entrenamiento con equipos específicos.

- **JEFIT:** Ofrece rutinas de entrenamiento personalizadas y seguimiento del progreso. JEFIT también proporciona una base de datos de ejercicios y permite a los usuarios registrar sus entrenamientos y compartirlos con la comunidad.

Ventajas:

- Base de datos extensa de ejercicios.
- Comunidad activa para compartir y comparar rutinas.

Limitaciones:

- Requiere suscripción para acceder a algunas funciones avanzadas.
- Personalización limitada sin la versión premium.

2.3.3. Plataformas Combinadas

Algunas plataformas buscan integrar tanto la nutrición como el ejercicio para ofrecer una solución completa.

- **8fit:** Combina planes de entrenamiento y nutrición personalizados, adaptados a los objetivos del usuario, como perder peso o ganar músculo. 8fit ofrece recetas, listas de compras y rutinas de ejercicio con videos instructivos.

Ventajas:

- Integración de planes de nutrición y ejercicio.
- Videos instructivos y listas de compras.

Limitaciones:

- Suscripción necesaria para acceder a todas las funciones.
- Requiere consistencia en el ingreso de datos por parte del usuario.

- **Noom:** Además de ofrecer planes de dieta personalizados, Noom incluye entrenamientos físicos adaptados a las necesidades del usuario. Utiliza la psicología del comportamiento para ayudar a los usuarios a adoptar hábitos saludables a largo plazo.

Ventajas:

- Enfoque en la psicología del comportamiento para cambios duraderos.
- Integración de dieta y ejercicio en un solo plan.

Limitaciones:

- Suscripción necesaria para acceder a las funciones completas.
- Puede ser más caro en comparación con otras aplicaciones.

2.3.4. Limitaciones y Oportunidades de Mejora

A pesar de los avances en la personalización de planes de nutrición y ejercicio, las soluciones actuales todavía enfrentan desafíos:

- **Adaptabilidad:** La capacidad de ajustar automáticamente los planes basados en el progreso y los cambios en los objetivos del usuario sigue siendo limitada.
- **Accesibilidad Financiera:** Muchas de las plataformas personalizadas requieren suscripciones pagas, lo que puede ser una barrera para algunos usuarios.
- **Integración de Datos:** La integración efectiva de datos de múltiples fuentes, como dispositivos de seguimiento de actividad y aplicaciones de salud, aún necesita mejoras.
- **Interacción Humana:** La falta de interacción directa con profesionales de la salud y el fitness puede limitar la efectividad de las recomendaciones personalizadas.

Estas limitaciones abren oportunidades para el desarrollo de nuevas plataformas que puedan ofrecer un mayor nivel de personalización, accesibilidad y apoyo profesional.

2.4. Tecnologías Utilizadas.

En esta sección, se explican los principales frameworks, bibliotecas y tecnologías utilizadas para el desarrollo del proyecto *FitFuelBalance*.

Django

Django es un framework de alto nivel para el desarrollo web en Python que fomenta el desarrollo rápido y un diseño limpio y pragmático. Fue creado en 2005 por Adrian Holovaty y Simon Willison mientras trabajaban en el periódico *Lawrence Journal-World*. Django se basa en la idea de DRY (Don't Repeat Yourself) y ofrece herramientas poderosas como un ORM (Object-Relational Mapping), un sistema de plantillas y una interfaz administrativa automáticamente generada.

Ventajas:

- Administración automática de la base de datos.
- Sistema de plantillas potente y flexible.
- Gran cantidad de bibliotecas y paquetes disponibles.

Limitaciones:

- Puede ser más complejo para proyectos pequeños.

- Requiere familiaridad con Python y el ecosistema de Django.

React

React es una biblioteca de JavaScript para crear interfaces de usuario para sistemas distribuidos, especialmente aquellos que trabajan con datos y actualizaciones en tiempo real. Utiliza JSX, una sintaxis muy similar a HTML, lo que facilita su implementación. El sistema se basa en componentes que contienen su propia lógica y estado.

Ventajas:

- Facilita la creación de interfaces de usuario dinámicas y responsivas.
- Gran comunidad y ecosistema de bibliotecas y herramientas.
- Modularización y reutilización de componentes.

Limitaciones:

- Requiere un proceso de configuración inicial.
- La gestión del estado en aplicaciones grandes puede ser compleja sin herramientas adicionales.

React Native

React Native es un marco de desarrollo de aplicaciones móviles creado por Facebook en 2015. Permite a los desarrolladores construir aplicaciones móviles utilizando JavaScript y React, compilando a código nativo para iOS y Android.

Ventajas:

- Reutilización de código entre plataformas iOS y Android.
- Ecosistema robusto y activo.
- Componentes nativos que garantizan un rendimiento óptimo.

Limitaciones:

- Algunas funcionalidades nativas pueden requerir desarrollo adicional.
- La actualización de versiones puede requerir ajustes significativos.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional y objeto-relacional, conocido por su estabilidad, extensibilidad y cumplimiento de estándares. Es una elección popular para aplicaciones web y sistemas de información que requieren una base de datos robusta y escalable.

Ventajas:

- Altamente extensible y soporte para tipos de datos avanzados.

- Transacciones ACID completas garantizando la integridad de los datos.
- Amplia comunidad y soporte de herramientas.

Limitaciones:

- Puede requerir configuración y ajuste para un rendimiento óptimo.
- Complejidad en la administración para usuarios sin experiencia previa en bases de datos.

Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, conocido por su sintaxis legible y facilidad de uso. En este proyecto, Python se utiliza tanto para el desarrollo de Django como para otras funciones del backend.

Ventajas:

- Sintaxis sencilla y clara, ideal para desarrollo rápido.
- Gran cantidad de bibliotecas y frameworks disponibles.
- Comunidad activa y extensa documentación.

Limitaciones:

- Puede no ser tan rápido como lenguajes compilados en ciertas tareas.
- Requiere un buen manejo de entornos virtuales para la gestión de dependencias.

JavaScript

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para el desarrollo de aplicaciones web dinámicas. Es el lenguaje más común para el desarrollo frontend y se utiliza en este proyecto junto con React.

Ventajas:

- Ampliamente soportado por todos los navegadores web.
- Gran cantidad de frameworks y bibliotecas para desarrollo web.
- Ideal para la creación de aplicaciones web interactivas y dinámicas.

Limitaciones:

- Problemas de compatibilidad entre diferentes navegadores.
- Requiere una gestión adecuada del código para evitar problemas de rendimiento.

Bootstrap y CSS

Bootstrap es un framework de front-end de código abierto que permite diseñar sitios y aplicaciones web de forma rápida y sencilla. CSS (Cascading Style Sheets) es el lenguaje utilizado para describir la

presentación de un documento escrito en HTML o XML.

Ventajas:

- Facilita la creación de interfaces responsivas y atractivas.
- Gran cantidad de componentes predefinidos.
- Compatible con todos los navegadores modernos.

Limitaciones:

- Puede resultar en sitios web que se ven similares si no se personaliza adecuadamente.
- Dependencia de archivos externos puede afectar el rendimiento si no se gestiona bien.

Estas tecnologías han sido seleccionadas y utilizadas en el proyecto *FitFuelBalance* para asegurar un desarrollo eficiente, una experiencia de usuario óptima y una alta escalabilidad del sistema.

DISEÑO

En esta sección se describirá el diseño del sistema FitFuelBalance, abordando las directrices y metodologías empleadas durante el ciclo de vida del proyecto, así como su arquitectura. El objetivo es proporcionar una visión detallada de cómo se estructuró y organizó el desarrollo para garantizar la eficiencia y funcionalidad del sistema.

3.1. Definición del proyecto y alcance

El proyecto FitFuelBalance tiene como objetivo desarrollar una plataforma integrada de nutrición y ejercicio, accesible tanto en formato web como móvil. Esta plataforma permite a los usuarios recibir planes personalizados de nutrición y ejercicio, creados por profesionales en base a sus necesidades individuales. El sistema no solo servirá como una herramienta práctica para usuarios individuales, sino también como una demostración de cómo la tecnología puede facilitar la gestión de la salud y el bienestar.

El proyecto abarca el desarrollo completo de una aplicación web y móvil, desde la fase de análisis y diseño, pasando por la implementación y pruebas, hasta su despliegue final en un entorno de producción. La plataforma está diseñada para ser accesible desde cualquier dispositivo con conexión a Internet, asegurando que los usuarios puedan interactuar con sus planes y profesionales en cualquier momento y lugar.

El ciclo de vida del proyecto está basado en un enfoque incremental, lo que permite la adición progresiva de funcionalidades y asegura que cada incremento deje el sistema en un estado funcional y robusto. Esto es especialmente beneficioso para adaptarse a nuevos requisitos y mejoras continuas basadas en la retroalimentación de los usuarios.

Con este proyecto, se busca no solo ofrecer una solución práctica a los usuarios, sino también demostrar el potencial de la integración de tecnologías modernas como Django, React, React Native y PostgreSQL en la creación de sistemas distribuidos eficientes y escalables.

3.2. Metodología

Para el desarrollo de FitFuelBalance, se ha optado por utilizar un modelo incremental. Esta metodología permite el desarrollo del proyecto en pequeños incrementos funcionales, lo que facilita la incorporación de nuevas funcionalidades de manera progresiva, esto permite el desarrollo de un sistema robusto y haciendo que los incrementos no afecten a otros subsistemas al desarrollarse.

Fases del proyecto

El proyecto se ha dividido en varias fases clave, cada una con objetivos específicos y entregables definidos:

- **Análisis de requisitos:** Se recopilaron los requisitos funcionales y no funcionales a través de reuniones con los stakeholders y encuestas a usuarios potenciales. Esta fase incluyó la creación de historias de usuario y la definición del backlog del producto.
- **Diseño:** En esta fase se definió la arquitectura del sistema, incluyendo la estructura de la base de datos, la arquitectura de la aplicación web y móvil, y la integración de servicios externos. Se crearon diagramas UML para visualizar la arquitectura y las interacciones entre los componentes del sistema.
- **Desarrollo:** El desarrollo se realizó en incrementos, cada uno añadiendo nuevas funcionalidades al sistema. Las tecnologías utilizadas en esta fase incluyen Django para el backend, React para el frontend web, React Native para la aplicación móvil, y PostgreSQL gestionado a través de Neon para la base de datos.
- **Pruebas:** Se realizaron pruebas unitarias, de integración y funcionales para garantizar la calidad del código y el correcto funcionamiento de la aplicación. También se llevaron a cabo pruebas de usabilidad con un grupo de usuarios seleccionados para obtener feedback directo y realizar mejoras en la interfaz y la experiencia de usuario.
- **Despliegue:** La aplicación se desplegó en entornos de prueba y producción utilizando Render para el alojamiento tanto del frontend como del backend. Se implementaron procesos de CI/CD (Integración Continua/Despliegue Continuo) para automatizar las pruebas y el despliegue.
- **Mantenimiento:** Se establecieron procedimientos para el mantenimiento y actualización continua de la aplicación, incluyendo la monitorización del rendimiento y la gestión de incidencias a través de un sistema de tickets.

Herramientas y técnicas

Durante el desarrollo del proyecto se utilizaron diversas herramientas y técnicas para apoyar el modelo incremental:

- **Git y GitHub:** Para el control de versiones y la colaboración en el código fuente.
- **Visual Studio Code:** Como entorno de desarrollo integrado (IDE) para la programación.
- **Figma:** Para el diseño de interfaces de usuario y la creación de prototipos.
- **React Native CLI:** Para la prueba y depuración de la aplicación móvil.
- **Neon:** Para la gestión de la base de datos PostgreSQL.
- **Render:** Para el despliegue y alojamiento del frontend y backend.

Este modelo incremental ha permitido un desarrollo progresivo y adaptable, asegurando que el

producto final cumpla con los requisitos de los usuarios y sea de alta calidad.

3.3. Arquitectura

La arquitectura de FitFuelBalance se ha diseñado para ser modular, escalable y mantener una separación clara entre las distintas capas del sistema. A continuación, se describen los principales componentes de la arquitectura y sus interacciones.

El sistema está estructurado en tres capas principales:

- **Capa de Presentación:**
 - **Frontend Web:** Desarrollado utilizando React, el frontend web se encarga de la interacción con el usuario a través de una interfaz intuitiva y responsive. Utiliza Bootstrap y CSS para el diseño y estilo visual.
 - **Aplicación Móvil:** Desarrollada con React Native, esta aplicación permite a los usuarios acceder a los servicios de FitFuelBalance desde dispositivos móviles iOS y Android. La interfaz móvil está diseñada para ser amigable y fácil de usar.
- **Capa de Lógica de Negocio:**
 - **Backend:** Desarrollado con Django, el backend maneja la lógica de negocio de la aplicación, las reglas de negocio y la gestión de datos. Implementa APIs RESTful para la comunicación con el frontend y la aplicación móvil.
- **Capa de Datos:**
 - **Base de Datos:** Se utiliza PostgreSQL para almacenar y gestionar los datos de la aplicación. Neon gestiona la base de datos, proporcionando alta disponibilidad y escalabilidad.

Diagrama de Arquitectura

[width=0.8]FUTURA_IMAGEN_ARQUITECTURA.png

Figura 3.1: Diagrama de Arquitectura de FitFuelBalance

Figura 3.2:

Descripción de Componentes

- **Frontend Web:**
 - **React:** Utilizado para construir una interfaz de usuario dinámica y eficiente. React permite la creación de componentes reutilizables y un manejo eficiente del estado de la aplicación.
 - **Bootstrap y CSS:** Utilizados para estilizar la aplicación y asegurar una presentación visual consistente y atractiva.
- **Aplicación Móvil:**
 - **React Native:** Permite el desarrollo de aplicaciones móviles multiplataforma con una única base de

código. Proporciona una experiencia de usuario nativa y acceso a funcionalidades específicas del dispositivo.

- **Backend:**

- **Django:** Framework de alto nivel que facilita el desarrollo rápido y limpio de aplicaciones web. Proporciona una estructura robusta y una gran cantidad de funcionalidades integradas.
- **APIs RESTful:** Utilizadas para la comunicación entre el frontend, la aplicación móvil y el backend. Permiten un intercambio de datos eficiente y seguro.

- **Base de Datos:**

- **PostgreSQL:** Base de datos relacional utilizada para almacenar datos estructurados de manera eficiente. Neon gestiona la base de datos proporcionando capacidades avanzadas de administración y escalabilidad.

Despliegue y Infraestructura

El despliegue de FitFuelBalance se ha realizado utilizando Render, que facilita la gestión y el despliegue continuo tanto del frontend como del backend. La infraestructura está diseñada para ser escalable y manejar un alto volumen de usuarios y datos.

- **Render:** Plataforma utilizada para el despliegue y alojamiento de la aplicación. Permite gestionar el frontend y backend de manera eficiente, asegurando una alta disponibilidad y rendimiento.
- **CI/CD:** Se han implementado pipelines de Integración Continua y Despliegue Continuo para automatizar las pruebas y el despliegue, garantizando que las nuevas versiones del software se lancen de manera rápida y segura.

En resumen, la arquitectura de FitFuelBalance está diseñada para ser modular, escalable y altamente eficiente, utilizando tecnologías modernas y probadas para asegurar un rendimiento óptimo y una experiencia de usuario superior.

3.4. Roles

3.5. Requisitos Funcionales y No Funcionales

3.6. Incrementos

ELEMENTOS INTERNOS

En $\text{\LaTeX}_{2\epsilon}$ se denomina entorno flotante a aquel en el que es el compilador el que decide el lugar más conveniente para situarlos. Por eso es importante que todos estos elementos tengan una etiqueta (*label*) y que en el texto sean referenciados. Nunca hay que utilizar las expresiones ‘La siguiente figura ...’ o ‘La siguiente tabla ...’ sino que hay que utilizar expresiones como ‘En la figura 1.3’ o ‘En la tabla 2.7’. Para ello es necesario utilizar el comando `\ref` como se indica en la sección ???. Muchos de los elementos vistos en este capítulo son entornos flotantes y es necesario tenerlo en cuenta a la hora de diseñar. Cuando corresponda se indicará si el elemento es flotante o no lo es.

4.1. Figuras

Con el fin de simplificar el uso de las figuras se ha definido un conjunto de funciones y modificado el entorno **figure**. Las figuras son elementos flotantes. Dentro de estos entornos se pueden incluir imágenes, gráficas o diagramas de Gantt. El entorno tiene tres parámetros. El primero es opcional y es el pie de figura para la lista de figuras. Al ser opcional debe ir entre corchetes y no es necesario indicarlo, se utilizará el pie de figura si no se indica este parámetro. El segundo parámetro es la etiqueta para posteriores referencias. El tercero es el pie de figura. En la figura ?? se puede ver su uso.



Figura 4.1: Figura de ejemplo creada a partir del código

Con este estilo, dentro de cada figura se pueden crear subfiguras con el comando `\subfigure` que tiene tres parámetros. El primero es opcional y por tanto, si existe, irá entre corchetes y es la etiqueta para ser referenciado y se puede dejar vacío, el segundo es el pie de subfigura que se añade a la numeración en letras latinas que ya crea automáticamente el comando (puede dejarse vacío). El tercero es el elemento que va a estar dentro de la subfigura. En la figura ?? puede verse un ejemplo y

se puede ver su uso en los fuentes de este documento.



Figura 4.2: Figura de ejemplo. El pie de figura debe ser suficientemente explicativo y con el tamaño que haga falta mientras que el de las subfiguras debe reducirse al mínimo y hacer referencia a las figuras en este pie de figura, como por ejemplo haciendo referencia a la figura ??

Para organizar las subfiguras se pueden utilizar saltos de párrafo, saltos de línea u, horizontalmente con los comandos `\quad`, `\qquad` o `\hspace*{espacio}`.

4.1.1. Gráficas

La creación de gráficas científicas es uno de ls elementos más complejos a la hora de crear un documento científico. Utilizando el paquete de $\text{\LaTeX 2}_{\epsilon}$ **pgfplots** es relativamente sencillo. Sin embargo, aún así este paquete es demasiado complejo. Para simplificar su uso se han creado un conjunto de entornos y funciones con el fin de facilitar esta tarea en la mayoría de los casos.

Otro paquete de gráficas disponible es el **gnuplottex** que permite diseñar gráficas utilizando gnuplot. En este caso también se aportan algunos entornos y funciones para simplificar su uso.

En cualquier caso el autor puede acudir a estos paquetes directamente si esta clase no le aporta la funcionalidad necesaria. Este es un elemento que permanece en desarrollo y el autor debería estar atento a los cambios en futuras versiones.

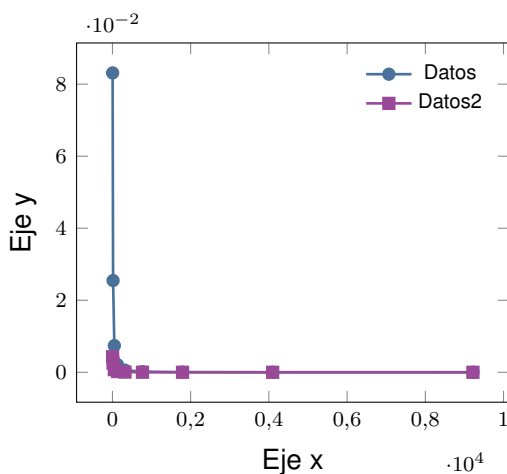
4.1.2. Gráficas con pgfplots

Gráficas xy

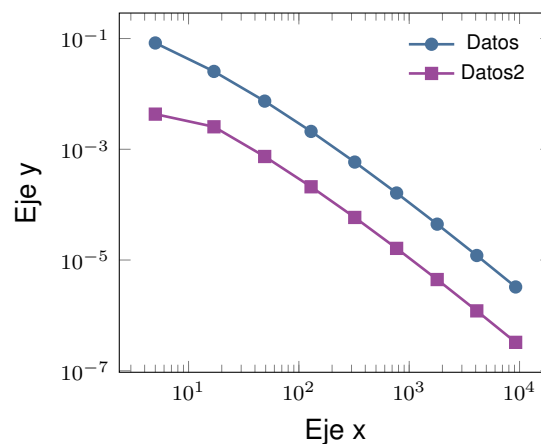
Tipo de gráfica

Para crear una gráfica xy lo primero que es necesario es definir los ejes y títulos. Para ello se pueden utilizar distintos entornos dependiendo del tipo de ejes a utilizar, sin embargo todos los entornos tienen los mismos cinco parámetros que son, por este orden, el título de la gráfica, el título del eje x, el título del eje y y el ancho y el alto de la gráfica.

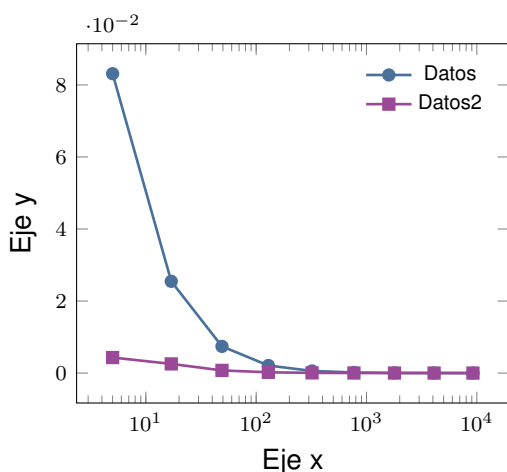
Los entornos disponibles son los siguientes: **xyplot**, **semilogxplot**, **semilogyplot** y **loglogplot**. En la figura ?? pueden verse los distintos esquemas.



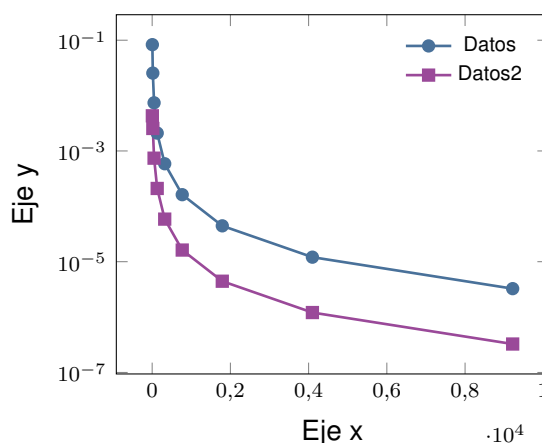
(a) xyplot



(b) loglogplot



(c) semilogxplot



(d) semilogyplot

Figura 4.3: En esta figura se pueden ver los resultados de aplicar los distintos entornos de gráficas xy.

Datos

Para presentar los datos se tienen tres funciones posibles, ambas con dos parámetros, el primero es el nombre completo del fichero con los datos y el segundo es la etiqueta que van a tener estos datos. Los ficheros de datos son ficheros con dos columnas de valores y la primera fila es siempre una fila de etiquetas. Aunque estas etiquetas no se usan, el paquete utilizado permite operaciones de representación muy complejas con los datos y con ficheros con más de dos columnas de datos, para simplificar los comandos se ha preferido hacer así. Si se desea algo más fino y complejo es necesario utilizar los comandos del paquete **pgfplots**.

Los comandos existentes son:

\plotdata{file}{label} Pone los puntos asociados a los datos. El tipo de marca y el color los selecciona la función internamente.

\plotline{file}{label} Une con una recta los puntos asociados a los datos. El color los selecciona la función internamente.

\plotline{file}{label} Es la combinación de los dos anteriores.

Expresiones matemáticas

Para representar expresiones matemáticas se dispone de la función **\plotfunction[n-samples] {expresion}{label}{xmin}{xmax}**. El primero es un parámetro opcional y por tanto debe ir entre corchetes si es que se pone; este parámetro es el número de muestreos de la función utilizados. Por otro lado 'label' es la etiqueta en la gráfica y 'xmin' y 'xmax' es el rango que se va a utilizar para la gráfica en x. Las funciones que se pueden utilizar en la expresión son un poco limitadas e incluyen -, *, /, abs, round, floor, mod, <, >, max, min, sin, cos, tan, deg (conversión de radianes a grados), rad (conversión de grados a radianes), atan, asin, acos, cot, sec, cosec, exp, ln, sqrt, ^ (potencia), ! (factorial), rand (aleatorio entre -1 y 1), rnd (aleatorio entre 0 y 1); sqrt, las constantes pi y e; las conversiones de formato numérico hex, Hex, oct, bin y algunas funciones más. Las funciones trigonométricas funcionan en grados.

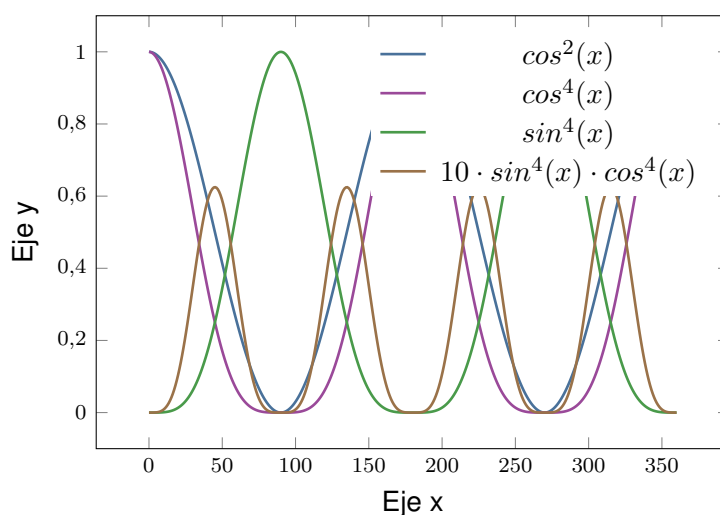


Figura 4.4: En esta figura se pueden ver los resultados de representar varias expresiones matemáticas.

4.1.3. Gráficas con gnuplot

Para realizar gráficas también se puede utilizar gnuplots, sin embargo tiene varias restricciones y es que la compilación debe hacerse en un ordenador y no se puede hacer en overleaf. Así mismo hay

que permitir que pdf_latex escape a la terminal con la opción de compilación **-shell-escape** o poner la opción de **shell** en la compilación con arara. Así mismo es necesario poner en la declaración de la clase la opción **gnuplots** al principio del documento. Si no se pone la opción, en el texto aparecerá un error en lugar de la gráfica. Para introducir código gnuplot se utiliza el entorno **gnuplot** que tiene un parámetro opcional que se recomienda que valga **terminal=epslatex**, sin embargo pueden utilizarse otros valores según se indica en el manual del paquete gnuplottex.

Un ejemplo del resultado puede verse en la ?? y, por supuesto, puede verse el código en los fuentes de este documento.

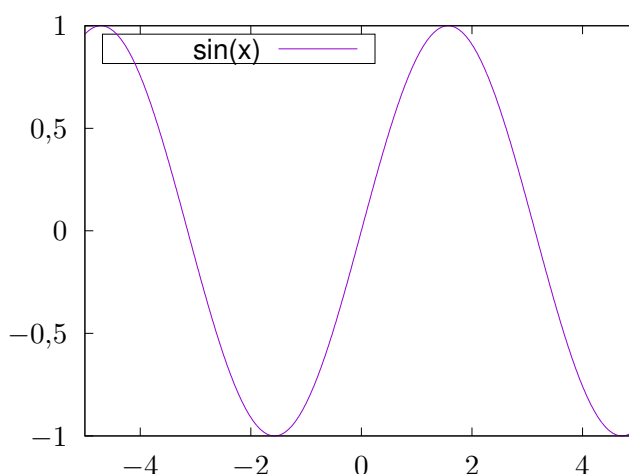


Figura 4.5: En esta figura se pueden ver los resultados de aplicar los distintos entornos de gráficas xy.

4.1.4. Imágenes

Como elemento previo es interesante recordar la resolución mínima necesaria de una imagen para ser impresa correctamente. Una resolución mínima implica 120 puntos por centímetro (o lo que es lo mismo 300 ppp). Es decir, si vamos a presentar una imagen con pixel cuadrado con un ancho de 12 cm en el papel deberá tener como mínimo $120 \times 12 = 1440$ puntos. Cualquier imágenes con una resolución inferior no se verá con la nitidez adecuada. Por otro lado, para una resolución de impresión con calidad fotográfica de 1200 ppp, o lo que es lo mismo 480 puntos por centímetro la resolución necesaria para el caso anterior sería $480 \times 12 = 5760$ puntos, sin embargo resoluciones tan altas si se tienen muchas imágenes se puede relentizar mucho la compilación del documento.

Para introducir una imagen se utiliza la función **\image** que tiene tres parámetros obligatorios de los cuales los dos primeros se pueden dejar vacíos. el primer parámetro es el ancho de la imagen; el segundo es la altura de la imagen y el tercero es el nombre de la imagen sin extensión. Se admiten múltiples formatos de ficheros entre los que se incluyen JPG, GIF y PNG. Si se deja vacío el ancho o el

alto, la dimensión no definida se calcula internamente para mantener la relación de aspecto original. Si se dejan los dos vacíos la imagen utiliza el 90 % del ancho del texto y la altura se calcula para mantener la relación de aspecto. Un del uso de este comando puede verse en el código de las figuras ?? y ??.

Además se dispone de un comando para introducir imágenes *in line* que van a ser rodeadas por el texto. Este comando es el `\imageIL` que tiene dos parámetros, el primero es el nombre de la imagen y el segundo es su anchura en cualquier unidad permitida por $\text{\LaTeX 2}_{\epsilon}$.

4.1.5. Diagramas de Gantt

Los diagramas de Gantt pueden llegar a ser muy complejos, sin embargo se ha preparado una versión simple que permita generar la mayor parte de los diagramas de Gantt que se puedan necesitar para trabajos para los que está diseñado este estilo.

En este caso para simplificar sólo se permiten diagramas de Gantt por semanas con un límite máximo de 26 semanas aunque si se necesitan más semanas basta con crear un segundo diagrama de Gantt o un tercero con la numeración de semanas necesaria.

En la figura ?? puede verse un ejemplo de uso cuyo fuente puede verse en los fuentes de este documento.

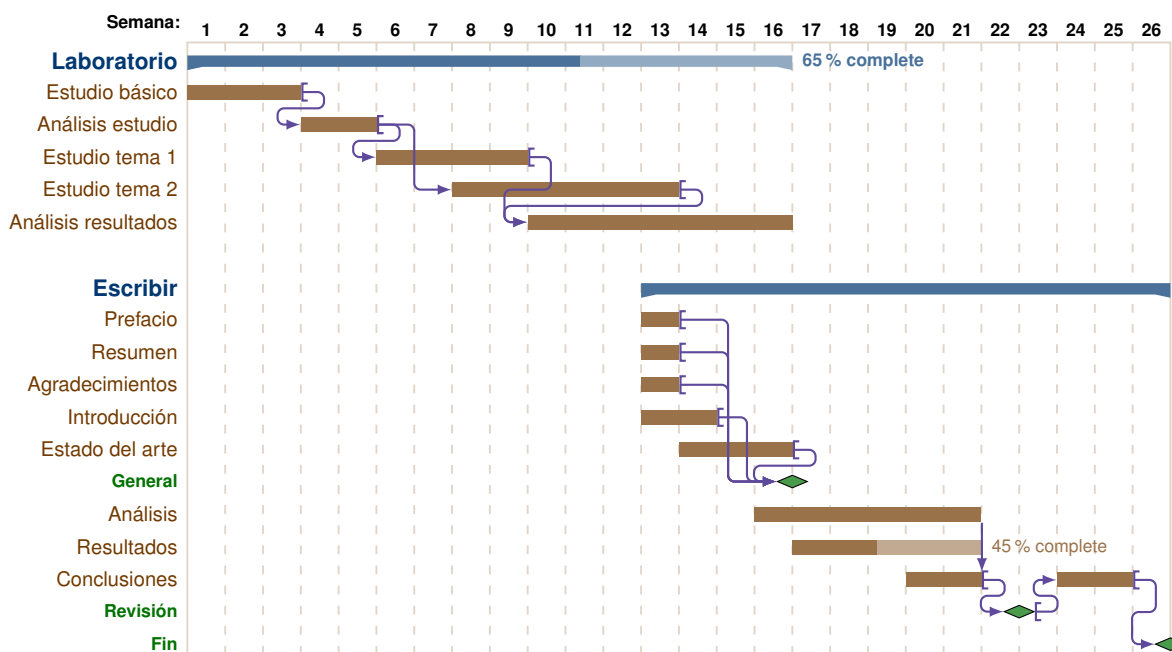


Figura 4.6: Diagrama de Gantt ejemplo en el que se muestran la mayoría de las posibilidades de un diagrama de Gantt con los comandos indicados en el texto.

En primer lugar, para definir un diagrama de Gantt se utiliza el entorno **gantt** con dos parámetros,

el primero es la numeración de la primera semana y el segundo el de la última semana a representar.

Dentro de este entorno se pueden usar varios comandos. Para definir grupos de tareas se dispone del comando `\taskgroup[name][progress]{grouptitle}{start}{end}`. Se puede no indicar ninguno de los parámetros opcionales; si se indica uno de ellos (entre corchetes) siempre se corresponde con el primero de los parámetros opcionales y si se quiere indicar sólo el segundo de los parámetros opcionales entonces deberá aparecer también el primero aunque esté vacío. El primer parámetro opcional es una etiqueta para los enlaces descritos más adelante y el segundo es un porcentaje de progreso que es un valor entre 0 y 100 (sin ningún símbolo más, sólo el número). Los parámetros obligatorios son el título del grupo, su semana de inicio y su semana de finalización respectivamente.

Para definir barras de tareas normales se utiliza el comando `\taskbar` que tiene los mismos parámetros y en el mismo orden que `\taskgroup`.

Para definir hitos se utiliza la función `\milestone[name]{milestonetitle}{week}` que pone un hito en la semana indicada y con el título indicado. En nombre es opcional y vale para los enlaces.

Se dispone también de tres funciones para crear los enlaces, todas con dos parámetros obligatorios y son el inicio y el fin del enlace. Las funciones son las siguientes:

FtoSlink Enlaza el final de la preimera barra con el principio de la segunda barra.

FtoFlink Enlaza el final de las dos barras.

StoSlink Enlaza el inicio de las dos barras.

En todos los casos de barras o hitos debe colocarse un `\\` al final de cada línea si se desea que la siguiente barra vaya en una nueva línea. Eso se debe a que a veces querremos más de una barra en la misma línea en cuyo caso se pondrán una a continuación de la otra sin `\\` y la segunda y posteriores sin título.

4.2. Tablas

Para realizar una tabla se puede usar cualquiera de los entornos de \LaTeX diseñados a tal efecto como pueden ser **tabular**, **tabbing**, **longtable**, etc. Sin embargo estas tablas tienen que ser elementos flotantes con pie de tabla, referenciables y listables. Para ello se dispone del entorno **table**[short]{label}{caption}. El primer parámetro es opcional y por tanto, si aparece, debe ir entre corchetes y es texto corto que aparecerá en la lista de cuadros de texto; el segundo parámetro es una etiqueta para ser referenciada y no es opcional aunque puede dejarse en blanco; y el tercero es el texto que aparecerá como pie de tabla. Ejemplo de un cuadro de tabla ??.

La estética interna de las tablas es responsabilidad del autor pero se recomiendan diseños minimalistas y si se usan colores es aconsejable utilizar los colores complementarios definidos para la gama

no	puedo	decir	nada
1.23	2.32	1.15	3.5
10.2	2.2	4.5	5.7
8.3	1.56	2.78	8.91

Tabla 4.1: Esta es una tabla de ejemplo en la que, internamente, se usa el entorno **tabular**.

de colores correspondiente.

Así mismo, muchas veces es necesario introducir varias tablas juntas con sus correspondientes subpies de tabla. Para ello se puede utilizar el comando **\subtable** dentro del entorno **table** en el que el primer parámetro es la etiqueta para ser referenciado, el segundo el pie de la subtabla y el tercero es el elemento que se quiere presentar en la subtabla. Un ejemplo con subtablas puede verse en la tabla ??.

no	puedo	decir	nada	no	puedo	decir	nada
1.23	2.32	1.15	3.5	11.39	1.21	5.15	2.9
10.2	2.2	4.5	5.7	5.2	4.8	9.43	1.7
8.3	1.56	2.78	8.91	7.3	6.35	0.11	3.13

(a) Esta es una subtabla, y su pie no debe ser demasiado largo.

(b) Y este es otro ejemplo aunque con demasiado texto ambos.

Tabla 4.2: Esta es una tabla de ejemplo en la que se definen subtablas. Los pies de las subtablas no deben ser excesivos y se debe cargar toda la explicación posible en el pie de la tabla.

4.2.1. Presupuestos

Se han diseñado una serie de comandos sencillos para la creación de presupuestos. Si se necesita un diseño de presupuestos más complejo puede ser creado por el autor manteniendo una estética similar a la de los presupuestos creados con estos comandos. Normalmente esta tabla no será indexada como una tabla del documento aunque si se considera conveniente puede ser introducida dentro del entorno **table** explicado en la sección ??.

Los comandos para diseñar un presupuesto deben utilizarse dentro del entorno diseñado para el presupuesto que es el entorno **budget**. En ningún caso es necesario indicar la unidad monetaria ya que ya la añade internamente. Se dispone de los siguientes comandos:

\budgettitle{título} cada una de las secciones en las que está dividido un presupuesto tiene su título y debe indicarse con este comando. El final de cada sección deberá acabar con un comando **subtotal**.

\concept{nombre}{unitario}{cantidad}{total} permite definir cada uno de los conceptos con

el texto indicado en el primer parámetro, el precio unitario en el segundo y el total en el tercero. Pueden dejarse elementos vacíos si se desea y es necesario dejar los tres últimos vacíos si el concepto está dividido en subconceptos

\subconcept{nombre}{unitario}{cantidad}{total} funciona exactamente igual que el comando **\concepto** pero se expresa de forma distinta en el presupuesto.

\subtotal{valor} valor del subtotal.

\separator añade una separación por motivos estéticos.

\total{valor}] valor total del presupuesto.

Un ejemplo de cómo puede quedar un presupuesto es el siguiente y cuyo código puede verse en los fuentes de este documento:

Concepto	Coste Unitario	Cantidad	Cantidad Total
Materiales			
Hierro	12.00 €	1	12.00 €
Derivados del cobre			
Latón	12.00 €	2	24.00 €
Bronce	12.15 €	3	36.45 €
Estaño	10.11 €	2	20.22 €
SUBTOTAL			82.67 €
Personal			
Ingenieros			
Informático	45,000.00 €	3	135,000.00 €
Telecomunicaciones	45,000.00 €	3	135,000.00 €
Técnicos			
Telecomunicaciones	28,000.00 €	3	84,000.00 €
Informática	28,000.00 €	3	84,000.00 €
Administrador	50,000.00 €	1	50,000.00 €
SUBTOTAL			488,000.00 €
TOTAL			488,082.67 €

4.3. Cuadros de texto

Para poner un texto dentro de un cuadro de texto se dispone del entorno **textbox[short] {label}{caption}**. El primer parámetro es opcional y por tanto, si aparece, debe ir entre corchetes siendo texto corto que aparecerá en la lista de cuadros de texto; el segundo parámetro es una etiqueta para ser referenciado y no es opcional aunque puede dejarse en blanco; y el tercero es el texto que aparecerá bajo el cuadro. Los cuadros de texto son elementos flotantes. Ejemplo de un cuadro de texto es el que puede verse en el cuadro ??.

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

Cuadro 4.1: Este es un cuadro de texto en el que usando el paquete lipsum se genera el texto internamente

4.4. Ecuaciones

El uso de ecuaciones se basa en el paquete **amsmath** y sólo cambia en los entornos **equation** y **equation***. Estos entornos tienen dos parámetros, el primero es opcional y por tanto, si aparece debe ir entre corchetes. Este parámetro es la etiqueta que después servirá para referenciar la ecuación con el comando **\ref**. El segundo parámetro es un texto que es el texto que aparecerá en la lista de ecuaciones si se selecciona su aparición. Como su aparición no es opcional debe aparecer siempre entre llaves aunque si está vacío no introduce ninguna entrada en la lista de ecuaciones.

Un ejemplo de la ecuación resultante puede verse en la ecuación ??. En la que puede verse que la ecuación aparece recuadrada. Si se quiere que aparezca recuadrada es necesario usar el comando **\boxed**.

$$\sum c_{ij} = \frac{a}{\int adx} \quad (4.1)$$

También pueden definirse múltiples ecuaciones con subnumeración si es necesario utilizando en entorno **multiequations** que no tiene parámetros y en cuyo interior se utiliza el entorno **equation** antes descrito. Las ecuaciones ?? y ?? son un ejemplo de ello.

$$\sum c_{ij} = \frac{a}{\int adx} \quad (4.2a)$$

$$\sum c_{ij} = \frac{a}{\int adx} \quad (4.2b)$$

El código del ejemplo puede verse en los fuentes de este documento.

4.5. Código

Hay múltiples funciones para presentar el código. La primera de ellas es genérica y es **\Code** que tiene 8 parámetros, el primero de ellos opcional. El resto de las funciones que se listarán más adelante tienen los primeros 7 parámetros idénticos a esta función y no tienen un octavo parámetro. Los parámetros significativos son los siguientes:

- 1.— Parámetro opcional (entre corchetes si aparece) y es la etiqueta para referenciar el código en otra parte del texto.
- 2.— Es la descripción corta que aparecerá en el listado de códigos. Si se deja vacía se utilizará la descripción larga.
- 3.— Es la descripción larga del código.
- 4.— Nombre (incluyendo el camino relativo) del fichero con el código a presentar.
- 5.— Primera línea del código que se presentará en el texto.
- 6.— Última línea del código que se presentará en el texto.
- 7.— Número que aparecerá como primer número de línea del código presentado. Puede tomar el valor que desee el autor pero si se deja vacío el código no se numerará. Habitualmente estará vacío o valdrá 1 o lo mismo que el quinto parámetro.
- 8.— El lenguaje tal y como se indica en el paquete listings.

Para el autor no tenga que leer el paquete listings se han creado una serie de funciones sin el octavo parámetro y que ya indican el lenguaje correspondiente y son de uso más cómodo. Estas funciones son:

\AdaCode[etiqueta]{short cap}{long caption}{file}{init num}{end num}{first num} Código ADA

\ASMCode[etiqueta]{short cap}{long caption}{file}{init num}{end num}{first num} Código ensamblador de Intel.

\ASMMotCode[etiqueta]{short cap}{long caption}{file}{init num}{end num}{first num} Código

ensamblador de Motorola.

`\CCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código en C.

`\CPPCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código C++.

`\CSharpCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código C#.

`\GnuplotCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código gnuplot.

`\HaskellCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Haskell.

`\HTMLCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código HTML.

`\JavaCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Java.

`\LaTeXCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código $\text{\LaTeX} 2_{\epsilon}$.

`\LispCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Lisp.

`\MakeCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Ficheros makefile.

`\MathematicaCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Mathematica.

`\MatlabCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Matlab.

`\OctaveCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Octave.

`\PascalCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Pascal.

`\PerlCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Perl.

`\PHPCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código PHP.

`\PythonCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Python.

`\RCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código R.

`\RubyCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Ruby.

`\ScilabCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código Scilab.

`\SQLCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código SQL.

`\VHDLCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código VHDL.

`\XMLCode[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código XML.

Es importante tener en cuenta que si el código es muy largo será el autor quien deberá partirlo manualmente si no cabe en la misma página y se recomienda que esto se haga en la edición final del documento. Un ejemplo de su uso se puede ver en el ?? y como en todos los casos se puede acudir a los fuentes de este manual para ver su uso en $\text{\LaTeX 2}_{\epsilon}$. Se puede ver otro ejemplo en ??

Código 4.1: En esta figura se presenta el código correspondiente a la conexión de los usuarios. Realmente este texto es por poner algo a modo de ejemplo en C.

```

1 long IRCMsg_ErrAlreadyRegistred(char **command, char *prefix, char *nick)
2 {
3     *command=0;
4     if(nick == NULL) return IRCERR_NONICK;
5     __CPM__(command,prefix,COM_ERR_ALREADYREGISTRED,nick,"Unauthorized_command_
        (already_registered)");
6     return IRC_OK;
7 }

```

Código 4.2: En esta figura se presenta el código correspondiente a la conexión TCP. Realmente este texto es por poner algo a modo de ejemplo en Python.

```

1 class TCPClient:
2     def __init__(self, server, port, function):
3         self.clientsock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4         self.clientsock.connect((server, port))
5         function(self,self.clientsock)
6
7     def receive(self, socks):
8         msg = socks.recv(2048)
9         return msg
10
11     def send(self, msg, socks):
12         socks.sendall(msg)

```

4.6. Algoritmos

Describir algoritmos suele ser complejo y para ello se dispone del estilo (**algorithm2e**). Sin embargo, para simplificar un poco su uso e incluirlo en los listados correspondientes y añadir un pie de algoritmo se han creado dos entornos **algorithm** y **algorithmN**, ambos tienen tres parámetros y sólo se diferencian en que uno numera las líneas del algoritmo (algorithmN) y el otro no. De los tres parámetros el primero es opcional y por tanto si aparece debe hacerlo entre corchetes, el segundo puede dejarse vacío y es la explicación corta que es la que aparecerá en el listado de algoritmos y la tercera

es el pie de algorithmo que se utilizará también para el listado de algoritmos si se deja vacío el segundo parámetro. Como siempre puede verse un ejemplo en el

```

input : A bitmap  $Im$  of size  $w \times l$ 
output: A partition of the bitmap

1  special treatment of the first line;
2  for  $i \leftarrow 2$  to  $l$  do
3      special treatment of the first element of line  $i$ ;
4      for  $j \leftarrow 2$  to  $w$  do
5           $left \leftarrow \text{FindCompress}(Im[i, j-1])$ ;
6           $up \leftarrow \text{FindCompress}(Im[i-1, j])$ ;
7           $this \leftarrow \text{FindCompress}(Im[i, j])$ ;
8          if left compatible with this then
9              if  $left < this$  then  $\text{Union}(left, this)$ ;
10             ;
11             else  $\text{Union}(this, left)$ ;
12         end
13         if up compatible with this then
14             if  $up < this$  then  $\text{Union}(up, this)$ ;
15             ;
16             else  $\text{Union}(this, up)$ ;
17         end
18     end
19     foreach element  $e$  of the line  $i$  do  $\text{FindCompress}(p)$ ;
20 end

```

Algoritmo 4.1: Ejemplo de algoritmo. Para ver cómo se usa deben editarse los fuentes de este documento. Además, debido a la complejidad de los comandos internos no se ha hecho ninguna simplificación del estilo y por tanto es aconsejable leerse la documentación del paquete algorithm2e.

4.7. Listas

En este estilo se han modificado los entornos tradicionales para crear listas (itemize, enumerate o description) cambiando el espaciado, el tamaño de letra o los márgenes derecho e izquierdo. Sin embargo su uso es el tradicional.

4.7.1. Listas tradicionales de $\text{\LaTeX} 2_{\epsilon}$

El aspecto con el que quedan las distintas listas tradicionales es el que se muestra a continuación.

Punteado con anidación

- Un ejemplo anidado.
 - Este es el primer punto anidado.

- Y este el segundo.
- Se sale de la anidación.

Enumerado con anidación

- 1.– Un ejemplo anidado.
 - 1.1.– Este es el primer punto anidado.
 - 1.2.– Y este el segundo.
- 2.– Se sale de la anidación.

Definiciones con anidación

Primer elemento: Un ejemplo anidado.

Primer elemento interno: Este es el primer punto anidado.

Segundo interno: Y este el segundo.

Segundo: Se sale de la anidación.

4.7.2. Nuevos tipos de listas

Dadas las necesidades específicas que pueden tener estos documentos se han desarrollado una serie de tipos de listas específicos con el fin de cubrir esas necesidades.

Se disponen de cinco entornos específicos que son **simplelist**, **functional**, **nonfunctional**, **functionality** y **objective**. Es aspecto de este tipo de listas puede verse a continuación.

Simplelist con anidación

Los elementos no llevan ningún tipo de marca ni están destacados de ninguna forma, sólo cambia la indentación de los elementos.

- Un ejemplo anidado.
 - Este es el primer punto anidado.
 - Y este el segundo.
- Se sale de la anidación.

Functional con anidación

Se utiliza para definir requisitos funcionales.

- RF-1.–** Un ejemplo anidado.
 - RF-1.1.–** Este es el primer punto anidado.
 - RF-1.2.–** Y este el segundo.

RF-2.— Se sale de la anidación.

Nonfunctional con anidación

Se utiliza para definir requisitos no funcionales.

RNF-1.— Un ejemplo anidado.

RNF-1.1.— Este es el primer punto anidado.

RNF-1.2.— Y este el segundo.

RNF-2.— Se sale de la anidación.

Functionality con anidación

Se utiliza para definir funcionalidades.

F-1.— Un ejemplo anidado.

F-1.1.— Este es el primer punto anidado.

F-1.2.— Y este el segundo.

F-2.— Se sale de la anidación.

Objective con anidación

Se utiliza para definir objetivos.

O-1.— Un ejemplo anidado.

O-1.1.— Este es el primer punto anidado.

O-1.2.— Y este el segundo.

O-2.— Se sale de la anidación.

4.8. Referencias internas e hiperenlaces

Todos los enlaces internos se realizan utilizando el comando de $\text{\LaTeX 2}_{\epsilon}$ `\ref{label}`, donde label es la etiqueta del elemento referenciado.

Para los hiperenlaces se utiliza el comando `\href` que tiene dos parámetros, el primero es la URI del objeto correspondiente y el segundo es el texto que aparecerá. Por ejemplo, para contactar conmigo basta con pinchar [aquí](#). Es interesante comentar que los hiperenlaces aparecen con distinto color que el texto en pantalla pero aparecen del mismo color del texto cuando se imprimen.

COMPILACIÓN

Para compilar el documento se puede hacer de múltiples formas y aquí se van a contar dos posibles formas. Una compilación manual basada en `pdflatex` y otra basada en `arara`.

5.1. `Pdflatex`

Cuando se compila manualmente es importante saber qué hacer, en qué orden y por qué. El primer paso es compilar el fuente principal con `pdflatex` usando como parámetro el nombre del fuente principal (no es necesario poner la extensión `.tex`). Esta compilación permite guardar el en fichero del mismo nombre pero con extensión `.aux` información acerca de elementos sin resolver. Si no hay elementos especiales, una segunda compilación resuelve todos los elementos que habían quedado sin resolver gracias a la información almacenada en el `.aux`. Sin embargo, si introducimos bibliografía o acrónimos, definiciones o glosario no se realizará esa segunda compilación.

Si se usa bibliografía es necesario ejecutar el comando `bibtex` con el nombre del fuente principal por parámetro (ahora siempre sin extensión) que recurrirá a la información de varios ficheros auxiliares par construir la bibliografía.

De igual forma, si se usan acrónimos o definiciones es necesario ejecutar el comando `glossaries` con el nombre del fuente principal sin extensión que, al igual que en el caso de la bibliografía utilizará múltiples ficheros auxiliares para crear el capítulo de acrónimos o de definiciones.

Para finalizar, si se quiere crear el glosario es necesario ejecutar `makeindex` con el nombre del principal como parámetro y como estilo el `.ist` creado al compilar. El fichero con el estilo será un segundo parámetro precedido por la bandera `-s`.

Una vez realizados estos pasos se realizará una doble compilación con `pdflatex` como en el primer paso de este proceso.

Es importante acudir al `.log` para ver el proceso de compilación, errores y *warnings* con el fin de resolver los problemas que sea necesario resolver. \LaTeX 2_ε es extremadamente preciso en la creación de textos y muchos *warnings* no es necesario resolverlos pero sí hay que revisarlos todos.

5.2. Arara

Otra forma de compilar es usando el programa arara con el nombre del principal como paámetro. En el programa principal, cada línea de comentario que contenga tras el símbolo de comentario la palabra arara será el indicador de una acción de compilación. En el fuente principal de este documento puede verse un ejemplo que además se copia en el ??.

Al igual que en la compilación manual, comprobar los ficheros .log es importante.

```
% arara: clean: {files: [tfgtfmthesisuam.aux, tfgtfmthesisuam.idx, tfgtfmthesi-
suam.ilg, tfgtfmthesisuam.ind, tfgtfmthesisuam.bbl, tfgtfmthesisuam.bcf, tfgtfmthe-
sisuam.blg, tfgtfmthesisuam.run.xml, tfgtfmthesisuam.fdb_latexmk, tfgtfmthesi-
suam.fls, tfgtfmthesisuam.loe, tfgtfmthesisuam.lof, tfgtfmthesisuam.lol, tfgtfmthesi-
suam.lot, tfgtfmthesisuam.ltb, tfgtfmthesisuam.out, tfgtfmthesisuam.toc, tfgtfmthe-
sisuam.upa, tfgtfmthesisuam.upb, tfgtfmthesisuam.acn, tfgtfmthesisuam.acr,
tfgtfmthesisuam.alg, tfgtfmthesisuam.glg, tfgtfmthesisuam.glo, tfgtfmthesisuam.gls,
tfgtfmthesisuam.glsdefs, tfgtfmthesisuam.idx, tfgtfmthesisuam.ilg, tfgtfmthe-
sisuam.xdy, tfgtfmthesisuam.loa, tfgtfmthesisuam.gnuploterrors , tfgtfmthesi-
suam.mw]}
% arara: pdflatex: {shell: yes}
% arara: makeglossaries
% arara: makeindex: {style: tfgtfmthesisuam.ist }
% arara: bibtex
% arara: pdflatex: {shell: yes}
% arara: pdflatex: {shell: yes}
% arara: clean: {files: [tfgtfmthesisuam.aux, tfgtfmthesisuam.idx, tfgtfmthesi-
suam.ilg, tfgtfmthesisuam.ind, tfgtfmthesisuam.bbl, tfgtfmthesisuam.bcf, tfgtfmthe-
sisuam.blg, tfgtfmthesisuam.run.xml, tfgtfmthesisuam.fdb_latexmk, tfgtfmthesi-
suam.fls, tfgtfmthesisuam.loe, tfgtfmthesisuam.lof, tfgtfmthesisuam.lol, tfgtfmthesi-
suam.lot, tfgtfmthesisuam.ltb, tfgtfmthesisuam.out, tfgtfmthesisuam.toc, tfgtfmthe-
sisuam.upa, tfgtfmthesisuam.upb, tfgtfmthesisuam.acn, tfgtfmthesisuam.acr,
tfgtfmthesisuam.alg, tfgtfmthesisuam.glg, tfgtfmthesisuam.glo, tfgtfmthesisuam.gls,
tfgtfmthesisuam.glsdefs, tfgtfmthesisuam.idx, tfgtfmthesisuam.ilg, tfgtfmthe-
sisuam.xdy, tfgtfmthesisuam.loa, tfgtfmthesisuam.gnuploterrors , tfgtfmthesi-
suam.mw]}
```

Cuadro 5.1: Ejemplo de compilación con arara. Estos comentarios deben introducirse al principio del documento principal y permiten que arara realice las tareas indicadas en el orden que se indica.

5.3. Overleaf

En overleaf no se puede controlar la compilación y por tanto no se puede utilizar xindy ni gnuplot. En este sentido es necesario indicar la opción overleaf y no utilizar la opción gnuplot. Si se utiliza Mendeley o Zotero puede importarse la bibliografía en un solo click si se enlazan las cuentas en la configuración de Overleaf. En el menú del proyecto de Overleaf se selecciona “bibliografía” y se puede importar directamente el .bib de cualquiera de estos sistemas.

5.4. Permitir el uso de la shell al compilador

Si se utiliza gnuplot para crear gráficas es necesario permitir que el compilador pueda ejecutar comandos en una shell, para ello, en el caso de pdflatex es necesario usar la bandera -shell-escape y en el caso de arara utilizar la opción de shell como puede verse en el ??.

BIBLIOGRAFÍA

APÉNDICES

WORD[®] vs. L^AT_EX 2_ε

A.1. Ventajas e inconvenientes de L^AT_EX 2_ε

El gusto por el L^AT_EX depende de la forma de trabajar de cada uno. La principal virtud es la facilidad de formatear cualquier texto y la robustez. Incluir referencias a capítulos, secciones, figuras, tablas, etc. es inmediato. Las ecuaciones quedan estupendamente, la escritura se puede realizar modularizada y estructurada. Con estilos como el presentado se reduce considerablemente la utilización de paquetes complejos reduciendo su uso a comandos simples aunque eliminatos

El principal inconveniente de L^AT_EX 2_ε radica en la necesidad de aprender un conjunto de comandos para generar los elementos que queremos. Cuando se está acostumbrado a un entorno “lo que veo es lo que obtengo” (WYSIWYG) es difícil cambiar la mentalidad a un entorno del tipo “lo que pienso es lo que obtengo” (WYTIWYG) como L^AT_EX 2_ε.

Por otro lado, en general será muy complicado cambiar el formato para desviarnos de la idea original de sus creadores del estilo. No es imposible, pero sí muy difícil. En muchos casos, como en el tipo de documentos a los que está dirigido este estilo de L^AT_EX 2_ε es una ventaja y no un inconveniente en el caso de querer obtener una imagen corporativa en los documentos.

A.2. Ventajas e inconvenientes de Word[®]

La ventaja mayor del Word[®] es que permite configurar el formato muy fácilmente. Para las ecuaciones tradicionalmente ha proporcionado pésima presentación. Sin embargo, el software adicional Mathtype[®] solventa este problema, incluyendo una apariencia muy profesional y cuidada. Incluso permite utilizar un estilo similar al de L^AT_EX 2_ε. Además, aunque el Word[®] incluye sus propios atajos para escribir ecuaciones, Mathtype[®] admite también la escritura de ecuaciones utilizando los mismos comandos que L^AT_EX 2_ε.

Trabajar con títulos, referencias cruzadas e índices es un engorro, por no decir nada sobre la creación de una tabla de contenidos. Resulta muy frecuente que alguna referencia quede pérdida o huér-

fana y aparezca un mensaje en negrita indicando que no se encuentra. Algunos autores hacen todas estas referencias manualmente lo significa que cualquier cambio supone un arduo trabajo rehaciendo las referencias de todo el documento. Los estilos permiten trabajar bien definiendo la apariencia, pero también puede desembocar en un descontrolado incremento de los mismos. Además, es muy probable que Word® se quede colgado, sobre todo al trabajar con copiar y pegar de otros textos y cuando se utilizan ficheros de gran extensión, como es el caso de un libro.

A.3. ¿Cuál elijo?

En cualquier caso las tipografías, colores, distancias entre párrafos, interlineados, encabezamientos y estructura del documento debe coincidir con el aquí presentado. Dado que sólo se aporta el estilo de L^AT_EX 2_ε se recomienda su uso aunque no es obligatorio y es decisión del autor elegir.

INSTALACIÓN

B.1. Linux

Este paquete en Linux puede instalarse de tres formas diferentes. La primera de ellas es a través de los sistemas de paquetes usados habitualmente en Linux, para ello basta con configurar el repositorio con los siguientes comandos: **sudo deb http://metis.ii.uam.es asignaturas main**. Una vez configurado es necesario validarlo obteniendo la clave pública del repositorio. Para ello basta con acceder a uno de los servidores del anillo de claves de ubuntu para obtenerla. La forma de hacerlo es ejecutando como superusuario el siguiente comando: **sudo apt-key adv --keyserver keyserver.ubuntu.com --recv C95B8FCEC5A57017**. Para finalizar basta con ejecutar: **sudo apt-get update && sudo apt-get install tfgtfmthesisuam**. Una vez realizados todos estos comandos el paquete estará instalado y funcional y con acceso a las actualizaciones que se realicen sobre el paquete. En este caso la documentación estará en el directorio `/usr/share/doc/tfgtfmthesisuam`.

La segunda forma es descargar el fichero `tfgtfmthesisuam.deb` de la página moodle correspondiente e instalarlo con el comando **sudo dpkg -i tfgtfmthesisuam.deb** estando en el mismo directorio en el que se ha descargado el fichero. Sin embargo esta opción no es tan sencilla dado que si no se tienen los paquetes de los que depende este paquete producirá un error hasta que estén todos instalados. Además, si se instala de esta forma no está disponible el acceso a actualizaciones de forma automática. En este caso la documentación estará en el directorio `/usr/share/doc/tfgtfmthesisuam`.

La tercera posibilidad consiste en descargar el fichero `.tgz` o el `.zip` del estilo y descomprimirlo en cualquier directorio. Si se ejecuta **sudo installLinux** el estilo será instalado en el sistema y estará disponible desde cualquier directorio y usuario y puede eliminarse del directorio donde ha sido descomprimido. Si no se tiene acceso como superusuario se puede no ejecutar el comando pero entonces es necesario que el estilo y el documento se encuentren en el mismo directorio y que además sea el mismo directorio desde el que se compila. Si se utiliza este método no se tiene acceso a las actualizaciones automáticas. En este caso la documentación estará en el directorio `/usr/share/doc/tfgtfmthesisuam`.

B.2. Windows™

Es necesario instalar MiKTeX completo e instalar manualmente algunos paquetes así como este estilo.

Las indicaciones que se presentan a continuación no han sido probadas y sólo son indicaciones que teóricamente deberían funcionar pero si no lo hacen, el creador de este estilo agradecería que se comunicase qué instrucción no funciona.

- 1.– Crear el directorio `c:\localtexmf` como administrador de Windows.
- 2.– Descomprimir el estilo zip en ese directorio.
- 3.– Activar dicho directorio como directorio de estilos de latex para ello es necesario utilizar una de las siguientes dos opciones:
 - 3.1.– Usando el GUI de MiKTeX:
 - 3.1.1.– En el menú Inicio ve a la entrada MiKTeX y abre la configuración “Configuración (Administrador)”, por supuesto. Se abrirá la ventana “Opciones MiKTeX”.
 - 3.1.2.– Ve a la pestaña “Raíces”. Haz clic en “Añadir” y elije `c:\localtexmf`.
 - 3.1.3.– Ahora la parte más importante: ve a la pestaña “General” y haz clic en “Actualizar FNDB” (FNDB = File Name Data Base). En algunos casos, especialmente si hay nuevas fuentes instaladas, hay que pulsar también el botón “Actualizar Formatos”.
 - 3.2.– En la línea de comandos (siempre añadiendo `--admin` para actuar como administrador y opcionalmente `--verbose`):
 - 3.2.1.– Ejecuta `initexmf --register-root=c:\localtexmf`
 - 3.2.2.– Ejecuta `initexmf --update-fndb`

Dado que no dispongo de ningún ordenador en este sistema operativo este apartado se actualizará adecuadamente en el momento en el que algún estudiante me comunique cómo lo ha realizado o me solicite ayuda para instalarlo.

B.3. Mac OS X

En el caso de querer instalar el estilo en este sistema es necesario instalar MacTex. Para ello se puede ir a la página oficial de MacTex pinchando <http://tug.org/mactex> y seguir las instrucciones correspondientes con todas las actualizaciones necesarias. Dependiendo de la versión de MacTex este se instala en el directorio `/usr/local/texlive/YYYY` donde YYYY es el año de la versión de MacTex que se esté instalando y cuyo valor es necesario saber para instalar correctamente el estilo.

En Mac OS X es necesario descargar el fichero `.tgz` o el `.zip` del estilo y descomprimirlo en cualquier directorio. Si se ejecuta **`sudo installMaC YYYY`** el estilo será instalado en el sistema y estará disponible desde cualquier directorio y usuario y puede eliminarse del directorio donde ha sido descomprimido. Si

no se tiene acceso como superusuario se puede no ejecutar el comando pero entonces es necesario que el estilo y el documento se encuentren en el mismo directorio y que además sea el mismo directorio desde el que se compila. Cualquier actualización debe realizarse manualmente realizando el mismo procedimiento.

B.4. Overleaf o ShareLatex

Para utilizar este estilo en alguno de estas aplicaciones web es necesario bajarse el archivo .tgz o .zip, descomprimirlo y subir los ficheros tfgtfmthesisuam.cls, tfgtfmthesisuam.ist, y todas las imágenes. Pueden borrarse todos los logos e imágenes que no se correspondan con la escuela o facultad correspondiente. En general estos sistemas en su versión gratuita tienen limitado el número de archivos que se pueden subir por cada proyecto y por tanto es necesario no desperdiciar espacio con ficheros innecesarios, sobre todo si se va a estructurar mucho el documento o se utilizan muchas imágenes o fuentes de código. Si se dispone de una cuenta de pago en alguno de estos sistemas entonces hay muchas menos limitaciones y se pueden copiar todos los ficheros.

B.5. ¿Dónde está el manual?

Dónde se encuentre el manual depende mucho del sistema operativo y el tipo de instalación realizada por ello se recomienda que se busque el fichero tfgtfmthesis.pdf. En ese mismo directorio se encontrarán los fuentes del manual a los que se hace referencia a lo largo de todo este documento.

B.6. Corrección ortográfica y codificación de caracteres

La corrección ortográfica depende exclusivamente del editor que se esté utilizando y por tanto es necesario acudir a la documentación del editor que se esté utilizando para configurarla correctamente.

Por otro lado todo el estilo se ha creado utilizando la codificación UTF8 y por tanto la codificación de los fuentes debe estar también en UTF8. Debe seleccionarse dicha codificación en el editor que se esté utilizando.

B.7. ¿Qué editor utilizo?

En todos los sistemas operativos hay múltiples editores de \LaTeX 2_ε e incluso algunos entornos de desarrollo integrados como eclipse o netbins así como editores como vi, emacs, sublime o atom tienen

plugins o paquetes que pueden ser instalados para que reconozcan la sintaxis de $\text{\LaTeX} 2_{\epsilon}$ y pueden compilar los documentos. La elección depende de cada uno y depende de los gustos, habilidades y conocimientos de cada uno. Lo recomendable es probar con varios hasta encontrar el adecuado.

PACKETES INCLUIDOS

Este estilo utiliza múltiples paquetes que se indican a continuación con enlaces a sus manuales en la WEB. Estos paquetes se utilizan con ciertos parámetros que se indican a continuación de cada elemento del listado. Para acceder a la documentación pincha en documentación en cada paquete.

algorithm2e (Documentación).

alltt (Documentación).

amsmath (Documentación).

babel (Documentación).

calc (Documentación).

caption small,bf,margin={5em,5em}. (Documentación).

cite (Documentación).

cleveref (Documentación).

etex (Documentación).

etoolbox (Documentación).

eurosym (Documentación).

fancyhdr (Documentación).

fancybox (Documentación).

filecontents (Documentación).

float (Documentación).

gnuplottex shell,cleanup,subfolder. (Documentación).

glossaries acronyms,nonumberlist,shortcuts,toc. (Documentación).

graphicx (Documentación).

hyperref (Documentación).

ifpdf (Documentación).

inputenc utf8. (Documentación).

lipsum (Documentación).

listings (Documentación).

longtable (Documentación).

makeidx (Documentación).

morewrites (Documentación).

multicol (Documentación).

multirow (Documentación).

pgf (Documentación).

pgfgantt (Documentación).

pgfplots (Documentación).

subfigure `hang,TABBOTCAP`. (Documentación).

tocloft `subfigure`. (Documentación).

twoopt (Documentación).

verbatim (Documentación).

wrapfig (Documentación).

xcolor (Documentación).

RESUMEN DE OPCIONES DEL ESTILO

En este apéndice se presentan las múltiples opciones de estilo (clase) y sus funciones de forma resumida aunque todas ellas ya han sido presentadas donde corresponde es aconsejable disponer de un resumen de ellas. Estas opciones son las siguientes:

- Tamaño de la página

normalbook Opción por defecto, no hace falta indicarla. Utiliza A4 como tamaño de página.

smallbook Con esta opción se utiliza el tamaño B5 como tamaño de página y se reduce la tipografía de acuerdo con la reducción de tamaño.

tinybook Con esta opción se utiliza el tamaño C5 como tamaño de página, siendo el tamaño más pequeño posible y se reduce la tipografía de acuerdo con la reducción de tamaño.

- Idioma

spanish Opción por defecto, no hace falta indicarla. Usa el español como lenguaje base. El abstract y el resumen se ordenan de acuerdo con ello.

english Usa el idioma inglés como lenguaje base. El abstract y el resumen se ordenan de acuerdo con ello.

- Tipo de documento

tfg Opción por defecto, no hace falta indicarla. El documento será un trabajo fin de grado.

tfm El documento será un trabajo fin de máster.

thesis El documento será una tesis.

- Tipo de copyright

copyright Opción por defecto, no hace falta indicarla. Muestra el copyright normal en el reverso de la portada.

copyleft Muestra el copyleft en el reverso de la portada.

nocopyright No muestra ni copyright ni copyleft en el reverso de la portada.

- Tipo de índice.

normalindex Opción por defecto, no hace falta indicarla. Muestra partes, capítulos y apartados.

extendedindex Muestra también los subapartados. Es aconsejable pensar si en el documento creado tiene sentido o no mostrar ese nivel.

fullindex Muestra hasta el nivel de subsubapartados. Su uso no se aconseja pero está disponible por si en algún caso se considera necesario dada la extensión del documento y la necesidad de mostrar en el índice la organización del documento hasta ese nivel.

- Listados de figuras, ecuaciones, algoritmos ...

loall Muestra todos los listados.

nonelo No muestra ningún listado.

loa Muestra el listado de algoritmos y es compatible con cualquier otra opción menos con loall y nonelo. Se recomienda que no se utilice si el número de algoritmos en todo el documento es menor de tres salvo que se considere adecuado hacerlo.

loc Muestra el listado de códigos y es compatible con cualquier otra opción menos con loall y nonelo. Se recomienda que no se utilice si el número de códigos en todo el documento es menor de tres salvo que se considere adecuado hacerlo.

loe Muestra el listado de ecuaciones y es compatible con cualquier otra opción menos con loall y nonelo. Se recomienda que no se utilice si el número de ecuaciones en todo el documento es menor de tres salvo que se considere adecuado hacerlo.

lof Muestra el listado de figuras y es compatible con cualquier otra opción menos con loall y nonelo. Se recomienda que no se utilice si el número de figuras en todo el documento es menor de tres salvo que se considere adecuado hacerlo.

lot Muestra el listado de tablas y es compatible con cualquier otra opción menos con loall y nonelo. Se recomienda que no se utilice si el número de tablas en todo el documento es menor de tres salvo que se considere adecuado hacerlo.

lotb Muestra el listado de cuadros de texto y es compatible con cualquier otra opción menos con loall y nonelo. Se recomienda que no se utilice si el número de cuadros de texto en todo el documento es menor de tres salvo que se considere adecuado hacerlo.

- Aspecto global

covers Presenta las cubiertas institucionales.

final Se prepara para la versión final eliminando la marca de agua.

printable Por defecto se compila el documento con márgenes simétrico y esta opción lo que hace es añadir margen en la parte derecha de las páginas pares y en la izquierda de las páginas impares para dejar espacio para la encuadernación. Se recomienda una compilación con esta opción para la entrega electrónica y otra con ella para la entrega en papel.

firstnumbered Por defecto las primeras páginas de cada capítulo no se numeran. Si se desea que sean numeradas debe ponerse esta opción.

- Imagen institucional

epsbased Dado que este estilo ha sido diseñado inicialmente es la opción por defecto y no hace falta indicarla. Utiliza los colores institucionales de la Escuela Politécnica Superior y los logos y textos adecuados.

uambased Es una opción genérica que utiliza los colores institucionales de la Universidad Autónoma de Madrid. Es necesario indicar el logo, su ancho, la facultad y cuantas variables sean necesarias para el correcto formato del documento.

cienciasbased Utiliza los colores institucionales de la Facultad de Ciencias y los logos y textos adecuados.

economicasbased Utiliza los colores institucionales de la Facultad de Ciencias Económicas y Empresariales y los logos y textos adecuados.

derechobased Utiliza los colores institucionales de la Facultad de Derecho y los logos y textos adecuados.

-
- filosofiabased** Utiliza los colores institucionales de la Facultad de Filosofía y Letras y los logos y textos adecuados.
 - medicinabased** Utiliza los colores institucionales de la Facultad de Medicina y los logos y textos adecuados.
 - enfermeriabased** Utiliza los colores institucionales de las Escuelas de Enfermería, sin embargo al haber varias y los logos y textos deberán ser modificados utilizando las funciones adecuadas.
 - fisioterapiabased** Utiliza los colores institucionales de las Escuelas de Fisioterapia, sin embargo al haber varias y los logos y textos deberán ser modificados utilizando las funciones adecuadas.
 - Colores globales. Dado que sólo controlan los colores del documento es necesario utilizar todos los comandos necesarios para indicar los textos y logos del documento.
 - blackbased** Todas las decoraciones estarán en negro.
 - graybased** Todas las decoraciones estarán en tonos de gris.
 - redbased** Todas las decoraciones estarán en rojo.
 - greenbased** Todas las decoraciones estarán en verde.
 - bluebased** Todas las decoraciones estarán en azul.
 - yellowbased** Todas las decoraciones estarán en amarillo.
 - magentabased** Todas las decoraciones estarán en magenta.
 - cyanbased** Todas las decoraciones estarán en cian.
 - orangebased** Todas las decoraciones estarán en naranja.
 - Control de compilación
 - overleaf** Permite compilar en overleaf pero no se dispone de xindy que es quien permite una ordenación de los acrónimos y las definiciones teniendo en cuenta que las letras con acento se ordenan con las letras sin acento. No se podrá utilizar tampoco la opción gnuplot.
 - gnuplot** Permite compilar código gnuplot.

FUNCIONES Y ENTORNOS

En este apéndice se presenta un listado de funciones y otro de los entornos. Se presentan en formato de tabla a modo de resumen y para que puedan ser impresas a modo de referencia. En cada función o entorno se indica si debe usarse en el cuerpo del texto o en el preámbulo, los parámetros opcionales si tiene y los obligatorios si tiene. También se presentan aquellos comandos de $\text{\LaTeX 2}_{\epsilon}$ que no sólo ha sido modificado su comportamiento sino que también han cambiado los parámetros.

E.1. Comandos en el preámbulo

Autoría	
<code>\advisor</code>	{tutor}
<code>\author</code>	{titulo}
<code>\coadvisor</code>	{cotutor}
<code>\copyrightdate</code>	{fecha}
<code>\faculty</code>	{facultad/escuela}
<code>\levelin</code>	{titulacion}
<code>\title</code>	[título corto] {titulo}
<code>\speaker</code>	{ponente}
Decoraciones	
<code>\coverdata</code>	{texto}
<code>\facultylogo</code>	{fichero}
<code>\facultylogowide</code>	{dimension}
Directorios	
<code>\codesdir</code>	{directorio}
<code>\datadir</code>	{directorio}
<code>\graphicsdir</code>	{directorio}
<code>\logosdir</code>	{directorio}
Prefacio	
<code>\abstractfile</code>	{fichero}
<code>\ackfile</code>	{fichero}
<code>\dedication</code>	{dedicatoria}
<code>\famouscite</code>	{cita}
<code>\keywords</code>	{palabras}
<code>\palabrasclave</code>	{palabras}
<code>\prefacefile</code>	{fichero}
<code>\privateaddress</code>	{direccion}
<code>\resumenfile</code>	{fichero}

E.2. Commandos en el cuerpo del texto

Código		
<code>\Code</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}{lenguaje}
<code>\AdaCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\ASMCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\ASMMotorolaCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\CCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\CPPCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\CSharpCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\GnuplotCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\HaskellCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\HTMLCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\JavaCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\LaTeXCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\LispCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\MakeCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\MathematicaCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\MatlabCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\OctaveCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\PascalCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\PerlCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\PHPCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\PythonCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\RCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\RubyCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\ScilabCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\SQLCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\VHDLCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
<code>\XMLCode</code>	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}

Ecuaciones	
<code>\boxed</code>	{ecuación}

Estructura		
<code>\cleardoublepage</code>		
<code>\part</code>	[título corto]	{título}{fichero}
<code>\part</code>	[título corto]	{título}
<code>\chapter</code>	[título corto]	{título}{fichero}
<code>\chapter</code>	[título corto]	{título}
<code>\section</code>	[título corto]	{título}{fichero}
<code>\section</code>	[título corto]	{título}
<code>\subsection</code>	[título corto]	{título}{fichero}
<code>\subsection</code>	[título corto]	{título}
<code>\subsubsection</code>	[título corto]	{título}{fichero}
<code>\subsubsection</code>	[título corto]	{título}
<code>\paragraph</code>	[título corto]	{título}{fichero}
<code>\paragraph</code>	[título corto]	{título}
<code>\subparagraph</code>	[título corto]	{título}{fichero}
<code>\subparagraph</code>	[título corto]	{título}

Figuras y tablas		
<code>\subfigure</code>	[etiqueta]	{pie}{contenido}
<code>\subtable</code>	[etiqueta]	{pie}{contenido}

Gantt		
<code>\milestone</code>	[etiqueta enlace]	{etiqueta gantt}{tiempo}
<code>\taskbar</code>	[etiqueta enlace][porcentaje fin]	{etiqueta gantt}{tiempo inicio}{tiempo fin}
<code>\taskgroup</code>	[etiqueta enlace][porcentaje fin]	{etiqueta gantt}{tiempo inicio}{tiempo fin}
<code>\FtoFlink</code>		{etiqueta inicio}{etiqueta fin}
<code>\FtoSlink</code>		{etiqueta inicio}{etiqueta fin}
<code>\StoSlink</code>		{etiqueta inicio}{etiqueta fin}

Gráficas		
<code>\plotlined</code>		{fichero datos}{título datos}
<code>\plotdatalined</code>		{fichero datos}{título datos}
<code>\plotdata</code>		{fichero datos}{título datos}
<code>\plotfunction</code>		{fichero datos}{título datos}
Imágenes		
<code>\image</code>		{ancho}{alto}{fichero}
<code>\imageL</code>	ancho	fichero
Presupuestos		
<code>\budgettitle</code>		{título presupuesto}
<code>\concept</code>		{título}{precio unitario}{cantidad}{coste total}
<code>\separator</code>		
<code>\subconcept</code>		{título}{precio unitario}{cantidad}{coste total}
<code>\subtotal</code>		{subtotal}
<code>\total</code>		{total presupuesto}
Texto citado		
<code>\onlinecitation</code>		{autor}{texto}

E.3. Entornos

De igual forma en la siguiente tabla se presentan los distintos entornos creados o modificados para este estilo. Todos los entornos deben usarse dentro del cuerpo del documento.

General		
<code>algorithm</code>	[pie corto]	{etiqueta}{pie completo}
<code>algorithmN</code>	[pie corto]	{etiqueta}{pie completo}
<code>budget</code>		
<code>equation</code>	[etiqueta]	{título}
<code>figure</code>	[pie corto]	{etiqueta}{pie completo}
<code>gantt</code>		{tiempo inicio}{tiempo fin}
<code>largecitation</code>		{autor}
<code>multiequation</code>		
<code>table</code>	[pie corto]	{etiqueta}{pie completo}
<code>textbox</code>	[pie corto]	{etiqueta}{pie completo}
Listados especiales		
<code>functionality</code>		
<code>objective</code>		
<code>functional</code>		
<code>nonfunctional</code>		
<code>simplelist</code>		
Gráficas		
<code>gnuplot</code>	[opciones]	
<code>loglogplot</code>	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}
<code>semilogxplot</code>	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}
<code>semilogyplot</code>	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}
<code>xyplot</code>	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}



Universidad Autónoma
de Madrid