

Trabajo fin de grado

Sensor de viento en Dron



Daniel Serena Sanz

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

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería de Textos

TRABAJO FIN DE GRADO

Sensor de viento en Dron

Si hace falta subtítulo

Autor: Daniel Serena Sanz
Tutor: Daniel Serena Sanz

junio 2019

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, nº 1
Madrid, 28049
Spain

Daniel Serena Sanz
Sensor de viento en Dron

Daniel Serena Sanz
C\ Francisco Tomás y Valiente Nº 11

A mi mujer y a mis hijos

*Lo peor es cuando has terminado un capítulo
y la máquina de escribir no aplaude.*

Orson Welles

PREFACIO

Este estilo de $\text{\LaTeX} 2_{\varepsilon}$ 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 sis 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

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

En este sentido se ha intentado diseñar un estilo de $\text{\LaTeX} 2_{\varepsilon}$ que mantenga una imagen corporativa y con comandos simples que permitan mantener la imagen corporativa con la calidad necesaria sin olvidar las necesidades del autor. Para ello se han creado un conjunto de comandos simples en torno a paquetes complejos. Estos comandos permiten realizar la mayoría de las operaciones que un documento de este tipo pueda necesitar.

Así mismo se puede controlar un poco el diseño del documento a través de las opciones del estilo pero siempre manteniendo la imagen institucional.

PALABRAS CLAVE

Diseño de documento, $\text{\LaTeX} 2_{\varepsilon}$, thesis, trabajo fin de grado, trabajo fin de master

ABSTRACT

In our School a considerable number of documents are produced, as many educational as research. Our students also contribute to this production through his final degree, master and thesis projects. The objective of this material is to facilitate the editing of all these documents and at the same time to promote our corporate image, facilitating the visibility and recognition of our center.

In this sense we have tried to design a style of $\text{\LaTeX}2_{\varepsilon}$ that maintains a corporate image and with simple commands that allow to maintain the corporate image with the necessary quality without forgetting the needs of the author. For this, a set of simple commands have been created around complex packages. These commands allow you to perform most of the operations that a document of this type may need.

Likewise, you can control a little the design of the document through the options of the style but always maintaining the institutional image.

KEYWORDS

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

ÍNDICE

1	Introducción	1
2	Estado del arte	3
3	Teoría	5
4	Sistema	7
4.1	Sensor de viento	7
4.2	Dron	9
4.2.1	Sistema de propulsion	9
4.2.2	Sistema de control	15
4.2.3	Mecánica	22
4.3	Sensor de viento en Dron	26
5	Resultados experimentales y discusión	29
6	Conclusión y outlook	31
7	Como continuar el proyecto a partir de mis resultados	33
8	Bibliografía	35
9	Estética	37
9.1	Tipo de documento	37
9.2	Gama de colores	37
9.3	Colores	38
9.4	Uso de los colores	38
10	Estructura	43
10.1	Título, autor, tutor y otras variables	43
10.2	Índices	44
10.3	Copyright, dedicatoria y cita inicial	44
10.4	Prefacio, resumen	45
10.5	Partes, capítulos	45
10.6	Glosario, acrónimos y definiciones	46
10.6.1	Glosario	46
10.6.2	Acrónimos	47
10.6.3	Definiciones	47
10.7	Referencias	48

10.8 Bibliografía	48
11 Primeros pasos	49
11.1 Estructurar el documento	49
11.1.1 Organización de $\text{\LaTeX} 2_{\varepsilon}$	49
11.1.2 Organización de la documentación	49
11.1.3 Otros elementos de la estructura del documento	50
11.2 Enlazar la bibliografía	51
12 Elementos internos	53
12.1 Figuras	53
12.1.1 Gráficas	54
12.1.2 Gráficas con pgfplots	54
12.1.3 Gráficas con gnuplot	56
12.1.4 Imágenes	57
12.1.5 Diagramas de Gantt	58
12.2 Tablas	59
12.2.1 Presupuestos	60
12.3 Cuadros de texto	61
12.4 Ecuaciones	62
12.5 Código	63
12.6 Algoritmos	66
12.7 Listas	66
12.7.1 Listas tradicionales de $\text{\LaTeX} 2_{\varepsilon}$	67
12.7.2 Nuevos tipos de listas	67
12.8 Referencias internas e hiperenlaces	68
13 Compilación	71
13.1 Pdfflatex	71
13.2 Arara	72
13.3 Overleaf	73
13.4 Permitir el uso de la shell al compilador	73
Bibliografía	75
Definiciones	77
Acrónimos	79
Apéndices	81
A Word® vs. $\text{\LaTeX} 2_{\varepsilon}$	83

A.1	Ventajas e inconvenientes de L ^A T _E X 2 _{<}	83
A.2	Ventajas e inconvenientes de Word®	83
A.3	¿Cuál elijo?	84
B	Instalación	85
B.1	Linux	85
B.2	Windows™	86
B.3	Mac OS X	86
B.4	Overleaf o ShareLatex	87
B.5	¿Dónde está el manual?	87
B.6	Corrección ortográfica y codificación de caracteres	87
B.7	¿Qué editor utilizo?	87
C	Paquetes incluidos	89
D	Resumen de opciones del estilo	91
E	Funciones y entornos	95
E.1	Comandos en el preámbulo	95
E.2	Comandos en el cuerpo del texto	97
E.3	Entornos	98

LISTAS

Listas de algoritmos

12.1 Ejemplo de algoritmo.....	66
--------------------------------	----

Listas de códigos

12.1 Respuesta a la conexión de los usuarios.....	65
12.2 Conexión TCP.....	65

Listas de cuadros

12.1 Este es un cuadro de texto en el que usando el paquete lipsum se genera el texto internamente	62
13.1 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.	72

Listas de ecuaciones

4.1a Primer ejemplo de subequación	19
4.1b Segundo ejemplo de subequación	19
4.1c Segundo ejemplo de subequación	19
12.1 Esto es un ejemplo de título de ecuación incluida la propia ecuación $\sum c_{ij} = \frac{a}{J^{adx}}$	62
12.2a Primer ejemplo de subequación	62
12.2b Segundo ejemplo de subequación	62

Listas de figuras

4.1	8
4.2	8

4.3	8
4.4	Ejes MPU9250	9
4.5	9
4.6	10
4.7	12
4.8	14
4.9	16
4.10	18
4.11	21
4.12	Chasis X y H	23
4.13	23
4.14	25
4.15	26
4.16	Chasis raspi dron	27
4.17	Zona sensorización dron	27
4.18	Pata dron	27
4.19	28
12.1	Ejemplo de uso de figure	53
12.2	Ejemplo de uso de figure	54
12.3	Esquemas de ejes de gráficas XY	55
12.4	Representaci�n de expresiones matem�ticas	56
12.5	Esquemas de ejes de gr�ficas XY	57
12.6	Ejemplo de diagrama de Gantt	58

Lista de tablas

9.1	Gamas de colores	38
9.2	Colores complementarios	39
9.3	Colores complementarios claros	39
9.4	Colores complementarios oscuros	40
12.1	Tabla de ejemplo	59
12.2	Tabla de ejemplo con subtablas	60

Lista de cuadros

INTRODUCCIÓN

ESTADO DEL ARTE

TEORÍA

SISTEMA

sistema

4.1. Sensor de viento

Actualmente los sensores de vientos habituales cuentan con dos elementos básicos. Uno de ellos se encarga de medir la velocidad del viento y el otro su dirección, a distinguir el anemómetro y la veleta respectivamente. Estos dos elementos pueden encontrarse montados de forma separada o conjuntamente denominándose veleta potencio métrica. Hay que tener en cuenta que este sistema, aun siendo muy preciso, tiene un coste elevado y necesita de una instalación en una localización fija, además de necesitar una altura y distancia mínima de cualquier superficie para poder realizar las medidas correctamente.

Diariamente utilizamos dispositivos electrónicos que cuentan con sensores que miden la inclinación, aceleración y dirección del dispositivo, podemos diferenciar desde un teléfono móvil a un coche o incluso un avión. Estos sensores son utilizados para diversas funciones como, el nivel de un objeto (un avión volando), la aceleración de un objeto (coche de fórmula 1), etc.

¿Por que no usar estos sensores para medir el viento?

En esta sección trataré de como utilizar un sensor de aceleración, giroscopio y magnetómetro para medir el viento. Para realizar este experimento he utilizado el sensor MPU9250 [1] que cuenta con la integración del sensor MPU6050, aceleración y giroscopio, y el sensor AK8963, magnetómetro, componiendo de esta forma un microchip que mide en función de 3 ejes la alteración de su posición respecto de la tierra.

Los datos obtenidos por el sensor MPU6050, son datos sin tratar que indican la aceleración en el instante de la medición y la inclinación respecto de la anterior medida tomada, mientras que el sensor AK8963 indica la orientación del sensor.

Para poder saber claramente en que posición se encuentra el chip respecto de una superficie plana y no su inclinación respecto de la anterior medida, he utilizado el algoritmo de madgwick que utiliza

WARNING FOTO

Figura 4.1: PONER FOTO DE LOS SENSORES.

cuaterniones para indicar la inclinación absoluta respecto de la tierra.

Los cuaterniones es un conjunto de cuatro componentes complejas que sirven para la teoría de números, rotaciones en el espacio y para diseño de gráficos, en mi caso lo usaré para las rotaciones en el espacio, los cuaterniones se pueden expresar de la siguiente forma [2].

$$\mathbb{H} = \{a + bi + cj + dk : a, b, c, d \in \mathbb{R}\} \subset \mathbb{C}^2$$

Figura 4.2: Fórmula de cuaternion, podemos diferenciar w, x, y, z que son los valores reales que componen el cuaternion Q y el conjunto i, j, k que componen la parte imaginaria de las unidades.

Mediante el algoritmo de madgwick, convertimos los datos obtenidos desde el giroscopio, acelerómetro y magnetómetro en un componente de 3 elementos, yaw, pitch y roll. Cada uno de estos elementos nos indica su rotación respecto de un eje del aeromodelo.



Figura 4.3: Avión con los ejes de giro, yaw, pitch y roll, obtenidos mediante cuaterniones.

Mediante la utilización de estas rotaciones respecto de sus ejes podemos determinar como hay que actuar en los motores para que se mantenga volando estable y horizontal.

En este punto contamos con la inclinación del sensor respecto de la tierra, su aceleración, dirección en grados, con este conjunto de datos podemos calcular la variación de su posición respecto de un punto.

Para calcular el viento he planteado un sistema en el que el sensor está situado en un "poste flexible" inclinado al suelo, de esta forma si el sensor está orientado al norte y se inclina hacia delante obtenemos una dirección de viento de sur a norte, y en caso de inclinarse hacia delante y la izquierda, obtenemos un viento de sur-este. La fuerza del viento viene indicado directamente por los datos obtenidos del acelerómetro en función de los tres ejes.

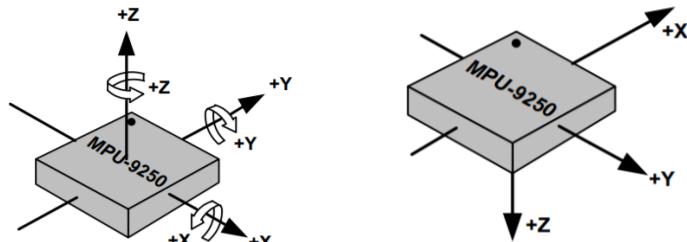
(a) Sensor MPU9250 con ejes del sensor
MPU6050(b) Sensor MPU9250 con ejes del sensor
AK8963

Figura 4.4: Sensor MPU9250 con ejes de funcionamiento. Figura 4.4(a) muestra el sentido de los ejes del sensor MPU6050. Figura 4.4(b) muestra el sentido de los ejes del sensor MPU6050.

4.2. Dron

Para la construcción del dron sobre el que se han realizado las pruebas, se ha seguido una lista de componentes, partiendo de los mas restrictivos y completando con aquellos mas flexibles dependiendo del sistema montado. A continuación indicaré en grupos los componentes elegidos con una previa descripción de su funcionamiento e importancia en el aeromodelo final.

WARNING
FOTO

Figura 4.5: PONER FOTO DE DRON VIRGEN.

4.2.1. Sistema de propulsion

Motores

Los motores son una parte fundamental de un dron ya que han de proporcionar la potencia suficiente para hacer girar la hélices y por consecuente hacer volar al dron.

He tenido en cuenta varios parámetros como por ejemplo, la potencia, el consumo y la fuerza máxima de empuje. Los motores vienen clasificados según su velocidad indicándose como KV, revoluciones por minuto por voltio y su dirección de giro, distinguiendo entre CW y CCW, es necesario tener el mismo numero de motores CW y CCW ya que estos giran en sentido contrario, el motivo de tener el mismo número de motores que giren en sentido horario y anti horario es debido a que si no provocaría un efecto de vórtice, haciendo girar el dron sobre si mismo sin ningún control. A su vez se pueden diferenciar entre dos clases de motores, con escobillas o sin ellas, este factor afecta a la forma de cambio de giro de los motores, los motores sin escobillas llevan un sistema de carga de polos magnéticos para realizar el cambio de dirección mientras que los motores con escobillas hacen circular corriente por unas bobinas generando un campo magnético y en consecuencia haciendo atraer o repeler el motor en un giro u otro, cabe destacar que es necesario un total de 3 bobinas para hacer girar el motor, puesto que si tuviésemos solo dos podría provocar que el motor se quedase en perpendicular cuando se produjese el cambio de giro.

He elegido unos motores sin escobillas por su rendimiento, menor desgaste que los motores con escobillas y sencillez a la hora de cambiar el sentido de giro.



Figura 4.6: Motores de dron, 2 CW y 2 CCW.

Los motores elegidos constan de 2300 KV y sin escobillas, contando así con 3 cables: alimentación, masa y potencia. Se varía la velocidad de los motores mediante la utilización de una señal PWM, variándose desde un 0 % a un 100 % de potencia modificando su ciclo de trabajo desde 1 ms a 2'5 ms, siendo a 1'5 ms funcionaría a mitad de la potencia y a 2 ms funcionaría al 100 %.

Un PWM es la modulación de una señal digital provocando en sistemas con menos velocidad de lectura la llegada de una señal analógica. Un PWM con una salida de 3,3 V y con un ciclo de trabajo del 50 % produciría en el sistema receptor una señal analógica de $3,3 \text{ V} / 50 = 0,66 \text{ V}$ de señal de entrada, reduciendo su potencia un 50

fsafsa

Controlador de velocidad

El ESC, Electronic Speed Controller, es un controlador esencial que determina la potencia suministrada al motor, es necesario instalar uno por cada motor controlandolos asi de forma independiente. Puede contar con 2 cables o 3 dependiendo si es un controlador de velocidad para un motor con escobillas o sin ellas.



Figura 4.7: Controladores de velocidad dron.

Los controladores de velocidad comprados cuentan con una capacidad de pico de descarga máxima de 35 amperios, la cantidad de descarga máxima limita la potencia máxima que pueden generar los motores. Los ESC se deben elegir en función de la batería que vayas a utilizar, teniendo en cuenta el número de celdas y su ratio de descarga continua.

Los ESC pueden recibir la señal que indique la potencia que debe de suministrar al motor, a partir del controlador de vuelo o directamente desde el receptor del mando. Es aconsejable recibirla desde el controlador de vuelo ya que de esta forma gestiona un microchip la estabilización del dron, favoreciendo a un control mucho mas fácil del aeromodelo a control remoto. [3]

Hélices

Las hélices son un punto menos crítico a la hora de elegirlas y montarlas sobre el dron, contamos con diferentes formatos, formas y número de aspas. Podemos encontrar hélices con las siguientes características:

Tamaño: Puede variar desde 2 a 19 pulgadas, el tamaño de la hélices se ve limitado por el tamaño de los brazos del chasis donde se monte. Hay que tener en cuenta que a mayor tamaño mas superficie y por tanto mas empuje, hoy en día se utilizan mayormente las hélices entre 4 y 6 pulgadas, estas hélices favorecen el funcionamiento de drones con motores rápidos como los utilizados en drones de carreras, no obstante también se utilizan las hélices con tamaño de 10 o mas pulgadas para aeromodelos que no priorizan la velocidad y necesiten estabilidad y llevar cargas elevadas.

Número de aspas: Varía entre 2 a 6 aspas, el número de aspas afecta a la superficie de empuje, aumentando la fricción y por tanto la fuerza que ejerce para levantar el drone. En hélices de tamaño pequeño, hasta 6 pulgadas, podemos encontrar hasta un total de 6 aspas mientras que en hélices de tamaño grande, 10 pulgadas o más, no solemos superar las 2 aspas.

Forma: La forma es un punto crítico de este elemento, tienen una forma de pala corvada para conseguir el empuje del aire en el sentido contrario de giro del motor. La terminación del aspa también afecta a la hora de volar un dron, se pueden diferenciar acabadas en punta, Bullnose e Híbridas bullnose, determinando mayor o menor empuje y por consecuente mayor o menor consumo. Podemos contar con hélices de 2 a 6 aspas, con un tamaño variable desde 2 hasta 19 pulgadas.



Figura 4.8: Hélices dron, con 3 aspas y con un tamaño de 5 pulgadas.

En el dron diseñado he priorizado el tamaño, como he explicado anteriormente el tamaño de las hélices afecta a la fuerza de empuje vertical del dron para poder levantar su carga, no obstante dado que contamos con un dron con un peso máximo de NOSECUANTOS gramos y con una carga añadida de 200g máx., he elegido unas hélices de 5 pulgadas con un total de 3 aspas, obteniendo así un punto óptimo entre empuje o fuerza máxima y duración de vuelo.

4.2.2. Sistema de control

Placa distribuidora de potencia

Un punto crucial de los componentes electrónico es su alimentación, para favorecer la durabilidad y estabilidad del circuito o circuitos electrónico debemos aislar el circuito de potencia del circuito lógico mediante elementos con la placa distribuidora de potencia (PDB) que cuenta con elementos como diodos, resistencias y transistores para evitar picos de tensión y derivaciones.

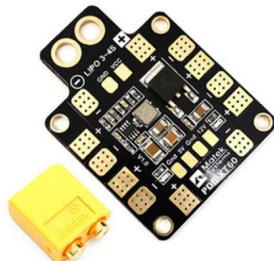


Figura 4.9: Placa distribuidora de potencia con conector XT60.

Esta placa se encarga de conectarse con la batería mediante un conector, en este caso, XT60; este conector impide conectar cada polo con su opuesto debido a su forma evitando cortocircuitos y daños al sistema.

Por otro lado, la PDB distribuye la tensión necesaria al resto de los componentes, entre los que destacamos:

ESC: Hasta un total de 6 controladores de velocidad, en mi caso son 4 ESC que funcionan con baterías de 2 a 4 celdas y por tanto desde 6 voltios a 16 voltios.

Circuito 5V: Circuitos de 5 voltios de tensión con un sistema de BEC para evitar picos y conseguir un circuito con alimentación continua y estable, con una potencia continua de 2 amperios y una potencia máxima de 2'5 amperios.

Circuito 12V: Circuitos de 12 voltios de tensión, con el mismo funcionamiento que los circuitos a 5 voltios, con una potencia continua de 500 mA y máxima de 0'8 amperios.

Controlador de vuelo

La placa controladora de vuelo es la unidad de procesamiento del dron, es la encargada de gestionar la señales provenientes del receptor, leer los parámetros de los diferentes sistemas de estabilización y su posterior modificación para proporcionar la potencia adecuada a cada ESC.

Placa comercial

La elección de la controladora de vuelo viene determinada solamente por el protocolo de comunicación que uses, aunque a día de hoy la mayoría de controladoras cuentan con software re programable que se ajusta a los diferentes sistemas de comunicación, como los mas conocidos iBUS o sBUS.



Figura 4.10: Placa controladora mateksys.

En la controladora de vuelo puede conectarse el sistema de comunicación, sistema de vídeo, controladores de velocidad de los motores, bocina de emergencia, antena de comunicación de largo alcance.

REF

Placa desarrollada

En este TFG he reproducido el sistema de control del dron en una Raspberry Pi 3 B+, con el fin de unificar todos los controles en un solo punto y poder utilizar todas señales entrantes y salientes para su uso en mediciones posteriores.

Para representar el sistema de la controladora de vuelo he partido del sistema de descifrado del protocolo de comunicación elegido, iBUS, este protocolo te manda paquetes de información por el un puerto serie. La raspberry cuenta con dos pines de comunicacion por puerto serie, Rx para recibir la información y Tx para transmitir la información. El paquete recibido tiene un tamaño total de 31 bytes, con un byte inicial standard con contenido igual a , a continuacion encontramos un total de 14 canales con 2 bytes por canal y un checksum al final con un tamaño de 1 byte. El paquete es transmitido cada 7ms con una tasa de baudios de 115200, es decir, un total de 115200 bits por segundo. Para desglosar el paquete recibido por el mando he codificado un algoritmo en python donde diferencio y convierto cada uno de los bytes a decimal, obteniendo de esta forma un valor de cada canal entre 1000 y 2000, siendo 1000 el mínimo valor y 2000 el máximo valor posible.

Para poder producir el PWM correcto y mandarselo a los controladores de velocidad, he contado con un chip conversor, PCA9685, de digital a PWM hasta un máximo de 8 canales, ha sido necesario dicho chip ya que la raspberry no cuenta con el numero suficiente de canales como para controlar 4 motores. La configuración del controlador PCA9685 se ha realizado mediante comunicación por I2C y su respectiva librería de adafruit obtenida por github. He tenido que indicar la frecuencia de salida de los PWM, 2KHz, y el ciclo de trabajo previamente obtenido por nuestro rango de entrada entre 1000 y 2000. [4]

Sistema

Input: mínimo de 1000 y máximo de 2000

Output: señal PWM a 2Khz de frecuencia y un ciclo de trabajo de 125us hasta un máximo de 250us

PCA9685

Input: canal de 1 a 8, mínimo entre 3296 y 2498, máximo rango fijo 4095

Output sistema: señal PWM a 2Khz de frecuencia y un ciclo de trabajo de 125us hasta un máximo de 250us

$$input = [1000, 2000] \quad (4.1a)$$

$$rango = 3296 - (input - 1000) * \frac{(3296 - 2498)}{1000} \quad (4.1b)$$

$$pwm.set_pwm(canal, rango, 4095) \quad (4.1c)$$

Una vez tengo la señal de salida PWM en función de la entrada del mando, modifco la señal a partir de los datos obtenidos por el sensor MPU9250 el cual nos indica la inclinación y fuerzas que se ejercen sobre el dron. De esta forma mandamos la potencia adecuada a cada motor para conseguir un vuelo estable y horizontal. Debido a las dificultades de probar el sistema de estabilización y su ajuste, decidí por trabajar con la placa comercial.

Sistema de comunicación

El sistema de comunicación cuenta con dos componentes, receptor y transmisor, se pueden elegir diferentes tipos teniendo en cuenta su protocolo de comunicación.

El sistema de comunicación tiene diferentes protocolos para transmitir la información, utiliza un protocolo de comunicación entre emisor y receptor y otro distinto entre receptor y unidad de procesamiento o controladora de vuelo. [5]



Figura 4.11: Mando FlySky para dron.

El protocolo de comunicación entre receptor y emisor tiene que ser siempre iguales, es por eso que la mayoría de mandos, emisores, vienen con un receptor standard. En mi caso he utilizado un pack FlySky entre emisor y receptor.

Entre el receptor y la unidad de procesamiento podemos encontrar protocolos como PWM, PPM, SBUS, iBus entre otros. La diferencia entre estos protocolos es la forma de estructurar el paquete de datos o la conexión que hay que realizar a nivel de hardware. Por su simplicidad y comodidad he utilizado el protocolo PWM que cuenta con un paquete de transmisión de 31 bytes.

4.2.3. Mecánica

Chasis

El chasis de un dron es la estructura donde se van a montar todos los componentes anteriormente descritos, contamos con diferentes características a elegir a la hora de comprar o fabricar nuestro chasis:

Número y longitud de brazos El número de brazos determina el número de motores que podemos instalar en nuestro aeromodelo, hay chasis con 4 hasta 8 brazos, siempre números pares. La longitud de los brazos nos ayuda a instalar unas hélices de mayor tamaño.

Distribución La distribución de los brazos es una característica no muy importante y más bien estética. En chasis con 4 brazos podemos encontrar distribuciones en H o en X.



(a) Chasis drone en forma de X



(b) Chasis drone en forma de H

Figura 4.12: Diferencia entre formas de chasis en X 4.12(a) y H 4.12(b) en drones.

Material El material del chasis dispone su rigidez y peso al dron, el material mas utilizado es la fibra de carbono por su leve peso y su rigidez que permite recibir golpes sin llegar a romperse.



Figura 4.13: Chasis dron.

El dron elegido cuenta con un chasis con un total de 4 brazos en disposición H con un tamaño total de cada brazo de 220 milímetros permitiéndome instalar unas hélices de hasta 5 pulgadas, este chasis de fibra de carbono pesa un total 127 gramos.

Batería

Las baterías para cualquier radio control son indispensables ya que es elemento que proporciona la electricidad para que todo funcione, por eso es indispensable calcular el gasto de nuestro sistema para escoger una batería que nos proporcione una durabilidad media elevada. A la hora de elegir una batería podemos encontrar diferentes parámetros donde escoger, como:

Composición: Podemos encontrar múltiples baterías en función de su composición. Hay diferentes tipos como Niquel Cadmio(NiCd), Ion Litio (Li-ion), Ion Litio Polímero (LiPo).

Capacidad: A mas capacidad de carga de la batería mas duración de uso tiene pero a su vez mas peso. Afectando el peso al tiempo de vuelo en el caso del dron.

Número de celdas: Cuanto mayor es el número de celdas, generalmente, más capacidad de la batería y más potencia. Las celdas tienen una tensión de 3'3 V y al estar conectadas en serie proporcionan un total de 11V.

Tasa de descarga: La tasa de descarga indica la tasa máxima de descarga que ofrece la batería, a mayor tasa de descarga mas potencia se transmite a los motores y por tanto mayor empuje.

Para el dron utilizado he escogido una batería de 3 celdas de Ion Litio Polímero (LiPo) con una capacidad de 1500 miliamperios (mAh) con un peso de 107 gramos (g).



Figura 4.14: Batería dron.

La configuración de esta batería nos ayuda a tener una capacidad aceptable en relación con el peso. Está compuesta por Ion Litio de polímero proporcionando una durabilidad de 300 a 500 horas (h) de uso sin necesidad de mantenimiento por parte del usuario. Cuenta con 3 celdas en serie consiguiendo una mayor capacidad de almacenamiento y mayor tensión de salida para alimentar el sistema.

4.3. Sensor de viento en Dron

En esta sección describiré como uno el sensor de viento con el dron y que es necesario para que todo funcione como un único sistema. Para poder llevar a cabo esta unión se ha de tener en cuenta el peso del sistema de viento, raspberry y sensorización, junto con el peso del dron para hacer una estimación media de la duración de vuelo.

Desde donde parto Para poder diseñar un sensor de viento que funcione con un dron primero hay que estudiar que sistema de estabilización usa el dron, podemos diferenciar entre dos principales sistemas. Un primer sistema realiza una estabilización del dron que registra la desviación de su posición desde un punto y lo corrige para mantenerse siempre en el mismo punto. El segundo sistema de estabilización intenta que el dron no se caiga modificando la potencia de los motores acorde con la fuerza que se ejerza sobre él, de esta forma consigue que el dron esté los más horizontal posible.

En mi caso, uso un sistema comercial que funciona como el segundo sistema descrito y por tanto uso un giroscopio para poder medir dichas variaciones. En el caso de usar el primer sistema se debería usar una medición de la potencia suministrada a los motores para ver que potencia esta ejerciendo para contrarrestar la fuerza que se ejerce sobre él.



Figura 4.15: PONER FOTO DE DRON VIRGEN Y RASPY/SENSOR.

Por otro lado contamos con un chasis base de un dron de 220 mm de longitud de hélice a hélice donde es necesario montar un soporte para instalar nuestro ordenador o unidad de procesamiento, en mi caso una Raspberry Pi 3 B+, y nuestro sistema de sensorización.

Como lo junto Para poder unir los dos sistemas y realizar medidas en vuelo, he tenido que diseñar una serie de piezas 3D para sus posterior impresión y acoplamiento en el chasis del dron.

Entre las piezas de impresión podemos contar con:

Soporte Raspberry Pi Para el soporte de la Raspberry Pi 3 B+, he diseñado una pieza que se acopla en la parte inferior del chasis del dron.

Zona de sensorización: Una vez instalado el soporte de la Raspberry he diseñado una tapa intermedia donde se puede instalar el sistema de sensorización necesario. Esta zona esta diseñada con suficiente espacio para instalar mas sensores a parte del sensor base diseñado, sensor de viento, como por ejemplo un sensor de gas con sus componentes necesarios.

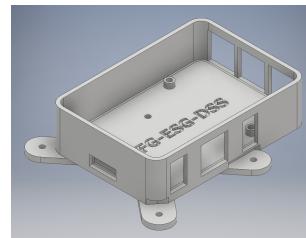


Figura 4.16: Soporte Raspberry dron.

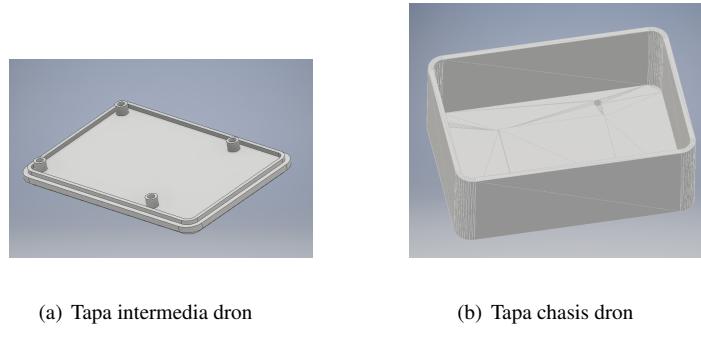


Figura 4.17: Zona de sensorización. Tapa intermedia 4.17(a) acoplada al soporte de la raspberry 4.16 y a ella se ancla la tapa 4.17(b) para cerrar nuestro modelo.

Al diseñarlo de forma modular, por piezas, podemos modificar la tapa del chasis del dron 4.17(b) y diseñar otra forma o ampliar el espacio para que se acople a la tapa intermedia 4.17(a).

Elementos de aterrizaje Una vez instalado las anteriores piezas, hemos prolongado el dron por su parte inferior un total de TANTOS cm y es necesario instalar un tren de aterrizaje.

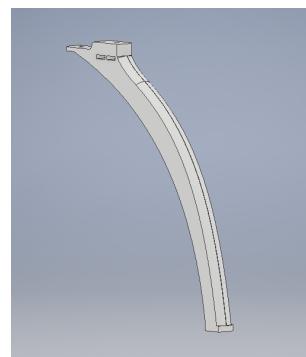


Figura 4.18: Diseño de pata del dron del tren de aterrizaje.

Para el tren de aterrizaje se ha probado instalar un total de 4 patas, una por cada motor. Por problemas de estabilidad se eliminó una de ellas y la tercera se instaló en la parte delantera-media del dron. Con 3 patas nos aseguramos aterrizar el dron en una superficie plana.

Como funciona todo junto Una vez unido todo como un solo sistema, dron y sensorización junto con la raspberry, procederé a realizar las mediciones con el dron volando.

Para tomar las medidas correctamente el dron ha de ponerse en un sitio y mantenerlo volando. Se procede a la toma de medidas y con la variación de su posición en cuanto a inclinación que sufre por fuerzas como el viento obtenemos los datos. Para la realización de esta prueba contamos con los cálculos de vuelo previamente hechos sobre un dron con un peso máximo de 900 gramos y con una batería de 3 celdas y 1500 mAh de capacidad. El único manejo que se ha de hacer con el dron a la hora de medir es modificar su altura ya que no tiene un sistema automático de mantenimiento de distancia respecto del suelo.

**WARNING
FOTO**

Figura 4.19: FOTO DEL SISTEMA FINAL.

RESULTADOS EXPERIMENTALES Y DISCUSIÓN

resultados y discusion

CONCLUSIÓN Y OUTLOOK

conclusion

COMO CONTINUAR EL PROYECTO A PARTIR DE MIS RESULTADOS

continuarproy

BIBLIOGRAFÍA

Para la creación de la bibliografía se ha elegido el uso de BibTex dado que es el que permite más versatilidad a la hora de crear bibliografía.

La bibliografía siempre aparecerá al final del documento aunque es necesario configurar su funcionamiento. Para ello se utilizará el comando **\bibliographyconfig[estilo]{fichero}** que tiene un parámetro opcional que es el estilo de la bibliografía y el paraámetro obligatorio es el nombre del fichero.bib en el que está la bibliografía en formato BibTex obviando la extensión de este fichero. El estilo por defecto es el siam. Por ejemplo se debe citar con el comando **\cite{label}** como en este ejemoplo de cita [6] en el que se usa la etiqueta Narendra1990 que es la etiqueta asociada al artículo en el fichero .bib correspondiente.

En la mayoría de las entradas de la base de datos se puede introducir un campo **note**. Si en ese comando se indica algo como: **note = “\href{file://unfichero.pdf}{Leer}”** se puede realizar el hiperenlace al documento correspondiente almacenado junto al documento creado. También se pueden realizar enlaces a páginas web [7].

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.

9.1. 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}`.

9.2. 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 las 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**, **economicsbased**, **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]{tfgtfmthesisuam}.

9.3. Colores

El uso de los colores es muy simple, basta con usar el comando \color{nombrericolor}. Hay muchos colores predefinidos dependiendo de la gama de colores elegida. En la tabla 9.1 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	I	vl	ul	bg
uam							
ciencias							
derecho							
economicas							
enfermeria							
eps							
filosofia							
fisioterapia							
medicina							
profesorado							
psicologia							

Tabla 9.1: Gamas 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 9.2 para los colores básicos, para los colores ‘vl’ (verylight) se pueden ver en la tabla 9.3 y para los colores ‘vd’ (verydark) se pueden ver en la tabla 9.4.

9.4. Uso de los colores

Se puede usar cualquier color usando su nombre o con el comando \color{nombrericolor} 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	[color swatch]							
ciencias	[color swatch]							
derecho	[color swatch]							
economicas	[color swatch]							
enfermeria	[color swatch]							
eps	[color swatch]							
filosofia	[color swatch]							
fisioterapia	[color swatch]							
medicina	[color swatch]							
profesorado	[color swatch]							
psicologia	[color swatch]							

Tabla 9.2: 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	[color swatch]							
ciencias	[color swatch]							
derecho	[color swatch]							
economicas	[color swatch]							
enfermeria	[color swatch]							
eps	[color swatch]							
filosofia	[color swatch]							
fisioterapia	[color swatch]							
medicina	[color swatch]							
profesorado	[color swatch]							
psicologia	[color swatch]							

Tabla 9.3: Colores complementarios claros. El nombre del color se construye concatenando el nombre de la fila con el prefijo 'vl', el sufijo 'based' y se añade como segundo sufijo el nombre de la columna.

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

Tabla 9.4: 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.

ESTRUCTURA

En este capítulo se explican todos los elementos que conforman la estructura de un documento. Para elementos internos a la estructura como son las figuras, las tablas o las ecuaciones es necesario leer el capítulo 12. Si lo que se busca es cómo diseñar y estructurar el documento debe leer el capítulo 11.

10.1. Título, autor, tutor y otras variables

Se dispone de una colección de variables que es necesario definir en el preámbulo del texto (antes de `\begin{document}`). Estas variables permiten que el documento esté estructurado correctamente. Las variables que es obligatorio definir son las siguientes:

`\title[valor]{Valor}` ‘Valor’ es el título del trabajo mientras que ‘valor’ es el título del trabajo reducido que es el que se utilizará en los pies de página de las páginas pares pero no es necesario definirlo si el título es suficientemente corto.

`\author{Valor}` ‘Valor’ es el nombre del autor del documento.

`\advisor{Valor}` ‘Valor’ es el nombre del tutor o tutores o del director o directores.

`\levelin{Valor}` ‘Valor’ es el grado que se pretende obtener.

`\copyright{Valor}` ‘Valor’ es la fecha del copyright.

También se dispone de otro conjunto de variables que no es necesario definir pero que son útiles en algunas situaciones:

`\speaker{Valor}` ‘Valor’ es el nombre en caso de existir. Si no existe no es necesario definir esta variable.

`\privateaddress{Valor}` ‘Valor’ es la dirección privada del autor. Esta variable se utiliza en el copyright y puede no definirse.

`\faculty{Valor}` ‘Valor’ es el nombre de la facultad o escuela pero no es necesario definirlo salvo que difiera del predefinido a través de la variable de la opción asociada a la imagen

corporativa que se explica en la sección 9.2.

\facultylogo{Valor} ‘Valor’ es el nombre del fichero con el logo de la facultad o escuela sin extensión pero no es necesario definirlo salvo que difiera del predefinido a través de la variable de la opción asociada a la imagen corporativa que se explica en la sección 9.2.

\facultylogowide{Valor} ‘Valor’ es el ancho del logo de la facultad o escuela pero no es necesario definirlo salvo que difiera del predefinido a través de la variable de la opción asociada a la imagen corporativa que se explica en la sección 9.2.

10.2. Índices

Además del tradicional índice de contenidos se disponen de otros muchos índices controlables a través de las opciones de la clase. Si se utiliza la opción **loall** se presentarán todos los índices. Por el contrario la opción **lonone** elimina todas estas listas. Si no se usa ni estas opciones ni las que se describen a continuación se presentarán todos los índices. Las siguientes opciones no se excluyen entre si de tal forma que es posible indicar varias simultáneamente:

loa Presenta la lista de algoritmos.

loc Presenta la lista de códigos.

loe Presenta la lista de ecuaciones.

lof Presenta la lista de figuras.

lot Presenta la lista de tablas.

ltb Presenta la lista de cuadros de texto.

Como norma de estilo se recomienda no añadir una lista si no tiene al menos 3 elementos aunque queda a discreción del autor.

10.3. Copyright, dedicatoria y cita inicial

El texto del copyright aparecerá siempre por defecto en el reverso de la portada, sin embargo se puede cambiar usando opciones de la clase. Estas opciones pueden tomar tres valores, **copyright**, **copyleft** y **nocopyright**. La primera de ellas muestra un texto de copyright y es el valor por defecto, la segunda un texto de copyleft y la tercera elimina el texto de copyright.

En este tipo de documentos suele hacerse una dedicatoria corta y citar alguna frase celebre. Ambas cosas se pueden hacer con los comandos \dedication{} y \famouscite{} en el preámbulo del documento y que llevan como parámetro la dedicatoria y la cita celebre. En los fuentes de este documento se puede encontrar un ejemplo de estos dos comandos. Debido a cómo son estos comandos

no deben introducirse líneas en blanco en los textos y los saltos de línea o párrafo deberán hacerse con `\` o `\[height]`.

10.4. Prefacio, resumen, abstract, agradecimientos y palabras clave.

Existen varios elementos previos al texto como son el abstract, el resumen, los agradecimientos y el prefacio. Suele ser habitual que todos ellos sean obligatorios salvo el prefacio. Sin embargo no aparecerán los que no se indiquen usando los comandos correspondientes. Todos ellos deberán escribirse en ficheros aparte y el parámetro de los comandos será el fichero (incluyendo el *path* relativo) sin el ‘.tex’ del final. Los comandos son los siguientes:

- `\prefacefile{fichero}` Prefacio.
- `\resumenfile{fichero}` Resumen en castellano.
- `\abstractfile{fichero}` Resumen en inglés.
- `\ackfile{fichero}` Agradecimientos.

Todos estos comandos deben usarse en el preámbulo, antes de `\begin{document}`.

Así mismo se deben usar otros dos comandos para introducir las palabras clave al final del resumen y del abstract. Ambos comandos tienen un sólo parámetro y son las palabras clave separadas por comas. Estos comandos son `\palabrasclave{}` y `\keywords{}`. Estos comandos deben usarse al final del fichero con el resumen o con el abstract según corresponda.

10.5. Partes, capítulos, apartados, subapartados, subsubapartados, párrafos y subpárrafos

Las partes, capítulos, apartados, subapartados y demás elementos de este tipo pueden usarse de dos formas, una es la tradicional de L^AT_EX 2_E con un parámetro opcional que se corresponde con el título de la parte, capítulo o sección que aparecerá en el índice y en las cabeceras o pies de página. El primer parámetro obligatorio será el nombre completo de la parte, capítulo o sección. El parámetro opcional sólo se utilizará si el nombre completo es demasiado largo.

La segunda forma de usarlo es idéntica a la tradicional pero añadiendo un segundo parámetro obligatorio en el que se pondrá el nombre del fichero (sin ‘.tex’) en el que esté el texto de esa parte, capítulo, sección, etc.

De forma resumida los comandos completos, incluyendo el segundo parámetro obligatorio son en orden decreciente:

\part[shorttitle]{title}{file} Partes.

\chapter[shorttitle]{title}{file} Capítulos.

\section[shorttitle]{title}{file} Apartados.

\subsection[shorttitle]{title}{file} Subapartados (normalmente no se presenta en el índice si no se utiliza la opción de índice extendido).

\subsubsection[shorttitle]{title}{file} Subsubapartados (no se presenta en el índice si no se utiliza la opción de índice completo).

\paragraph[shorttitle]{title}{file} Párrafos (no se presenta en el índice).

\ subparagraph[shorttitle]{title}{file} Subpárrafos (no se presenta en el índice).

10.6. Glosario, acrónimos y definiciones

Tres elementos que también están disponibles para los autores son el glosario, la lista de acrónimos y la lista de definiciones.

10.6.1. Glosario

Para realizar el glosario y simplificar su creación se han diseñado seis comandos. En todos ellos el primer parámetro es opcional (por tanto si se indica debe hacerse entre corchetes) y representa el elemento referenciado al estilo *see also*. Los comandos que empiezan por Ph (phantom), introducen la palabra en el índice pero sin escribirla en el texto mientras que los que no empiezan por Ph también escriben la palabra en el texto. La combinación de ambos comandos es imprescindible porque el diseño del glosario es muy crítico con la tipología y la misma expresión o palabra con un espacio de más o de menos o una letra en mayúscula o sin mayúscula hacen que haya entradas distintas en el glosario.

\Index[]{}{} El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra a indexar.

\Subindex[]{}{}{} El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra sobre la que se indexa y el tercero la palabra a indexar.

\Subsubindex[]{}{}{}{} El primer parámetro es opcional y se corresponde con el *see also*. El segundo y tercer parámetro es el punto de indexación y el cuarto la palabra a indexar.

\PhIndex[]{}{} El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra a indexar.

\PhSubindex[]{}{} El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra sobre la que se indexa y el tercero la palabra a indexar.

\PhSubsubindex[]{}{}{} El primer parámetro es opcional y se corresponde con el *see also*. El segundo y tercer parámetro es el punto de indexación y el cuarto la palabra a indexar.

10.6.2. Acronimos

Para definir un nuevo acrónimo se puede hacer en cualquier lugar del texto. Así, lo normal, es realizar la definición del acrónimo donde se use por primera vez, dicha definición será añadida a la sección de definiciones al final del texto. Para realizar la definición hay que utilizar el comando **\newacronym{label}{acron}{extended}** donde label es la etiqueta para hacer referencia al acrónimo, acron es el acrónimo en si mismo y extended es lo que significa el acrónimo.

Para hacer referencia a los acrónimos se pueden utilizar las siguientes funciones:

\ac{label} La primera vez que se use el acrónimo en el texto aparecerá en su forma extendida y entre paréntesis el acrónimo.

\acs{label} Se presenta el acrónimo.

\acl{label} Se presenta la forma extendida del acrónimo.

Un ejemplo es por ejemplo la definición de Institute of Electrical and Electronics Engineers (IEEE), que si repito aparece sólo como IEEE o puedo utilizar la forma extendida de esta forma: Institute of Electrical and Electronics Engineers. El formato corto siempre se presentará así: IEEE. Para ver ejemplos de su uso lo mejor es ver los fuentes de este manual justo en este mismo punto.

10.6.3. Definiciones

Las definiciones se realizar de forma similar a los acrónimos. La diferencia está en los comandos utilizados. En este caso el comando tiene un parámetro más en el que se introduce el elemento definido en plural y el comando a utilizar es **\newdefinition{label}{defined}{plural}{extended}**. Los comandos para referenciar la definición serán:

\dfn{label} Se pone la palabra definida.

\dfnpl{label} Se pone la palabra definida en plural.

\Dfn{label} Se pone la palabra definida en mayúscula.

\Dfnpl{label} Se pone la palabra definida en plural y mayúsucula.

Es importante no poner un punto al final de la definición dado que se añade automáticamente al final de las definiciones.

Al igual que con los acrónimos voy a realizar lo mismo con las definiciones incluida la definición de acrónimo. Para ver cómo se usan es importante editar los fuentes de este documento.

10.7. Referencias

Para referenciar elementos etiquetados en el documento se pueden utilizar cualquiera de los siguientes comandos:

\ref{label} Indica la numeración del elemento indicado por la etiqueta.

\pageref{label} Es idéntico a \ref pero indicando el número de la página.

\cref, \Cref, \crefrange Comandos del paquete cleveref. En el caso de usarlos en su versión con * previenen de que sean hiperreferenciados.

Vamos a ver el uso de algunos de estos comandos en este párrafo. Como en todos los casos, se dispone de los fuentes para ver cómo se usan. Como ejemplo podemos referenciar a la sección 10.5 o sin poner a mano lo de sección se puede referenciar como apartado 10.5 o, para finalizar, indicando que está en la página 45.

El uso de los comandos del paquete cleveref pueden verse en su manual, sin embargo no es necesario que se busque dado que se dispone de un enlace a este manual en el apéndice C.

10.8. Bibliografía

Para la creación de la bibliografía se ha elegido el uso de BibTex dado que es el que permite más versatilidad a la hora de crear bibliografía.

La bibliografía siempre aparecerá al final del documento aunque es necesario configurar su funcionamiento. Para ello se utilizará el comando **\bibliographyconfig[estilo]{fichero}** que tiene un parámetro opcional que es el estilo de la bibliografía y el parámetro obligatorio es el nombre del fichero.bib en el que está la bibliografía en formato BibTex obviando la extensión de este fichero. El estilo por defecto es el siam. Por ejemplo se debe citar con el comando **\cite{label}** como en este ejemoplo de cita [6] en el que se usa la etiqueta Narendra1990 que es la etiqueta asociada al artículo en el fichero .bib correspondiente.

En la mayoría de las entradas de la base de datos se puede introducir un campo **note**. Si en ese comando se indica algo como: **note = “\href{file://unfichero.pdf}{Leer}”** se puede realizar el hiperenlace al documento correspondiente almacenado junto al documento creado. También se pueden realizar enlaces a páginas web [7].

PRIMEROS PASOS

Este capítulo está dedicado a aquellos que no tienen experiencia en diseñar documentos utilizando $\text{\LaTeX} 2_{\varepsilon}$.

11.1. Estructurar el documento

A la hora de crear un documento es muy importante organizar correctamente todos los archivos que van a contener las distintas partes del documento. En este capítulo se va a presentar cómo está organizado $\text{\LaTeX} 2_{\varepsilon}$ y cómo debe organizarse correctamente un documento con este estilo.

11.1.1. Organización de $\text{\LaTeX} 2_{\varepsilon}$

Un documento de $\text{\LaTeX} 2_{\varepsilon}$ comienza siempre por comando `\documentclass` aunque previamente puede haber comentarios previos. En ese comando siempre es necesario indicar el estilo que se va a utilizar y si es necesario las opciones modificadoras del estilo. Para este estilo y poniendo algunas opciones a modo de ejemplo podría ser: `\documentclass[epsbased,lof,loc]{tfgtfmthesisuam}`.

Tras este comando tenemos todos los comandos que deben presentarse en el preámbulo como se indica en los distintos capítulos y apartados.

Y para finalizar, el documento en sí mismo debe estar dentro del entorno `document`, es decir, el documento empezará con un `\begin{document}` y terminará con un `\end{document}`.

Como indica en otros capítulos hay una gran cantidad de variables del documento que deben estar en el preámbulo, es decir, antes de declarar el principio del documento con `\begin{document}`.

11.1.2. Organización de la documentación

Al declarar cada capítulo, apartado o subapartado el segundo parámetro determina el fichero que será utilizado para ese capítulo, apartado, etc. Por ello se recomienda organizar los distintos capítulos

o apartados en subdirectorios de forma adecuada. En este caso el nombre del fichero que se utilizará estará precedido del *path* relativo de dicho fichero al fichero principal. También puede estar precedido de un *path* absoluto. Se puede ver a modo de ejemplo cómo está estructurado este documento.

Además hay algunas funciones importantes que permiten poner los logos, imágenes, código o datos que van a ser usados en el documento en distintos directorios. Estas funciones son:

\codesdir {} Directorio donde se pondrán los códigos que se van a utilizar. Las funciones para incluir código admiten a su vez en el nombre del fichero subdirectorios de este directorio.

\logosdir {} Directorio donde se pondrán los logos que se van a utilizar. Las funciones para incluir logos admiten a su vez en el nombre del fichero subdirectorios de este directorio. Este directorio no debe indicarse si se ha instalado este estilo como parte del sistema operativo.

\graphicsdir {} Directorio donde se pondrán las imágenes que se van a utilizar. Las funciones para incluir imágenes admiten a su vez en el nombre del fichero subdirectorios de este directorio.

\datadir {} Directorio donde se pondrán los datos para ser graficados. Las funciones para crear gáficas **no** admiten en el nombre del fichero subdirectorios de este directorio.

La función para determinar dónde están los logos no debe ser usada si este estilo ha sido instalado en el sistema operativo, sin embargo sí es necesario si este estilo es usado en un directorio de usuario y los logos se ponen en un directorio distinto del del documento principal. En cualquier caso si no se indica ninguno de estos directorios, todos estos elementos deberán estar en el mismo directorio donde está el documento principal que es el mismo donde debe realizarse la compilación.

11.1.3. Otros elementos de la estructura del documento

Cuando se compila el documento a PDF existe una serie de valores que aparecen en los metadatos de dicho fichero. Para controlar estos datos es necesario ejecutar la función **\pdfmetavalue**s que tiene cuatro parámetros obligatorios. El primero es el autor del documento, el segundo el título, el tercero el tipo de documento y el cuarto una serie de palabras clave separadas por comas. por ejemplo: **\pdfmetavalue{Eloy Anguiano Rey}{Manual de la clase LaTeXe tfgtfmthesisuam}{Tesis}{manual, tfgtfmthesisuam, TFG, TFM, Tesis}**.

De igual forma hay un recuadro en la portada en la que se puede poner información variada. Típicamente esta información será la escuela o facultad, la dirección, teléfono, email, etc. Para definir el contenido de este recuadro hay que utilizar la función **\coverdata** que sólo tiene un parámetro, el texto a incluir en ese recuadro en el que es necesario indicar cada final de línea con un ****.

11.2. Enlazar la bibliografía

El formato de bibliografía para este estilo siempre debe ser BIBTEX. Este formato de bibliografía puede crearse con múltiples programas o exportarse desde programas de gestión bibliográfica on line como Mendeley. Algo interesante del formato bibliográfico es que sólo se pondrán en la bibliografía los artículos libros o webs que hayan sido referenciados en el texto y lo hará en el orden correcto de acuerdo con el estilo de bibliografía elegido. Por tanto, en los programas de gestión bibliográfica basta con exportar a formato BIBTEX toda la bibliografía.

ELEMENTOS INTERNOS

En \LaTeX 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 12.8. 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.

12.1. 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 12.1: 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 12.2 puede verse un ejemplo

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



(a) Esta es una subfigura



(b) Esta es otra subfigura

Figura 12.2: 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 12.2(b)

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

12.1.1. Gráficas

La creación de gráficas científicas es uno de los 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.

12.1.2. 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 12.3 pueden verse los distintos esquemas.

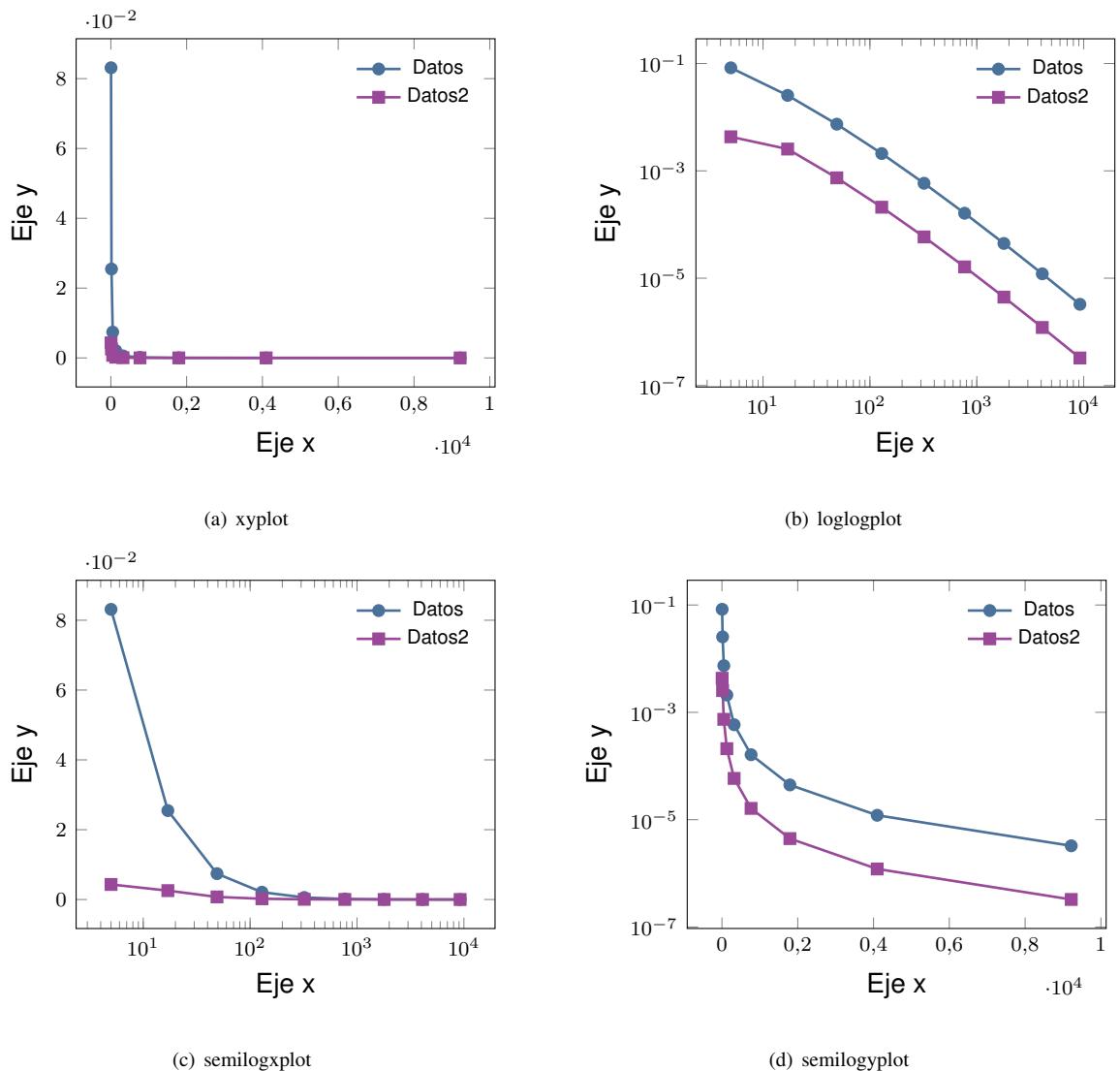


Figura 12.3: En esta figura se pueden ver los resultados de aplicar los distintos entornos de gráficas xy.

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] {expresión}{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.

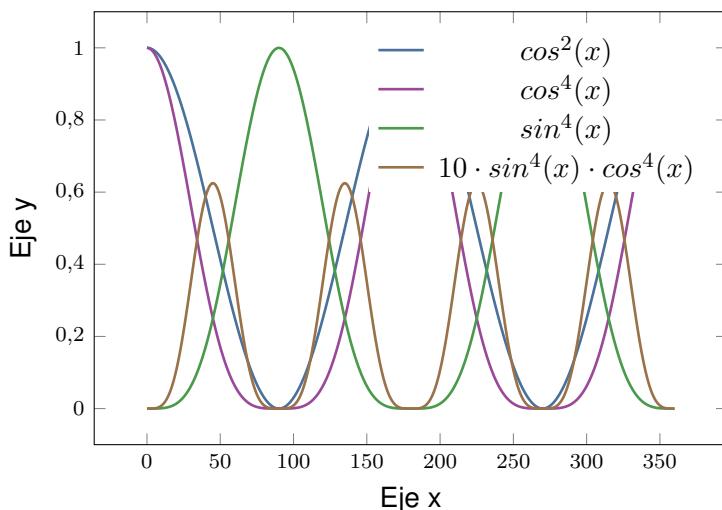


Figura 12.4: En esta figura se pueden ver los resultados de representar varias expresiones matemáticas.

12.1.3. 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 pdflatex 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 figura 12.5 y, por supuesto, puede verse el código en los fuentes de este documento.

Figura 12.5: En esta figura se pueden ver los resultados de aplicar los distintos entornos de gráficas xy.

12.1.4. 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 12.2.

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

12.1.5. Diagramas de Gantt

Los diagramas de Gantt pueden llegar a ser muy complejos, sin embargo se ha preparado una versión simple que permite 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 12.6 puede verse un ejemplo de uso cuyo fuente puede verse en los fuentes de este documento.

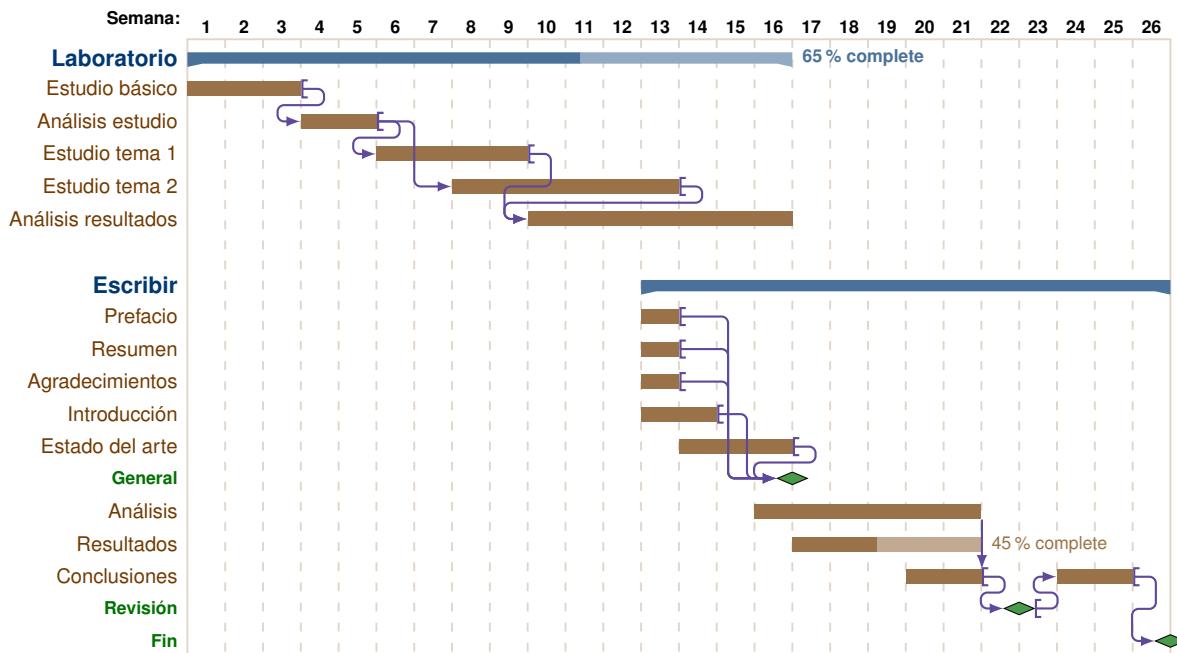


Figura 12.6: Diagrama de Gantt ejemplo en el que se muestran la mayoría de las posibilidades de un diagrama de Gantt con los comandos indicados en el texto.

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 primera 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.

12.2. Tablas

Para realizar una tabla se puede usar cualquiera de los entornos de L^AT_EX 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 12.1.

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 12.1: Esta es una tabla de ejemplo en la que, internamente, se usa el entorno `tabular`.

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 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 12.2.

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 12.2: 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.

12.2.1. 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 12.2.

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 **concept** 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 €

12.3. 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 12.1.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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 12.1: Este es un cuadro de texto en el que usando el paquete lipsum se genera el texto internamente

12.4. 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 12.1. En la que puede verse que la ecuación aparece recuadrada. Si se quiere que aparezca recuadrada es necesario usar el comando **\boxed**.

$$\sum c_{ij} = \frac{a}{\int adx} \quad (12.1)$$

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 12.2a y 12.2b son un ejemplo de ello.

$$\sum c_{ij} = \frac{a}{\int adx} \quad (12.2a)$$

$$\sum c_{ij} = \frac{a}{\int adx} \quad (12.2b)$$

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

12.5. 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 significan lo siguiente:

- 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[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código ADA
- `\ASMCODE[label]{short cap}{long caption}{file}{init num}{end num}{first num}` Código ensamblador de Intel.
- `\ASMMotCode[label]{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 partirllo 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 apartado 12.5 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 apartado 12.5

Código 12.1: 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_ErrAlreadyRegistered(char **command, char *prefix, char *nick)
2 {
3     *command=0;
4     if(nick == NULL) return IRCERR_NONICK;
5     _CPM_(command,prefix,COM_ERR_ALREADYREGISTERED,nick,"Unauthorized command"
6         (already registered));
7     return IRC_OK;
8 }
```

Código 12.2: 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)
```

12.6. 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 algoritmo 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        $up \leftarrow \text{FindCompress}(Im[i - 1, ]);$ 
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 12.1: 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.

12.7. 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.

12.7.1. Listas tradicionales de L^AT_EX 2_&

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.

12.7.2. 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.

Objetivo 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.

12.8. Referencias internas e hiperenlaces

Todos los enlaces internos se realizan utilizando el comando de \LaTeX `\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.

13.1. 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 para 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. L^AT_EX 2_E es extremadamente preciso en la creación de textos y muchos *warnings* no es necesario resolverlos pero sí hay que revisarlos todos.

13.2. 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 cuadro de texto 13.1.

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

```
% arara: clean: {files: [tfgtfmthesisuam.aux, tfgtfmthesisuam.idx, tfgtfmthesisuam.ilg, tfgtfmthesisuam.ind, tfgtfmthesisuam.bbl, tfgtfmthesisuam.bcf, tfgtfmthesisuam.blg, tfgtfmthesisuam.run.xml, tfgtfmthesisuam.fdb_latexmk, tfgtfmthesisuam.fl, tfgtfmthesisuam.loe, tfgtfmthesisuam.lof, tfgtfmthesisuam.lol, tfgtfmthesisuam.lot, tfgtfmthesisuam.lt, tfgtfmthesisuam.out, tfgtfmthesisuam.toc, tfgtfmthesisuam.upa, tfgtfmthesisuam.upb, tfgtfmthesisuam.acn, tfgtfmthesisuam.acr, tfgtfmthesisuam.alg, tfgtfmthesisuam.glg, tfgtfmthesisuam.glo, tfgtfmthesisuam.gls, tfgtfmthesisuam.glsdefs, tfgtfmthesisuam.idx, tfgtfmthesisuam.ilg, tfgtfmthesisuam.xdy, tfgtfmthesisuam.loa, tfgtfmthesisuam.gnuploterrors , tfgtfmthesisuam.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, tfgtfmthesisuam.ilg, tfgtfmthesisuam.ind, tfgtfmthesisuam.bbl, tfgtfmthesisuam.bcf, tfgtfmthesisuam.blg, tfgtfmthesisuam.run.xml, tfgtfmthesisuam.fdb_latexmk, tfgtfmthesisuam.fl, tfgtfmthesisuam.loe, tfgtfmthesisuam.lof, tfgtfmthesisuam.lol, tfgtfmthesisuam.lot, tfgtfmthesisuam.lt, tfgtfmthesisuam.out, tfgtfmthesisuam.toc, tfgtfmthesisuam.upa, tfgtfmthesisuam.upb, tfgtfmthesisuam.acn, tfgtfmthesisuam.acr, tfgtfmthesisuam.alg, tfgtfmthesisuam.glg, tfgtfmthesisuam.glo, tfgtfmthesisuam.gls, tfgtfmthesisuam.glsdefs, tfgtfmthesisuam.idx, tfgtfmthesisuam.ilg, tfgtfmthesisuam.xdy, tfgtfmthesisuam.loa, tfgtfmthesisuam.gnuploterrors , tfgtfmthesisuam.mw]}
```

Cuadro 13.1: 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.

13.3. 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.

13.4. 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 cuadro de texto 13.1.

BIBLIOGRAFÍA

- [1] InvenSense, “MPU9250 Product Specification Revision 1.1,” 2016.
- [2] T. O. J. T. Graves, D. R. Wilkins, and M. Corr, “COPY OF A LETTER FROM SIR WILLIAM R . HAMILTON On Quaternions ; or on a new System of Imaginaries in Algebra . By Sir,” *Philosophical Magazine*, vol. 25, no. 1844, 1999.
- [3] Oscar Serrano, “Variador electrónico (ESC): Qué es y cómo funciona | FpvMax.”
- [4] A. Industries, “PCA9685 Adafruit.”
- [5] Eric, “No Title,” 2017.
- [6] K.S.Narendra and K.Parthsarathy, “Identification and Control of Dynamical System using Neural Networks,” *IEENN*, vol. 1, no. 1, pp. 4–27, 1990.
- [7] R. W. Zurek and L. J. Martin, “Interannual Variability {OF} planet-encircling dust activity on {M}ars,” *J. Geophys. Res.*, vol. 98, no. E2, pp. 3247–3259, 1993.

DEFINICIONES

acrónimo Sigla cuya configuración permite su pronunciación como una palabra; por ejemplo, ovni: objeto volador no identificado; TIC, tecnologías de la información y la comunicación.

definición Proposición que expone con claridad y exactitud los caracteres genéricos y diferenciales de algo material o inmaterial.

opción de estilo Son los valores que modifican el funcionamiento del estilo. Se ponen entre corchetes y separadas por comas en el comando \documentclass y antes de el nombre del estilo que irá entre llaves.

ACRÓNIMOS

IEEE Institute of Electrical and Electronics Engineers.

WYSIWYG What You See Is What You Get.

WYTIWYG What You Think Is What You Get.

APÉNDICES

WORD® vs. LATEX 2 ε

A.1. Ventajas e inconvenientes de LATEX 2 ε

El gusto por el LATEX 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 limitados.

El principal inconveniente de LATEX 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 LATEX 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 LATEX 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 LATEX 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 LATEX 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 perdida 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 párrados, interlineados, encabezamientos y estructura del docuemnto debe coincidir con el aquí presentado. Dado que sólo se aporta el estilo de LATEX 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 ella 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 validarla 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:textbackslashlocaltexmf 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 elige 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-fnbd`

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/XXXX donde XXXX 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 XXXX** 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 las 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 $\text{\LaTeX} 2_{\varepsilon}$ e incluso algunos entornos de desarrollo integrados como eclipse o netbns 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 porque 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	
\advisor	{tutor}
\author	{título}
\coadvisor	{cotutor}
\copyrightdate	{fecha}
\faculty	{facultad/escuela}
\levelin	{titulacion}
\title	[título corto] {título}
\speaker	{ponente}

Decoraciones	
\coverdata	{texto}
\facultylogo	{fichero}
\facultylogowide	{dimension}

Directorio	
\codesdir	{directorio}
\datadir	{directorio}
\graphicsdir	{directorio}
\logosdir	{directorio}

Prefacio	
\abstractfile	{fichero}
\ackfile	{fichero}
\dedication	{dedicatoria}
\famouscite	{cita}
\keywords	{palabras}
\palabrasclave	{palabras}
\prefacefile	{fichero}
\privateaddress	{direccion}
\resumenfile	{fichero}

E.2. Comandos en el cuerpo del texto

Código		
\Code	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}{lenguaje}
\AdaCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\ASMCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\ASMMotorolaCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\CCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\CPPCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\CSharpCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\GnuplotCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\HaskellCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\HTMLCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\JavaCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\LaTeXCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\LispCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\MakeCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\MathematicaCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\MatlabCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\OctaveCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\PascalCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\PerlCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\PHPCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\PythonCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\RCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\RubyCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\ScilabCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\SQLCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\VHDLCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
\XMLCode	[etiqueta]	{pie corto}{pie largo}{fichero}{línea inicial}{línea final}{numeración inicial}
Ecuaciones		
\boxed		{ecuación}
Estructura		
\cleardoublepage		
\part	[título corto]	{título}{fichero}
\part	[título corto]	{título}
\chapter	[título corto]	{título}{fichero}
\chapter	[título corto]	{título}
\section	[título corto]	{título}{fichero}
\section	[título corto]	{título}
\subsection	[título corto]	{título}{fichero}
\subsection	[título corto]	{título}
\subsubsection	[título corto]	{título}{fichero}
\subsubsection	[título corto]	{título}
\subsubsubsection	[título corto]	{título}
\subsubsubsection	[título corto]	{título}
\paragraph	[título corto]	{título}{fichero}
\paragraph	[título corto]	{título}
\ subparagraph	[título corto]	{título}{fichero}
\ subparagraph	[título corto]	{título}
Figuras y tablas		
\subfigure	[etiqueta]	{pie}{contenido}
\subtable	[etiqueta]	{pie}{contenido}
Gantt		
\milestone	[etiqueta enlace]	{etiqueta gantt}{tiempo}
\taskbar	[etiqueta enlace][porcentaje fin]	{etiqueta gantt}{tiempo inicio}{tiempo fin}
\taskgroup	[etiqueta enlace][porcentaje fin]	{etiqueta gantt}{tiempo inicio}{tiempo fin}
\FtoFlink		{etiqueta inicio}{etiqueta fin}
\FtoSlink		{etiqueta inicio}{etiqueta fin}
\StoSlink		{etiqueta inicio}{etiqueta fin}

Gráficas		
\plotlined		{fichero datos}{título datos}
\plotdashed		{fichero datos}{título datos}
\plotdata		{fichero datos}{título datos}
\plotfunction		{fichero datos}{título datos}

Imágenes		
\image		{ancho}{alto}{fichero}
\imageL	ancho	fichero

Presupuestos		
\budgettitle		{título presupuesto}
\concept		{título}{precio unitario}{cantidad}{coste total}
\separator		
\subconcept		{título}{precio unitario}{cantidad}{coste total}
\subtotal		{subtotal}
\total		{total presupuesto}

Texto citado		
\onlinecitation		{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		
algorithm	[pie corto]	{etiqueta}{pie completo}
algorithmN	[pie corto]	{etiqueta}{pie completo}
budget		
equation	[etiqueta]	{título}
figure	[pie corto]	{etiqueta}{pie completo}
gantt		{tiempo inicio}{tiempo fin}
largecitation		{autor}
multiequation		
table	[pie corto]	{etiqueta}{pie completo}
textbox	[pie corto]	{etiqueta}{pie completo}

Listados especiales		
functionality		
objetive		
functional		
nonfunctional		
simplelist		

Gráficas		
gnuplot	[opciones]	
loglogplot	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}
semilogxplot	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}
semilogyplot	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}
xyplot	[posicion leyenda]	{título}{título eje x}{título eje y}{ancho}{alto}

ÍNDICE TERMINOLÓGICO

`budgettitle`, 46

colores, 24

predefinidos, 24

`eigenvalue`, 68

opciones, 75

