**MEMORIA FINAL DEL PROYECTO**

*Desarrollo de Aplicación Multiplataforma para Speedcubing*

**CICLO FORMATIVO DE GRADO SUPERIOR**

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

CURSO 2023-2024

**AUTORES:**

**Sergio Misas Vírseda – Emma Fernández Barranco**

**TUTOR:**

**José Carlos Núñez**

**DEPARTAMENTO DE INFORMÁTICA Y COMUNICACIONES**

**I.E.S. LUIS VIVES**

Resumen

Desarrollo de Aplicación Multiplataforma para Speedcubing

Aplicación multiplataforma de cronometrado de resoluciones de speedcubing.

Cross-platform Speedcubing Application Development

Cross-platform timer for speedcubing resolutions.

Índice de contenidos

Resumen 3

Índice de contenidos 4

Índice de figuras 6

Índice de tablas 7

Capítulo 1 - Introducción 1

1.1 Justificación del proyecto 1

1.2 Objetivos 1

1.3 Alcance 1

Capítulo 2 - Desarrollo del proyecto 3

2.1 Investigación de tecnologías 3

2.2 Diseño y planificación de la aplicación 3

2.2.1 Requisitos funcionales, no funcionales y de información 4

2.2.2 Diagrama de clases 7

2.3 Aprendizaje de Jetpack Compose 8

2.4 Implementación 9

Capítulo 3 - Capítulos varios 11

3.1 Secciones de un capítulo (Título 2) 11

3.1.1 Subsecciones (Título 3) 11

3.2 Listas numeradas y con viñetas 11

3.3 Figuras y tablas 11

Capítulo 4 - Conclusiones y trabajo futuro 13

Bibliografía 15

Apéndices 17

Índice de figuras

[Figura 3‑1. Ejemplo de figura 6](file:///C:\Users\crust\Downloads\TFM-TFG-Word-v1(2019).docx#_Toc11912417)

Índice de tablas

[Tabla 3‑1. Ejemplo de tabla 6](file:///C:\Users\crust\Downloads\TFM-TFG-Word-v1(2019).docx#_Toc11912424)

# Introducción

## Justificación del proyecto

El tema elegido es el desarrollo de una aplicación para asistir en el desempeño del speedcubing, que consiste en resolver cubos de Rubik (o puzles similares) en el menor tiempo posible. Dentro de la comunidad del speedcubing existen distintas herramientas como csTimer (cronómetro online) o TwistyTimer (cronómetro para Android). Sin embargo, no existe ninguna que soporte una mayor variedad de plataformas de forma cómoda de usar.

## Objetivos

El principal objetivo del proyecto es desarrollar una aplicación que realice las funciones necesarias para el speedcubing (cronómetro, generación de mezclas, estadísticas, algoritmos, datos de competiciones oficiales) de forma centralizada y disponible en múltiples plataformas. Para ello hemos decidido usar Kotlin Multiplatform para la propia aplicación y un sistema de almacenamiento online como Firebase para sincronizar los datos entre los distintos dispositivos.

De esta forma, esperamos adquirir conocimientos más profundos sobre Kotlin, más específicamente la vertiente multiplataforma, así como entender el proceso de desarrollo y publicación del software en las distintas plataformas.

## Alcance

Las funcionalidades mínimas de la aplicación serán las siguientes:

• Debe ser multiplataforma (mínimo Android y escritorio).

• Implementa un cronómetro para registrar el tiempo de resolución del puzle.

• Habilita la modificación de cada resultado para aplicar distintas penalizaciones (+2, DNF)

• Genera mezclas válidas para las categorías reconocidas como oficiales.

• Permite almacenar tiempos para cada categoría de forma independiente, así como la creación de

nuevas categorías.

• Sincroniza los tiempos almacenados entre los distintos dispositivos.

• Genera estadísticas relevantes (ao5, ao12, récords personales).

Adicionalmente, se incluirán las siguientes funciones opcionales para complementar los requisitos

mínimos:

• Soporte para más plataformas (web, iOS)

• Generación de imágenes de mezclas

• Base de datos de algoritmos comunes

• Listado de resultados de competiciones oficiales

# Desarrollo del proyecto

## Investigación de tecnologías

Como está escrito en el anterior capítulo, para poder realizar el proyecto se debe escoger lenguaje, IDE y otras tecnologías. Dado que en el grado se ha utilizado Kotlin durante la totalidad del primer curso y en la asignatura de móviles del presente, se ha decidido utilizar este lenguaje para desarrollar el proyecto.

Es de importancia señalar que como es necesario que la aplicación sea multiplataforma vamos a utilizar la novedosa tecnología de Kotlin Multiplatform para poder desarrollar en todas plataformas y dispositivos. Para desarrollar el código de la aplicación con Kotlin Multiplatform es necesario utilizar Android Studio como IDE.

Para el guardado de información en la nube se utilizarán 2 aplicaciones de Google Firebase:

* Authentication: Guardado de cuentas para los usuarios.
* Firestore: Base de datos NoSQL para guarda todos los datos pertenecientes a los usuarios.

Se utiliza Google Firebase ya que es una tecnología que se ha utilizado en la asignatura de Móviles y es gratuita hasta tener una afluencia de tráfico grande.

Para control de versiones se utilizará Github, usando Gitflow como flujo de trabajo para llevar a cabo las tareas y su distribución entre los miembros del equipo.

Entre las tecnologías que se han barajado hay algunas que no han llegado a ser seleccionadas:

* MAUI: No permite hacer aplicaciones en Linux.
* Flutter: Kotlin es un lenguaje que al que se están acostumbrados los miembros del equipo, al contrario que Dart.
* Fleet: Este IDE de JetBrains no tiene la funcionalidad necesaria en este momento para adaptarse al flujo de trabajo del equipo con comodidad y rapidez.
* Google Drive: Se consideró como almacenamiento de datos pero Firebase gestiona autenticación y almacenamiento en un solo servicio.

## Diseño y planificación de la aplicación

Antes de empezar a codificar la aplicación se debe hacer un diseño previo, en este caso se harán requisitos y diagramas de clase de la aplicación.

### Requisitos funcionales, no funcionales y de información

Se han dividido los requisitos según la vista que haya en la aplicación.

#### Vista de cronómetro

Esta es la vista principal, tiene por lo tanto la mayor cantidad de funcionalidades.

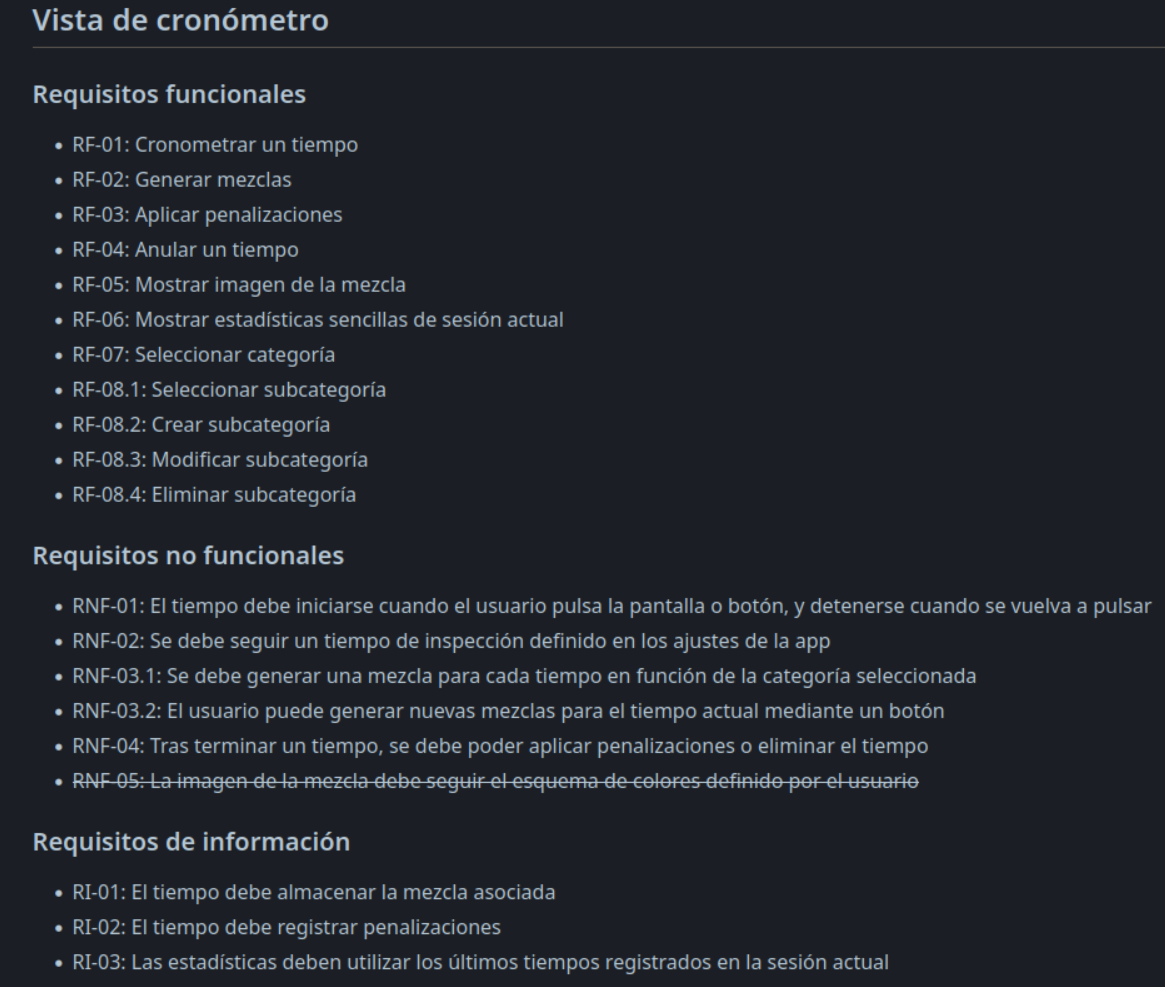


Figura 2.1 Requisitos de vista de cronómetro

#### Lista de tiempos

Esta vista permitirá visualizar y modificar tiempos.

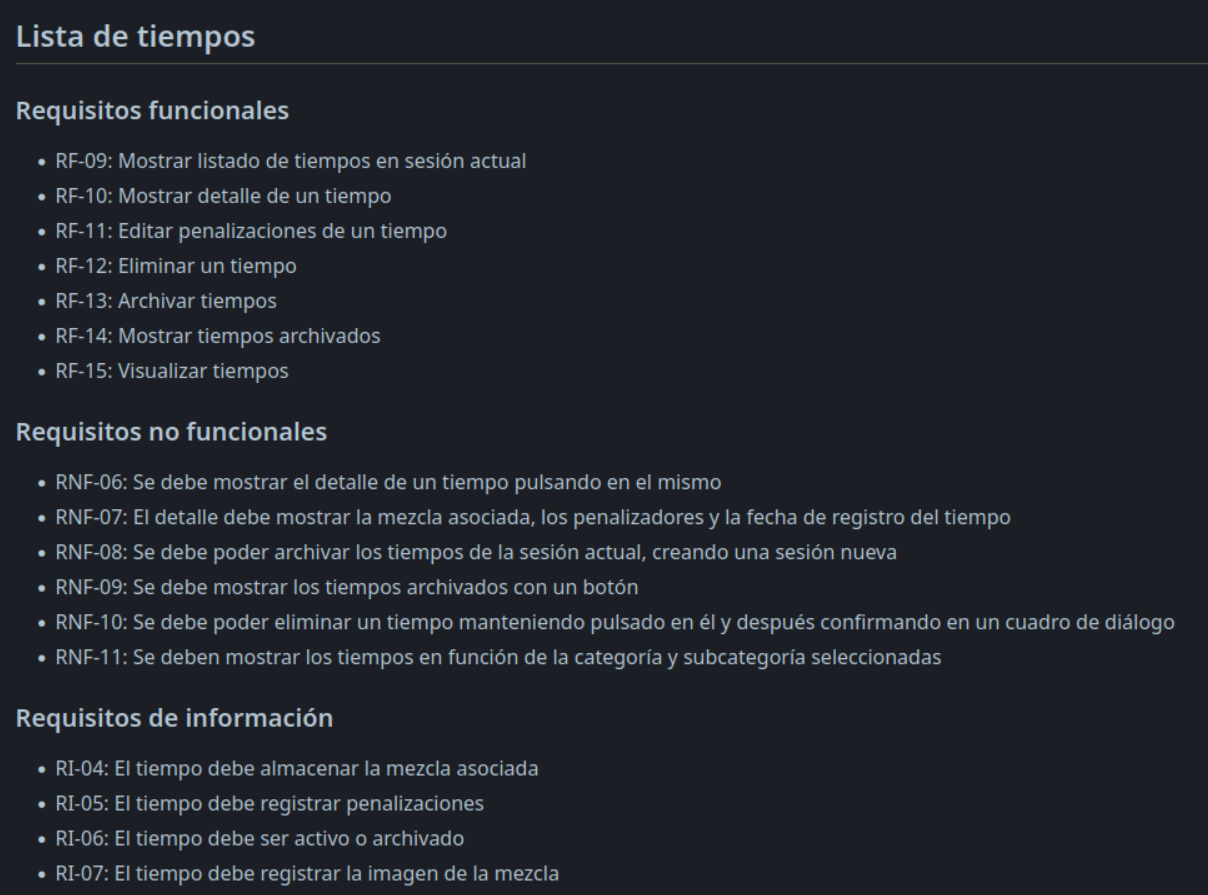


Figura 2.2 Requisitos de vista de listado de tiempo

#### Vista de ajustes

En esta vista se podrán cambiar los ajustes de la aplicación.



Figura 2.3 Requisitos de la vista de ajustes

### Diagrama de clases

Para organizar y definir las clases se realizó el siguiente diagrama de clases:

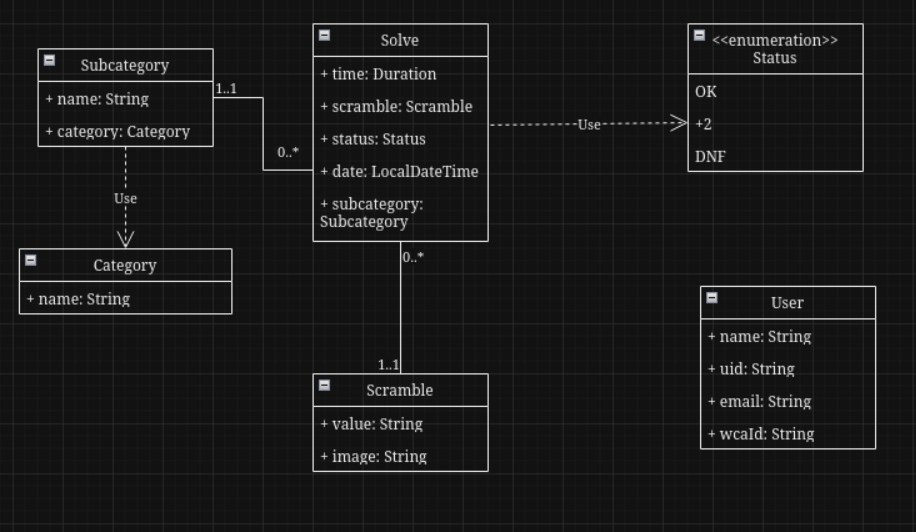


Figura 2.4 Diagrama de clases de la aplicación

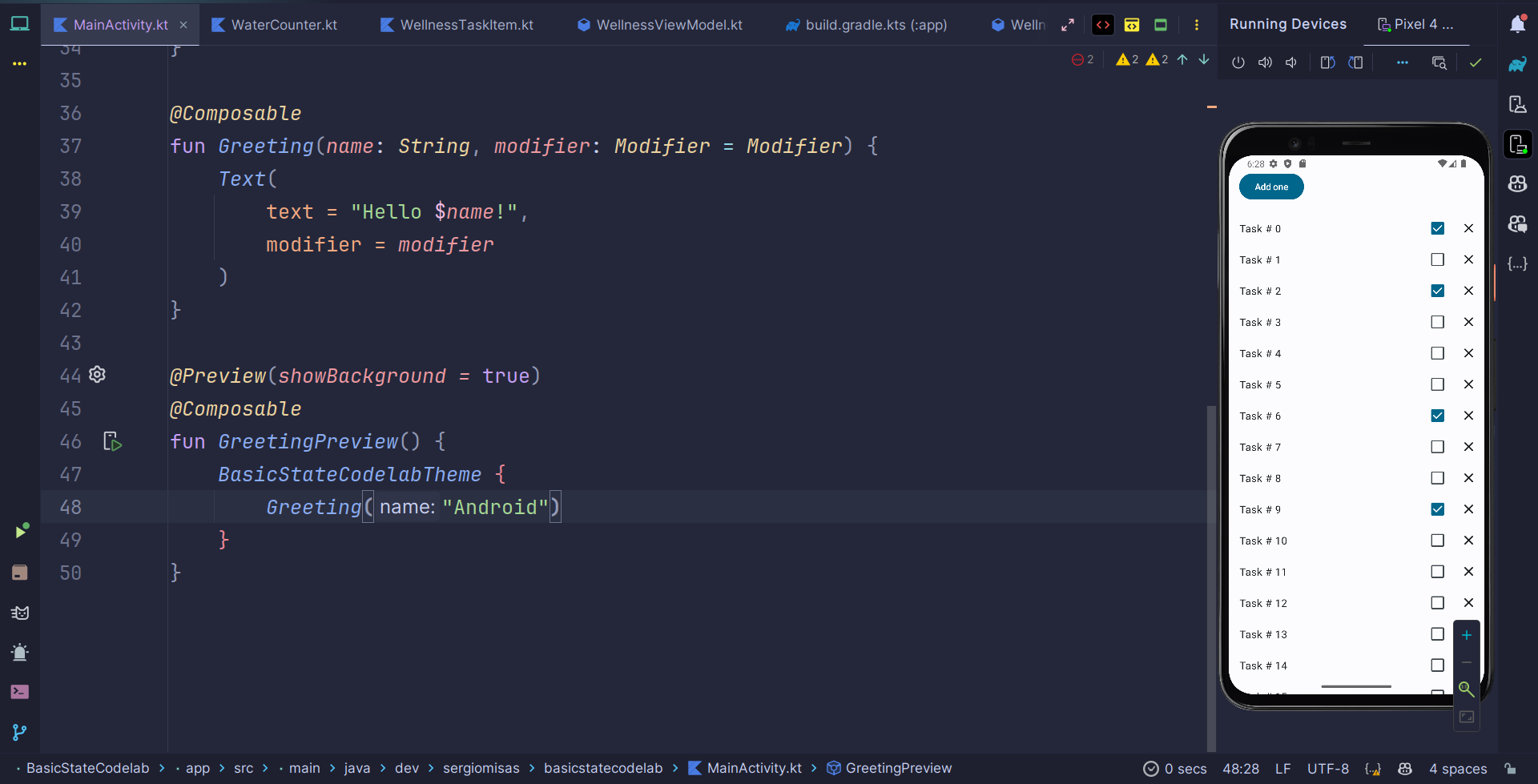
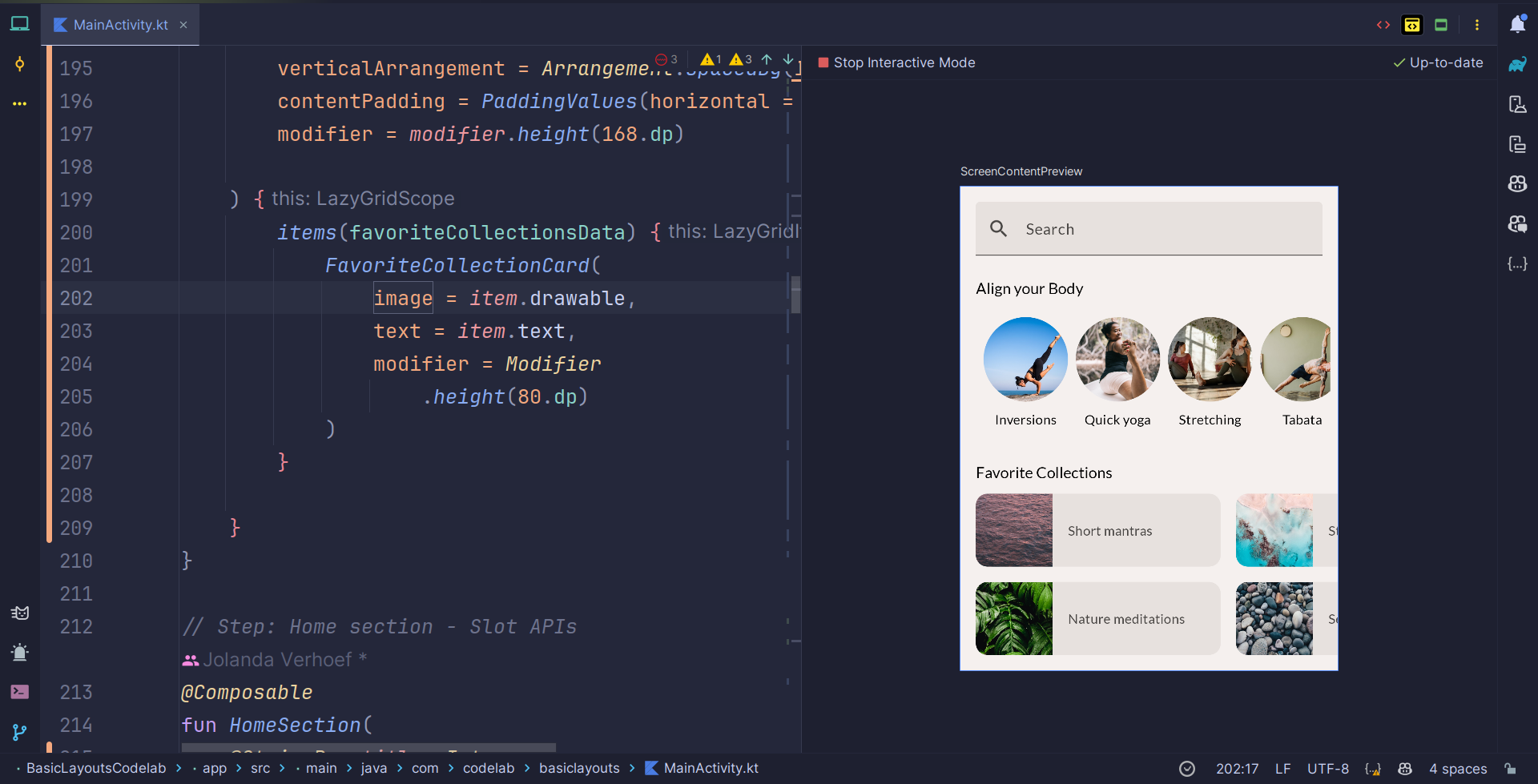
Se ha planteado así para que una solución (cuando se resuelve el cubo y se para el tiempo) que pertenece a una subcategoría (normal, a una mano, a ciegas) que a su vez esta tenga una categoría (el tipo de cubo, es decir, 3x3, 4x4, etc...). Una solución tiene un status que depende de si se ha completado de forma normal un cubo, se ha aplicado una penalización o no es válida.

También se ha creado una clase usuario que será personalizable en la parte de ajustes por si el usuario quiere cambiar el correo u otras posibilidades.

## Aprendizaje de Jetpack Compose

Antes de proceder a implementar la aplicación se han realizado varios cursos Codelab ofrecidos por Google Developers para formarse en Compose. Están más enfocados a móvil pero ya que Compose es multiplataforma servirán para el proyecto.

Principalmente los cursos realizados han formado al equipo en la formación de interfaces funcionales y que tengan un estado que se mantenga entre recomposiciones (giros de pantalla, deslizar hacia abajo en una vista, etc...)

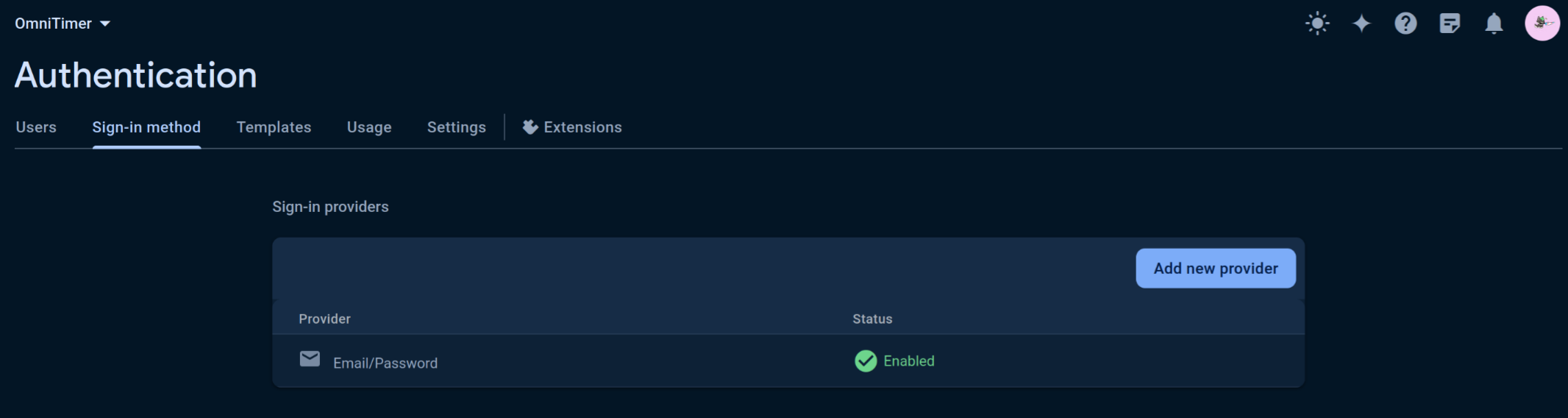
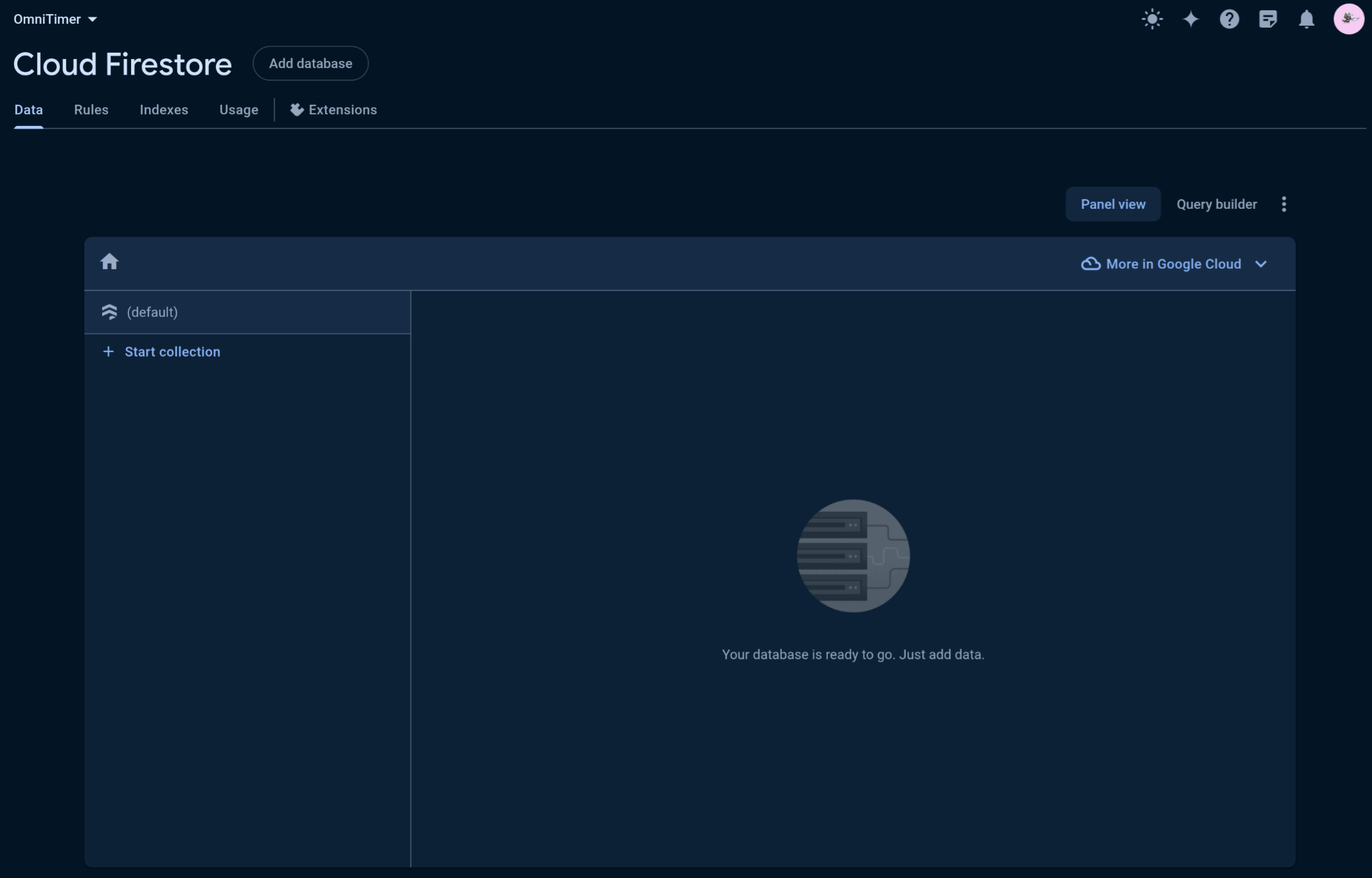


Figuras 2.5 y 2.6 Prácticas de Codelab

## Implementación

Primero, se procede a la creación de un proyecto de Firebase para usar sus servicios al conectarlo a la aplicación.

Estos servicios serán los siguientes:



Figuras 2.7 y 2.8 Servicios de Firebase

# Capítulos varios

Este es un ejemplo de capítulo. Cada capítulo comienza con el nombre del capítulo. Este párrafo ha de usar el estilo Título 1, con el fin de que aparezca automáticamente en el Índice de contenidos (el índice se puede actualizar en cualquier momento pulsando sobre él con el botón derecho y seleccionando la opción *Actualizar campos*.

Cada capítulo ha de terminar con un “*Salto de sección – Página impar*” con el fin de que cada capítulo comience siempre en la misma página (como si de un libro se tratase), dejando una página en blanco si fuese necesario.

## Secciones de un capítulo (Título 2)

Las secciones de un capítulo usan el estilo Título 2. Si necesitamos tener subsecciones entonces utilizaremos el estilo Título 3, como se ve a continuación.

### Subsecciones (Título 3)

Las subsecciones usan el estilo Título 3. Es posible que una subsección tenga también sub-subsecciones, como veremos a continuación.

#### Sub-subsecciones (Título 4)

Las sub-subsecciones utilizan el estilo Título 4. Éstas no aparecen en el Índice de contenidos. No se recomienda bajar más de nivel. El estilo Título 5 está reservado para los nombres de los capítulos en inglés, mientras que el resto de estilos de títulos (Título 6, Título 7…) se reservan para las secciones de los apéndices.

## Listas numeradas y con viñetas

Para hacer listas numeradas (como las que aparecen en la Introducción de este documento) se recomienda utilizar los estilos Lista con números y Lista con Viñetas, respectivamente.

En caso de listas con varios niveles se recomienda hacer uso de las opciones de creación de listas multinivel que proporciona Word y que ayudan a gestionar el aspecto y los estilos de numeración de cada uno de los niveles.

## Figuras y tablas

En la memoria es conveniente incorporar alguna figura explicativa que ayude a la comprensión del texto escrito. En la Figura 3‑1 se muestra un ejemplo.

Se recomienda insertar la imagen y, a continuación, pulsar con el botón derecho sobre ella y seleccionar la opción Insertar Título. De este modo, generaremos de manera el texto “Figura X-Y”, lo que permite que la numeración de las figuras sea gestionada de manera automática, que las figuras creadas aparezcan automáticamente en el Índice de figuras y que podamos hacer referencia a ellas desde el texto utilizando la opción *Referencia Cruzada* del menú *Insertar – Vínculos.* Para que la figura aparezca centrada se puede utilizar el estilo Descripción, que es el mismo que el usado en el pie de la figura.



Figura 3‑1. Ejemplo de figura

Del mismo modo, las tablas nos ayudan a mostrar ciertos resultados de nuestro trabajo. El estilo de la tabla depende de los contenidos de la misma. De todos modos, se ha creado un estilo Texto Tabla de referencia. Se puede ver un ejemplo de este estilo en la Tabla 3‑1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A-D | A | B | C | D |
| 1 | A1 | B1 | C1 | D1 |
| 2 | A2 | B2 | C2 | D2 |
| 3 | A3 | B3 | C3 | D3 |

Tabla 3‑1. Ejemplo de tabla

Al igual que con las figuras, es recomendable insertar un pie de tabla explicativo. El título de la tabla se añade con la opción *Insertar Título* del menú *Referencias – Títulos*. Al igual que antes, este título aparecerá con el estilo Descripción. Podremos hacer referencias a las tablas desde el texto utilizando para ello las Referencias cruzadas. Por último, todas las tablas aparecerán referenciadas automáticamente en el Índice de tablas.

# Conclusiones y trabajo futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de máster, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF. En la portada de la misma deberán figurar, como se ha señalado anteriormente, la convocatoria y la calificación obtenida. Asimismo, el estudiante también entregará todo el material que tenga concedido en préstamo a lo largo del curso.

Bibliografía

|  |  |
| --- | --- |
| [1] | L. A. Bucki, Word 2013 Bible, John Wiley & Sons, 2013. |
| [2] | CFI, «Cursos de Formación en Informática,» [En línea]. Available: http://cursosinformatica.ucm.es. [Último acceso: 01 06 2019]. |

Apéndices

###### Título del primer apéndice

Los apéndices son secciones al final del documento en los que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Para el título del apéndice se utiliza el estilo Título 6. Los títulos de los apéndices no aparecen en el Índice de contenidos. Las secciones de los apéndices usan el estilo Título 7, como se ve a continuación.

Sección del apéndice (Título 7)

Para los títulos de las secciones de los apéndices se utiliza el estilo Título 7. Para las subsecciones se usa el estilo Título 8, como aparece a continuación.

Subsección del apéndice (Título 8)

Para los títulos de las subsecciones de los apéndices se utiliza el estilo Título 8. Para las sub-subsecciones se usa el estilo Título 9, como aparece a continuación.

Sub-subsecciones del apéndice (Título 9)

Para las sub-subsecciones de un apéndice se utiliza el estilo Título 9. No se recomienda añadir más niveles de subsección.

###### Título del segundo apéndice

Se pueden añadir los apéndices que se consideren oportunos. Si se quieren separar por páginas entonces se recomienda, al igual que con los capítulos, y utilizar los “*Saltos de sección – página impar*”