



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Flutter**



Presentado por Samuel Casal Cantero  
en Universidad de Burgos — 15 de julio  
de 2020

Tutor: César García Osorio y Francisco Javier  
Diez Pastor







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. José Francisco Díez Pastor y D. César García Osorio. profesores del departamento de Ingeniería Informática. área de Lenguajes y Sistemas Informáticos. Expone:

Que el alumno D. Samuel Casal Cantero, con DNI 71301273p, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Flutter.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 15 de julio de 2020

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor





## Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## Descriptores

*Android, Flutter, Dart, VisualStudio Code, Github, ...*

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

*Android, Flutter, Dart, VisualStudio Code, Github, ...*



---

# Índice general

---

Índice general	III
Índice de figuras	IV
Índice de tablas	V
Introducción	1
1.1. Estructura de la memoria . . . . .	2
Objetivos del proyecto	3
2.1. Objetivos generales . . . . .	3
2.2. Objetivos técnicos . . . . .	3
2.3. Objetivos personales . . . . .	4
Conceptos teóricos	5
3.1. Framework . . . . .	5
3.2. Flutter . . . . .	8
Técnicas y herramientas	9
Aspectos relevantes del desarrollo del proyecto	11
Trabajos relacionados	13
Conclusiones y Líneas de trabajo futuras	15
Bibliografía	17

---

# Índice de figuras

---

3.1. Encuesta . . . . .	6
-------------------------	---

---

## Índice de tablas

---



---

# Introducción

---

A principios de diciembre de 2019, fue la primera de las reuniones con los tutores, a fin de explicar las ideas propias o de barajar la opción de hacer el trabajo de fin de grado en la empresa, ITCL, donde estaba cursando las prácticas extracurriculares. Entre mis ideas estaba hacer un traductor de jeroglíficos o crear una base de datos de los graffitis de la calle. Al final me decanté por hacerlo en la empresa, ya que vi opciones para ello.

La primera idea de proyecto era hacer un detector de moscas de la fruta. Este primer enfoque, no se llegó a iniciar, debido a que la empresa no estaba participando de una manera fluida y estaba en una fase muy temprana de desarrollo. El inicio de las conversaciones fue a mediados de enero del 2020, y a finales de febrero al no ver esperanzas, busqué otro trabajo entre los disponibles de la plataforma.

La segunda idea del proyecto, si que la comencé, dedicando numerosas horas al desarrollo e investigación, pero acabé dejándola de lado porque no me sentía motivado, ya que se reunieron muchos factores que me bloquearon.

Brevemente, consistía en la implementación de un formulario de login dentro de un nodo de Knime[?], con el fin de poder conectarse a la base de datos de Moodle, recogiendo los datos pertinentes, para hacer minería de datos, con las herramientas de la aplicación.

Finalmente a un mes de la entrega para la segunda convocatoria, vi la necesidad de hacerlo, decantandome por una aplicación en Android. Una gran carga de trabajo, en caso de hacer algo que tenga entidad suficiente como para aprobar, pero la idea era intentarlo. Uno de los pasos iniciales para comenzar a programar para estos dispositivos, fue buscar cuáles de los lenguajes de programación actuales son más interesantes para el desarrollo.

Pensé en varios candidatos pero finalmente me decante por Flutter, ya que está desarrollado y respaldado por Google.

## 1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

- **Introducción:** breve descripción del problema a resolver y la solución propuesta. Estructura de la memoria y listado de materiales adjuntos.
- **Objetivos del proyecto:** exposición de los objetivos que persigue el proyecto.
- **Conceptos teóricos:** explicación de los conceptos teóricos clave para el entendimiento de la aplicación.
- **Técnicas y herramientas:** listado de técnicas metodológicas y herramientas utilizadas para gestión y desarrollo del proyecto.
- **Aspectos relevantes del desarrollo:** exposición de aspectos destacables que tuvieron lugar durante la realización del proyecto.
- **Conclusiones y líneas de trabajo futuras:** conclusiones obtenidas tras la realización del proyecto y posibilidades de mejora o expansión de la solución aportada.

---

# Objetivos del proyecto

---

A continuación, se detallarán los objetivos que han motivado la realización de este proyecto así como los resultados que se desean conseguir.

## 2.1. Objetivos generales

- Desarrollar una aplicación para *smartphone*.
- Implementación y despliegue de la app en la tienda de Google.
- Hacer que los usuarios pasen un buen rato.
- Mostrar quien es el creador de la aplicación, con el fin de poder usar esta como herramienta o portfolio.
- Desarrollar juegos.

## 2.2. Objetivos técnicos

- Aprender una alternativa moderna a javascript mediante Dart.
- Comprender el funcionamiento de Flutter.
- Control de versiones con la herramienta GitHub, mediante comandos a través de GitBash.
- Generar documentación de todo el proceso en  $\text{\LaTeX}$ .

- Realizar una planificación mediante *Scrum* eficiente, a través de la herramienta ZenHub, integrada en GitHub.
- Comenzar a usar las herramientas telemáticas para las reuniones con los tutores del trabajo final de grado.
- Comunicación de la aplicación mediante WebServices.
- Conocer y aprender a usar las herramientas que proporciona el *Cloud Services* de Google.
- Diseñar la arquitectura de la aplicación.

## 2.3. Objetivos personales

- Adquirir el conocimiento necesario para desarrollar aplicaciones móviles y multiplataforma. Es decir, para tres entornos, Android, iOS y web.
- Aprobar el trabajo de fin de grado, ya que es un reto.
- Estudiar como generar documentación en  $\text{\LaTeX}$ .
- Aprender que sin esfuerzo no hay recompensa.
- Comprender el funcionamiento de la comunicación entre la una aplicación y los servicios en la nube.
- Reforzar los conocimientos adquiridos durante la carrera.
- Investigar diferentes herramientas para solventar los problemas que salgan.
- Adquirir las nociones necesarias para poder llevar proyectos.
- Conseguir manejarme en entornos de incertidumbre.



---

# Conceptos teóricos

---

La parte del proyecto más importante es el proceso de lanzar una aplicación Android desde cero, para ello se ha tenido que realizar una investigación para determinar que herramientas, lenguajes de programación, entornos de desarrollo son los más apropiados para un despliegue ágil.

Todo esto se sitúa en un mercado altamente competitivo, con una gran variabilidad (usuarios, empresas, desarrolladores ...) y con un constante cambio.

## 3.1. Framework

Un framework (de origen anglosajón, marco de trabajo), según lo que dice la wikipedia [?], es una estructura conceptual y tecnológica de soporte definido, normalmente por módulos de software concretos, que sirve de base para la organización y el desarrollo de software. Las ventajas que ofrece son varias, entre las que se puede destacar:

- **Evita repetición de código:** las partes más usadas, pasan a ser algo del *core* del framework.
- **Uso de buenas prácticas:** muchos de estos están basados en patrones de diseño que nos obligan a usar.
- **Elementos avanzados integrados:** cosas complejas y que implementarlas llevaría mucho tiempo, suelen venir integradas.
- **Desarrollo ágil:** por los factores anteriores, podemos centrarnos más en la lógica de negocio de la aplicación que se desea hacer, de una manera más rápida y segura.

Por lo tanto, es necesario trabajar mediante frameworks, ya que nos garantizar una aplicación de mayor calidad, que si la hacemos desde cero, en código nativo.

## Opciones disponibles

En el mercado existen muchos frameworks disponibles para el desarrollo de aplicaciones móviles, por lo que decantarse por uno no es tarea sencilla, ya que como se aprecia cada uno tiene sus pros y contras. En una encuesta realizada en un foro [?], a mediados del 2019, la opinión de los más recomendados era 3.1:



Figura 3.1: Encuesta

Algo ideal es que se intento buscar un framework *crossplatform*, con el que abarcar un mayor mercado, a nivel de usuarios y dispositivos.

## Ionic

Ionic [?] se basa en lenguaje de programación javascript, html y css. Internamente esta basado en otro framework, como es Angularjs. La licencia Open source, fue lanzado en el 2013, permite el desarrollo para iOS y Android. A sufrido gran cantidad de cambios desde entonces.

Ventajas:

- Una amplia comunidad de usuarios.
- Documentación extensa y de calidad.

Algunas de las desventajas:

- Rendimiento algo menor, ya que no se desarrolla de forma nativa.

- Bibliotecas en constante cambio y evolución, más que ser algo bueno, puede que deje a la aplicación fuera de versión y se tenga que hacer desde cero.

### **React native**

React native [?] fue creado en 2015, de la mano de Facebook. La licencia que tiene es MIT. La diferencia que tiene con React, es que no manipula el DOM, por lo que tampoco usa HTML o CSS. Se programa en javascript. Las aplicaciones más conocidas que usan este framework es instagram o facebook.

Ventajas:

- Gran comunidad de usuarios y herramientas.
- Mejora de las funcionalidades constantemente.

Algunas de las desventajas:

- Rendimiento algo menor.
- No es código nativo, pero casi, por lo que no tiene soporte oficial de Google y Apple.

### **Xamarin**

Xamarin [?] creado por Microsoft en el 2011, por lo que está desarrollado en .NET, siendo propietario.

Ventajas:

- Permite el desarrollo de iOS, Android, web pero nativas de escritorio también.
- Puede llamar a fragmentos de código usados en otras plataformas.
- Soporte para wearables.

Algunas de las desventajas:

- Acceso limitado a las bibliotecas.
- Soporte y actualizaciones lento / tarde.

- Comunidad grande pero pequeña comparada con otras.
- Aplicaciones de mayor tamaño.
- Coste.

### **Otros**

Hay muchos otros, cmomo kotlin, Apache Cordova, jQuery Mobile, Native script ... Pero no me convencieron por diversas razones.

## **3.2. Flutter**

---

## Técnicas y herramientas

---

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.



---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.





---

## Trabajos relacionados

---

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.



---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.



---

## **Bibliografía**

---