



TITULACIÓN EN MAYÚSCULAS

Curso Académico 20XX/20XX

Trabajo Fin de Grado/Máster

TÍTULO DEL TRABAJO EN MAYÚSCULAS

Autor : Nombre del Alumno/a

Tutor : Nombre del Tutor del TFG

Trabajo Fin de Grado/Máster

Título del Trabajo con Letras Capitales para Sustantivos y Adjetivos

Autor : Nombre del Alumno/a

Tutor : Nombre del Tutor del TFG

La defensa del presente Proyecto Fin de Carrera se realizó el día de
de 20XX, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 20XX

*Dedicado a
mi familia / mi abuelo / mi abuela*

Agradecimientos

Aquí vienen los agradecimientos... Aunque esté bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.

Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.

Índice general

1. Introducción y Motivación	1
1.1. Mercado actual de las aplicaciones móviles	1
1.2. Tipos de aplicaciones móviles	4
1.3. Análisis del entorno	5
2. Objetivos	9
3. Tecnologías, Herramientas y Metodologías	11
4. Descripción informática	13
4.1. Requisitos	13
4.2. Arquitectura y Análisis	13
4.3. Diseño e Implementación	13
4.4. Pruebas	14
5. Conclusiones y trabajos futuros	15
Bibliografía	17
A. Manual de usuario	19

Índice de figuras

1.1. Aplicaciones gratuitas y de pago, Android vs iOS.	2
1.2. Apps por categoría Google Play	2
1.3. Apps por categoría App Store	2
1.4. Nuevas aplicaciones por mes en cada tienda de aplicaciones (27/01/2024) . . .	3
1.5. Mapa mundial de Android e iOS (27/01/2024)	4

Capítulo 1

Introducción y Motivación

Contexto en el que se enmarca el proyecto y la justificación del mismo. Este capítulo es muy importante porque permite al lector conocer qué sentido tiene el proyecto, qué ofrece, por qué es relevante su implementación, los objetivos que persigue, etc. Este capítulo debería tener una extensión de entre 4 y 8 páginas.

1.1. Mercado actual de las aplicaciones móviles

Las aplicaciones móviles han transformado por completo nuestra vida cotidiana. En la actualidad están presentes en nuestra vida diaria y tienen un gran impacto en cómo nos comunicamos, cómo trabajamos, cómo nos divertimos y cómo realizamos nuestras tareas como la compra.

Según los datos de 42matters, compañía que se encarga de recopilar y ofrecer datos a diferentes empresas, el 27 de enero de 2024 hay 2 millones aplicaciones Android en Google Play y algo más de 1.900.000 aplicaciones iOS en App Store, figura 1.1. De las cuales, el porcentaje de aplicaciones que los usuarios se pueden descargar de manera gratuita es aproximadamente de 95 % en ambas tiendas de aplicaciones.



Figura 1.1: Aplicaciones gratuitas y de pago, Android vs iOS.

Fuente: <https://42matters.com/stats#available-apps-count>

En cuanto a categorías, figura 1.2 y figura 1.3, la educación lidera en Google Play, mientras que los juegos son predominantes en App Store, apareciendo el “negocio” como segunda categoría en ambos casos.



Figura 1.2: Apps por categoría Google Play

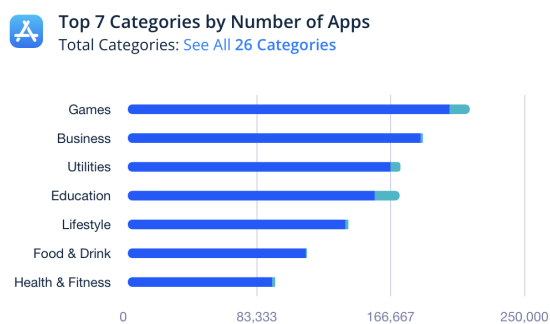


Figura 1.3: Apps por categoría App Store

Fuente: <https://42matters.com/stats#apps-by-category>

Además, se lanzan diariamente más de 2.300 nuevas aplicaciones en Google Play y unas 1.100 en App Store. Esto representa más de 90.000 aplicaciones nuevas al mes en la platafor-

ma de Android, lo que pone de manifiesto la alta competencia y dinamismo de este mercado, figura 1.4.

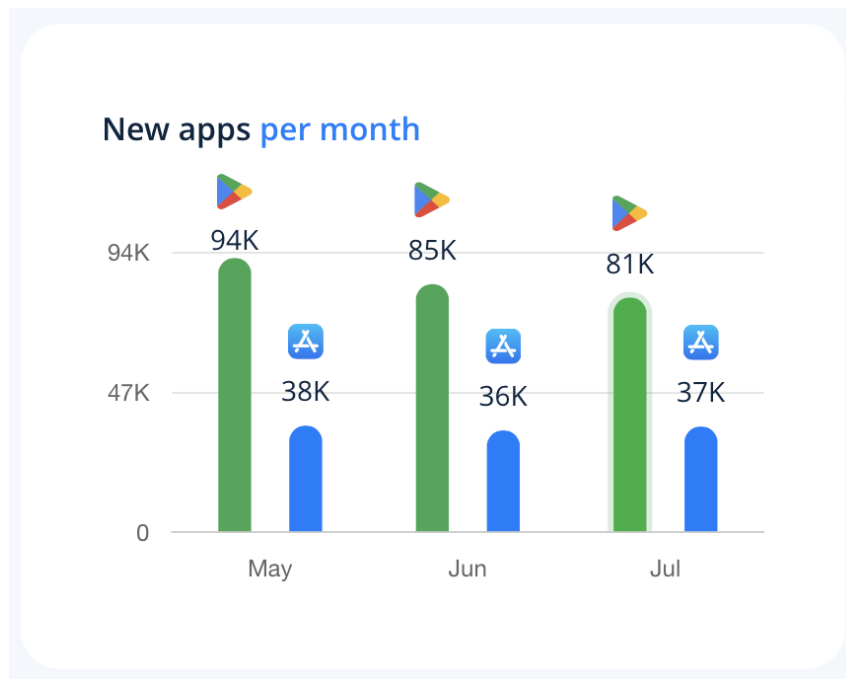


Figura 1.4: Nuevas aplicaciones por mes en cada tienda de aplicaciones (27/01/2024)

Fuente: <https://42matters.com/stats#apps-by-category>

En cuanto a los usuarios, como se puede ver en la figura 1.5, Android domina el panorama global. Según Statista (junio de 2024), su cuota de mercado alcanza el 72,15 %, frente al 27,19 % de iOS. Aunque en países como Estados Unidos e Irlanda iOS tiene más presencia, en regiones como América Latina, África, Asia y, especialmente, España, Android es el sistema operativo predominante.



Figura 1.5: Mapa mundial de Android e iOS (27/01/2024)

Fuente: <https://www.statista.com>

En el caso concreto de España, que es donde se publicaría esta aplicación, el 77 % de los smartphones son Android frente a un 23 % de iOS (Statista, diciembre 2023). Este dato resulta decisivo a la hora de seleccionar la plataforma de desarrollo, ya que permite orientar el producto a la mayoría de los usuarios potenciales.

1.2. Tipos de aplicaciones móviles

El desarrollo de software para dispositivos móviles puede abordarse desde diferentes enfoques según el tipo de aplicación que se desea construir. En general, se distinguen tres tipos principales: aplicaciones nativas, aplicaciones web y aplicaciones híbridas.

Una aplicación nativa es aquella que se crea específicamente para un sistema operativo móvil, utilizando su lenguaje y herramientas oficiales. Esto permite aprovechar al máximo las capacidades del hardware del dispositivo, lo que se traduce en un mayor rendimiento y una

experiencia de usuario más fluida. Por ejemplo, una aplicación desarrollada en Kotlin para Android o en Swift para iOS es considerada nativa. Este tipo de desarrollo requiere distribuir la app a través de plataformas oficiales como Google Play o App Store y, en caso de querer abarcar varios sistemas operativos, implica desarrollar una versión distinta para cada uno.

Las aplicaciones web, por el contrario, son páginas web optimizadas para dispositivos móviles, a las que se accede desde el navegador sin necesidad de instalación. Se desarrollan con tecnologías como HTML, CSS y JavaScript, y su principal ventaja es la portabilidad entre sistemas. Sin embargo, su rendimiento es inferior al de una aplicación nativa y tienen acceso limitado a las funcionalidades del dispositivo.

Las aplicaciones híbridas combinan elementos de ambas. Básicamente, se trata de aplicaciones web que se ejecutan dentro de un contenedor nativo, lo que permite distribuir las desde tiendas oficiales y acceder a algunas funcionalidades del hardware. No obstante, la integración entre la parte web y la parte nativa puede generar complejidades, sobre todo en términos de rendimiento y mantenimiento.

Tras analizar las características de cada enfoque, se optó por desarrollar una aplicación nativa para Android. Esta decisión responde tanto al interés por aprender en profundidad el desarrollo específico para esta plataforma, como al deseo de ofrecer una experiencia más fluida, potente y adaptada a los dispositivos que predominan en el mercado español.

1.3. Análisis del entorno

Antes de iniciar el desarrollo de la aplicación, se llevó a cabo un análisis del entorno para identificar qué soluciones digitales existen actualmente orientadas a la organización de la compra doméstica. Esta revisión permitió detectar tanto buenas prácticas como carencias que justifican el presente proyecto.

Una de las aplicaciones más conocidas en este ámbito es *Bring!*. Su funcionamiento se basa en la creación de listas de la compra mediante una interfaz visual con iconos organizados por categorías (por ejemplo, frutas, lácteos, panadería). Es intuitiva y estética, lo que la hace accesible para usuarios de cualquier perfil. Sin embargo, no permite especificar cantidades con precisión, no admite comentarios o notas personalizadas para cada producto, ni se integra con recetas o planificación semanal, lo que limita su utilidad para quienes desean gestionar la

compra de forma más detallada.

Otra opción popular es *Listonic*, que apuesta por un diseño más clásico: listas de verificación con productos agrupados por secciones del supermercado. Es rápida y práctica, y ofrece sugerencias inteligentes basadas en el texto introducido por el usuario. No obstante, su interfaz resulta más básica y no incorpora funcionalidades como la planificación de menús, el control del consumo habitual o la personalización de listas según hábitos domésticos.

Ambas aplicaciones ofrecen soluciones útiles para el usuario promedio, pero presentan limitaciones cuando se busca una herramienta más flexible y adaptada a la realidad diaria de una persona o familia que necesita controlar mejor lo que compra, cuándo lo compra y por qué. Esta falta de personalización y de conexión con hábitos reales de consumo doméstico justifica el desarrollo de una alternativa más completa y adaptable.

A diferencia de otras soluciones del mercado, Pinche aborda el problema de la organización de la compra doméstica desde una perspectiva práctica, personalizada y centrada en la realidad cotidiana de quienes gestionan los menús y las compras del hogar.

La aplicación se estructura en tres secciones funcionales: listas de la compra, recetas e invitados. En la sección de listas, el usuario puede crear múltiples listas adaptadas a distintas ocasiones —por ejemplo, “Lista semanal” o “Cumpleaños de María”— y añadir productos con su cantidad, e incluso el supermercado donde se prefiere comprarlos. En la sección de recetas, se pueden almacenar platos con su elaboración detallada e ingredientes para un número determinado de comensales, además de añadir fácilmente esos ingredientes a cualquier lista activa. Por último, en la sección de invitados, se puede registrar información personalizada sobre las personas que suelen comer en casa, como sus intolerancias, preferencias y qué platos se les han preparado recientemente.

Esta estructura modular convierte a Pinche no solo en una herramienta para digitalizar listas de la compra, sino en una solución doméstica integral que facilita la planificación, optimiza el tiempo y mejora la experiencia culinaria y organizativa en el hogar. Al permitir registrar hábitos, preferencias e información nutricional, contribuye también a evitar errores comunes como compras duplicadas, olvidos o preparación de menús inadecuados para los invitados.

Desde el punto de vista académico, el proyecto ofrece un caso completo para aplicar competencias clave del grado: desarrollo de interfaces modernas con Jetpack Compose, gestión de datos con Firebase, diseño orientado al usuario y trabajo bajo metodologías ágiles.

En definitiva, Pinche es un proyecto que trasciende el ejercicio académico. Es una propuesta aplicable y escalable, que puede mejorar la calidad de vida de los usuarios al resolver un problema doméstico frecuente mediante una herramienta accesible, inteligente y bien estructurada.

Capítulo 2

Objetivos

El objetivo principal de este proyecto es comprender y aplicar todo el proceso que conlleva la creación de una aplicación desde cero. En este caso, nos centraremos en cómo facilitar al usuario la tarea de realizar la compra doméstica, tanto de alimentos como de productos del hogar, de manera eficiente y organizada.

Este trabajo se basa en el desarrollo de una aplicación móvil nativa para el sistema operativo Android, denominada *Pinche*, que busca ayudar a los usuarios a planificar su compra semanal, registrar los productos que necesitan y gestionar su consumo.

La forma de alcanzar este objetivo consistirá en recorrer cada una de las etapas del proceso de desarrollo, adoptando los distintos roles que forman parte de un equipo digital: experiencia de usuario (UX), producto y desarrollo. Aunque el núcleo del trabajo se centrará en la parte técnica, también se tendrán en cuenta aspectos clave de diseño y definición del producto.

Adoptar esta perspectiva integral permite no solo aprender a programar, sino también a tomar decisiones informadas sobre las necesidades reales de los usuarios, los problemas que enfrentan en su día a día y cómo una solución tecnológica puede aportar valor real.

La motivación del proyecto radica precisamente en cubrir este vacío mediante el desarrollo de una aplicación intuitiva, personalizada y eficiente, centrada en mejorar la experiencia del usuario. Además, permite aplicar y reforzar conocimientos adquiridos durante el grado, desde el diseño de interfaces hasta la integración de servicios en la nube.

Desde el punto de vista técnico, se ha optado por emplear tecnologías modernas del ecosistema Android: el lenguaje de programación Kotlin, el framework Jetpack Compose para la construcción de interfaces declarativas y los servicios de Firebase —específicamente Firestore

como base de datos y Firebase Authentication para la gestión de usuarios—. Estas tecnologías permiten un desarrollo flexible y están alineadas con las demandas actuales del sector.

En cuanto a la metodología, se ha adoptado una combinación de Design Thinking para la definición inicial del problema, Lean Startup para validar hipótesis de producto, y metodologías ágiles como Scrum para organizar el desarrollo en iteraciones.

El presente trabajo ofrece una oportunidad para adquirir experiencia práctica en un entorno controlado, simulando las condiciones reales del desarrollo de software en equipos multidisciplinares. Asumir diferentes roles permite entender los retos de la comunicación entre perfiles técnicos y no técnicos, la importancia de la empatía hacia el usuario final, y el valor de trabajar con un enfoque centrado en el producto.

En resumen, el proyecto se plantea como una oportunidad para consolidar conocimientos técnicos, mejorar habilidades de diseño y comunicación, y resolver una necesidad real con impacto práctico. A través del desarrollo de esta aplicación, se busca no solo alcanzar los objetivos académicos del Trabajo de Fin de Grado, sino también generar una solución útil que podría ser utilizada por muchas personas en su vida diaria.

Capítulo 3

Tecnologías, Herramientas y Metodologías

Descripción de los lenguajes de programación, entornos de desarrollo, herramientas auxiliares, librerías de terceros, sistemas operativos, navegadores web, etc. utilizados para la realización del proyecto, así como la metodología empleada. El grado de profundidad a la hora de explicar cada tecnología dependerá de lo relevante que ha sido para el proyecto y lo conocida que es. Por ejemplo, si se usa el lenguaje de programación Java, no es necesario entrar en tanto detalle como si se usa un lenguaje mucho menos usado como Scala, por ejemplo.

Respecto a la metodología, dada la naturaleza de los proyectos, se suele describir una metodología iterativa e incremental en espiral, en la que se van sucediendo reuniones con el profesor que van definiendo el ámbito del proyecto.

Este capítulo puede tener una extensión entre 10 y 15 páginas.

Capítulo 4

Descripción informática

Descripción del proyecto realizado. Después de unos párrafos introductorios el capítulo se divide en subcapítulos. (de 25 a 35 páginas)

4.1. Requisitos

Descripción detallada de las funcionalidades que tendría que implementar la aplicación (pues se asume que los requisitos se escriben antes de empezar el desarrollo). Pueden tener forma de historias de usuario o bien ser una lista de requisitos funcionales y no funcionales.

4.2. Arquitectura y Análisis

Descripción de los aspectos de alto nivel de la aplicación. Diagramas de clases de análisis, diagramas de clases de diseño, etc. Se debe incluir la suficiente información para que el lector pueda entender la estructura de alto nivel del software desarrollado. Se pueden incluir diagramas de casos de uso si se considera útil.

4.3. Diseño e Implementación

Descripción de algún aspecto relevante de la implementación que quiera mencionarse. Por ejemplo se podría incluir alguno de los siguientes aspectos:

- Algoritmo complejo que se haya tenido que desarrollar.

- Integración entre librerías problemática.
- Resolución de algún bug que haya sido especialmente problemático.
- Focalizar en alguna parte del desarrollo y describirla en más detalle.

En esta sección se pueden incluir fragmentos de código fuente. También se pueden incluir algunas métricas del proyecto (número de clases, líneas de código, etc.). También se puede incluir la evolución del repositorio de GitHub (gráfico de commits por día).

4.4. Pruebas

En esta sección se describen las pruebas automáticas que han sido implementadas para el proyecto. Sobre los tests, conviene indicar la cobertura del código. Si no se han implementado pruebas automáticas, deberían haberse implementado y describirse aquí o tener una buena justificación de por qué no se han implementado.

Capítulo 5

Conclusiones y trabajos futuros

Reflexión sobre el trabajo realizado, qué objetivos se han cumplido y qué aspectos quedan pendientes para una futura ampliación del proyecto. Además, se deben incluir unas conclusiones personales indicando lo que ha supuesto para el alumno la realización del trabajo. Entre 2 y 4 páginas.

Bibliografía

Apéndice A

Manual de usuario