Logotipo

El contenido generado por IA puede ser incorrecto.

Memoria de prácticas

App Taskshare

Descripción breve

TaskShare es una app para gestionar tareas compartidas, enviando recordatorios y motivando con un sistema de puntos, ideal para la organización en el hogar.

*IES GONZALO CHACÓN*

*9 de junio de 2025*

*Adrian Carretero Alcazar*

*adrian.carretero16@educa.madrid.org*

INDICE

[1. INTRODUCCIÓN 2](#_Toc195442998)

[1.1 JUSTIFICACION DEL PROYECTO 2](#_Toc195442999)

[1.2 OBJETIVOS GENERALES Y ESPECIFICOS 2](#_Toc195443000)

[1.2.1 OBJETIVO GENERAL 2](#_Toc195443001)

[1.2.2 OBJETIVOS ESPECÍFICOS 2](#_Toc195443002)

[1.3 PUBLICO OBJETIVO 2](#_Toc195443003)

[1.4 VISIÓN GENERAL | APP A DESARROLLAR 2](#_Toc195443004)

[2 ESTUDIO PREVIO 3](#_Toc195443005)

[2.1 TECNOLOGÍAS UTILIZADAS Y ELECCIÓN 3](#_Toc195443006)

[3 HERRAMIENTAS UTILIZADAS 3](#_Toc195443007)

[4 DISEÑO DEL SISTEMA 4](#_Toc195443008)

[5 DESARROLLO E IMPLEMENTACIÓN 4](#_Toc195443009)

[6 PRUEBAS Y VALIDACIÓN 4](#_Toc195443010)

[7 CONCLUSIONES Y MEJORAS FUTURAS 4](#_Toc195443011)

[8 BIBLIOGRAFÍA 5](#_Toc195443012)

# INTRODUCCIÓN

## JUSTIFICACION DEL PROYECTO

En la vida cotidiana, especialmente en hogares o grupos de convivencia, la gestión de tareas puede volverse complicada debido a la falta de organización y comunicación. Esta situación puede causar olvidos, acumulación de responsabilidades e incluso conflictos. Para abordar este problema, surge **TaskShare**, una aplicación móvil que busca optimizar la administración de tareas compartidas, facilitando la coordinación entre los miembros del hogar o grupo mediante tecnología accesible, funcional y motivadora.

## OBJETIVOS GENERALES Y ESPECIFICOS

### OBJETIVO GENERAL

* Desarrollar una aplicación móvil eficiente y fácil de usar que permita la gestión, asignación y seguimiento de tareas compartidas en entornos domésticos o colaborativos.

### OBJETIVOS ESPECÍFICOS

* Permitir a los usuarios crear y asignar tareas a otros miembros del grupo.
* Implementar recordatorios automáticos para evitar olvidos.
* Incluir un sistema de incentivos mediante puntos para fomentar el cumplimiento de tareas.
* Integrar recordatorios a través de plataformas externas como WhatsApp y Telegram.
* Ofrecer opciones de personalización como modo oscuro/claro.
* Incluir funcionalidades de recuperación de contraseña mediante correo electrónico.
* Utilizar Kotlin para el desarrollo de la app y Firebase para la gestión de autenticación, almacenamiento y notificaciones.

## PUBLICO OBJETIVO

**TaskShare** está dirigida a:

* Familias o personas que viven juntas y necesitan coordinar tareas del hogar.
* Grupos de compañeros de piso, residencias estudiantiles o espacios compartidos.
* Usuarios que buscan una solución práctica, motivadora y centralizada para organizar tareas cotidianas entre varias personas.

## VISIÓN GENERAL | APP A DESARROLLAR

**TaskShare** es una aplicación móvil desarrollada en Kotlin con backend en Firebase, pensada para facilitar la administración de tareas compartidas. Permitirá a los usuarios crear, asignar y seguir tareas entre contactos, incorporando recordatorios automáticos, notificaciones por mensajería externa, personalización de la interfaz, sistema de puntos como incentivo y recuperación de acceso por correo. El objetivo es ofrecer una herramienta práctica, moderna y efectiva para mejorar la convivencia y la organización en grupos.

# ESTUDIO PREVIO

Existen apps populares para la gestión de tareas como Todoist, Google Tasks y Trello.  
Todoist ofrece muchas funciones, pero requiere pago para opciones avanzadas. Google Tasks es muy simple pero limitada. Trello es excelente para trabajo en equipo, aunque más compleja y enfocada a proyectos.

Mi aplicación busca un punto intermedio, ofreciendo una herramienta sencilla, gratuita y colaborativa, centrada en la creación y asignación de tareas con fecha y hora.

## TECNOLOGÍAS UTILIZADAS Y ELECCIÓN

Existen varias opciones para desarrollar apps:

* Nativo (Kotlin, Swift): máximo rendimiento.
* Multiplataforma (React Native, Flutter): un solo código para Android e iOS.

Opté por Kotlin con Jetpack Compose para crear una app nativa Android moderna y fluida, siguiendo las recomendaciones oficiales de Google.  
Además, usé Firebase (Firestore y Auth) por su fácil integración, escalabilidad y porque evita mantener un servidor propio.

# HERRAMIENTAS UTILIZADAS

Para el desarrollo de esta aplicación se ha utilizado **Android Studio** como entorno de desarrollo integrado (IDE), que proporciona todas las herramientas necesarias para crear aplicaciones Android de forma eficiente.  
El lenguaje de programación principal es **Kotlin**, moderno, conciso y recomendado oficialmente por Google para el desarrollo en Android.

La interfaz de usuario está construida con **Jetpack Compose**, un *toolkit* declarativo que permite crear interfaces de forma más rápida, flexible y con menos código, en comparación con el enfoque tradicional basado en XML.

Como *backend* y sistema de almacenamiento de datos se ha utilizado **Firebase**, concretamente **Firebase Firestore** para la base de datos en la nube y **Firebase Authentication** para gestionar usuarios, lo cual permite una integración sencilla y escalable con Android.

Además, se ha utilizado **GitHub** como sistema de control de versiones, lo que ha permitido llevar un seguimiento ordenado del desarrollo, realizar copias de seguridad y mantener el código sincronizado entre distintos entornos.

Para las pruebas y validación de la aplicación se ha utilizado un **dispositivo físico Huawei**, lo que ha permitido verificar el correcto funcionamiento en un entorno real, más allá del uso de emuladores.

Estas herramientas permiten un desarrollo moderno, eficiente y orientado a la nube, lo que facilita tanto la escalabilidad de la app como su mantenimiento a largo plazo.

# DISEÑO DEL SISTEMA

 Arquitectura general de la app

 Estructura de pantallas y navegación (puedes añadir un diagrama)

 Diseño de la base de datos (estructura de Firestore)

 Casos de uso

 Mockups o wireframes si los hiciste

# DESARROLLO E IMPLEMENTACIÓN

 Desarrollo de la interfaz con Jetpack Compose

 Gestión de estado con ViewModels

 Conexión con Firebase (autenticación, guardado de tareas)

 Validaciones y lógica (añadir tareas, editar, eliminar)

 Gestión de fechas y horas (como lo que hiciste)

# PRUEBAS Y VALIDACIÓN

 ¿Cómo probaste la app?

 ¿Qué dispositivos/emuladores usaste?

 ¿Qué errores encontraste y solucionaste?

 Feedback de usuarios si lo tienes

# CONCLUSIONES Y MEJORAS FUTURAS

 ¿Qué aprendiste?

 ¿Qué funcionó bien?

 ¿Qué cosas mejorarías si tuvieras más tiempo?

 Posibles funcionalidades futuras (notificaciones push, colaboraciones, etiquetas, modo oscuro…)

# BIBLIOGRAFÍA

* <https://firebase.google.com/> (Ayuda base de datos)
* <https://stackoverflow.com/> (Ayuda con problemas)
* https://www.google.com/ (Imágenes)
* [https://github.com/ (Control](https://github.com/%20(Control) de versiones)