

est.



2025

TASKSHARE APP

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

1 INDICE

1. INTRODUCCIÓN 2

1.1 JUSTIFICACION DEL PROYECTO 2

1.2 OBJETIVOS GENERALES Y ESPECIFICOS 2

1.2.1 OBJETIVO GENERAL 2

1.2.2 OBJETIVOS ESPECÍFICOS..... 2

1.3 PUBLICO OBJETIVO 2

1.4 VISIÓN GENERAL | APP A DESARROLLAR 2

2 ESTUDIO PREVIO 3

2.1 TECNOLOGÍAS UTILIZADAS Y ELECCIÓN 3

3 HERRAMIENTAS UTILIZADAS 3

4 DISEÑO DEL SISTEMA 4

5 DESARROLLO E IMPLEMENTACIÓN..... 4

6 PRUEBAS Y VALIDACIÓN 4

7 CONCLUSIONES Y MEJORAS FUTURAS 5

8 BIBLIOGRAFÍA..... 5

1. INTRODUCCIÓN

1.1 JUSTIFICACION DEL PROYECTO (DEFINICION DEL PROBLEMA)

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.

1.2 OBJETIVOS GENERALES Y ESPECIFICOS

1.2.1 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.

1.2.2 OBJETIVOS ESPECÍFICOS

- Permitir a los usuarios crear y asignar tareas para realizar.
- Implementar recordatorios automáticos para evitar olvidos.
- Incluir un sistema de incentivos mediante puntos para fomentar el cumplimiento de tareas.
- Poder contactar con los usuarios agregados por WhatsApp, teléfono y/o mensaje(sms).
- 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.

1.3 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.

1.4 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, etc. El objetivo es ofrecer una herramienta práctica, moderna y efectiva para mejorar la convivencia y la organización en grupos.

2 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.

2.1 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.

3 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.

4 DISEÑO DEL SISTEMA

- Arquitectura general de la app
- Estructura de pantallas y navegación (puedes añadir un diagrama) DIAGRAMA UML Y E/R
- Diseño de la base de datos (estructura de Firestore)
- Casos de uso
- Mockups o wireframes si los hiciste

5 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)

6 PRUEBAS Y VALIDACIÓN

- ¿Cómo probaste la app?
- ¿Qué dispositivos/emuladores usaste?
- ¿Qué errores encontraste y solucionaste?
- Feedback de usuarios si lo tienes

7 ENTREGA Y DOCUMENTACIÓN

En esta memoria, en la que está todo documentado, se hace entrega de lo siguiente:

- Memoria
- Diario en pdf
- Archivo zip que contiene el código de la app

8 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...)

9 BIBLIOGRAFÍA

- <https://firebase.google.com/> (Ayuda base de datos)
- <https://stackoverflow.com/> (Ayuda con problemas)
- <https://www.google.com/> (Imágenes)
- <https://github.com/> (Control de versiones)