



**UNIVERSIDAD
DON BOSCO**

DESARROLLO DE SOFTWARE PARA MÓVILES

APLICACIÓN DE GESTIÓN DE EVENTOS COMUNITARIOS

[REPOSITORIO EN GITHUB](#)

[MOCKUPS EN FIGMA](#)

[BOARD TRELLO](#)

DOCENTE:

ING. ALEXANDER SIGUENZA

PRESENTAN:

HENRY VLADIMIR NAJERA GUERRA

NG110680

JONATHAN RAFAEL SEÑORA REYES

SR232918

JARLY LEONEL VIGIL VELASQUEZ

VV233222

NELSON VICENTE LUNA HERNANDEZ

LH150774

FECHA DE ENTREGA: XX/XX/2025

ÍNDICE

1. INTRODUCCIÓN3

2. OBJETIVOS4

 2.1 OBJETIVO GENERAL4

 2.2 OBJETIVOS ESPECÍFICOS4

3. CONTENIDO5

 3.1 DESCRIPCIÓN GENERAL DEL SISTEMA5

 3.2 HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.....6

 3.3 METODOLOGÍA DE DESARROLLO.....6

 3.4 ARQUITECTURA DEL SISTEMA.....7

 3.5 FUNCIONALIDADES DEL SISTEMA8

 3.6 ROLES Y APORTES DEL EQUIPO10

 3.7 MOCKUPS Y DESARROLLO VISUAL11

 3.8 REPOSITORIO Y CONTROL DE VERSIONES12

 3.9 LICENCIA DEL PROYECTO14

4. CONCLUSIONES Y RECOMENDACIONES15

5. BIBLIOGRAFÍA / REFERENCIAS16

1. INTRODUCCIÓN

El presente documento tiene como finalidad exponer de forma ordenada y detallada el proceso de diseño y desarrollo de la aplicación móvil que hemos construido como parte del trabajo final de la asignatura. Este proyecto, realizado de manera colaborativa por los cinco integrantes del equipo, surge como una propuesta funcional para gestionar eventos, facilitando la creación, asistencia, calificación y retroalimentación de cada uno de ellos. Nos propusimos no solo resolver una necesidad práctica, sino también aplicar los conocimientos adquiridos a lo largo del ciclo en una solución concreta, que combinara aspectos técnicos, visuales y de experiencia de usuario.

Desde el inicio, trabajamos con un enfoque que priorizó tanto la organización interna del equipo como la claridad de roles. Cada uno asumió una sección del sistema, permitiendo avanzar en paralelo y luego integrar los distintos módulos en un entorno unificado. La implementación se llevó a cabo utilizando **Jetpack Compose** como tecnología central para el desarrollo de interfaces, junto con Firebase como backend para la gestión de usuarios, eventos, autenticación y almacenamiento de datos. El uso de herramientas modernas, junto con prácticas básicas de control de versiones y reuniones de seguimiento por videollamada, nos permitió mantener el proyecto en ritmo y con buena comunicación.

Este informe presenta, en las secciones que siguen, no solo las características técnicas de la aplicación, sino también las decisiones que guiaron su diseño, los retos que surgieron en el camino y las soluciones que aplicamos. Más allá del resultado final, el proyecto fue también una oportunidad para fortalecer nuestras habilidades de trabajo en equipo, resolución de problemas y adaptación frente a nuevas herramientas.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar una aplicación móvil funcional, intuitiva y moderna que permita la gestión integral de eventos, facilitando a los usuarios la creación, administración y seguimiento de eventos sociales

o corporativos, integrando funcionalidades como RSVP, comentarios, calificaciones y control de usuarios mediante autenticación.

2.2 OBJETIVOS ESPECÍFICOS

- Implementar un sistema de registro e inicio de sesión mediante correo electrónico y autenticación con Google.
- Permitir la creación, edición y eliminación de eventos por parte de administradores.
- Desarrollar un sistema de confirmación de asistencia (RSVP) para usuarios invitados a eventos.
- Incorporar funcionalidades de calificación y comentarios para evaluar los eventos una vez finalizados.
- Ofrecer un módulo de estadísticas para visualizar métricas relacionadas a la participación y evaluación de los eventos.
- Establecer un sistema de roles que distinga entre usuarios comunes y administradores.
- Aplicar una arquitectura organizada basada en el patrón MVVM para mantener una estructura limpia y escalable.
- Utilizar herramientas de colaboración como Trello y Google Meet para planificar, coordinar y dar seguimiento al trabajo en equipo.

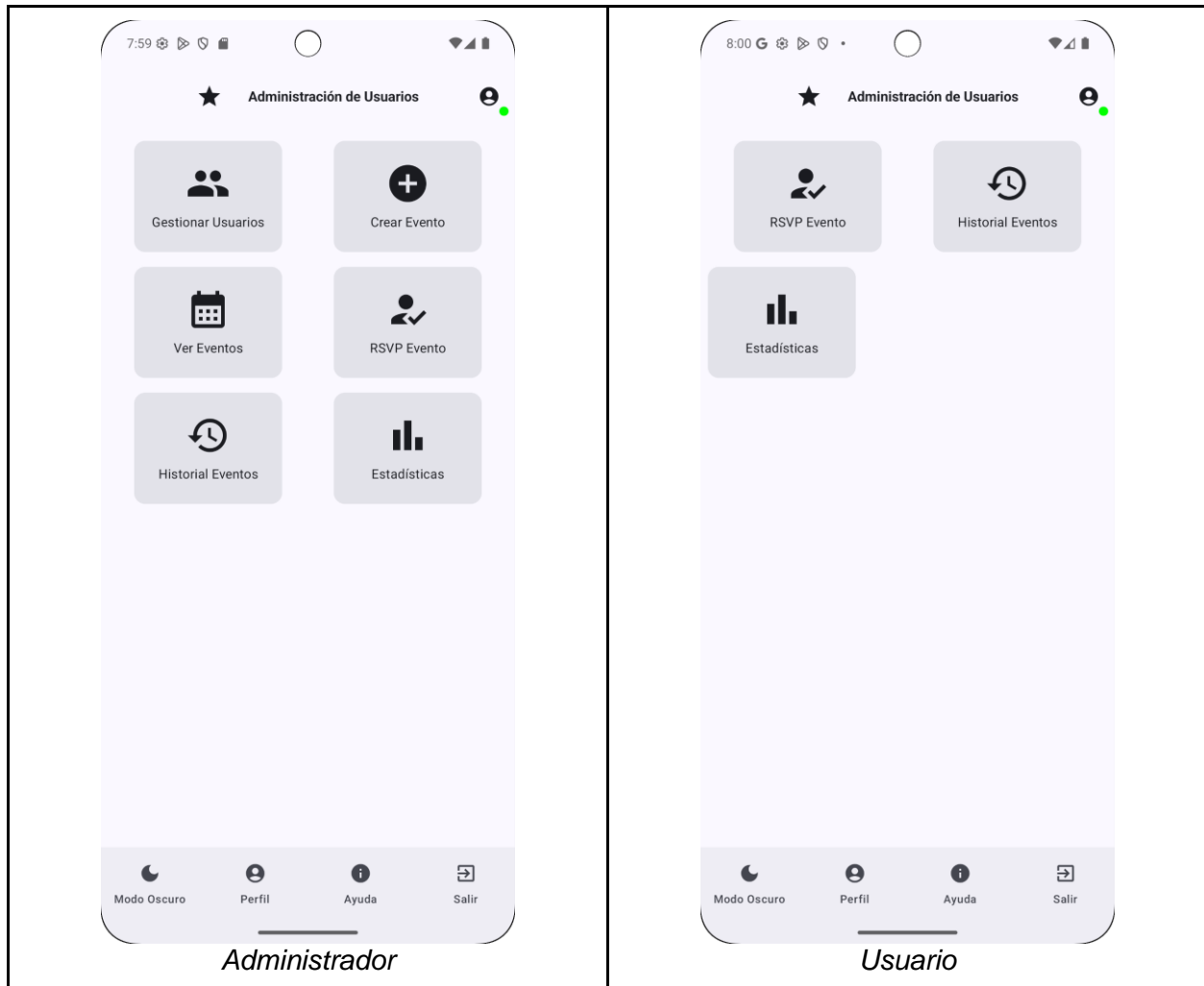
3. CONTENIDO

3.1 DESCRIPCIÓN GENERAL DEL SISTEMA

La aplicación desarrollada tiene como propósito brindar una plataforma práctica y funcional para la gestión de eventos. Está orientada tanto a usuarios comunes como a administradores, permitiendo crear, modificar, eliminar y visualizar eventos, así como también participar en ellos mediante confirmación de asistencia, calificación y comentarios. De esta forma, se cubren

diferentes aspectos que intervienen en la experiencia del usuario, desde la organización hasta la retroalimentación posterior a cada evento.

Uno de los elementos clave es que la aplicación se adapta al tipo de usuario que accede. Los administradores cuentan con un panel que les da control sobre la gestión de usuarios y eventos, mientras que los usuarios pueden interactuar de forma más simple: revisando eventos disponibles, confirmando su asistencia o dejando sus opiniones. Esta lógica fue pensada para que el sistema sea intuitivo y útil tanto en contextos académicos como en eventos sociales o profesionales.



A nivel técnico, la aplicación se construyó de manera modular, dividiendo responsabilidades según las secciones funcionales, lo cual permitió avanzar de forma ordenada, integrando progresivamente las distintas partes hasta formar un sistema cohesionado.

3.2 HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

La construcción de esta aplicación fue posible gracias a una combinación de herramientas modernas que facilitaron tanto el desarrollo como la integración de funcionalidades. En el apartado visual y estructural, empleamos **Jetpack Compose**, una tecnología de Google que permite diseñar interfaces en Android de forma declarativa. Esta herramienta resultó especialmente útil para adaptar visualmente las pantallas según el tipo de usuario y facilitar la navegación.

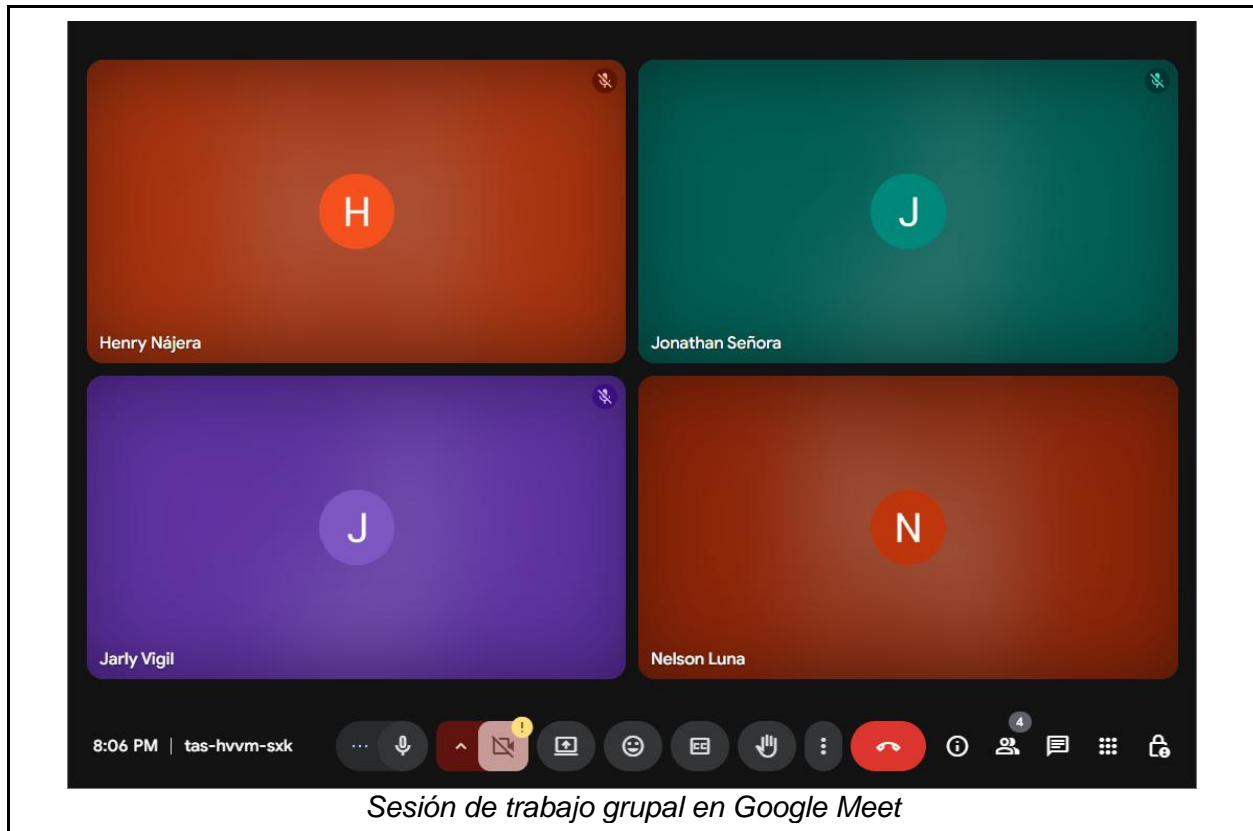
Para la gestión de datos, almacenamiento y autenticación, optamos por **Firestore**, lo que nos permitió registrar usuarios, guardar información de eventos, almacenar las calificaciones y comentarios, y manejar la lógica de autenticación tanto por correo electrónico como mediante cuentas de Google. Esto último contribuyó a simplificar el acceso a la aplicación.

El proyecto fue desarrollado en **Android Studio**, utilizando **Kotlin** como lenguaje principal. También configuramos el tema visual personalizado para unificar la apariencia de la app, aprovechando el sistema de temas, colores y tipografías.

3.3 METODOLOGÍA DE DESARROLLO

Aunque no aplicamos un marco formal de desarrollo como SCRUM, sí organizamos el trabajo siguiendo una estructura práctica que nos permitió avanzar por fases. Al inicio, se definió de forma clara el propósito de la aplicación y se establecieron las funcionalidades mínimas que debía cubrir. Luego se distribuyeron tareas entre los miembros del equipo, asignando a cada uno una parte del sistema, de acuerdo con sus habilidades y preferencias.

Durante el desarrollo, se realizaron reuniones periódicas mediante **Google Meet** para revisar el estado del proyecto, tomar decisiones conjuntas y resolver dudas. Cada integrante compartía avances, planteaba ideas o posibles mejoras, y recibía retroalimentación del resto del equipo. Esta dinámica permitió mantener una comunicación fluida, anticipar problemas y resolverlos antes de integrarlos a la base de código.



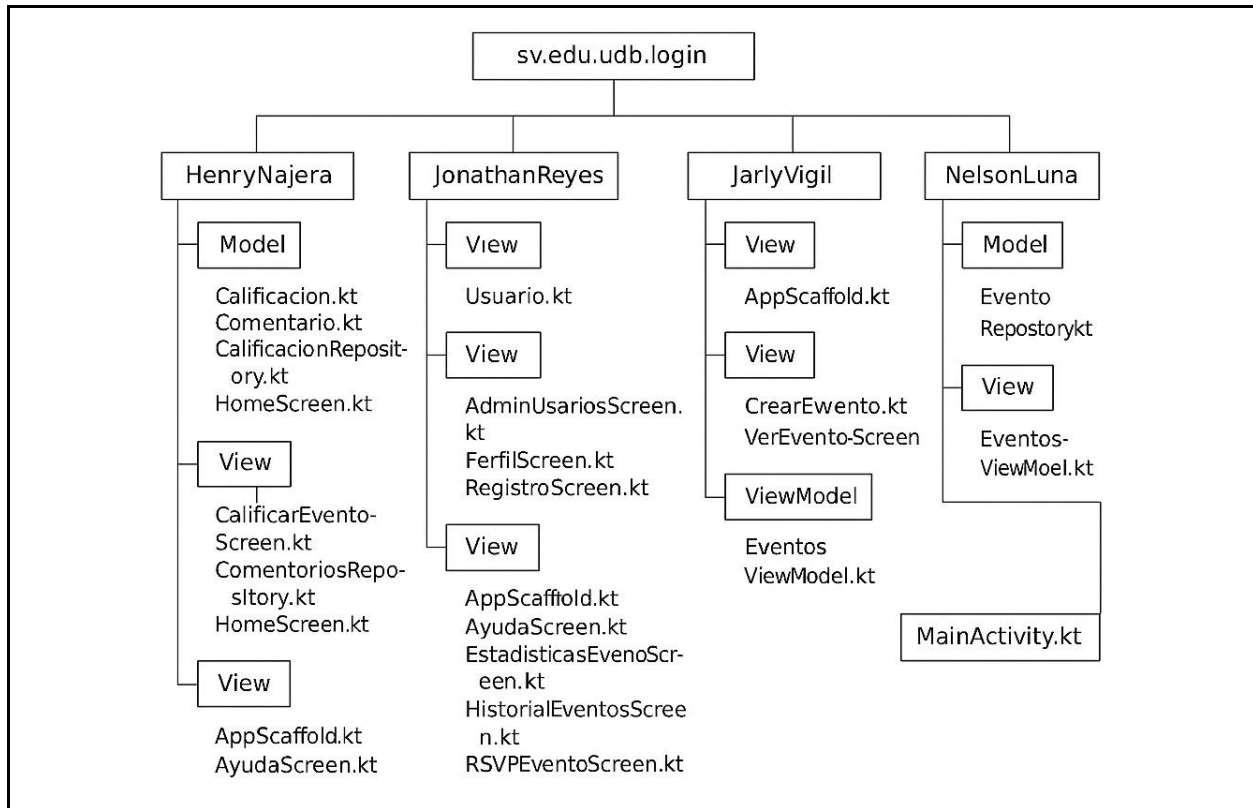
En las últimas etapas, nos enfocamos en pruebas, ajustes visuales, mejora del rendimiento y corrección de errores menores, asegurando que la experiencia final fuera lo más fluida posible para el usuario.

3.4 ARQUITECTURA DEL SISTEMA

La aplicación fue construida siguiendo una arquitectura basada en el modelo **MVVM (Modelo-Vista-ViewModel)**, lo que permitió una clara separación entre los datos, la lógica del negocio y la interfaz. Esta decisión se tomó para facilitar el mantenimiento del código, evitar repeticiones innecesarias y mejorar la escalabilidad del sistema en caso de futuras ampliaciones.

A nivel de organización, el proyecto se estructuró en paquetes con los nombres de cada integrante, conteniendo las funcionalidades que correspondían a su responsabilidad. Este enfoque fue muy útil para mantener el orden del proyecto y para identificar fácilmente qué parte del código pertenecía a quién. Además, al tratarse de un trabajo colaborativo, esta organización permitió minimizar conflictos al momento de integrar los módulos en un solo repositorio común.

Árbol de archivos del proyecto



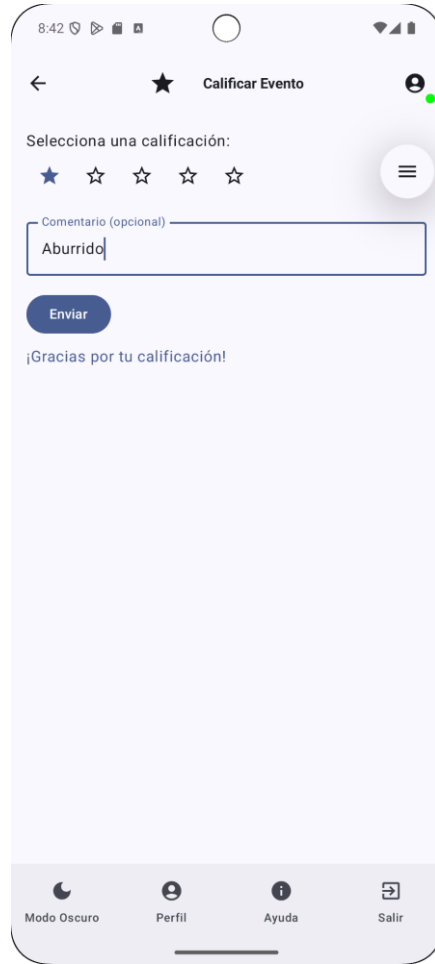
3.5 FUNCIONALIDADES DEL SISTEMA

El sistema cuenta con diversas funcionalidades que cubren tanto las necesidades del usuario regular como las del administrador. Entre las más relevantes se encuentran las siguientes:

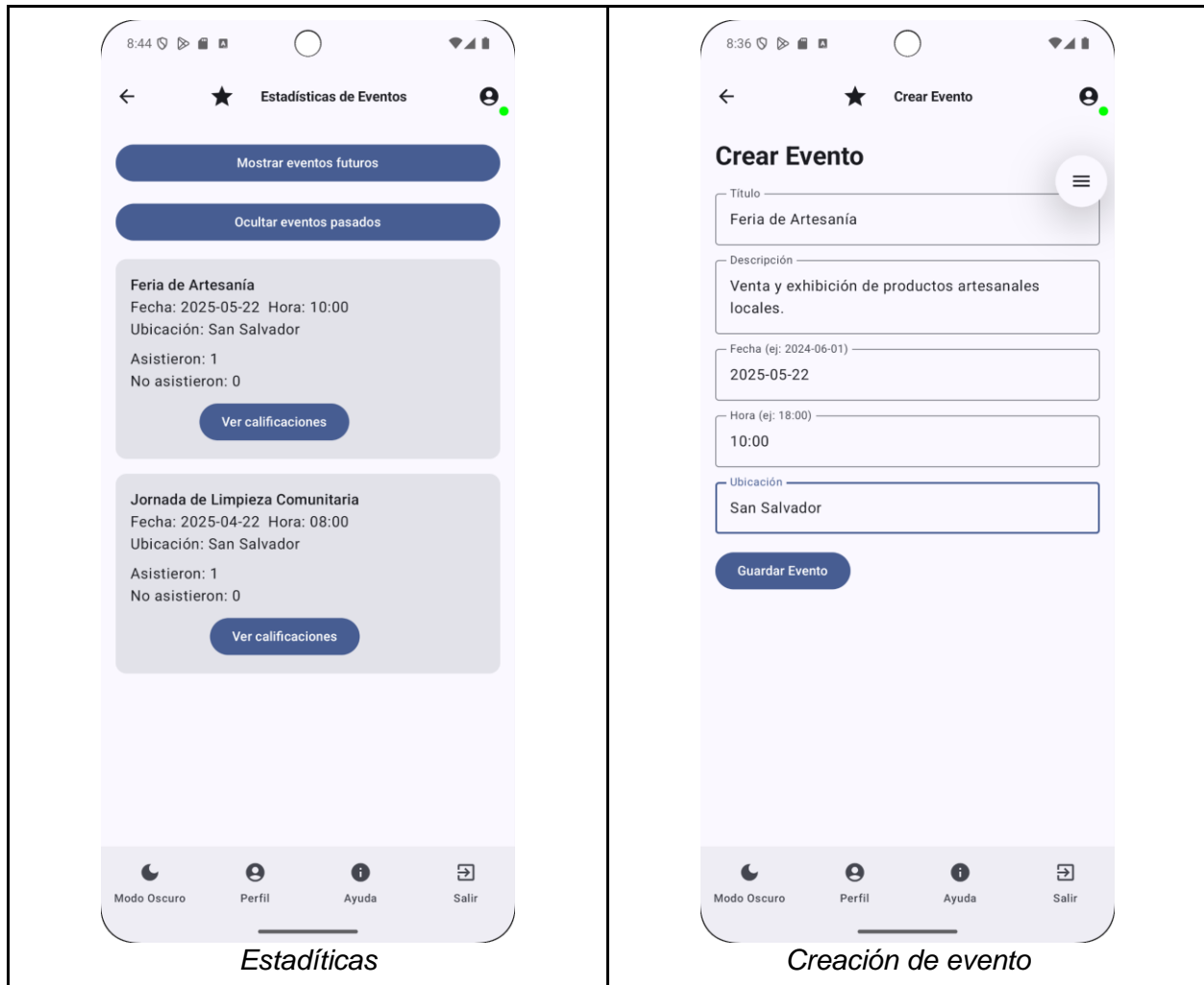
- Registro e inicio de sesión de usuarios, con posibilidad de autenticarse por correo o con cuenta de Google.
- Creación de eventos por parte de administradores, indicando título, fecha, hora, ubicación y descripción.
- Edición y eliminación de eventos, también accesible desde el panel de administración.
- Confirmación de asistencia (RSVP) por parte de los usuarios, con opción de cambiar su respuesta en cualquier momento.
- Visualización de estadísticas por evento, con detalles sobre la cantidad de asistentes confirmados, comentarios y calificaciones.
- Sección de historial para que cada usuario pueda ver los eventos pasados a los que asistió o no.
- Calificación de eventos utilizando un sistema de estrellas, con posibilidad de dejar comentarios.
- Panel de gestión de usuarios, accesible solo para administradores, que permite ver la lista de usuarios, cambiar sus roles o eliminar cuentas si es necesario.



RSVP



Calificación



3.6 ROLES Y APORTES DEL EQUIPO

El desarrollo de la aplicación fue resultado de una distribución clara del trabajo entre los cinco integrantes del equipo. Cada quien asumió una responsabilidad concreta, permitiendo que el proyecto avanzara en paralelo y luego fuera integrado como un solo sistema.

- **Henry Najera** fue responsable de las funcionalidades relacionadas con comentarios y calificaciones. Su trabajo incluyó tanto la interfaz donde los usuarios califican como la parte que muestra la retroalimentación en forma de lista.
- **Jarly Vigil** se encargó del diseño visual y la coherencia estética de la aplicación. También desarrolló las pantallas de ayuda, perfil de usuario, estadísticas y parte del historial, asegurando una experiencia fluida y atractiva.
- **Jonathan Señora** trabajó en el sistema de autenticación, el registro de nuevos usuarios, la gestión de roles (usuario o administrador) y la integración con Firebase para almacenar estos datos.

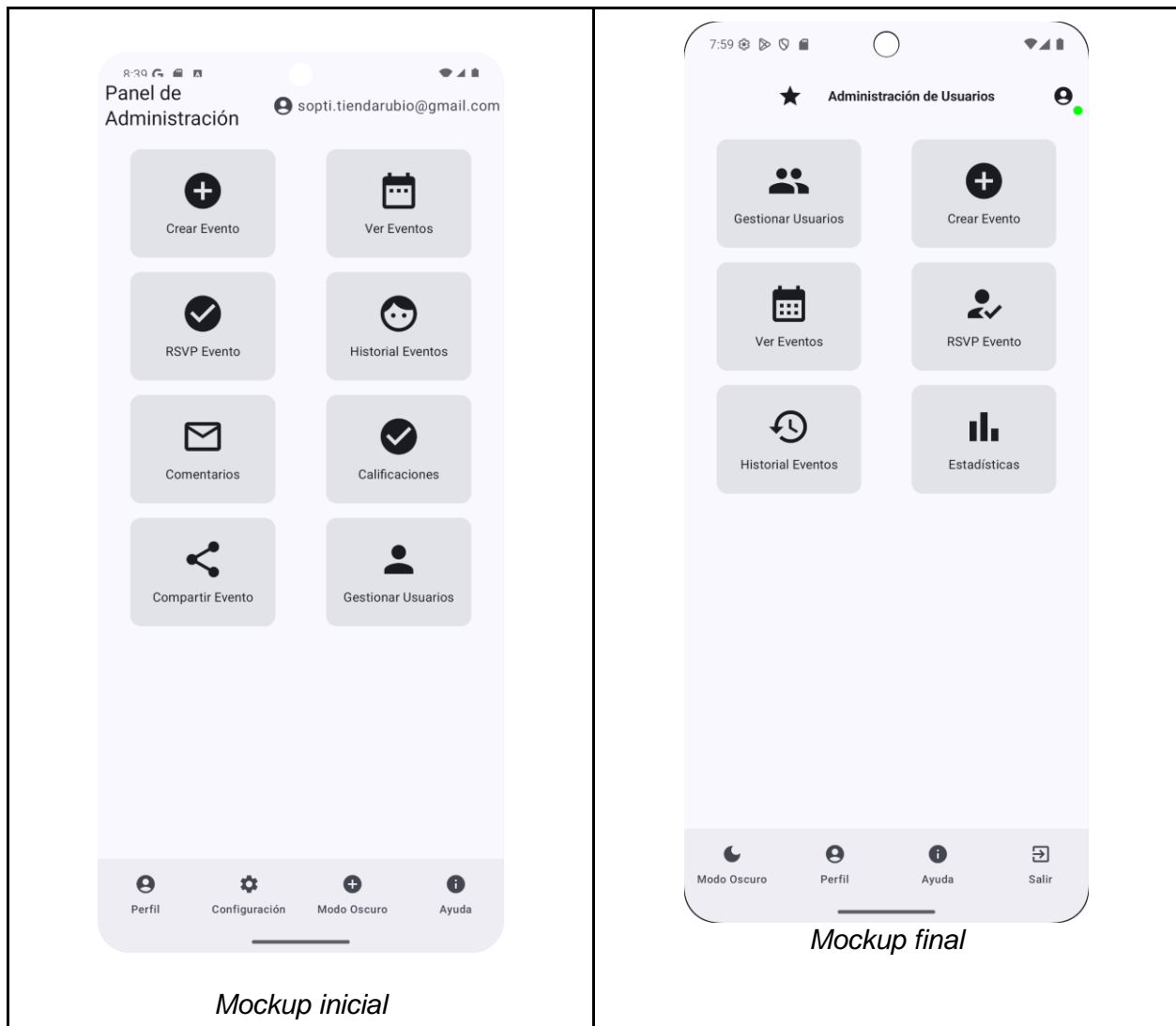
- **Nelson Luna** desarrolló las pantallas para crear, editar y listar eventos, así como el sistema RSVP. Además, colaboró en las pruebas finales y en la integración general de estas funcionalidades con la base de datos.

Si bien cada miembro tuvo un rol específico, todos participamos activamente en la revisión del código, las pruebas funcionales y las reuniones de seguimiento. Esta colaboración constante fue clave para alcanzar un resultado final sólido.

3.7 MOCKUPS Y DESARROLLO VISUAL

Antes de comenzar a codificar, trabajamos en bocetos visuales para representar cómo queríamos que se vieran las pantallas principales. Aunque no usamos una herramienta profesional de diseño, estos mockups nos sirvieron de guía para mantener una línea visual coherente entre los distintos módulos.

Durante el desarrollo, fuimos ajustando los colores, tamaños de texto y disposición de los elementos según la experiencia de uso. Optamos por una estética limpia, con íconos intuitivos y botones claramente visibles, sin sobrecargar las pantallas de información innecesaria.



Gracias a **Jetpack Compose**, pudimos aplicar temas personalizados, adaptar la app a modo claro y oscuro, y organizar mejor los estilos.

3.8 REPOSITORIO Y CONTROL DE VERSIONES

Para coordinar el trabajo entre los integrantes y llevar un control claro del avance del proyecto, utilizamos **Git** como sistema de versiones y **GitHub** como repositorio compartido. Esta herramienta fue fundamental para respaldar el código, integrar los diferentes módulos y mantener un registro claro de cada cambio realizado.

Desde la interfaz web, podíamos revisar los commits de cada compañero, identificar errores o retrocesos y evitar sobrescribir el trabajo ajeno. El hecho de trabajar con carpetas por integrante facilitó la integración y permitió que cada quien hiciera pruebas de su parte sin afectar el resto del sistema.

Commits

main

All users

All time

Commits on May 22, 2025

Add files via upload

Nqgh authored 3 days ago

Verifieddf2074b

Commits on May 21, 2025

Add files via upload

HenryGuerra14 authored 4 days ago

Verifiedf9b7e51

Commit

JarlyV authored 4 days ago

Verified8d5bdb5

Commits on May 20, 2025

Elimina archivos antiguos y sube solo la carpeta actual

JONHREYES95 committed 5 days ago

ce9e124

Guarda cambios locales antes de merge

JONHREYES95 committed 5 days ago

1654223

Initial commit

JONHREYES95 authored 5 days ago

Verified098016f

Commits on May 1, 2025

Actualiza README para reflejar autenticación con Firebase y Jetpack Compose

JONHREYES95 committed 3 weeks ago

16bd012

Actualiza README para reflejar autenticación con Firebase y Jetpack Compose

JONHREYES95 committed 3 weeks ago

b75e811

Primer commit del proyecto LoginFirebase

JONHREYES95 committed 3 weeks ago

e55136c

Commits on Apr 16, 2025

Merge branch 'main' of https://github.com/JONHREYES95/LoginForo1

JONHREYES95 committed on Apr 16

6de9251

Cambios y unificado actualizado

JONHREYES95 committed on Apr 16

2797fdd

Update README.md

JONHREYES95 authored on Apr 16

Verified6834d89

Initial commit

JONHREYES95 authored on Apr 16

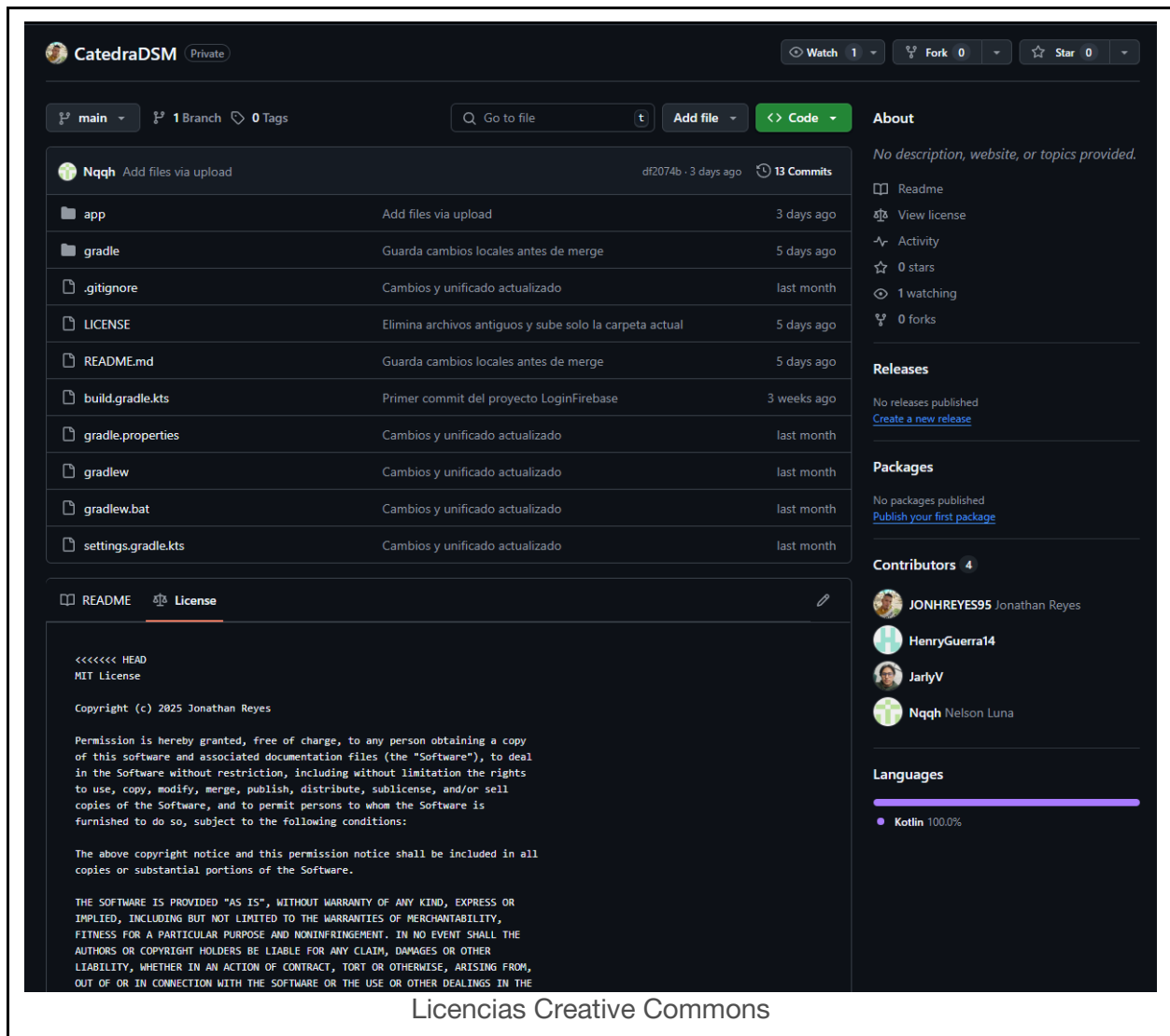
Verifiedd3b3ca4

Historial de commits en GitHub

Aunque al inicio el control fue más informal, a medida que avanzamos nos acostumbramos a hacer commits regulares, con mensajes claros, lo que ayudó a tener una trazabilidad del trabajo y resolver errores con más rapidez.

3.9 LICENCIA DEL PROYECTO

Al subir el proyecto a GitHub, decidimos incluir una **licencia de tipo** Licencias Creative Commons, lo que significa que cualquier persona puede utilizar el código, modificarlo o distribuirlo, siempre que se respeten los créditos originales. Esta decisión se tomó pensando en el carácter educativo del proyecto y en la posibilidad de que otros estudiantes puedan aprender de él en el futuro.



The screenshot shows a GitHub repository named 'CatedraDSM' (Private) with 1 branch and 0 tags. The repository contains 13 commits. The 'LICENSE' file is highlighted, showing the MIT License text. The repository also includes files like 'app', 'gradle', '.gitignore', 'README.md', 'build.gradle.kts', 'gradle.properties', 'gradlew', 'gradlew.bat', and 'settings.gradle.kts'. The right sidebar shows the repository's activity, including 0 stars, 1 watching, and 0 forks. The 'About' section is empty. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Contributors' section lists 4 contributors: JONHREYES95, HenryGuerra14, JarlyV, and Nqqh. The 'Languages' section shows Kotlin at 100.0%.

MIT License

Copyright (c) 2025 Jonathan Reyes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Licencias Creative Commons

Consideramos importante dejar en claro los términos de uso del código para evitar malentendidos y promover una cultura de responsabilidad digital dentro del desarrollo de software académico.

4. CONCLUSIONES Y RECOMENDACIONES

Al finalizar el desarrollo de esta aplicación, como equipo hemos reflexionado sobre el camino recorrido, las decisiones tomadas y las experiencias vividas. En primer lugar, una de las conclusiones más importantes que extraemos es que trabajar de forma colaborativa, con una organización clara y asignación de responsabilidades por carpetas, nos permitió avanzar con mayor fluidez, especialmente al tener definidos desde el inicio los roles de cada integrante y la visión general del producto que queríamos entregar. Esta estructura facilitó no solo el trabajo individual, sino también la integración de los componentes, evitando conflictos innecesarios en el código.

Por otro lado, la implementación de funcionalidades como el sistema de calificaciones, comentarios, autenticación y visualización de estadísticas, nos permitió aplicar muchos de los conocimientos aprendidos a lo largo del ciclo, especialmente en el uso de Firebase como backend y Jetpack Compose como herramienta principal de diseño de interfaces. A pesar de algunas dificultades técnicas, como la correcta sincronización de datos entre usuarios y eventos, logramos sortear los obstáculos apoyándonos mutuamente, aprendiendo nuevas herramientas y consultando documentación oficial cuando fue necesario.

Una de las lecciones más valiosas ha sido precisamente la gestión del tiempo y la importancia de tener un flujo de trabajo más constante. Aunque empezamos desarrollando en local, más adelante adoptamos el uso de GitHub para centralizar los avances, lo que sin duda facilitó el control de versiones y la revisión del trabajo ajeno. A futuro, recomendamos iniciar desde el principio con control de versiones en la nube para aprovechar todas sus ventajas desde el primer día.

Otra recomendación que queremos dejar es sobre el proceso de diseño visual y experiencia de usuario. Al usar Jetpack Compose, tuvimos la oportunidad de explorar un diseño más dinámico e interactivo, sin embargo, consideramos que sería enriquecedor contar con pruebas de usabilidad más tempranas que nos permitan validar con usuarios reales aspectos como la navegación, la claridad de los botones o la estructura de los formularios. Esto aportaría una capa más sólida a nivel de experiencia general.

Finalmente, creemos que esta aplicación tiene un gran potencial de crecimiento. Funcionalidades como la generación de reportes automáticos, el uso de notificaciones push, o incluso la integración con calendarios del dispositivo son aspectos que podrían considerarse para futuras versiones. El hecho de haber estructurado bien la base del sistema desde el inicio deja abierta esa posibilidad de escalar sin complicaciones estructurales.

En resumen, el desarrollo de esta aplicación ha sido una experiencia formativa integral, no solo en lo técnico sino también en lo humano. Aprendimos a comunicarnos mejor, a negociar decisiones y a valorar el esfuerzo de cada integrante, todo ello en función de un objetivo común: entregar una solución funcional, estética y útil que reflejara nuestras habilidades adquiridas a lo largo del curso.

5. BIBLIOGRAFÍA / REFERENCIAS

- Google Developers. (s.f.). Autenticación con Firebase. Firebase. <https://firebase.google.com/docs/auth?hl=es-419>
- Google Developers. (s.f.). Base de datos en tiempo real. Firebase. <https://firebase.google.com/docs/database?hl=es-419>
- Google Developers. (s.f.). Cloud Firestore. Firebase. <https://firebase.google.com/docs/firestore?hl=es-419>
- Google Developers. (s.f.). Jetpack Compose. Android Developers. <https://developer.android.com/jetpack/compose?hl=es-419>
- Material Design. (s.f.). Material 3 para Jetpack Compose. Material.io. <https://m3.material.io/develop/android/jetpack-compose>
- Mitchell, A. (2022). Cómo usar Git y GitHub: Guía para principiantes. GitHub Docs. <https://docs.github.com/es/get-started/using-git>
- Licencias Open Source Initiative. (s.f.). MIT License. Open Source Initiative. <https://opensource.org/license/mit/>
- JetBrains. (s.f.). Guía de estilo para Kotlin. Kotlinlang.org. <https://kotlinlang.org/docs/coding-conventions.html>
- Google Developers. (s.f.). Guía de arquitectura recomendada para apps Android. Android Developers. <https://developer.android.com/topic/architecture?hl=es-419>