

DAM: Desarrollo de Aplicaciones Multiplataforma

Modalidad: Presencial

Curso: 2024/2025

MEMORIA

PROOPARTY!

Autor: Alejandro Rodríguez Aragón

Tutor de proyecto: Sonia Suárez Pastur

Tutor individual: María del Mar Sánchez

Índice

1. INTRODUCCIÓN

- 1.1. Contexto
- 1.2. Presentación
- 1.3. Objetivos

2. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

- 2.1. Análisis de requisitos funcionales
 - 2.1.1. Introducción
 - 2.1.2. Tabla numerada de requisitos
- 2.2. Diagrama de casos de uso
 - 2.2.1. Introducción
 - 2.2.2. Diagramas y descripción
- 2.3. Diagrama de interfaces
 - 2.3.1. Introducción
 - 2.3.2. Diagramas y descripción
- 2.4. Análisis de las estructuras de datos utilizadas (bases de datos, ficheros...)
 - 2.4.1. Introducción
 - 2.4.2. Diagramas y descripción
- 2.5. Fase de pruebas
 - 2.5.1. Pruebas de validación de código
 - 2.5.2. Pruebas de implantación en distintos soportes

3. IMPLEMENTACIÓN

- 3.1. Tecnologías utilizadas en el desarrollo del proyecto

4. PUESTA EN MARCHA. DESPLIEGUE

5. APLICACIÓN EN EL ÁMBITO EMPRESARIAL

- 5.1. Idea de negocio
- 5.2. Planificación de necesidades de RRHH
- 5.3. Planificación de necesidades de producción
- 5.4. Planificación de la Prevención de Riesgos Laborales
- 5.5. Previsiones de inversión y financiación

6. CONCLUSIÓN

- 6.1. Valoración personal del trabajo realizado.
- 6.2. Posibles ampliaciones

7. BIBLIOGRAFÍA Y WEBGRAFÍA

8. ANEXOS

I – Manual de usuario

II – Código completo del proyecto

INTRODUCCIÓN

Contexto

En la actualidad, los eventos festivos, especialmente los populares como las fiestas de prado en Asturias, siguen siendo un punto de encuentro social muy importante, pero carecen de digitalización en su organización y difusión. A pesar de la gran afluencia de personas, la información sobre estos eventos suele transmitirse de manera informal o fragmentada, lo que genera confusión y dificulta la planificación de los asistentes.

En este contexto, surge la necesidad de una aplicación móvil que centralice y facilite el acceso a toda la información relativa a estas fiestas: fechas, ubicación, programación, imágenes y posibilidad de marcar eventos como favoritos. La digitalización de este proceso mejora la experiencia del usuario, promueve una mejor asistencia y ayuda a los organizadores a dar mayor visibilidad a sus eventos.

Presentación

PraoParty! es una aplicación móvil desarrollada para dispositivos Android que permite consultar información sobre fiestas populares, con especial enfoque en las que se celebran en zonas rurales y prados, típicas del norte de España.

La app muestra una lista de eventos con detalles como nombre, localización, descripción, imágenes y permite al usuario marcar sus fiestas favoritas para tenerlas rápidamente accesibles. Además, ofrece una interfaz clara e intuitiva, optimizada para una navegación sencilla, incluso para usuarios sin experiencia tecnológica.

El proyecto ha sido desarrollado con tecnologías modernas como Kotlin y Jetpack Compose, integrando funcionalidades prácticas que cubren tanto el entretenimiento como la utilidad.

Objetivos

Objetivo general:

- Desarrollar una aplicación móvil funcional e intuitiva que permita consultar y gestionar información sobre fiestas de Asturias, mejorando el acceso a estos eventos y fomentando su difusión.

Objetivos específicos:

- Crear una base de datos de eventos con su información principal.
- Diseñar una interfaz moderna y accesible con Jetpack Compose.
- Implementar una funcionalidad de favoritos para mejorar la experiencia del usuario.
- Garantizar la correcta visualización en distintos dispositivos Android.

- Validar el funcionamiento mediante pruebas en diferentes entornos.

ANÁLISIS Y DISEÑO DE LA APLICACIÓN

Análisis de requisitos funcionales

Introducción

El análisis de requisitos funcionales tiene como objetivo identificar todas las funcionalidades que la aplicación PraoParty! debe ofrecer al usuario final. Estos requisitos se han definido en base a las necesidades detectadas durante la fase de estudio del contexto, garantizando que la solución cubra los aspectos esenciales de consulta, gestión de eventos y personalización a través del uso de cuentas de usuario.

Tabla numerada de requisitos

N.º	Requisito funcional	Descripción
RF1	Inicio de sesión de usuarios	La aplicación cuenta con una pantalla de inicio de sesión gestionada con Firebase Authentication, permitiendo al usuario registrarse, iniciar sesión y mantener su sesión activa.
RF2	Gestión de sesiones	El sistema recuerda al usuario autenticado.
RF3	Seguridad básica de acceso	Algunas funcionalidades requieren que el usuario haya iniciado sesión para poder utilizarlas.
RF4	Visualización de fiestas	El usuario puede ver una lista de fiestas disponibles.
RF5	Detalle de una fiesta	El usuario puede pulsar sobre una fiesta para ver su descripción, localización e imagen.
RF6	Añadir a favoritos	El usuario puede marcar o desmarcar fiestas como favoritas.
RF7	Consultar favoritos	El usuario puede acceder a una lista con las fiestas marcadas como favoritas.
RF8	Navegación entre pantallas	La aplicación permite una navegación fluida entre la lista, los detalles y los favoritos.
RF9	Diseño responsive	La interfaz se adapta correctamente a diferentes tamaños de pantalla.
RF10	Almacenamiento local de favoritos	Las fiestas favoritas se guardan localmente en el dispositivo.

Diagrama de casos de uso

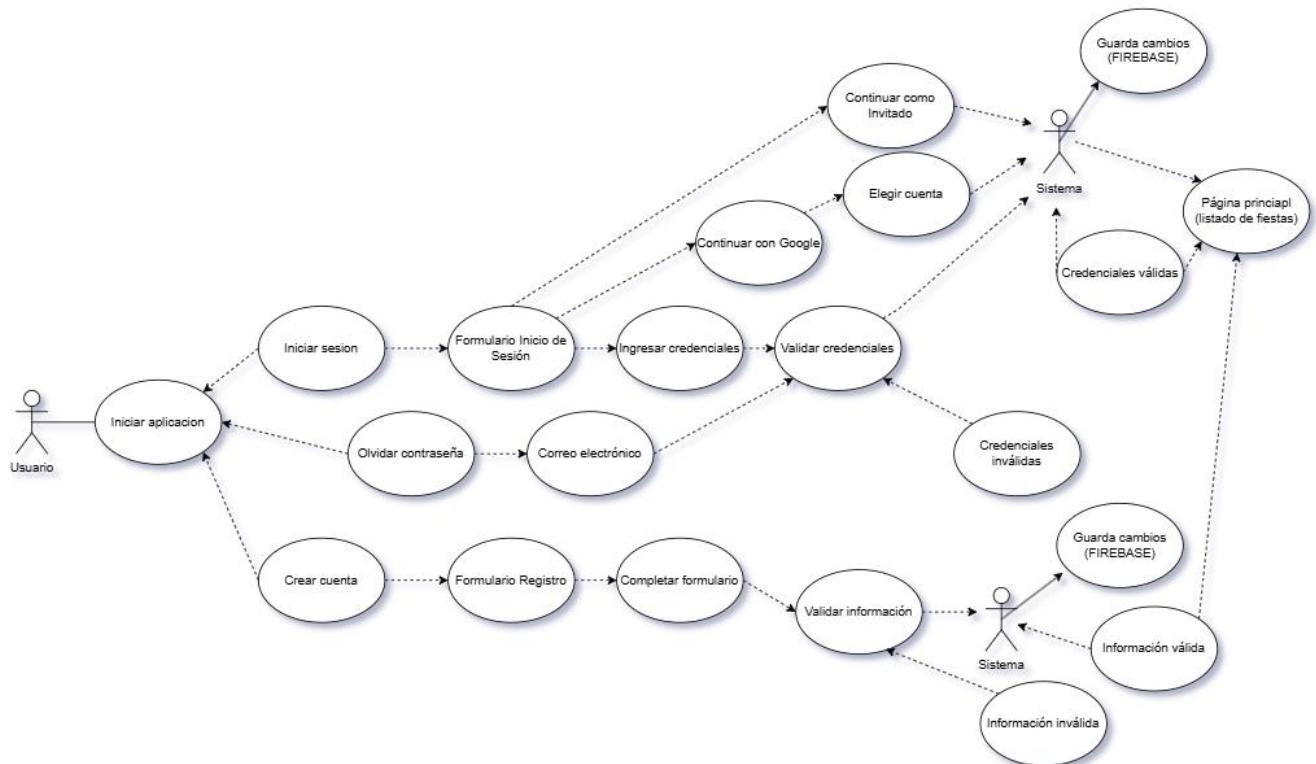
Introducción

Los diagramas de casos de uso permiten representar de forma gráfica las funcionalidades principales que ofrece una aplicación desde el punto de vista del usuario.

En el caso del proyecto PraoParty!, este tipo de diagrama se ha utilizado para identificar y estructurar las distintas acciones que un usuario puede realizar dentro de la aplicación, como iniciar sesión, consultar fiestas, gestionar favoritos o cerrar sesión.

Gracias a este modelo, es posible visualizar las interacciones entre el actor principal (usuario registrado) y los distintos casos de uso de forma clara, facilitando así el análisis y diseño funcional del sistema.

Diagramas y descripción



El diagrama representa el **flujo de autenticación y registro de usuarios** en la aplicación, abarcando los principales caminos que un usuario puede seguir al iniciar la app. El objetivo principal es mostrar cómo un usuario puede acceder a la aplicación, ya sea creando una cuenta, iniciando sesión, usando autenticación con Google, como invitado o recuperando su contraseña.

1. Inicio de la aplicación

El actor principal es el **Usuario**, quien interactúa inicialmente con el caso de uso "Iniciar aplicación". Desde aquí, el usuario dispone de tres opciones principales:

- **Iniciar sesión**

- **Crear cuenta**
- **Olvidar contraseña**

2. Inicio de sesión

Si el usuario elige “**Iniciar sesión**”, se le presenta el **Formulario de Inicio de Sesión**, donde puede:

- **Ingresar credenciales** manualmente (usuario y contraseña)
- Optar por “**Continuar con Google**”, lo que lleva a un flujo de autenticación externo para seleccionar una cuenta de Google.
- **Continuar como Invitado**, permitiendo el acceso sin autenticación.

Después de ingresar credenciales, se produce el caso de uso **Validar credenciales**:

- Si las credenciales son **válidas**, el sistema (actor secundario) permite el acceso y muestra la **Página principal (listado de fiestas)**.
- Si las credenciales son **inválidas**, el usuario es informado y debe intentarlo de nuevo.

Cuando se usa Google o el modo Invitado, el sistema también realiza una validación y, si es exitosa, lleva al usuario a la página principal.

3. Olvidar contraseña

Si el usuario selecciona “**Olvidar contraseña**”, se le solicita el **Correo electrónico** asociado a su cuenta para iniciar el proceso de recuperación de contraseña

4. Creación de cuenta

Si el usuario elige “**Crear cuenta**”, se presenta el **Formulario de Registro**, donde debe **Completar formulario** con sus datos.

Luego se ejecuta el caso de uso **Validar información**:

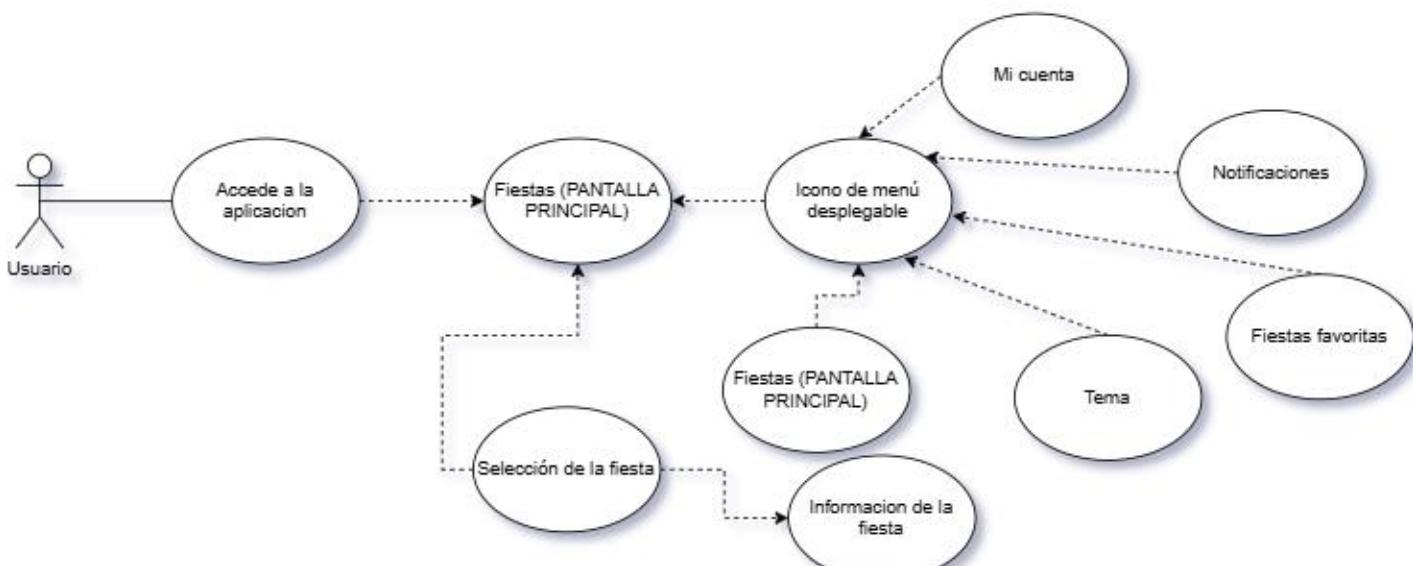
- Si la información es **inválida**, el sistema notifica al usuario para que corrija los datos.
- Si la información es **válida**, el sistema guarda los datos (indicado por **Guarda cambios (FIREBASE)**), señalando que se utiliza Firebase (como backend) y permite el acceso a la aplicación, mostrando la **Página principal**.

5. Guardado y acceso a la información

En varios momentos del flujo, el sistema realiza acciones internas como la **validación** y el **guardado de datos en Firebase**. Estas operaciones son representadas como casos de uso del actor secundario **Sistema**, indicando que algunas decisiones y procesos son automáticos y no requieren interacción directa del usuario.

6. Página principal

El objetivo final, tras cualquiera de los flujos de autenticación exitosos (inicio de sesión, registro, Google o invitado), es acceder a la **Página principal (listado de fiestas)**, que representa el punto de entrada al contenido de la app una vez autenticado o validado el usuario.



Este diagrama sintetiza y resume la **navegación básica** de la aplicación desde la perspectiva del usuario, centrándose en la exploración de fiestas y el uso del menú principal para acceder a opciones secundarias.

1. Acceso a la aplicación y pantalla principal

- El **Usuario** comienza el flujo accediendo a la aplicación, lo que le lleva directamente a la **Pantalla principal de fiestas (Fiestas - PANTALLA PRINCIPAL)**, punto de inicio central de la app.
- Desde esta pantalla, el usuario puede:
 - **Seleccionar una fiesta** específica del listado, lo que abre la opción de visualizar la **Información de la fiesta** seleccionada (por ejemplo, detalles, ubicación, fecha, etc.).

2. Navegación mediante menú desplegable

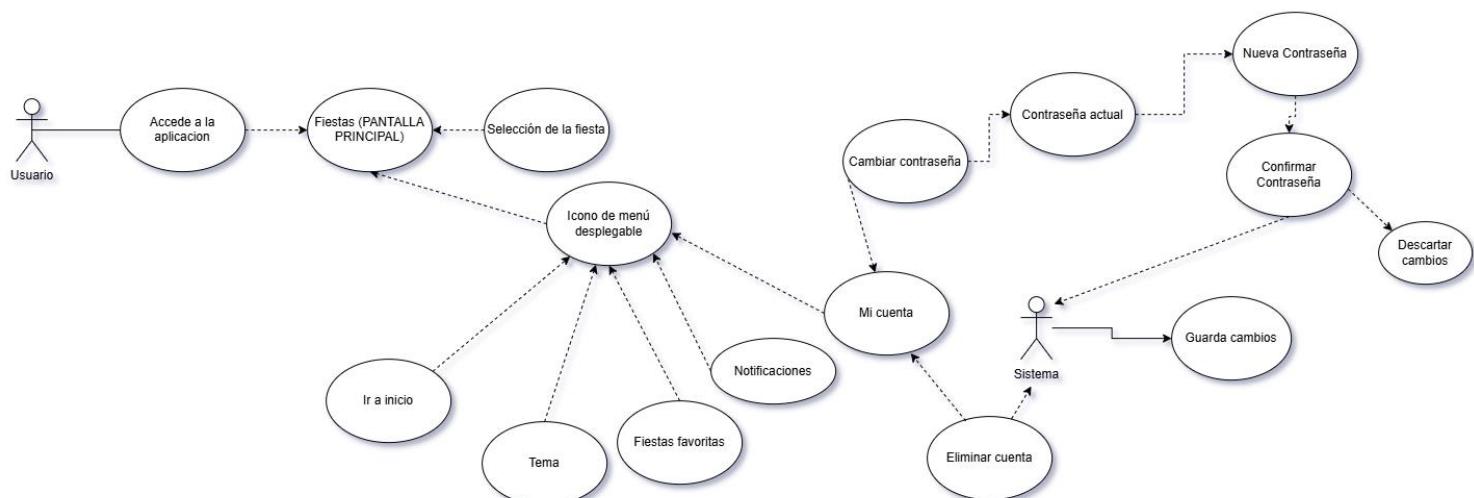
- Desde cualquier momento en la pantalla principal, el usuario puede acceder al **Icono de menú desplegable**, elemento recurrente y centralizador de la navegación en la app.
- El **menú desplegable** permite acceder de forma rápida a diversas funciones secundarias:
 - **Mi cuenta:** Gestión de la información de usuario (datos, contraseña, etc.).

- **Notificaciones:** Revisión y configuración de avisos y alertas que recibe el usuario.
- **Fiestas favoritas:** Acceso al listado de fiestas marcadas como favoritas.
- **Tema:** Cambio de la apariencia visual de la app (modo claro/oscuro, colores, etc.).

El menú también facilita la **vuelta a la pantalla principal** tras consultar otras opciones.

3. Detalle de exploración de fiestas

- El usuario puede navegar entre la **pantalla principal de fiestas** y la **información específica de cada fiesta** seleccionada, generando un flujo de ida y vuelta natural.
- Esta interacción directa es la base de la experiencia de usuario, pues la aplicación gira en torno al descubrimiento y consulta de eventos.



Este diagrama amplía el alcance funcional de la aplicación, centrándose en la **navegación interna y la gestión de la cuenta del usuario** una vez que ha accedido correctamente a la app, tras los procesos de autenticación descritos en el primer diagrama.

1. Acceso y navegación principal

- El **Usuario** accede a la aplicación y entra en la pantalla principal de la app: **Fiestas (PANTALLA PRINCIPAL)**, donde se listan los eventos disponibles.
- Desde esta pantalla, el usuario puede:

- Realizar una **Selección de la fiesta** (para ver más detalles de un evento).
- Acceder al **ícono de menú desplegable**, un componente típico en apps móviles que da acceso a diferentes opciones de configuración y navegación.

2. Menú desplegable y funcionalidades

Al pulsar el **ícono de menú desplegable**, el usuario dispone de varias opciones:

- **Ir a inicio**: Posiblemente, volver a la pantalla principal desde cualquier sección.
- **Tema**: Permite cambiar el tema visual de la app (por ejemplo, modo claro/oscuro).
- **Fiestas favoritas**: Acceso directo a una lista personalizada de fiestas marcadas como favoritas.
- **Notificaciones**: Acceso a la configuración o historial de notificaciones de la aplicación.
- **Mi cuenta**: Sección de gestión de cuenta de usuario (pivote fundamental de este diagrama).

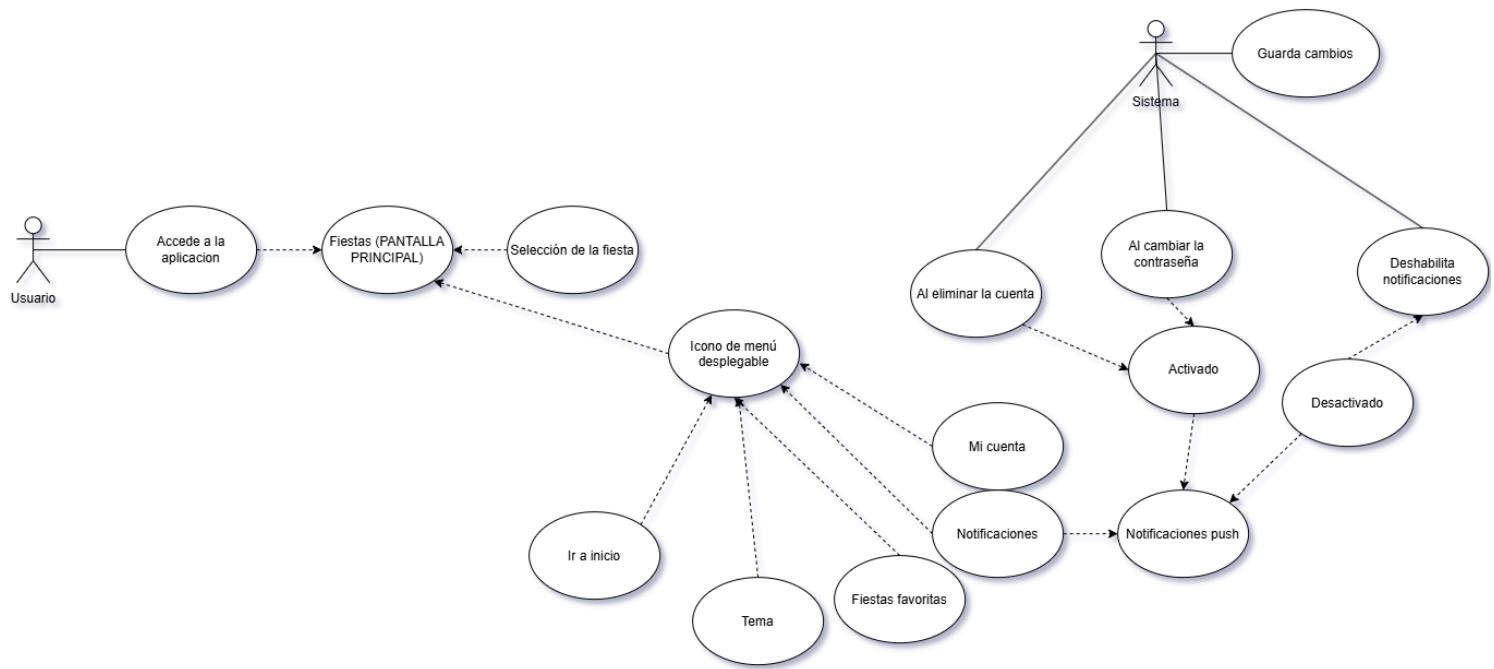
3. Gestión de la cuenta del usuario

Dentro de **Mi cuenta**, el usuario puede:

- **Cambiar contraseña**:
 - Se solicita la **Contraseña actual**.
 - El usuario introduce una **Nueva contraseña**.
 - Se le pide **Confirmar contraseña** para evitar errores de tipeo.
 - Puede **Descartar cambios** si decide no continuar.
 - Si la operación es válida, el **Sistema** ejecuta el caso de uso **Guarda cambios**, almacenando la nueva contraseña.
- **Eliminar cuenta**:
 - El usuario puede optar por eliminar su cuenta de forma definitiva.
 - El **Sistema** ejecuta la eliminación de los datos del usuario.

4. Rol del sistema

El actor **Sistema** vuelve a aparecer, representando los procesos automáticos internos, como guardar cambios en la cuenta o realizar la eliminación de usuario.



Este diagrama de casos de uso añade una **capa más completa de gestión de usuario y notificaciones**, mostrando cómo las acciones del usuario pueden activar respuestas automáticas del sistema, especialmente en lo relativo a la cuenta y las notificaciones.

1. Acceso y navegación principal

- El **Usuario** comienza accediendo a la aplicación, lo que le lleva a la **Pantalla principal de fiestas**.
- Desde la pantalla principal, puede:
 - **Seleccionar una fiesta** concreta para ver más información.
 - Acceder al **ícono de menú desplegable**, núcleo de navegación de la app, donde encuentra accesos a:
 - **Ir a inicio** (volver a la pantalla principal desde cualquier lugar).
 - **Tema** (cambiar el modo visual de la aplicación).
 - **Fiestas favoritas** (ver eventos marcados como favoritos).
 - **Notificaciones** (gestión de avisos).
 - **Mi cuenta** (gestión de información y seguridad del usuario).

2. Gestión de cuenta y notificaciones

Al acceder a **Mi cuenta** o **Notificaciones** desde el menú, el usuario puede realizar acciones que activan flujos automáticos:

a) Cambiar la contraseña

- Si el usuario cambia la contraseña, el sistema valida el cambio y lo guarda.
- Esta acción puede activar un flujo de notificaciones push, avisando al usuario de cambios en la seguridad de su cuenta (lo cual es una buena práctica de seguridad).

b) Eliminar la cuenta

- Al solicitar eliminar la cuenta, el sistema ejecuta la eliminación y guarda los cambios de manera irreversible.

c) Gestión de notificaciones push

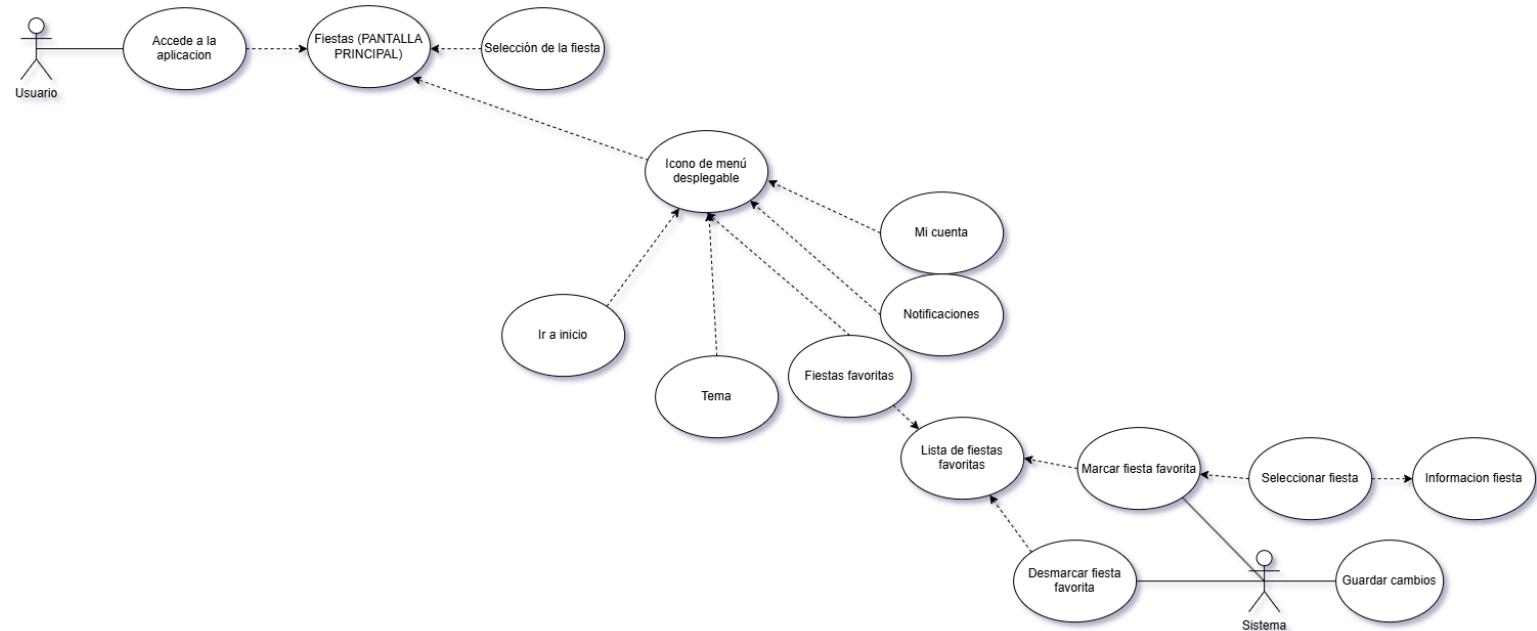
- Desde la sección **Notificaciones**, el usuario puede:
 - **Activar o desactivar las notificaciones push.**
 - Si el usuario las deshabilita, el sistema lo marca como **Desactivado**.
 - Si las activa, el sistema lo marca como **Activado** y podrá enviarle mensajes o alertas.

3. Automatización y lógica de sistema

- El actor **Sistema** es responsable de ejecutar las acciones automáticas (guardar cambios, gestionar notificaciones push, procesar eliminaciones, etc.).
- Todas estas operaciones están automatizadas y, en muchos casos, **desencadenan notificaciones push** para mantener al usuario informado en tiempo real sobre cambios críticos (contraseña, baja, etc.).

4. Flujo general

El usuario siempre puede volver a la pantalla principal (**Ir a inicio**), consultar fiestas favoritas, modificar la apariencia de la app, y gestionar su cuenta o notificaciones, siempre a través del menú desplegable.



Este diagrama amplía el módulo de **gestión de fiestas favoritas**, integrándolo dentro de la arquitectura general de la app y mostrando tanto la navegación como la interacción con el sistema para guardar los cambios de favoritos.

1. Acceso y navegación principal

- El **Usuario** accede a la aplicación y entra a la **Pantalla principal de fiestas**.
- Desde la pantalla principal puede:
 - **Seleccionar una fiesta** para ver detalles.
 - Acceder al **ícono de menú desplegable**, desde donde puede navegar hacia:
 - **Ir a inicio** (volver a la pantalla principal).
 - **Tema** (cambiar el estilo visual).
 - **Mi cuenta** (gestión del perfil).
 - **Notificaciones** (gestión de avisos).
 - **Fiestas favoritas** (acceso a la gestión de eventos favoritos).

2. Gestión de fiestas favoritas

Desde el menú, el usuario entra a **Fiestas favoritas**, lo que abre la **Lista de fiestas favoritas**.

Dentro de esta sección, el usuario puede realizar varias acciones:

- **Marcar fiesta como favorita:**
 - El usuario selecciona una fiesta en la lista general o desde el detalle, y puede marcarla como favorita.
 - El **Sistema** guarda este cambio, lo que se refleja en la **Lista de fiestas favoritas**.
- **Desmarcar fiesta favorita:**
 - El usuario puede quitar una fiesta de su lista de favoritas.
 - El **Sistema** actualiza y guarda el cambio.
- **Seleccionar fiesta:**
 - Desde la lista de favoritas, puede acceder directamente al detalle de la fiesta (**Información de la fiesta**).

3. Rol del sistema

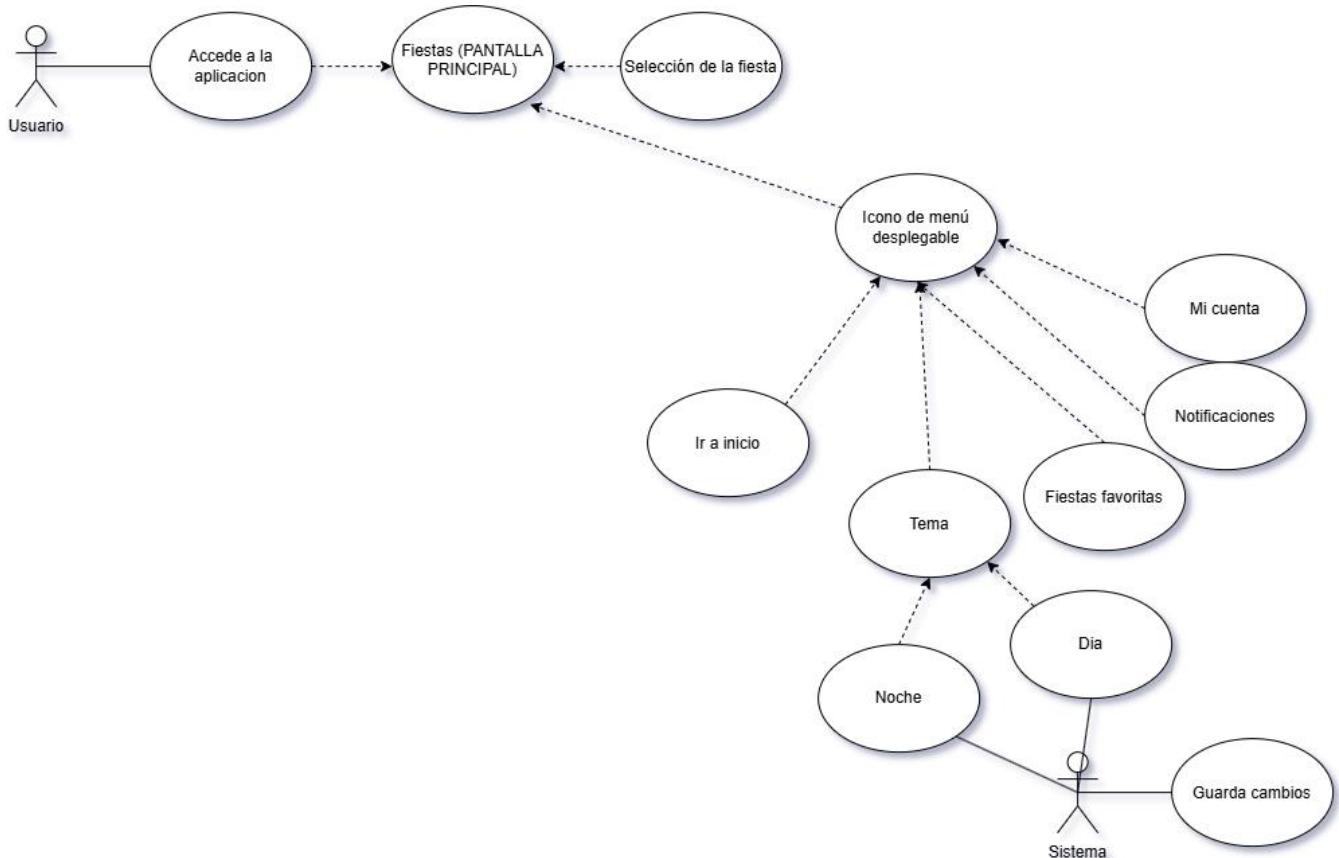
El **Sistema** automatiza las acciones de guardado cada vez que el usuario marca o desmarca una fiesta como favorita, asegurando que los cambios queden reflejados de forma inmediata y persistente (Firebase).

Esto mejora la experiencia del usuario porque:

- No necesita acciones adicionales para guardar manualmente.
- Se garantiza que las favoritas estén siempre actualizadas, incluso tras reiniciar la app.

4. Navegación transversal y experiencia de usuario

- El menú desplegable permite volver fácilmente al inicio, gestionar favoritos, cambiar el tema, gestionar la cuenta y notificaciones.
- El flujo de favoritos está totalmente integrado, haciendo muy fácil marcar/desmarcar y consultar rápidamente tus eventos preferidos.
- Desde cualquier favorita, puedes saltar a la **información detallada** de la fiesta, cerrando el ciclo de exploración y personalización.



Este diagrama detalla el **flujo de personalización del tema visual** de la aplicación, permitiendo al usuario elegir entre modo **Día** y **Noche** a través del menú de la app.

1. Acceso y navegación principal

- El **Usuario** comienza accediendo a la aplicación, entrando a la **Pantalla principal de fiestas**.
- Desde la pantalla principal puede:
 - **Seleccionar una fiesta** para ver más detalles.
 - Acceder al **icono de menú desplegable**, que centraliza la navegación.

2. Menú desplegable y opciones de navegación

El menú desplegable, como en los diagramas anteriores, ofrece varias rutas:

- **Ir a inicio** (volver a la pantalla principal).
- **Tema** (opción de personalización del aspecto visual).
- **Fiestas favoritas** (acceso a la lista personalizada).
- **Notificaciones** (gestión de avisos).
- **Mi cuenta** (gestión del perfil).

3. Personalización del tema visual

Al seleccionar la opción **Tema** desde el menú desplegable, el usuario puede elegir entre:

- **Día** (modo claro)
- **Noche** (modo oscuro)

Cuando el usuario elige uno de los modos:

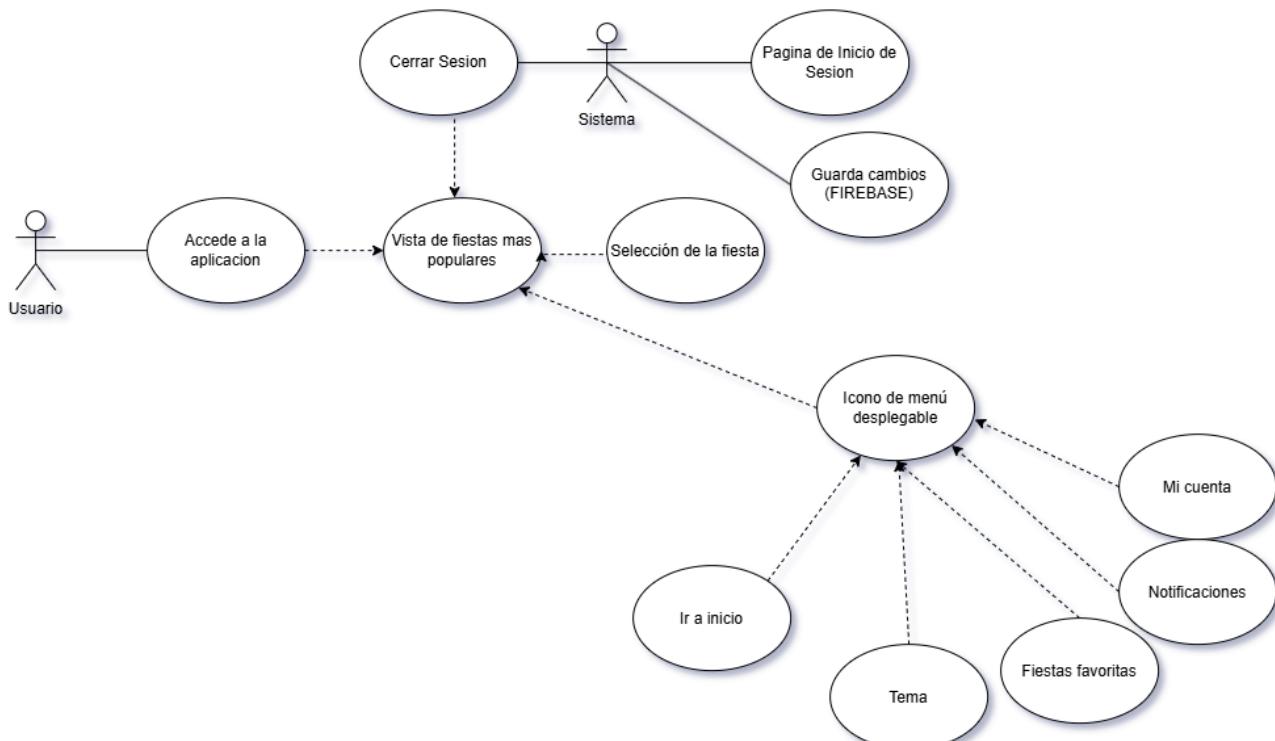
- El **Sistema** registra la selección y **guarda los cambios**.
- El cambio de tema se aplica de manera inmediata en la interfaz, mejorando la experiencia y adaptándose a las preferencias o necesidades visuales del usuario (por ejemplo, menor fatiga visual en modo noche).

4. Rol del sistema

El **Sistema** automatiza el guardado de la preferencia elegida, asegurando que se mantenga, aunque el usuario vuelva a iniciar sesión o utilice otro dispositivo sincronizado.

5. Integración en el flujo general de la app

Esta funcionalidad de personalización está perfectamente integrada en el menú central de navegación, igual que el resto de opciones principales, permitiendo acceder a ella desde cualquier lugar de la app y aportando coherencia de experiencia de usuario.



Este diagrama detalla el flujo de **inicio y navegación principal** de la aplicación, resaltando especialmente la funcionalidad de **cerrar sesión** y el uso de Firebase como sistema de persistencia. Además, introduce la “vista de fiestas más populares” como pantalla principal tras el acceso.

1. Acceso a la aplicación y navegación principal

- El **Usuario** inicia la app, lo que le lleva a la **Vista de fiestas más populares (Fiestas en la app)**. Esta pantalla es la principal y presenta los eventos o fiestas con más interacción o interés dentro de la aplicación.
- Desde la **vista de fiestas más populares**, el usuario puede:
 - Realizar la **selección de una fiesta** para ver sus detalles.
 - Acceder al **ícono de menú desplegable**, desde donde puede navegar a:
 - **Mi cuenta** (gestión de perfil y seguridad).
 - **Notificaciones** (gestión de avisos y mensajes).
 - **Fiestas favoritas** (acceso a sus eventos preferidos).
 - **Tema** (cambiar el modo Día/Noche).
 - **Ir a inicio** (volver a la pantalla principal de fiestas).

2. Cierre de sesión

- El usuario puede **cerrar sesión** desde esta vista.
- Al cerrar sesión, el **Sistema** ejecuta varias acciones:
 - Lleva al usuario a la **Página de inicio de sesión**, donde puede volver a autenticarse.
 - **Guarda cambios en FIREBASE**, asegurando que cualquier modificación pendiente (por ejemplo, marcadores de favoritos, cambios de tema, notificaciones, etc.) quede registrada antes de terminar la sesión.

3. Integración del menú desplegable

Como en los diagramas anteriores, el **ícono de menú desplegable** centraliza la navegación, permitiendo al usuario acceder a las secciones clave de configuración y personalización, manteniendo así una experiencia de usuario consistente.

4. Rol del sistema y persistencia de datos

- El **Sistema** automatiza la gestión de cierre de sesión y el guardado de cambios.
- **Firebase** aparece como plataforma de backend encargada de persistir la información del usuario, asegurando que la experiencia y las preferencias se mantengan, incluso después de salir y volver a entrar en la app.

5. Experiencia de usuario

- El flujo es sencillo y directo: el usuario entra, explora las fiestas más populares, navega por detalles, accede a favoritos o preferencias y puede cerrar sesión en cualquier momento.
- Todo esto sucede con persistencia y seguridad gracias al uso de Firebase.

Diagrama de interfaces

Introducción

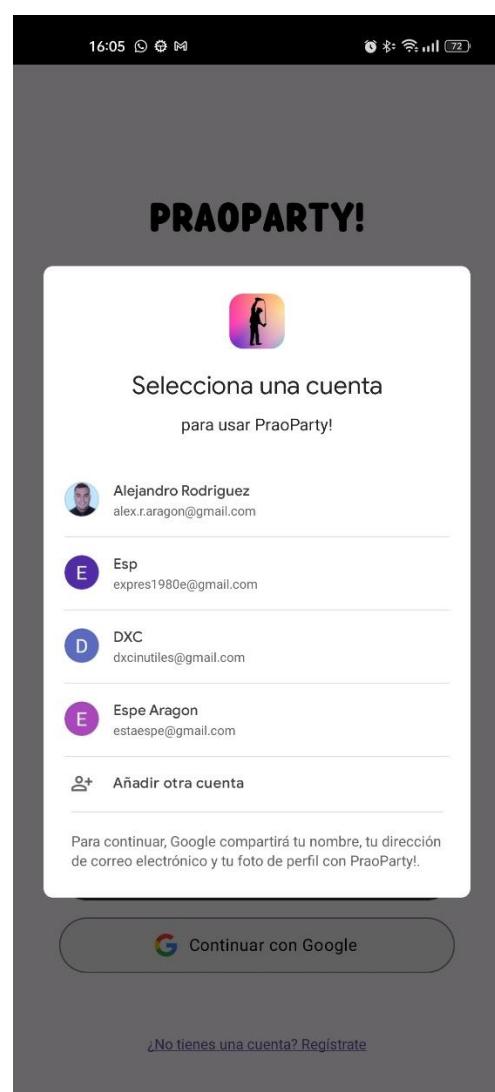
El diagrama de interfaces representa las distintas pantallas que componen la aplicación y cómo se relacionan entre ellas, mostrando el flujo de navegación del usuario.

En el caso de PraoParty!, la aplicación está compuesta por varias vistas principales. Este tipo de representación facilita la comprensión de la estructura de la app y su navegación, tanto para usuarios como para desarrolladores.

Diagramas y descripción

- Pantalla de Inicio de Sesión

En esta pantalla, los usuarios pueden iniciar sesión mediante correo electrónico y contraseña, usar su cuenta de Google o entrar como invitados. También ofrece enlaces para registrarse o recuperar la contraseña. Es el punto de acceso a la aplicación para todos los tipos de usuarios.



- Pantalla de Registro

Permite a los nuevos usuarios crear una cuenta completando su información personal, incluyendo nombre, correo y contraseña. Una vez registrados, son redirigidos automáticamente a la pantalla principal para empezar a usar la aplicación.



Crear cuenta

Correo electrónico

Contraseña

Repite la contraseña

Registrarse

[¿Ya tienes cuenta? Inicia sesión](#)

- Pantalla de Recuperación de Contraseña

Permite a los usuarios restablecer su contraseña ingresando su correo electrónico. Reciben un enlace de recuperación en su bandeja de entrada. Es útil en caso de olvido de la clave de acceso.

16:02



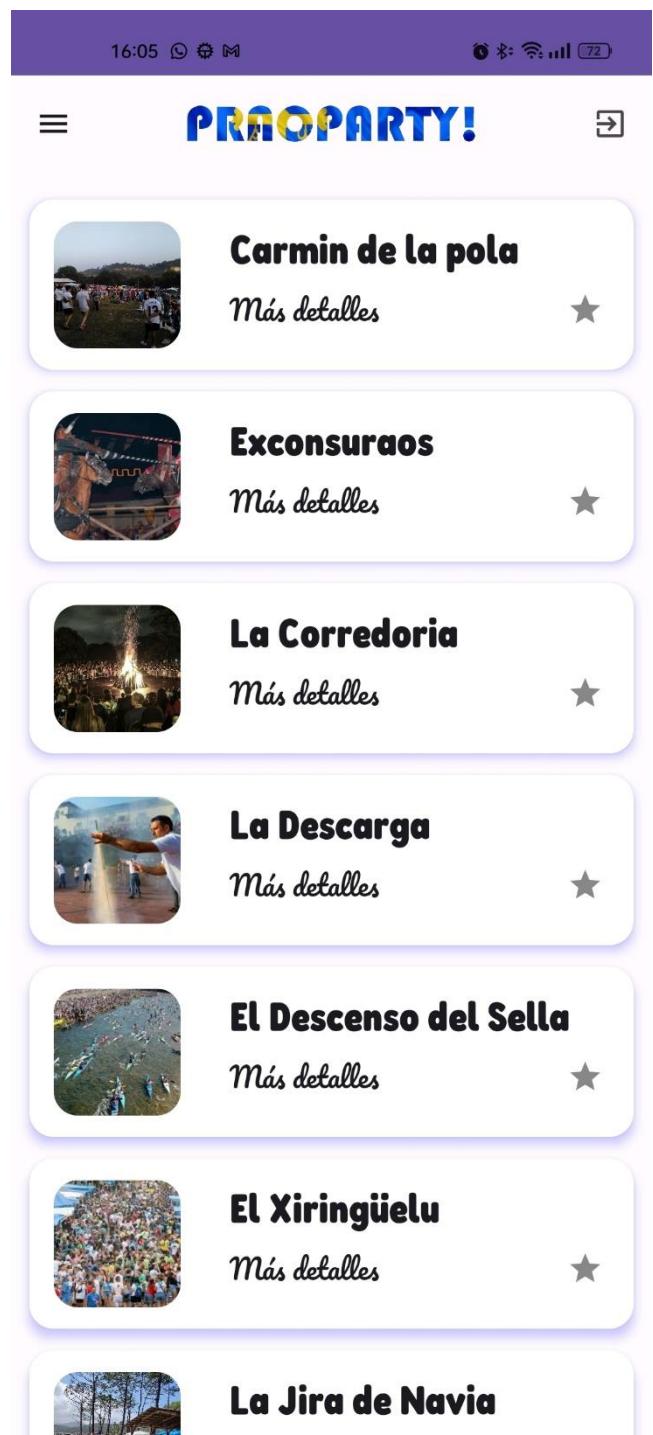
Recuperar contraseña

Correo electrónico

Recuperar contraseña

[¿Te acuerdas de la contraseña? Inicia sesión](#)

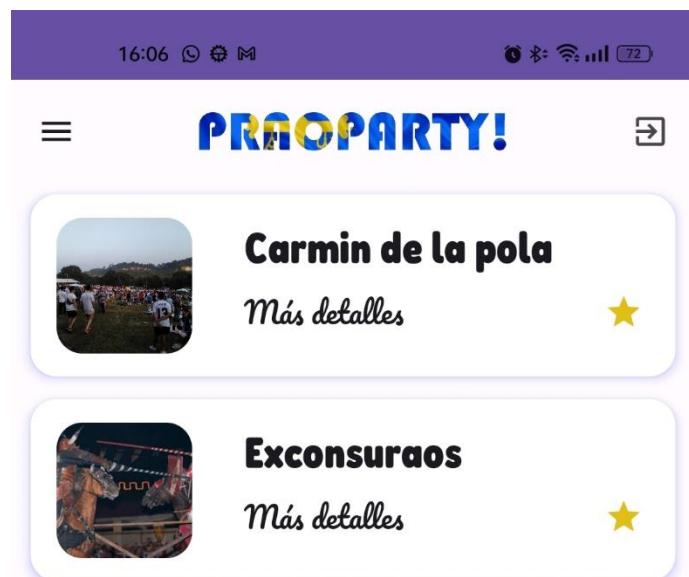
- Pantalla Principal (Listado de Fiestas)
Muestra una lista de las principales fiestas de Asturias, con su imagen, nombre y una opción para ver más detalles. Desde aquí, el usuario puede marcar fiestas como favoritas, acceder al menú lateral y navegar a otras secciones. Es el centro de la experiencia de usuario en la app.



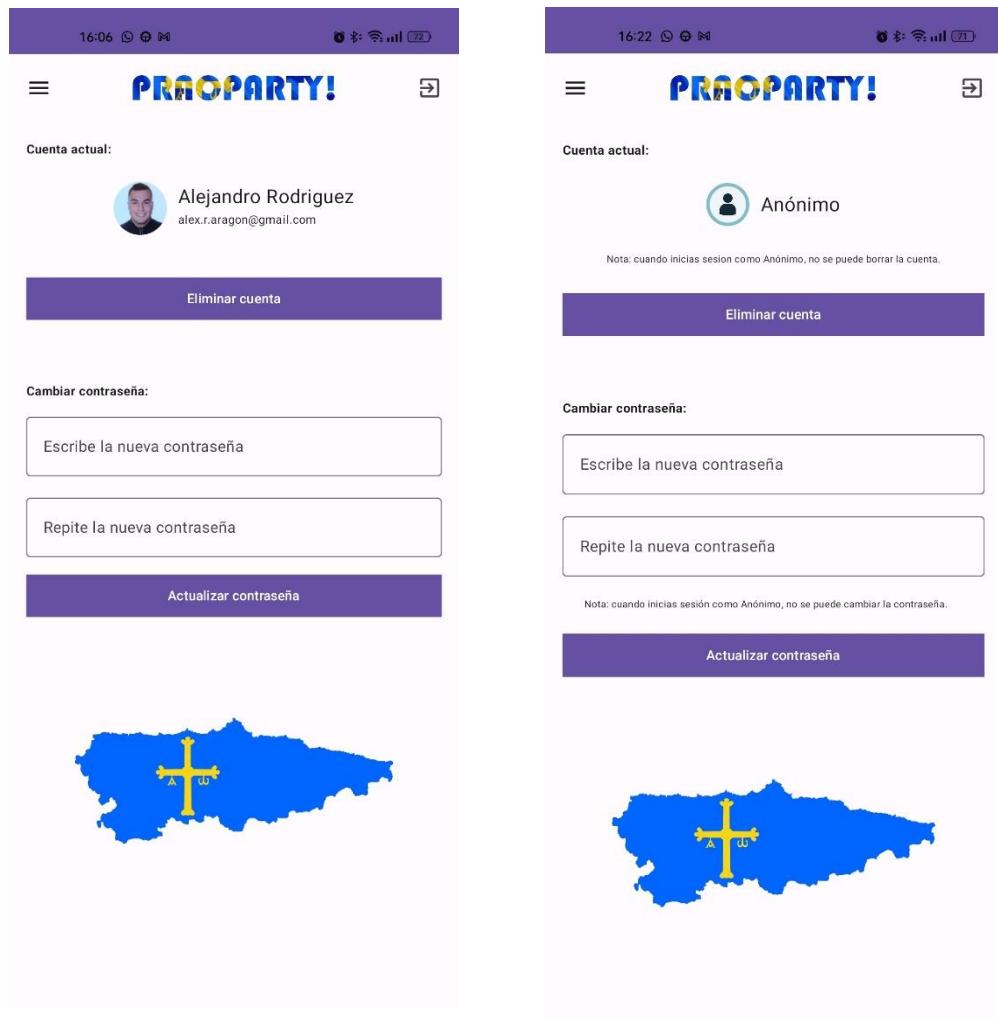
- Pantalla de Detalle de Fiesta (mediante AlertDialog)
Cuando se selecciona una fiesta del listado, se abre un cuadro de diálogo que muestra su imagen en grande, una descripción detallada del evento y un botón para abrir su ubicación en Google Maps. El contenido del diálogo es desplazable para asegurar su correcta visualización en todos los dispositivos.



- **Pantalla de Favoritos**
En esta pantalla se listan únicamente las fiestas que el usuario ha marcado como favoritas. Desde aquí pueden desmarcar fiestas y volver fácilmente a la pantalla principal. Es útil para acceder rápidamente a las celebraciones que más interesan al usuario.



- Pantalla de Cuenta / Perfil de Usuario
Muestra la información del usuario, incluyendo nombre, foto de perfil y correo electrónico. Ofrece opciones para cambiar la contraseña y eliminar la cuenta (si no es una cuenta anónima). Es la sección dedicada a la gestión de cuenta y seguridad.



- **Pantalla de Notificaciones**
Esta pantalla permite al usuario comprobar si las notificaciones de la app están habilitadas en su dispositivo. Se ofrece un acceso directo a los ajustes del sistema para activarlas, y se muestran mensajes informativos sobre su importancia.



PRAOPARTY!



Notificaciones de la app

Activar notificaciones

Al borrar la cuenta

Al cambiar la contraseña

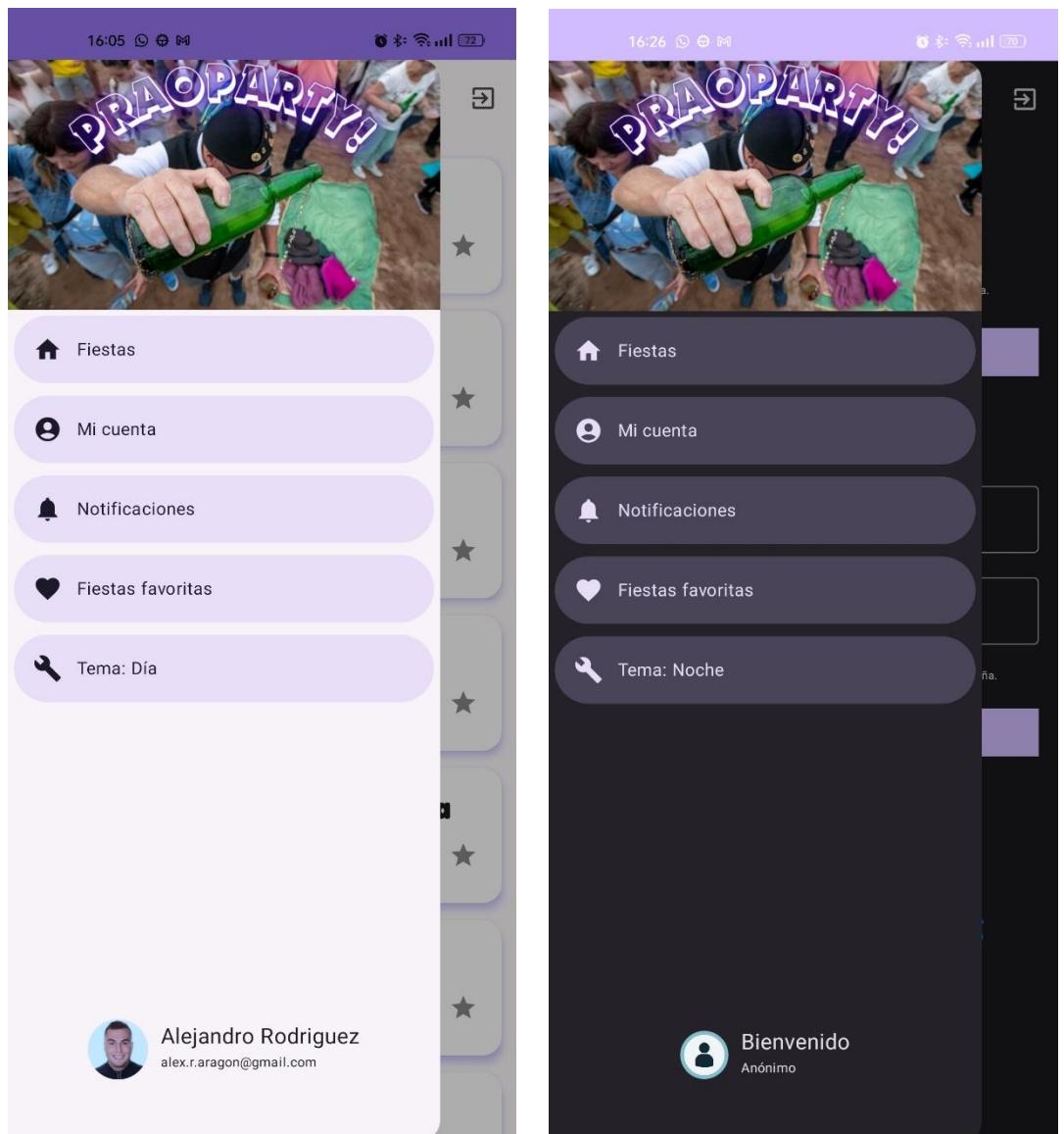
[Restablecer valores predeterminados](#)



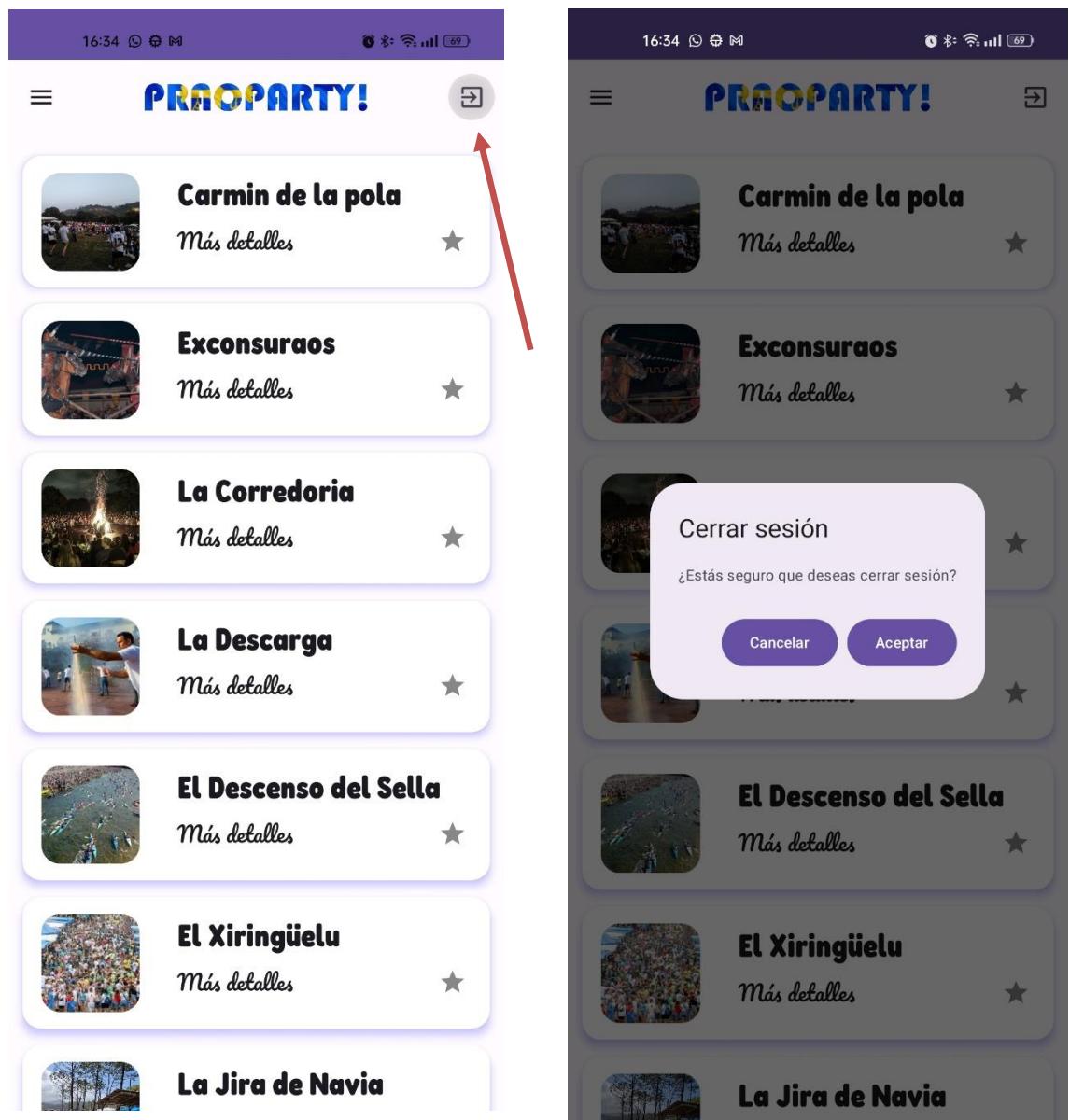
AJUSTES DEL SISTEMA

(comprueba que estén activadas las notificaciones de la app en tu sistema)

- Menú Lateral (Navegación)
Está disponible en todas las pantallas principales y permite acceder rápidamente a las secciones de inicio, favoritos, cuenta, cerrar sesión y cambiar al tema oscuro. Mejora la navegación general dentro de la aplicación.



- **Pantalla de Cierre de Sesión**
Desde cualquier parte de la aplicación, el usuario tiene la opción de cerrar sesión mediante el ícono de la esquina superior derecha. Al seleccionar esta opción, se muestra un diálogo de confirmación para evitar salidas accidentales. Si el usuario confirma, se cierra la sesión actual y se redirige automáticamente a la pantalla de inicio de sesión, garantizando así la seguridad de su cuenta.



Análisis de las estructuras de datos utilizadas (bases de datos, ficheros...)

Introducción

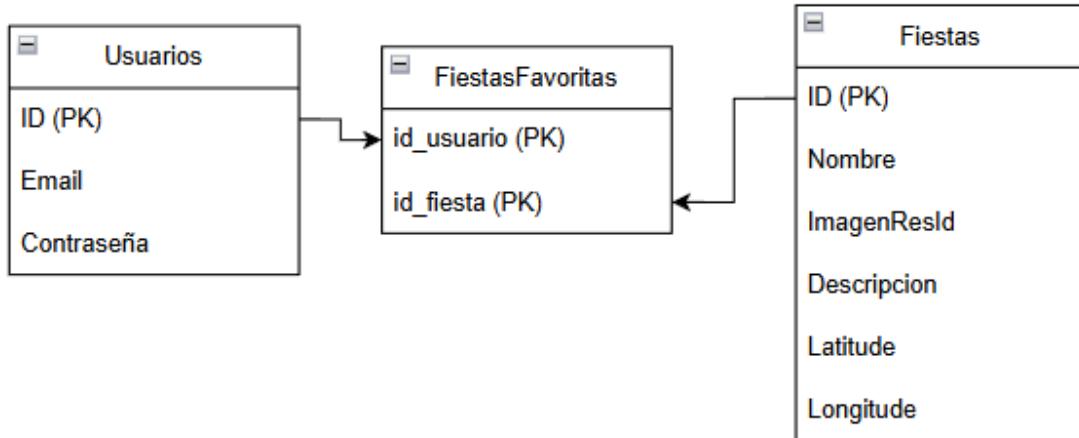
En esta sección se describen las estructuras de datos empleadas por la aplicación PraoParty! para almacenar y gestionar la información. La app hace uso tanto de servicios en la nube (Firebase) como de almacenamiento local para mantener la experiencia del usuario de forma eficiente y segura.

Se han utilizado las siguientes tecnologías y estructuras:

- Firebase Authentication
 - Se utiliza para registrar, autenticar e identificar a los usuarios que acceden a la aplicación.
 - Permite gestionar sesiones de usuario de forma segura.

- Firestore
 - Se utiliza para guardar en la nube la lista de fiestas favoritas marcadas por el usuario.
 - Esto permite que, al volver a entrar, sus favoritos se mantengan utilizando una base de datos externa.
- Estructuras internas
 - Clases de datos (data class) en Kotlin para representar:
 - Fiestas (nombre, descripción, imagen, ubicación...)
 - Listas (mutableListOf) para manejar el estado dinámico de la interfaz.

Diagramas y descripción



La gestión de las fiestas favoritas no se realiza mediante una base de datos relacional tradicional (como MySQL o SQLite), sino que se apoya en el sistema **NoSQL** proporcionado por **Firebase Firestore**. A continuación, se explica de forma detallada cómo funciona este almacenamiento y organización de datos.

Firebase Firestore organiza la información en **colecciones** y **documentos**. No existen tablas ni filas como en los modelos relacionales, sino que cada **colección** contiene varios **documentos** (que vendrían a ser como registros), y cada documento puede contener datos y otras subcolecciones.

En PraoParty!, la estructura que se utiliza para almacenar las fiestas favoritas es la siguiente:

- **Colección “usuarios”:**
Aquí se almacena un documento para cada usuario registrado en la aplicación. El identificador de cada documento suele ser el mismo que el uid del usuario en Firebase Authentication.
- **Subcolección “favoritos” dentro de cada usuario:**
Cada usuario tiene, dentro de su documento, una subcolección llamada

“favoritos”. En esta subcolección se almacenan documentos que representan cada fiesta que ese usuario ha marcado como favorita.

- **Colección “fiestas”:**

Aquí se almacenan los datos completos de cada fiesta disponible en la aplicación.

En resumen, PraoParty! utiliza Firebase Firestore para almacenar las fiestas favoritas de cada usuario. Cada usuario autenticado en la app dispone de una subcolección “favoritos” dentro de su propio documento en la colección “usuarios”. Cada documento de la subcolección representa una fiesta favorita, permitiendo una gestión eficiente y escalable, adaptada a la naturaleza NoSQL de Firestore y asegurando privacidad y rendimiento.

Fase de pruebas

Pruebas de validación de código

Esta sección recoge las pruebas realizadas durante el desarrollo de PraoParty! para garantizar la correcta funcionalidad de la aplicación, tanto a nivel de lógica de negocio como de experiencia de usuario. Se han llevado a cabo pruebas funcionales, de validación de datos, de navegación entre pantallas y de compatibilidad entre dispositivos.

Las siguientes pruebas se realizaron durante el desarrollo para verificar que la lógica interna de la aplicación funcionaba de acuerdo a los requisitos definidos:

- Navegación y flujo general
 - Se comprobó el correcto funcionamiento del flujo entre las distintas pantallas: desde la pantalla de inicio de sesión (LoginScreen), pasando por la pantalla principal con la lista de fiestas (HomeScreen), hasta el detalle de cada fiesta (DetailScreen) y la sección de favoritos (FavoritosScreen).
 - Se validó que el botón de retroceso y la navegación interna no generaban cierres inesperados ni comportamientos erráticos.
- Validación de formularios y errores
 - En la pantalla de login se validaron los siguientes casos:
 - Campo de email vacío o con formato incorrecto: muestra mensaje de error.
 - Campo de contraseña vacío: muestra mensaje de error.
 - Usuario no registrado: devuelve error desde Firebase Authentication.
 - Usuario válido: permite el acceso y redirige correctamente.
 - Se implementaron try-catch y estructuras condicionales para gestionar correctamente los errores lanzados por Firebase.
- Carga y visualización de datos

- Se probó la conexión con Firebase Firestore para asegurar que las fiestas están como favoritas y visualizan correctamente en la lista.
 - Se añadieron controles para evitar cuelgues si no se recuperan datos o si la lista está vacía.
- Gestión de favoritos
 - Se comprobó que al marcar una fiesta como favorita, esta se añade correctamente a una lista local persistente.
 - Al desmarcar una fiesta, desaparece de la lista de favoritos de inmediato y sin errores.
 - Se validó la persistencia de favoritos tras cerrar y volver a abrir la aplicación.
- Control de sesión
 - Se comprobó que el usuario permanece autenticado mientras no cierre sesión manualmente.
 - Si el usuario ya estaba autenticado, se accede directamente a la pantalla principal sin necesidad de volver a iniciar sesión.
- Gestión de estados en Compose
 - Se probaron estados dinámicos con `mutableStateListOf` para actualizar listas en tiempo real sin reiniciar la app.
 - Se validó que los recomposables se actualizan correctamente al cambiar los datos.

Pruebas de implantación en distintos soportes

Una vez validado el código funcional, se realizaron pruebas en varios entornos para asegurar que la aplicación se comporta correctamente en distintos dispositivos y configuraciones.

- Pruebas en emuladores
 - Se ejecutaron emuladores Android en distintos tamaños y resoluciones (Pixel 6 Pro, Pixel 7 Pro y Resizable Experimental).
 - Se verificó el comportamiento visual de la app en modo retrato y apaisado.
- Pruebas en dispositivos físicos
 - Se probó la app en un smartphone real (OPPO A74, Samsung Galaxy y A54 5G, Redmi note 13).
 - Se comprobó el rendimiento, carga de fiestas, fluidez en la navegación y uso de recursos.

- Modo oscuro y adaptabilidad
 - Se validó la compatibilidad con el modo oscuro del sistema.
 - La interfaz se adapta sin problemas según la configuración del sistema operativo.
- Compatibilidad de versiones
 - La app se diseñó y probó para funcionar desde Android 7.0 (API 24) en adelante.
 - No se observaron problemas de compatibilidad.

IMPLEMENTACIÓN

Tecnologías utilizadas en el desarrollo del proyecto

El desarrollo de la aplicación PraoParty! se ha llevado a cabo utilizando un conjunto de tecnologías modernas y adecuadas para el desarrollo de aplicaciones móviles en Android. A continuación, se detallan las principales herramientas y tecnologías utilizadas:

- Lenguaje de programación
 - Kotlin:
 - Lenguaje principal utilizado para el desarrollo de la aplicación. Ofrece una sintaxis moderna, segura y concisa, ideal para proyectos en Android. Su interoperabilidad con Java y su soporte oficial por parte de Google lo convierten en una opción óptima.
- Framework y arquitectura
 - Jetpack Compose:
 - Framework de diseño declarativo moderno de Android para construir interfaces de usuario. Permite una mayor flexibilidad y menos código que las vistas XML tradicionales. Facilita la creación de interfaces reactivas y adaptables al estado de la aplicación.
- Servicios BackEnd
 - Firebase Authentication:
 - Servicio utilizado para la autenticación de usuarios mediante correo electrónico y contraseña. Facilita la gestión de sesiones y la seguridad del acceso a funcionalidades personalizadas.
- Firebase Firestore:
 - Base de datos NoSQL en la nube, utilizada para almacenar las fiestas favoritas. Permite acceso rápido, sincronización en tiempo real y escalabilidad.

- Entorno de desarrollo
 - Android Studio:
 - Entorno de desarrollo oficial de Android, utilizado para codificación, ejecución de emuladores, pruebas y gestión de dependencias.
 - Gradle:
 - Sistema de automatización utilizado para compilar el proyecto, gestionar librerías externas y construir versiones de la app.

PUESTA EN MARCHA. DESPLIEGUE

Esta sección aborda los aspectos técnicos de la puesta en marcha de la aplicación PraoParty!, incluyendo los requisitos del sistema, el proceso de compilación, los métodos de distribución, las consideraciones de seguridad, así como la realización de pruebas reales en dispositivos físicos y entornos controlados. Todos estos aspectos han sido fundamentales para asegurar la estabilidad, funcionalidad y escalabilidad de la aplicación.

- Requisitos del sistema:

El objetivo al definir los requisitos del sistema fue asegurar una amplia cobertura de dispositivos Android activos, sin renunciar al uso de funcionalidades modernas ofrecidas por tecnologías actuales como Firebase o Jetpack Compose.

- Versión mínima de Android
 - Se ha establecido Android 7.0 (API 24) como versión mínima requerida.
 - Esta versión fue seleccionada por ser el punto de entrada a muchas APIs modernas (como almacenamiento scoped, optimizaciones de red y gráficos) y ofrecer una base estable y segura.
 - Es compatible con la versión mínima requerida por Jetpack Compose, la cual no es soportada en versiones anteriores a la API 33.
- Requisitos de hardware
 - Procesador ARM64 compatible con Android Runtime (ART).
 - Resolución mínima recomendada de 720p.
 - Conectividad Wi-Fi o datos móviles para sincronización con Firebase.
- Permisos necesarios
 - Acceso a red: para sincronización de datos con Firebase Firestore y autenticación.
 - Permiso de almacenamiento interno: en caso de caché de imágenes o almacenamiento local de datos temporales.

- Proceso de compilación y empaquetado
 - Herramientas utilizadas:
 - Android Studio Flamingo (o superior) con Gradle 7.3+
 - Lenguaje de programación: Kotlin 1.7+
 - Gradle build system, con configuración modular y uso de build.gradle.kts (en caso de uso de Kotlin DSL).

- Configuración del archivo build.gradle:

Se han definido los siguientes aspectos clave:

```
defaultConfig {
    applicationId "com.example.praoparty"
    minSdk 24
    targetSdk 33
    versionCode 1
    versionName "1.0"

    testInstrumentationRunner
    "androidx.test.runner.AndroidJUnitRunner"
    vectorDrawables {
        useSupportLibrary true
    }
}

buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    }
}
```

Además, se han añadido las siguientes dependencias relevantes:

```
dependencies {

    // Core
    implementation 'androidx.core:core-ktx:1.12.0'
    implementation platform('org.jetbrains.kotlin:kotlin-bom:1.8.0')

    // Lifecycle
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.7.0'

    // Jetpack Compose
    implementation 'androidx.activity:activity-compose:1.8.0'
    implementation platform('androidx.compose:compose-bom:2023.10.01')
    implementation 'androidx.compose.ui:ui'
    implementation 'androidx.compose.ui:ui-graphics'
    implementation 'androidx.compose.ui:ui-tooling-preview'
    implementation 'androidx.compose.material3:material3:1.2.1'

    // Firebase
    implementation platform('com.google.firebaseio:firebase-bom:32.2.0')
    implementation 'com.google.firebase:firebase-analytics-ktx'
    implementation 'com.google.firebase:firebase-auth-ktx'
    implementation 'com.google.firebase:firebase-database-ktx'
    implementation 'com.google.firebase:firebase-firebase-ktx'
    implementation 'com.google.firebase:firebase-storage-ktx'
    implementation 'com.google.firebase:firebase-crashlytics-ktx'
    implementation 'com.google.firebase:firebase-config-ktx'
```

```
implementation 'com.google.firebaseio:messaging-ktx'  
  
// Google Services  
implementation 'com.google.android.gms:play-services-auth:20.7.0'  
  
// Navigation  
implementation 'androidx.navigation:navigation-compose:2.7.5'  
  
// Image Loading  
implementation "io.coil-kt:coil-compose:2.4.0"  
  
// UI  
implementation 'androidx.compose.ui:ui:1.6.0'  
}
```

- Generación del APK
 - Se utilizó la función de “Build > Generate Signed Bundle/APK” en Android Studio.
 - El resultado es un archivo .apk listo para instalar en cualquier dispositivo que cumpla los requisitos del sistema.
- Distribución de la aplicación
 - Distribución actual (manual)
 - La distribución actual se realiza fuera de tiendas oficiales, mediante:
 - ❖ Instalación por ADB (Android Debug Bridge).
 - ❖ Envío del APK por email, Google Drive o almacenamiento USB.
- Consideraciones de seguridad
 - Gestión de usuarios:
 - Se emplea Firebase Authentication para el registro e inicio de sesión.
 - El acceso a funcionalidades clave como guardar favoritos o navegación personalizada está protegido tras autenticación.
 - Protección de datos:
 - Firebase Firestore respalda la autenticación mediante reglas de seguridad basadas en el UID del usuario.
 - Los favoritos se almacenan de forma local y privada, mediante mecanismos como DataStore.
- Pruebas en dispositivos reales
 - Entornos de prueba:
Se ha testeado la app en:
 - Emuladores Android x86 (API 26, 29, 33).
 - Dispositivos físicos (Pixel 6 Pro, Samsung Galaxy A51, OPPO CPH, Pixel 7 Pro)
 - Modo oscuro y claro.
- Escenarios probados

- Autenticación correcta / incorrecta.
- Persistencia de sesión tras cerrar la app.
- Guardado y borrado de favoritos.
- Funcionamiento sin conexión: los favoritos se mantienen, aunque Firebase no esté accesible.
- Compatibilidad UI en pantallas grandes y pequeñas.
- Resultados
 - Todos los flujos básicos se ejecutaron correctamente.
 - No se detectaron cuelgues ni errores críticos.
 - La carga desde Firebase fue fluida.

- Conclusión técnica del despliegue:

La puesta en marcha de PraoParty! ha sido planificada con una arquitectura clara, moderna y escalable. El uso de herramientas como Jetpack Compose y Firebase ha facilitado una implementación robusta y segura. La aplicación está completamente operativa y preparada para un despliegue oficial, tanto manual como automatizado, cumpliendo estándares técnicos actuales de desarrollo móvil profesional.

APLICACIÓN EN EL ÁMBITO EMPRESARIAL

Idea de negocio

PraoParty! es una aplicación móvil desarrollada en Android con el objetivo de digitalizar el acceso a la información sobre fiestas populares, especialmente aquellas celebradas al aire libre en entornos rurales del norte de España (conocidas como “fiestas de prao”). Estas fiestas suelen carecer de una plataforma centralizada que informe a los asistentes, lo que crea una oportunidad de negocio clara.

La app ofrece funcionalidades como autenticación de usuarios mediante Firebase, marcación de favoritos, consulta de fiestas, navegación intuitiva y soporte para futuras funcionalidades como notificaciones personalizadas.

Problemas detectados:

- Actualmente, la difusión de estas fiestas se hace mayoritariamente a través de carteles físicos, boca a boca o publicaciones dispersas en redes sociales. Esto genera:
 - Pérdida de visibilidad para muchas fiestas.
 - Dificultad para planificar asistencia (no hay horarios, ubicaciones exactas ni actualizaciones).

- Falta de profesionalización en la organización y promoción.

Solución propuesta:

PraoParty! actúa como plataforma centralizada donde los usuarios pueden:

- Consultar fiestas por fecha o ubicación.
- Ver detalles como descripción, imágenes o ubicación exacta.
- Marcar fiestas como favoritas para planificar su asistencia.
- Acceder con cuenta propia mediante login seguro vía Firebase.

Además, la estructura modular de la app permite escalar hacia funcionalidades como:

- Notificaciones personalizadas.
- Integración de mapas interactivos.
- Sistema de eventos destacados o promocionados.
- Publicidad geolocalizada.

Modelo de negocio (fases):

1. Fase inicial (validación de mercado):

- App gratuita para usuarios y organizadores.
- Inclusión de fiestas gestionadas por el administrador de la app.

2. Fase de crecimiento:

- Registro de organizadores para subir sus propias fiestas.
- Planes de suscripción para organizadores (gratis, básico, premium).
- Posicionamiento destacado mediante pago.

3. Fase de monetización:

- Publicidad segmentada: bares, DJs, food trucks locales.
- Sistema de donaciones o crowdfunding por fiesta.
- Venta de entradas (para fiestas privadas o con concierto).

Público objetivo:

- Jóvenes y adultos entre 16 y 55 años que asisten regularmente a fiestas populares.
- Comisiones de fiestas, peñas, asociaciones juveniles y ayuntamientos.
- Comercios locales interesados en patrocinar fiestas o productos.

Ventajas competitivas:

- Especialización: centrado exclusivamente en fiestas populares rurales.
- Simplicidad: interfaz clara y fácil de usar, sin funcionalidades innecesarias.
- Escalabilidad: arquitectura técnica preparada para crecer sin rehacer la app.
- Integración con Firebase: permite crecimiento seguro y sincronización real-time.

Planificación de necesidades de RRHH

Objetivo:

- Determinar los perfiles profesionales necesarios para desarrollar, mantener y hacer crecer la aplicación PraoParty!, teniendo en cuenta una planificación por fases y el aprovechamiento eficiente de recursos humanos.

Fase 1: Desarrollo inicial (prototipo funcional)

Esta fase ya está parcialmente cubierta por el trabajo personal realizado, pero si se quisiera replicar este desarrollo en un entorno profesional, serían necesarios los siguientes perfiles:

Perfil profesional	Nº Personas	Funciones principales
Desarrollador Android (Kotlin + Compose)	1	Diseño e implementación de pantallas, lógica de negocio, conexión con Firebase.
Diseñador UI/UX	1	Diseño visual de pantallas, flujo de usuario, iconografía y branding.
Backend Firebase Admin (opcional)	1/2	Configuración de Firestore, reglas de seguridad, autenticación y estadísticas.

Fase 2: Validación de mercado y expansión funcional

En esta etapa se pretende pasar de una app funcional a un producto mantenido y en crecimiento. Esto implica pulir la experiencia, añadir funcionalidades y empezar a captar usuarios reales.

<u>Perfil profesional</u>	<u>Nº Personas</u>	<u>Funciones principales</u>
Desarrollador Android	1-2	Mantenimiento de código, nuevas features, resolución de bugs.
Community Manager / Marketing Digital	1	Gestión de redes sociales, promoción de la app, captación de organizadores.
QA Tester (calidad)	1	Pruebas funcionales y de usabilidad en diferentes dispositivos.
Administrador Firebase	1	Gestión de base de datos, analítica, control de acceso.

Fase 3: Escalado y mantenimiento

Si PraelParty se lanza al mercado y empieza a crecer, hará falta un equipo más amplio para escalar tanto a nivel técnico como comercial:

<u>Perfil profesional</u>	<u>Nº Personas</u>	<u>Funciones principales</u>
Jefe de proyecto	1	Supervisión técnica, coordinación de tareas, planificación estratégica.
Desarrollador Android Senior	1	Mejora del rendimiento, arquitectura avanzada, nuevas integraciones.
Desarrollador Backend	1	(Solo si se decide sustituir Firebase por servidor propio).
Diseñador UI/UX	1	Evolución de diseño, diseño adaptativo para tablets y nuevas secciones.
Responsable de atención al usuario	1	Resolución de dudas, incidencias, soporte por email o en redes.
Legal / Protección de datos	1 externo	Asesoría para cumplir con la RGPD y protección de datos si se recopila información personal.

Observaciones:

- Gracias al uso de Firebase, se minimiza el personal técnico requerido en backend y administración de servidores.
- El uso de Jetpack Compose agiliza el diseño de interfaces, permitiendo que un solo desarrollador gestione más partes de la app.
- Con una planificación lean, PraoParty! puede escalar progresivamente sin requerir una plantilla grande al principio.

Planificación de necesidades de producción

Objetivo:

- Planificar los medios necesarios para que la app PraoParty pase de ser un proyecto técnico a una aplicación disponible en el mercado, accesible para usuarios y mantenida de forma continua.

Recursos materiales y técnicos

Recurso	Necesidad	Observaciones
Ordenadores con Android Studio	Sí	Mínimo un equipo con buena capacidad de RAM (16 GB) y SSD para compilar y ejecutar emuladores.
Dispositivos Android de prueba	Sí	Smartphones con diferentes versiones de Android para pruebas reales.
Conexión a internet estable	Sí	Requisito indispensable para Firebase, sincronización y testing.
Licencia de desarrollador Google Play	Sí	Pago único de 25 USD para publicar la app oficialmente.
Cuenta de Firebase	Sí	Uso gratuito en plan Spark; solo se paga si se supera el límite de uso.
Software de diseño (Figma/Adobe XD)	Opcional	Para mejorar los prototipos UI en etapas de expansión.
Herramientas de analítica (Firebase Analytics, Google Analytics)	Recomendado	Para entender el comportamiento del usuario y mejorar la app.

Producción inicial:

Duración estimada de desarrollo del MVP (mínimo producto viable):

De 2 a 3 meses (con un solo desarrollador a tiempo parcial)

Tareas clave en esta fase:

- Programación de las pantallas principales: login, lista de fiestas, detalle, favoritos.
- Configuración de Firebase (Firestore + Authentication).
- Validación del flujo completo: desde login hasta guardar favoritos.
- Pruebas y ajustes básicos para garantizar estabilidad.

Mantenimiento y actualizaciones

Una vez publicada la aplicación, se requiere una producción continua enfocada en:

- Corrección de errores detectados en dispositivos o versiones específicas.
- Mejora de rendimiento en dispositivos de gama baja.
- Actualización de contenido (fiestas nuevas, imágenes, etc.).
- Respuesta a comentarios y sugerencias de usuarios.

Frecuencia recomendada de actualizaciones: cada 2 a 4 semanas en fases activas.

Escalado y automatización

A medio/largo plazo, se recomienda implementar:

- Backend de apoyo (opcional) para gestionar fiestas desde un panel web para organizadores.
- Automatización de builds y testing con GitHub Actions y Firebase Test Lab.
- Sistema de moderación de contenido si se permite a terceros publicar fiestas.

Producción no técnica

- Gestión de redes sociales: crear perfiles en Instagram, TikTok, etc., para captar usuarios.
- Creación de contenido: vídeos cortos de promoción, tutoriales sobre cómo usar la app.
- Campañas promocionales: colaboración con organizadores reales de fiestas.

Conclusión:

La producción de PraoParty se puede abordar inicialmente con recursos muy reducidos, gracias al uso de herramientas gratuitas y escalables como Firebase y Jetpack Compose. Con una planificación adecuada, es posible lanzar la aplicación al mercado sin una infraestructura compleja ni grandes inversiones, e ir creciendo según la demanda y la respuesta del público.

Planificación de la Prevención de Riesgos Laborales

Riesgo potencial	Descripción / Contexto	Medidas preventivas
Fatiga visual	Uso continuado de pantallas	Descansos periódicos cada 1–2 horas. Ajuste de brillo y contraste.
Problemas musculoesqueléticos (espalda, cuello, muñeca)	Postura prolongada en escritorio	Uso de sillas ergonómicas, elevadores de portátil, teclado externo.
Estrés / carga mental	Plazos de entrega ajustados o bugs críticos	Planificación realista. Uso de metodologías ágiles. Descansos regulares.
Sobrecarga de trabajo en entorno multitarea	Varias tareas simultáneas (dev, tests, diseño)	Priorización de tareas. Comunicación clara.
Riesgo eléctrico o fallos en equipos	Cables sueltos, enchufes múltiples	Correcta instalación eléctrica. Revisiones periódicas.
Riesgos psicosociales (aislamiento, trabajo remoto)	Especialmente en modalidad freelance	Promover comunicación diaria, reuniones periódicas.
Seguridad digital	Acceso no autorizado a cuentas Firebase u otros servicios	Contraseñas seguras, doble autenticación, control de accesos.

Objetivo:

- Establecer medidas para garantizar la salud, seguridad y bienestar del equipo de trabajo durante el desarrollo y mantenimiento de la aplicación PraoParty, cumpliendo con la Ley de Prevención de Riesgos Laborales (LPRL).

Evaluación de riesgos asociados

Medidas organizativas y ergonómicas

- Horario flexible, pero con control de pausas para evitar fatiga.
- Espacio de trabajo ventilado, bien iluminado y con mobiliario ergonómico.
- Uso de aplicaciones como Stretchly o Pomodoro timers para promover descansos.
- Formación básica en ciberseguridad para cualquier miembro del equipo.

- Sistemas de backup automático del código (ej. GitHub) para evitar pérdida de datos.

Recomendaciones en trabajo remoto

Dado que este tipo de desarrollo suele realizarse en modalidad híbrida o 100% remota:

- Establecer rutinas de trabajo para evitar jornadas excesivas.
- Comunicación diaria por canales como Slack, Discord o Google Meet.
- Realizar sesiones informales tipo coffee break virtual para cuidar la salud mental.
- Planificar objetivos semanales para reducir presión diaria.

Conclusión:

Aunque los riesgos en el desarrollo de software no son físicos en la mayoría de los casos, sí existen factores ergonómicos, visuales y mentales a tener en cuenta. La correcta planificación y concienciación del equipo evita lesiones, mejora el rendimiento y contribuye a un entorno profesional saludable, incluso en equipos pequeños o unipersonales.

Previsiones de inversión y financiación

Objetivo:

- Estimar la inversión inicial necesaria para lanzar la aplicación al mercado y definir posibles vías de financiación para asegurar su sostenibilidad a corto y medio plazo.

Inversión inicial estimada

Concepto	Coste estimado (€)	Observaciones
Ordenador de desarrollo (si no se posee)	1.000 €	Portátil con 16 GB RAM, SSD y CPU moderna.
Dispositivos Android para pruebas	400 €	Dos dispositivos con diferentes versiones de Android.
Licencia Google Play Developer	25 € (pago único)	Obligatorio para publicar en Play Store.
Conexión a Internet	20 €/mes	Suponiendo tarifa básica doméstica.
Diseño gráfico e identidad visual (logo, icono, etc.)	150 €	Puede externalizarse por encargo único.

Concepto	Coste estimado (€)	Observaciones
Marketing inicial (redes, carteles, publicidad)	300–500 €	Campañas locales para dar a conocer la app.
Servicios externos (dominio, correo, alojamiento web auxiliar)	50–100 €/año	Si se crea landing page o panel de gestión web.
Margen de imprevistos y contingencia	200 €	Posibles ajustes, sustituciones, soporte técnico.

Total inversión estimada inicial: entre 2.000 € y 2.500 €

Costes recurrentes (mantenimiento anual)

- Firebase: plan gratuito (Spark) suficiente para empezar. Escalaría si se supera el uso (25 €/mes en plan Blaze).
- Renovación de dominio: 10 €/año.
- Mantenimiento y actualizaciones técnicas (si se externaliza): 600–1.000 €/año.
- Publicidad digital (Instagram, TikTok, etc.): variable según estrategia (~50–200 €/mes).

Fuentes de financiación posibles

Fuente	Detalle
Financiación personal (autoempleo)	Inversión inicial aportada por el creador del proyecto. Ideal si es un proyecto pequeño sin socios.
Subvenciones locales o regionales	Muchas comunidades autónomas ofrecen ayudas a proyectos tecnológicos o de digitalización cultural.
Ayudas para jóvenes emprendedores	Programas de entidades como INJUVE, Cámara de Comercio, etc.
Crowdfunding	Campaña para financiar el lanzamiento entre usuarios interesados en las fiestas.
Colaboración con ayuntamientos o asociaciones	Possible patrocinio institucional a cambio de promoción de fiestas oficiales.
Socios o inversores privados	Si se busca escalar el proyecto con mayor

Fuente	Detalle
	ambición.

Plan financiero simplificado (primer año)

Mes	Ingresos estimados	Gastos estimados	Comentario
1–3	0 €	2.000 €	Desarrollo, publicación y promoción inicial.
4–6	100–300 €	200–400 €	Publicidad, primeros usuarios, primeras colaboraciones.
7–12	500–1.000 €	400–700 €	Ingresos por patrocinadores u organizadores. Comienzo de retorno de inversión.

Conclusión:

PraoParty! es una aplicación que puede desarrollarse y lanzarse con una inversión inicial moderada, especialmente si se aprovechan recursos gratuitos como Firebase y se optimizan las tareas técnicas. Su monetización no requiere grandes volúmenes de usuarios para empezar a generar ingresos, lo cual la convierte en una propuesta viable a pequeña escala, con posibilidad de escalar según el éxito y la demanda.

CONCLUSIÓN

Valoración personal del trabajo realizado.

El desarrollo de PraoParty ha supuesto una experiencia enriquecedora tanto a nivel técnico como organizativo. A lo largo del proyecto, he podido aplicar y afianzar conocimientos de desarrollo móvil con Android, especialmente utilizando Jetpack Compose, un enfoque moderno y declarativo que ha facilitado la creación de interfaces dinámicas y adaptables.

Uno de los logros principales ha sido la integración de Firebase Authentication y Firestore, lo que me ha permitido manejar usuarios, sesiones y almacenamiento en la nube de forma segura y escalable, sin necesidad de mantener un backend propio. Esta experiencia ha sido especialmente valiosa al tratarse de servicios reales ampliamente utilizados en la industria.

También he trabajado la lógica de navegación, la organización modular del proyecto, la gestión de estados y el diseño de pantallas de forma coherente, siguiendo buenas prácticas de arquitectura.

A nivel personal, ha sido un reto gestionar todas las fases del proyecto por mí mismo: diseño, desarrollo, pruebas, documentación y planificación empresarial. Esta autonomía me ha obligado a tomar decisiones constantemente y a resolver

problemas técnicos de forma práctica, lo cual considero una de las partes más valiosas del proceso.

Además, el hecho de basar la app en un contexto real (las fiestas populares del norte de España) ha aumentado mi motivación, ya que el proyecto tiene potencial para ser útil y aplicable fuera del entorno académico.

Posibles ampliaciones

El proyecto actual sienta las bases de una aplicación funcional y bien estructurada, pero existen múltiples posibilidades de crecimiento, entre las que destacan:

1. **Notificaciones personalizadas:** para avisar al usuario cuando se acerque la fecha de una fiesta que ha marcado como favorita.
2. **Buscador avanzado y filtros por localización:** para encontrar fiestas por municipio, fechas concretas, tipo de música, etc.
3. **Panel web para organizadores:** plataforma externa donde los organizadores puedan registrar fiestas, añadir fotos, editar horarios, etc.
4. **Sistema de comentarios o valoraciones:** para fomentar la participación de los usuarios y generar comunidad.
5. **Versión para iOS:** desarrollada con Flutter o React Native para alcanzar un mayor número de usuarios.
6. **Monetización progresiva:** mediante publicidad local o planes de suscripción para organizadores, como se detalla en la idea de negocio.

Estas ampliaciones permitirían convertir PraoParty en un producto más completo, escalable y rentable, adaptado a las necesidades de los usuarios y del mercado.

BIBLIOGRAFÍA Y WEBGRAFÍA

Se incluyen recursos de documentación oficial, artículos técnicos, tutoriales y herramientas utilizadas.

-----Documentación oficial-----

Google. [Jetpack Compose Documentation](#)
Guía oficial para desarrollar interfaces declarativas en Android.

Firebase. [Firebase Documentation](#)
Documentación oficial de Firebase, especialmente Authentication y Firestore.

Google. [Material Design Guidelines](#)
Recomendaciones visuales para diseño de interfaces modernas.

Android Developers. [Navigation Component](#)
Sistema de navegación recomendado para Jetpack Compose.

Kotlin Lang. [Kotlin Language Documentation](#)
Documentación oficial del lenguaje utilizado

<https://developer.android.com>

<https://firebase.google.com>

<https://stackoverflow.com>

<https://material.io>

<https://developer.mozilla.org/>

-----ENLACES DE YOUTUBE-----

https://www.youtube.com/watch?v=LxABxtwhrDE&list=PL8ie04dqq7_PyM9t_U3qpOuuBw-UoX8Xi - FIREBASE desde CERO en ANDROID con KOTLIN y JETPACK COMPOSE - CURSO COMPLETO en Android Studio

<https://www.youtube.com/watch?v=liuKAmgRYeM> - FIREBASE ANDROID 🔥
Tutorial Android Studio + Firebase Analytics [2020]

https://www.youtube.com/watch?v=liuKAmgRYeM&list=PLNdFk2_brsRcaGhfeeiVkJW72qTYcn_nfQ - Tutorial Firebase para Android

<https://www.youtube.com/watch?v=HYzw8LFvmw4&list=PLS1QuWo1RlbKsL9GqxOLbToLNFFQFJW> - Firebase Tutorial for Android 1 - Getting Started + Add Firebase to Android Project

<https://www.youtube.com/watch?v=t5yyC1XfQrs> - FIRESTORE Android 🔥 BASE de DATOS con Firebase

<https://www.youtube.com/watch?v=qKAB2Q6RBCs> - NUEVO Firebase con Gemini 2.5 Crea APPs GRATIS en segundos 🔥 Tutorial FÁCIL

<https://www.youtube.com/watch?v=1Knr8ECB2sE&list=PLYx38U7gxBf3pmsHVTUwRTIGON6ZIBHi> - Android Firebase - 01 - How Create Firebase Project for Android

<https://www.youtube.com/watch?v=NMOdEHyMXBo> - Android App Development Full Course with PRACTICAL (Beginners to Advanced) | Full Course 2025

https://www.youtube.com/watch?v=XLt_moCoauw - Simple Kotlin App | Android Studio | 2024

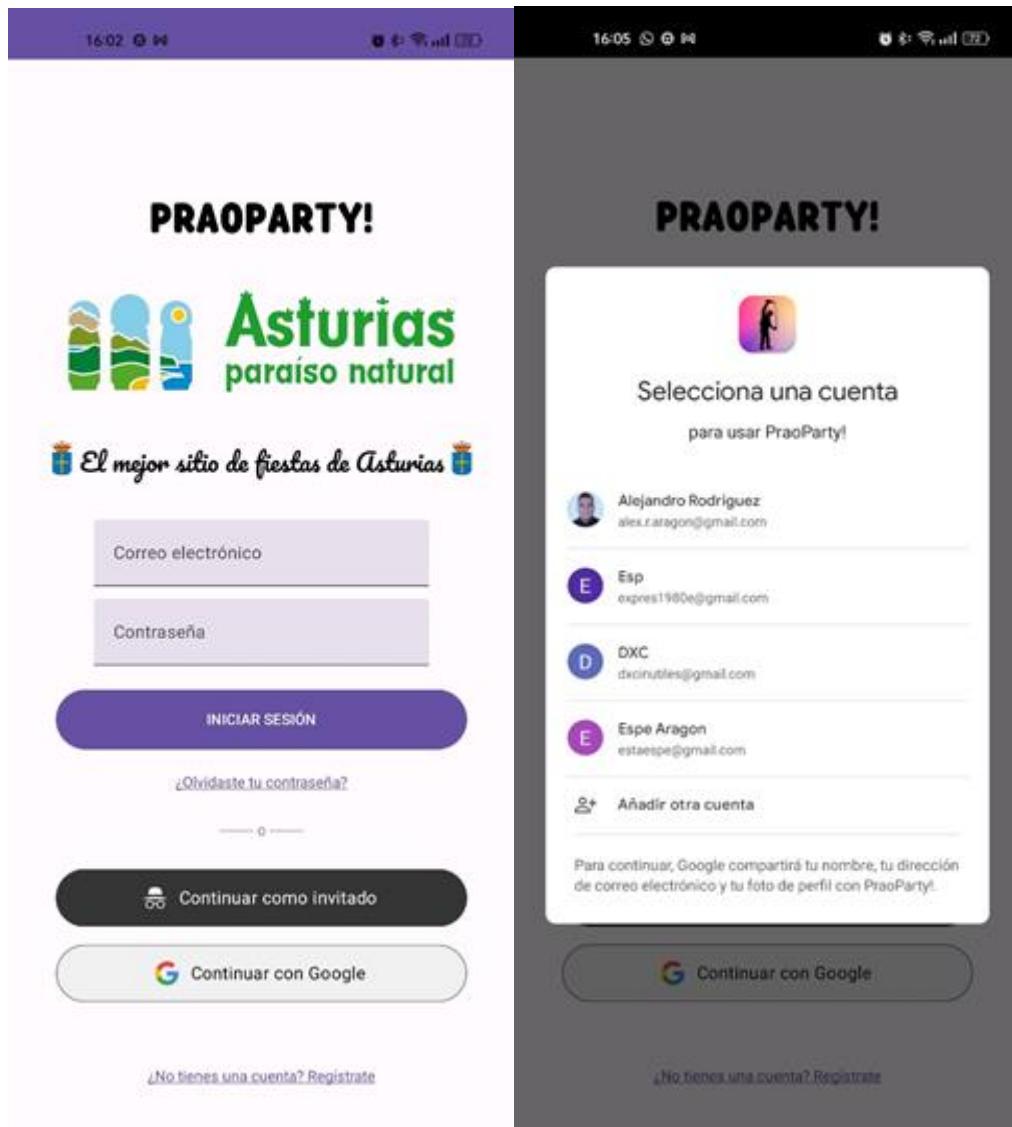
<https://www.youtube.com/watch?v=DX-Cldg3jWY> - Curso de ANDROID Studio con KOTLIN desde Cero COMPLETO 🌟

https://www.youtube.com/watch?v=BQaxPwZWboA&list=PLNdFk2_brsRdYF0FXDtSaGvluzBNHRbNe - ANDROID STUDIO: COMO Crear una APP (para Principiantes)



ANEXOS

I – Manual de usuario



Descripción general:

Esta es la pantalla principal de acceso a la aplicación PraoParty. Representa la primera interacción del usuario con la app y está diseñada para ofrecer múltiples formas de acceso: por correo/contraseña, como invitado o mediante Google. El diseño transmite identidad local con un enfoque cultural y moderno.

Contenido visual y componentes:

Encabezado:

- Texto superior: PRAOPARTY! — en mayúsculas, centrado y destacado como marca.
- Imagen del logotipo: un gráfico representativo de Asturias y el eslogan “Asturias paraíso natural”.
- Subtítulo: "El mejor sitio de fiestas de Asturias" con emojis festivos que refuerzan la temática local.

Campos de entrada:

- Correo electrónico: campo tipo TextField para introducir email.
- Contraseña: campo tipo PasswordField con ocultamiento de caracteres.

Botón de acción principal:

- INICIAR SESIÓN: botón con diseño personalizado (Button en Jetpack Compose) y color morado, que al pulsarlo valida las credenciales introducidas y autentica al usuario mediante Firebase Authentication (con email/password).
- Enlaces secundarios:
 - “¿Olvidaste tu contraseña?”: redirecciona al sistema de recuperación de contraseña de Firebase.
 - “¿No tienes una cuenta? Regístrate”: lleva al formulario de registro (probablemente otra pantalla RegisterScreen también implementada en Compose).

Botones de autenticación alternativa:

- Continuar como invitado: permite acceder sin crear cuenta, seguramente con un perfil temporal/local.
- Continuar con Google: integra el flujo de autenticación con una cuenta de Google mediante Firebase OAuth2.

Funcionamiento técnico

Validaciones:

- Los campos de email y contraseña están validados antes de enviar los datos a Firebase. Si alguno está vacío o no cumple el formato, se muestra un Toast o Snackbar.

Firebase Authentication:

- Al pulsar “Iniciar sesión”, se hace una llamada a Firebase para autenticar al usuario con las credenciales ingresadas.
- En caso de éxito, se guarda el UID del usuario y se redirige a la pantalla principal de la app (HomeScreen).
- En caso de error (credenciales incorrectas, red no disponible...), se lanza un mensaje informativo.

Modo invitado:

- Usa una cuenta anónima (signInAnonymously() de Firebase) o inicia la app sin persistencia de sesión.

Inicio de sesión con Google:

- Integrado a través del cliente de Google Sign-In, conectado a Firebase para manejar sesión segura y simplificada.

Función dentro de la aplicación

Esta pantalla centraliza y estandariza el acceso a todas las funciones de la app. Define el perfil del usuario (registrado o no), y es el punto de entrada al ecosistema de fiestas personalizado de PraelParty!



Descripción general:

Esta pantalla permite al usuario crear una nueva cuenta en PraoParty!, siendo parte del sistema de autenticación gestionado por Firebase Authentication. Es un paso fundamental para que el usuario pueda personalizar su experiencia, guardar favoritos y mantener una sesión persistente.

Contenido visual y componentes:

Encabezado gráfico:

- Se presenta el escudo oficial del Principado de Asturias, destacando la identidad regional de la app.
- Esto refuerza el enfoque local de la aplicación y genera familiaridad cultural con el usuario.

Título de la pantalla:

- Texto grande y centrado: "Crear cuenta", con estilo visual coherente con el resto de pantallas.

Campos de entrada de datos:

- Correo electrónico: campo TextField destinado a introducir una dirección válida.
- Contraseña: campo protegido (PasswordField) para establecer una clave.
- Repite la contraseña: segundo campo de confirmación para asegurar que no hay errores tipográficos.

Botón de acción principal:

- Registrarse: botón principal morado que, al pulsarse, realiza las validaciones necesarias y llama al método de creación de cuenta con Firebase.

Enlace de navegación secundaria:

- Texto “¿Ya tienes cuenta? Inicia sesión” que redirige al usuario de nuevo a la pantalla de login (LoginScreen) si ya está registrado.

Funcionamiento técnico:

Validaciones previas:

- El sistema comprueba que los tres campos estén rellenados.
- Verifica que el campo “Repite la contraseña” coincida con la contraseña inicial.
- Si algún campo falla, se muestra un mensaje de error (Snackbar o Toast).

Conexión con Firebase Authentication:

- Se llama a createUserWithEmailAndPassword (email, password) de Firebase.

- Si la operación es exitosa, se registra el nuevo usuario y se inicia sesión automáticamente.
- Si el correo ya está en uso, o la contraseña no cumple los requisitos mínimos (longitud, caracteres, etc.), se lanza un mensaje adecuado.

Redirección posterior:

- Al completarse el registro, el usuario es llevado a la pantalla principal de la aplicación (HomeScreen), ya autenticado.

Función dentro de la aplicación

Esta pantalla cumple el rol crítico de permitir la entrada al sistema personalizado de PraoParty!. Tener una cuenta permite al usuario:

- Guardar fiestas como favoritas.
- Mantener sus preferencias entre sesiones.
- En el futuro, recibir notificaciones personalizadas y participar en interacciones sociales (valoraciones, comentarios...).

El diseño limpio, centrado y con pocos pasos busca minimizar la fricción al registrarse, especialmente en usuarios poco habituados a apps complejas.



Recuperar contraseña

Correo electrónico

Recuperar contraseña

[¿Te acuerdas de la contraseña? Inicia sesión](#)

Descripción general:

Esta pantalla permite a los usuarios que han olvidado su contraseña recuperar el acceso a su cuenta mediante su dirección de correo electrónico registrada. El proceso se apoya completamente en la funcionalidad nativa de Firebase Authentication, lo que garantiza una solución segura y automática.

Contenido visual y componentes:

Encabezado visual:

- Escudo del Principado de Asturias centrado, reforzando la identidad regional de la app.
- Diseño limpio y sin distracciones, lo que da protagonismo al propósito de la pantalla: recuperar el acceso a la cuenta.

Título:

- Texto centrado en fuente grande: "Recuperar contraseña".

Campo de entrada:

- Campo único de tipo TextField para introducir el correo electrónico vinculado a la cuenta del usuario.

Botón de acción:

- Recuperar contraseña: botón grande, morado y centrado, que lanza la acción de recuperación.

Enlace de navegación secundaria:

- Texto: “¿Te acuerdas de la contraseña? Inicia sesión”, que permite volver a la pantalla principal de login si el usuario recuerda sus datos.

Funcionamiento técnico (basado en el análisis del proyecto):

Validación previa:

- Se comprueba que el campo de correo no esté vacío y que tenga formato válido.
- Si está vacío o mal formado, se muestra un mensaje de error (Snackbar, Toast o Text informativo).

Integración con Firebase:

- Al pulsar el botón, se llama al método FirebaseAuth.sendPasswordResetEmail(email).
- Firebase envía automáticamente un correo con enlace de restablecimiento al usuario.

- Si el correo no está registrado en Firebase, se notifica al usuario sin revelar que no existe (por seguridad).
- Si la operación es exitosa, se muestra un mensaje informando que el email ha sido enviado correctamente.

Función dentro de la aplicación

Esta pantalla es clave para garantizar la accesibilidad continua del usuario a su cuenta, incluso en caso de olvidar la contraseña. Refuerza la profesionalidad y usabilidad de la aplicación, ya que evita perder cuentas o tener que volver a registrarse.

Además, al tratarse de una función delegada a Firebase, se asegura:

- Proceso seguro y actualizado automáticamente por Google.
- Menor mantenimiento para el desarrollador.
- Mejora la experiencia del usuario sin necesidad de gestión manual.

The screenshot shows the PraoParty mobile application interface. At the top, there is a purple header bar with the time '16:05' and several status icons. Below the header, the app's logo 'PRAOPARTY!' is displayed in large, bold, blue letters. To the left of the logo is a menu icon (three horizontal lines), and to the right is a search icon (magnifying glass). The main content area consists of a vertical list of six festival entries, each enclosed in a rounded rectangular card:

- Carmin de la pola**: Includes a small thumbnail image of a crowd, a link 'Más detalles', and an empty star icon.
- Exconsuraos**: Includes a thumbnail image of a person on a horse, a link 'Más detalles', and an empty star icon.
- La Corredoria**: Includes a thumbnail image of a bonfire at night, a link 'Más detalles', and an empty star icon.
- La Descarga**: Includes a thumbnail image of a person performing, a link 'Más detalles', and an empty star icon.
- El Descenso del Sella**: Includes a thumbnail image of a river scene, a link 'Más detalles', and an empty star icon.
- El Xiringüelu**: Includes a thumbnail image of a large crowd, a link 'Más detalles', and an empty star icon.

Below the list, the text 'La Jira de Navia' is partially visible, suggesting another festival entry cut off by the screen edge.

Descripción general:

Esta es la pantalla de inicio tras el login, y representa el núcleo funcional de PraoParty. Aquí se muestra una lista vertical de fiestas populares disponibles, permitiendo a los usuarios descubrir eventos, acceder a detalles y marcar favoritos.

Es la sección más consultada por los usuarios, ya que ofrece acceso directo al contenido más importante: las fiestas.

Componentes visuales y funcionales

Barra superior (TopAppBar personalizada):

- Menú hamburguesa (esquina superior izquierda): abre un drawer lateral o despliega más opciones.
- Logo "PRAOPARTY!" centrado en la barra, con tipografía festiva y colorido acorde a la temática.
- Icono de perfil o configuración (esquina superior derecha): posiblemente accede a la pantalla de cuenta o ajustes.

Lista de fiestas (LazyColumn o LazyVerticalGrid):

Cada ítem contiene:

- Imagen de la fiesta: cargada desde local.
- Nombre del evento: texto en negrita y con fuente personalizada.
- Texto “Más detalles”: sugiere que al pulsar el ítem se accede a una pantalla de detalles (DetailScreen).
- Icono de estrella (favoritos): permite al usuario marcar la fiesta como favorita. Cambia de color o estado al pulsar.

Diseño del ítem:

- Cada fiesta está contenida en una tarjeta redondeada con sombreado sutil (elevación).
- Los márgenes y espaciados dan un diseño aireado y moderno.

Funcionamiento técnico:

Carga de datos:

- Las fiestas favoritas se cargan desde Firebase Firestore, donde cada fiesta tiene campos como nombre, descripción, imageUrl, ubicación, fecha.
- El uso de LaunchedEffect o collectAsState permite que la UI se actualice automáticamente al obtener los datos.

Gestión de favoritos:

- Cada usuario puede marcar o desmarcar fiestas como favoritas.
- El estado se gestiona mediante una lista reactiva (`mutableStateListOf`).
- Al pulsar la estrella, se añade o elimina el identificador de la fiesta de la lista de favoritos.
- Este estado se puede persistir en Firestore (por UID) o en almacenamiento local.

Navegación:

- Al pulsar “Más detalles” o la tarjeta completa, se realiza una navegación hacia la pantalla de detalles (NavController).

Función dentro de la aplicación

Esta pantalla representa el centro de exploración y descubrimiento de fiestas en la app:

- Es accesible por cualquier usuario autenticado (registro/login o invitado).
- Permite consultar rápidamente todas las fiestas, clasificadas visualmente.
- Es el punto desde el que se puede profundizar en cada evento.
- Marca el inicio de la personalización del contenido mediante el uso de favoritos.

Aspectos destacados de experiencia de usuario

- Diseño visual limpio y festivo, con imágenes atractivas.
- Interacción intuitiva con favoritos y navegación fluida.
- Ideal para sesiones cortas en móvil (descubrir una fiesta en segundos).



Descripción general:

Esta pantalla muestra la información completa y extendida de una fiesta específica seleccionada por el usuario desde la pantalla principal. Se despliega en forma de ventana modal flotante (dialogo), permitiendo que el usuario explore la descripción sin perder el contexto de navegación anterior.

Componentes visuales y diseño:

Encabezado con el nombre del evento:

- Nombre de la fiesta— el nombre de la fiesta se presenta en negrita y centrado, como título del modal.

Imagen representativa del evento:

- Ocupa el ancho del modal, justo debajo del título.
- Muestra una imagen contextual de la fiesta (procesiones, gente al aire libre, paisaje, etc.), que ayuda al usuario a identificar visualmente el tipo de celebración.

Descripción completa:

- Se ofrece un texto explicativo largo y detallado del evento seleccionado.
- Describe la tradición, localización, fecha, actividades, ambiente social y cultural.

Botón “Ver en Google Maps”:

- Integra un Intent que lanza Google Maps.
- Usa la ubicación almacenada (geoposición o nombre del lugar) para mostrar dónde se celebra la fiesta.
- Refuerza la utilidad de la app para planificación real.

Botón “Cerrar”:

- Cierra el modal y devuelve al usuario a la lista principal sin salir de la pantalla actual.

Funcionamiento técnico:

- Esta pantalla se muestra mediante un modal personalizado (Dialog, AlertDialog o Box with Surface en Jetpack Compose).
- Se alimenta de un objeto Fiesta que contiene propiedades como:
 - nombre, descripcion, imageUrl, latitud, longitud.
- La apertura de este diálogo se produce tras pulsar en “Más detalles” de la tarjeta de fiesta.
- El botón de Google Maps construye una geo: URI o una maps.google.com/?q= con la dirección o coordenadas de la fiesta.

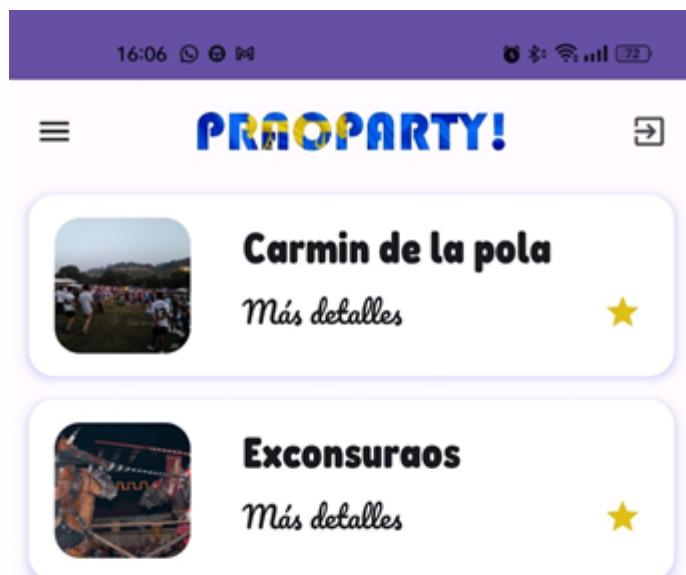
- El estilo es coherente con el diseño general: fondo claro, esquinas redondeadas, botones morados y enfoque en la legibilidad.

Función dentro de la app

- Esta pantalla permite profundizar en el contenido y conocer más sobre el evento antes de decidir asistir o marcarlo como favorito.
- Mejora la experiencia al ofrecer contexto histórico y logístico de la fiesta.
- Refuerza el aspecto cultural y turístico de la app, sirviendo tanto para asturianos como para visitantes que buscan planes populares.

Valor de diseño y UX

- El uso de un modal flotante permite explorar sin perder el scroll actual de la lista.
- Presenta la información de forma clara, jerarquizada y visualmente atractiva.
- El botón a Maps facilita la planificación y añade valor práctico.



Descripción general:

Esta pantalla muestra al usuario únicamente las fiestas que ha marcado como favoritas desde la pantalla principal. Es una sección personalizada, diseñada para que el usuario tenga acceso rápido a los eventos que más le interesan.

Componentes visuales y funcionales:

Barra superior (TopAppBar):

- Ícono de menú hamburguesa (izquierda): permite abrir el menú lateral o drawer, si está implementado.
- Logo "PRAOPARTY!": centrado y consistente con otras pantallas.
- Ícono de perfil o ajustes (derecha): mantiene la coherencia del diseño de navegación.

Lista de tarjetas (LazyColumn o LazyList):

Cada tarjeta representa una fiesta marcada como favorita e incluye:

- Imagen representativa de la fiesta.
- Nombre del evento en tipografía festiva personalizada.
- Texto “Más detalles” como indicativo de que se puede acceder al modal o pantalla de detalle.
- ícono de estrella en amarillo: representa visualmente que estas fiestas ya están seleccionadas como favoritas (diferencia respecto a la estrella gris sin seleccionar en la pantalla principal).

Funcionamiento técnico

- Esta pantalla filtra las fiestas cargadas desde Firebase para mostrar solo aquellas cuyo ID esté presente en la lista de favoritos del usuario.
- La gestión del estado de favoritos se realiza usando:
 - Un mutableListOf de IDs o fiestas favoritas.
 - Se sincroniza con Firestore en tiempo real.
- Al pulsar en cualquier tarjeta o en “Más detalles”, se lanza el mismo DetailDialog que en la pantalla principal, mostrando toda la información de esa fiesta.
- El usuario puede desmarcar la estrella desde esta pantalla, lo que elimina inmediatamente la fiesta de esta lista.

Función dentro de la aplicación

Esta pantalla permite al usuario crear su propio calendario o agenda de fiestas favoritas, sin necesidad de volver a buscar entre todas. Su propósito es:

- Ofrecer acceso directo a los eventos más importantes para el usuario.
- Facilitar la organización personal: saber a qué fiestas quiere ir y cuándo.
- Mejorar la experiencia mediante personalización real basada en el comportamiento.

Diseño y experiencia de usuario

- Visualmente, mantiene la coherencia de la pantalla principal, pero con un número reducido de ítems que el usuario ha elegido.
- El uso del icono de estrella en amarillo sirve como indicador inmediato de estado (favorita) sin necesidad de texto adicional.
- Es una pantalla rápida, limpia y útil, diseñada para repetidas visitas a lo largo del verano o temporada festiva.



PRAOPARTY!

Cuenta actual:



Anónimo

Nota: cuando inicias sesión como Anónimo, no se puede borrar la cuenta.

[Eliminar cuenta](#)

Cambiar contraseña:

Escribe la nueva contraseña

Repite la nueva contraseña

Nota: cuando inicias sesión como Anónimo, no se puede cambiar la contraseña.

[Actualizar contraseña](#)



PRAOPARTY!

Cuenta actual:



Alejandro Rodriguez
alex.r.aragon@gmail.com

[Eliminar cuenta](#)

Cambiar contraseña:

Escribe la nueva contraseña

Repite la nueva contraseña

[Actualizar contraseña](#)



Descripción general:

Esta pantalla permite al usuario consultar y gestionar su cuenta actual. Según el tipo de usuario (registrado o anónimo), las funciones disponibles se adaptan dinámicamente. Representa el centro de configuración personal de la app PraoParty.

Componentes visuales comunes

TopBar: se mantiene el diseño de la app con:

- Menú lateral (hamburguesa) a la izquierda.
- Logo centrado (“PRAOPARTY!”).

- Icono lateral derecho para navegación o cierre de sesión (dependiendo de la implementación)

Título “Cuenta actual”: encabezado fijo, centrado en la parte superior del contenido.

Avatar e información de usuario:

- Para usuarios registrados: avatar con su imagen (cargada desde Google si usa sign-in con Google), nombre y email.
- Para usuarios anónimos: ícono genérico y etiqueta “Anónimo”.

Botón “Eliminar cuenta”:

- Registrados: funcional, permite eliminar la cuenta.
- Anónimos: desactivado o funcionalmente bloqueado con mensaje informativo (“no se puede borrar la cuenta”).

Formulario para cambiar contraseña:

- Dos campos tipo PasswordTextField: nueva contraseña y repetición.
- Botón morado “Actualizar contraseña”.

Imagen inferior decorativa:

- Mapa de Asturias con la Cruz de la Victoria, coherente con la identidad regional de la app.

Funcionamiento técnico:

Si el usuario está autenticado (correo y contraseña o Google):

- Firebase detecta el FirebaseAuth con un email válido.
- Se permite:
 - Eliminar cuenta → user.delete().
 - Cambiar contraseña → user.updatePassword() si ambas coinciden y cumplen condiciones mínimas.
- Validaciones:
 - Ambas contraseñas deben coincidir y tener longitud mínima.
 - Se muestran errores con Toast si falla la operación.
- Seguridad:
 - Firebase puede pedir reautenticación antes de permitir estos cambios.

Si el usuario es anónimo (signInAnonymously):

- No se muestra email ni nombre personalizado.
- Se deshabilita el botón de eliminar cuenta.
- También se inhabilita el formulario de contraseña con un texto aclaratorio:
 - “Nota: cuando inicias sesión como Anónimo, no se puede cambiar la contraseña.”

Función dentro de la aplicación

Esta pantalla permite al usuario:

- Ver su identidad actual.
- Cambiar su contraseña fácilmente (seguridad y mantenimiento).
- Eliminar su cuenta si así lo desea (control total sobre sus datos).
- Entender las limitaciones del modo anónimo.

Ofrece transparencia y control, dos factores importantes para la confianza del usuario.

Diferenciación UX entre tipos de usuario

Usuario registrado	Usuario anónimo
Puede eliminar cuenta	No puede eliminar cuenta
Puede cambiar contraseña	No puede cambiar contraseña
Ve su nombre y correo	Solo aparece la palabra “Anónimo”
Imagen personalizada (si aplica)	Avatar genérico

The screenshot shows the 'Notificaciones de la app' (App Notifications) screen of the PRAOPARTY! app. At the top, there is a purple header bar with the time '16:06' and battery level '92'. Below the header, the app logo 'PRAOPARTY!' is displayed. On the left is a menu icon (three horizontal lines), and on the right is a user profile icon.

Notificaciones de la app

Activar notificaciones

trash Al borrar la cuenta

key Al cambiar la contraseña

[Restablecer valores predeterminados](#)

gear AJUSTES DEL SISTEMA
(comprueba que estén activadas las notificaciones de la app en tu sistema)

Descripción general:

Esta pantalla permite al usuario gestionar las notificaciones internas de la aplicación PraoParty.

Componentes visuales y funcionales:

TopBar de navegación:

- Menú hamburguesa (izquierda) y botón de acción (derecha) se mantienen como en otras pantallas.
- Logo "PRAOPARTY!" centrado.

Sección principal “Notificaciones de la app”:

- Presentada en una tarjeta (Card) con fondo lavanda claro y bordes redondeados.
- Texto en fuente personalizada para reforzar la identidad gráfica.

Opciones disponibles:

- Activar notificaciones (switch principal): habilita o deshabilita el resto de las opciones de notificación.
- Al borrar la cuenta: notificación interna que puede mostrarse cuando el usuario elimina su perfil (local o tipo Toast, Snackbar, Dialog).
- Al cambiar la contraseña: indica si se desea recibir confirmación local tras modificar los datos de acceso.

Todas las opciones son controladas por interruptores deslizantes (Switch), y están respaldadas por una estructura de rememberSaveable o persistidas mediante DataStore para mantener los estados entre sesiones.

Enlace “Restablecer valores predeterminados”:

- Restaura todos los switches a su estado inicial (por defecto: todo activado o todo desactivado).
- Internamente, restablece los valores guardados (reseteo de Preferences).

Pie de pantalla “AJUSTES DEL SISTEMA”:

- Enlace informativo que recuerda al usuario verificar que las notificaciones están activadas en la configuración de Android del dispositivo.
- Implementado con un botón que lanza un Intent a los ajustes de la app (Settings.ACTION_APP_NOTIFICATION_SETTINGS).

Funcionamiento técnico previsto

Esta pantalla permite al usuario controlar eventos internos que podrían generar avisos locales.

Sincronización de switches:

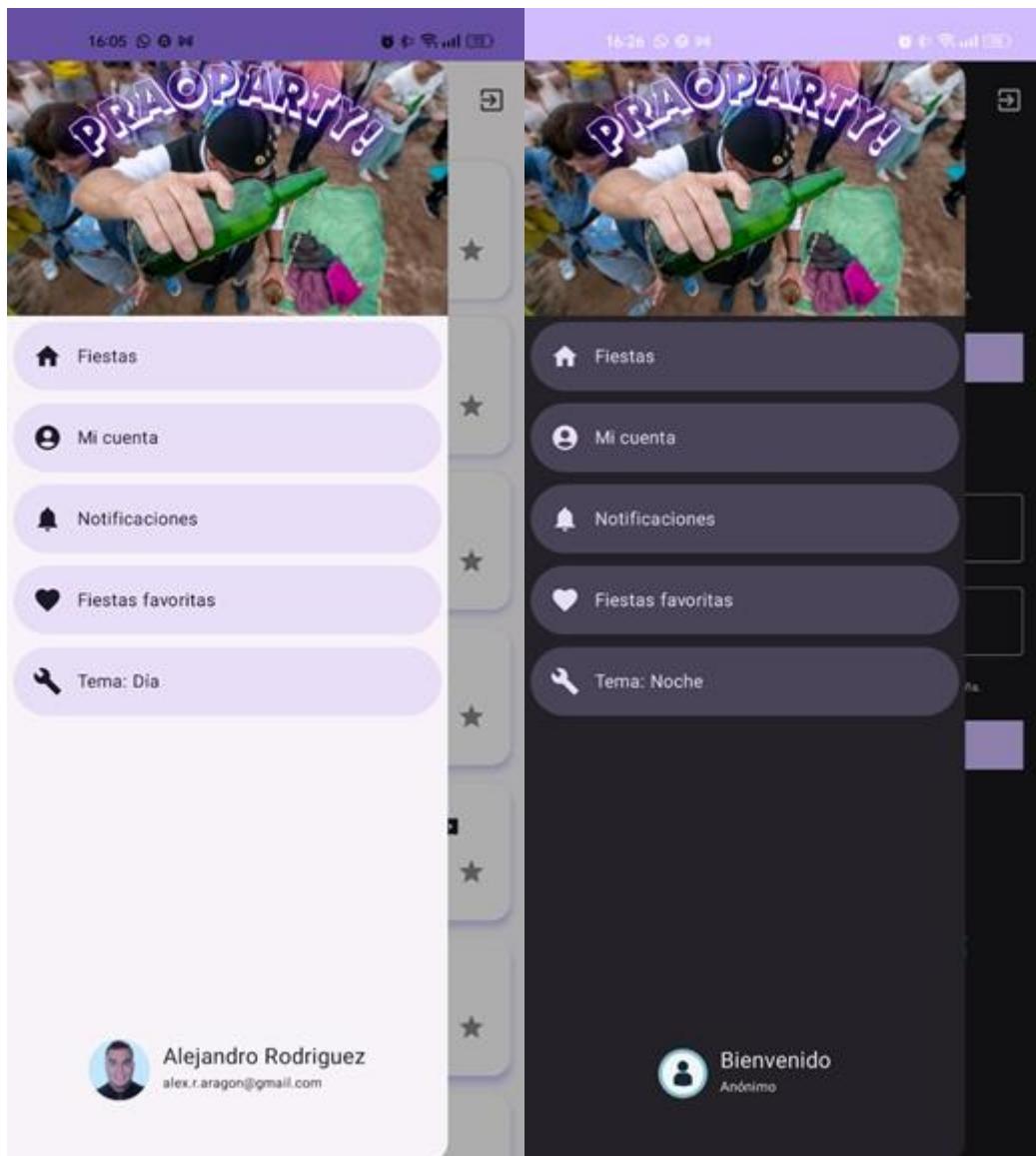
- Si se desactiva el switch general “Activar notificaciones”, todos los demás se deshabilitan visual y funcionalmente (bloqueo o enabled = false).
- Si se activa, se habilitan los secundarios para edición.

Función dentro de la app

- Esta pantalla anticipa funcionalidades futuras de comunicación personalizada con el usuario.
- Refuerza la idea de control y respeto por la privacidad, permitiendo al usuario decidir cómo y cuándo recibir alertas o confirmaciones.
- Mejora la experiencia de usuario ofreciendo transparencia y configuración personalizada.

Detalles UX destacados

- Diseño limpio, sencillo y accesible incluso para usuarios no técnicos.
- Iconografía clara, interruptores grandes, texto explicativo y jerarquía visual bien definida.
- Se alinea con buenas prácticas de accesibilidad y usabilidad móvil.



Descripción general:

El Drawer lateral de PraoParty proporciona acceso rápido y centralizado a las secciones más importantes de la aplicación. Se adapta tanto al modo Día/Noche como al tipo de usuario (registrado o anónimo), integrando funcionalidad y personalización visual.

Componentes visuales y funcionales:

Cabecera del Drawer:

- Imagen superior personalizada que muestra una foto festiva (asturiano escaciando sidra, gente de fiesta alrededor).
- Sobreimpreso el logo “PRAOPARTY!” en letras grandes y coloridas.

- Refuerza la identidad visual y el ambiente de la aplicación (alegre, cultural, local).

Opciones del menú:

Cada una es un botón visual con ícono, texto y estilo coherente. Las opciones son:

- Fiestas
 - Redirige a la pantalla principal con la lista completa de fiestas (HomeScreen).
- Mi cuenta
 - Lleva a la pantalla de perfil/cuenta donde se pueden modificar datos o eliminar el usuario (AccountScreen).
- Notificaciones
 - Abre la pantalla de configuración de notificaciones (NotificationsScreen), donde se pueden gestionar alertas locales.
- Fiestas favoritas
 - Redirige a la lista de fiestas que el usuario ha marcado como favoritas (FiestasFavoritasScreen).
- Tema: Día/Noche
 - Permite alternar entre modo claro y modo oscuro.
 - Este interruptor actúa sobre el Theme global de Jetpack Compose, actualizando toda la UI al instante.
 - Cambia visualmente entre:
 - Tema: Día (fondo blanco, componentes lavanda)
 - Tema: Noche (fondo gris oscuro/negro, componentes violetas apagados)

Pie del Drawer:

- Información del usuario autenticado:
 - Si está registrado: se muestra su nombre, correo y avatar.
 - Si es anónimo: aparece el texto “Bienvenido Anónimo” con ícono genérico.
- Esta sección proporciona contexto de sesión sin necesidad de ir a “Mi cuenta”.

Funcionamiento técnico

- El Drawer está implementado usando ModalDrawer y Scaffold con DrawerContent.

- Las acciones de navegación están gestionadas mediante NavController de Jetpack Compose Navigation.
- El cambio de tema se controla con un mutableStateOf en un ViewModel o CompositionLocal, que alterna entre MaterialTheme.light() y MaterialTheme.dark() personalizados.
- Los colores, tipografías e íconos están adaptados a cada modo mediante el uso de isSystemInDarkTheme() o variables controladas manualmente.

Función dentro de la app

- Este menú centraliza la navegación sin depender del Bottom Navigation.
- Refuerza la idea de estructura clara y jerarquía lógica
- Aporta personalización estética (tema visual) y mantiene al usuario informado sobre su sesión.

Característica	Beneficio
Modo oscuro alternable	Mejora la accesibilidad y personalización.
Menú lateral desplegable	Permite ahorrar espacio en pantallas pequeñas.
Avatar y nombre visibles	Genera confianza y sensación de pertenencia.
Imagen festiva superior	Refuerza la temática y el diseño único de la app.

II – Código completo del proyecto
<https://github.com/DAM2A-PDAM-24-25/PraoParty>