

PROYECTO INTEGRADO DE APLICACIONES MULTIPLATAFORMA

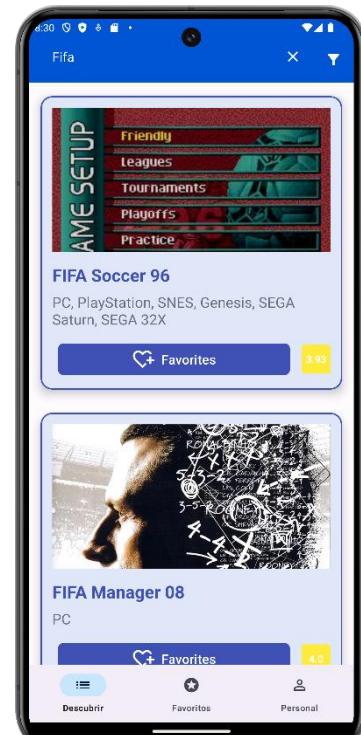
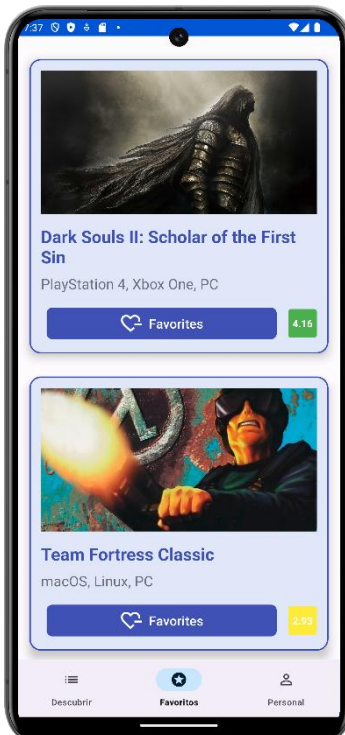
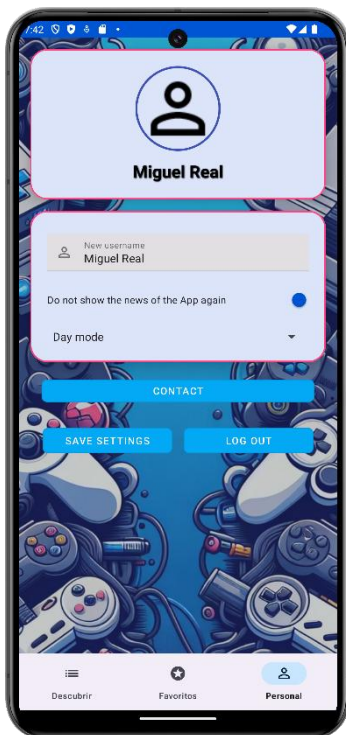
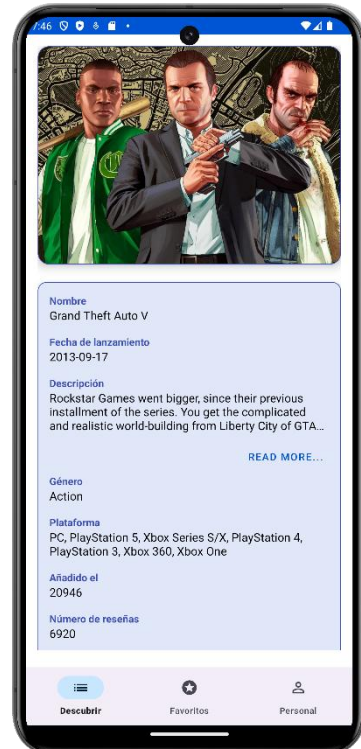
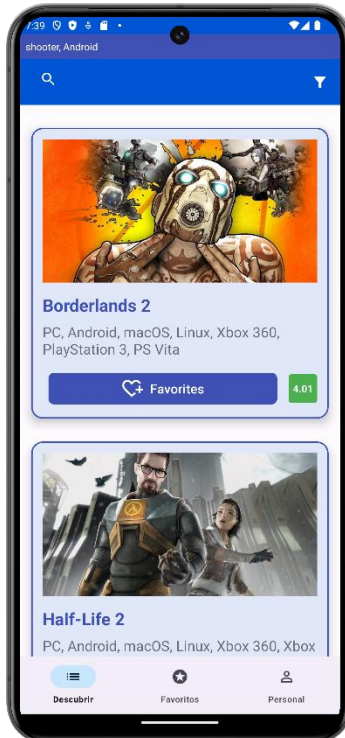
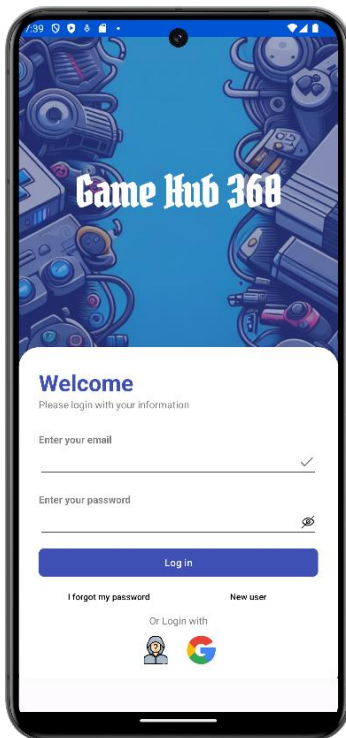


Contenido

Imágenes	4
Resumen	5
Introducción	6
Exposición del problema	6
Estudio de mercado.....	6
➔ Mi colección de Juegos.....	6
➔ Stash Video Game Manager.....	7
➔ Instant Gaming.....	7
Objetivos del proyecto	8
Desarrollo	8
Introducción	8
Tecnologías.....	9
Firebase.....	9
Base de datos ROOM	11
API de RAWG.IO	12
Data Store.....	13
Arquitectura de GameHub360.....	15
MVVM Model-View-ViewModel	15
MVVM aplicado a GameHub360	16
Interfaces gráficas.....	17
Inicio de Sesión con correo o Google	17
Inicio de Sesión como anónimo.....	18
Registro	19
Olvido de contraseña	20
Configuración de la aplicación.....	21
Lista de juegos.....	22
Buscador de juegos por filtro.....	24
Detalles del juego	25
Añadir comentario público	26
Lista de juegos favoritos	27
Detalles de un juego favorito.....	29
Añadir comentario privado	30

Novedades de la App.....	31
No Internet	32
Diagrama de clases.....	33
Diagrama de secuencia.....	33
Diagrama de casos de uso	34
Modelo de datos.....	35
Base de datos ROOM	35
Base de datos Realtime Database	36
Base de datos Cloud Firestore.....	37
Conclusiones	39
Dificultades encontradas.....	40
Documentación	40
Desarrollo de un Diseño Profesional	40
Webgrafía	41
Documentación de Google	41
Material Design	41
Youtube Firebase.....	41
API	41
Enlace APK de la aplicación	41

Imágenes



Resumen

GameHub360 es una aplicación para dispositivos móviles desarrollada en Android Studio cuyo objetivo principal es proporcionar a los usuarios una experiencia integral en la gestión y exploración de videojuegos. La aplicación se centra en la comodidad y la personalización, ofreciendo diversas características que permiten a los usuarios aprovechar al máximo su experiencia de juego.

Una de las características destacadas de GameHub360 es su sistema de inicio de sesión que utiliza la autenticación de Firebase para permitir a los usuarios acceder de manera segura a la aplicación mediante sus cuentas de Google. Esta integración asegura la confiabilidad del inicio de sesión y simplifica el proceso para los usuarios.

Una vez dentro de la aplicación, los usuarios tienen acceso a una amplia biblioteca de videojuegos obtenidos a través de una API. Esta biblioteca incluye detalles importantes de cada juego, lo que facilita a los usuarios explorar y descubrir nuevos títulos de manera eficiente. Además, los usuarios pueden guardar sus videojuegos favoritos dentro de la aplicación para poder consultarlos en cualquier momento y desde cualquier lugar.

La gestión personalizada de juegos es otra característica clave de GameHub360. Los usuarios pueden agregar anotaciones y comentarios personalizados a cada juego, lo que les permite llevar un registro de sus impresiones y opiniones sobre cada título. Esta función añade un elemento de personalización a la experiencia de juego y permite a los usuarios compartir sus opiniones con otros usuarios de la aplicación.

Para facilitar la búsqueda de juegos específicos, GameHub360 incluye un buscador que permite a los usuarios encontrar juegos por nombre, género o cualquier otro criterio relevante. Esta función agiliza el proceso de navegación dentro de la extensa biblioteca de videojuegos y ayuda a los usuarios a encontrar rápidamente los juegos que están buscando.

Además, la aplicación ofrece opciones de personalización para adaptarse a las preferencias individuales de los usuarios. Los usuarios pueden seleccionar entre el modo nocturno o diurno y pueden elegir si desean mostrar o no las novedades de la aplicación. Esta flexibilidad permite a los usuarios personalizar su experiencia de acuerdo a sus gustos y preferencias.

Para aquellos que prefieren una experiencia más discreta, GameHub360 permite el inicio de sesión anónimo con funcionalidades limitadas. Esto permite a los usuarios acceder a algunas funciones básicas de la aplicación sin la necesidad de crear una cuenta o compartir información personal. Además, los datos del usuario se guardan de forma local gracias a la base de datos de ROOM, permitiendo el acceso a los datos sin necesidad de internet.

En resumen, GameHub360 es una aplicación completa y personalizada que ofrece a los usuarios una experiencia integral en la gestión y exploración de videojuegos. Con características como el inicio de sesión seguro, la amplia biblioteca de juegos, la gestión personalizada de juegos y las opciones de personalización.

Introducción

Exposición del problema

La génesis de GameHub360 surge de una necesidad personal compartida por muchos aficionados a los videojuegos. La experiencia como jugador llevó a enfrentar la dificultad de gestionar y explorar una amplia variedad de videojuegos de manera eficiente, rápida y conveniente. Muchos usuarios desean probar la variedad de juegos disponibles en el mercado.

Buscando una solución, se necesitaba una aplicación que permitiera acceder fácilmente a una lista de juegos con sus detalles importantes, valoraciones, guardar juegos favoritos y compartir experiencias con otros usuarios para luego decidir si probar un juego.

Al analizar el mercado, se encontraron muchas webs que permitían estas funciones, pero se deseaba una aplicación móvil para acceder rápidamente a todos los juegos y favoritos sin necesidad de internet, conocer opiniones de otros usuarios y más características necesarias. Esta frustración compartida llevó a concebir la idea de GameHub360, una aplicación que aborda estas necesidades de manera integral y personalizada, permitiendo guardar juegos como si fueran fotos en el móvil y acceder a ellos en cuestión de segundos desde cualquier dispositivo, incluso sin conexión a internet.

Estudio de mercado

Para destacar entre las demás aplicaciones, se realizó un estudio de las mejores aplicaciones existentes que ofrecen funcionalidades similares para crear una aplicación diferente, añadiendo todo lo que las mejores aplicaciones tienen y más funcionalidades necesarias desde la experiencia de jugador de videojuegos. En esta larga búsqueda he encontrado estas aplicaciones (gratuitas y disponibles en Play Store)

→ Mi colección de Juegos

- **Características:** Permite gestionar y organizar la colección de videojuegos, realizar un seguimiento de los juegos poseídos, registrar juegos deseados y mantener un registro de los juegos jugados. Ofrece escaneo de códigos de barras para facilitar la entrada de datos.

- **Debilidades:** Carece de características avanzadas como la integración con bases de datos externas para obtener detalles de juegos automáticamente y de opciones de personalización y características sociales.

→ Stash Video Game Manager

- **Características:** Permite gestionar la colección de videojuegos, incluyendo detalles como título, plataforma, género y estado. Ofrece seguimiento de la progresión en los juegos y notas personalizadas.

- **Debilidades:** Carece de integración con bases de datos de videojuegos y de funciones sociales y de personalización.

→ Instant Gaming

- **Características:** Plataforma de venta de juegos en línea con una amplia gama de títulos para diversas plataformas. Permite comprar juegos a precios reducidos y recibir códigos de descarga instantáneamente.

- **Debilidades:** No ofrece características de gestión de colecciones y está más centrada en la venta de juegos.

Imágenes y enlaces

Mi colección de juegos



Stash Video Games Manager Instant Games



Instant Games

<https://play.google.com/store/apps/details?id=com.instantgaming.android&hl=es>

Mi colección de juegos

<https://play.google.com/store/apps/details?id=com.whynotapplications.micolecciondejuegos&hl=es>

Stash

<https://play.google.com/store/apps/details?id=io.stashteam.games.tracker.stashapp&hl=es>

Objetivos del proyecto

1. Proporcionar una plataforma completa y personalizada para explorar, gestionar y compartir videojuegos.
2. Ofrecer un inicio de sesión seguro mediante Firebase Authentication.
3. Integrar una amplia biblioteca de juegos con detalles completos obtenidos de la API.
4. Permitir la gestión personalizada de cada videojuego por parte del usuario.
5. Implementar un buscador avanzado para facilitar la búsqueda de juegos.
6. Ofrecer opciones de personalización para adaptar la experiencia a las preferencias individuales de los usuarios.
7. Incluir una opción ligera de inicio de sesión anónimo para permitir guardar datos localmente.
8. Facilitar el filtrado de juegos por categorías, plataformas y otros criterios relevantes.

Desarrollo

Introducción

Para el desarrollo de GameHub360 se utilizará Android Studio aprovechando el lenguaje de programación Kotlin. La fuente de datos principal será la API de RAWG.IO y para funciones clave como la autenticación y gestión de datos en tiempo real se empleará Firebase. Además, se utilizarán tecnologías avanzadas de almacenamiento local y gestión de preferencias como la base de datos local Room y DataStore.

Como fuente de datos fundamental, GameHub360 se conectará a la API de RAWG.IO, una plataforma líder en la industria de los videojuegos que proporciona una amplia gama de datos y detalles sobre miles de videojuegos. Esta integración permitirá a los usuarios acceder a una extensa biblioteca de videojuegos con todos sus detalles más importantes, simplificando así la exploración y gestión de sus juegos favoritos.

Además, para funciones clave como la autenticación de usuarios y la gestión de datos en tiempo real, GameHub360 hará uso de Firebase, una plataforma de desarrollo de aplicaciones

móviles y web desarrollada por Google. Firebase ofrece soluciones listas para usar y escalables, lo que simplifica el desarrollo de aplicaciones al permitir a los desarrolladores centrarse en la lógica de la aplicación en lugar de la infraestructura subyacente. En este proyecto se aprovecharán las herramientas de autenticación de usuarios y bases de datos en tiempo real proporcionadas por Firebase para ofrecer una experiencia segura y fluida a los usuarios de GameHub360.

Adicionalmente, para proporcionar una experiencia aún más personalizada y eficiente, GameHub360 emplea tecnologías avanzadas de almacenamiento local y gestión de preferencias. Utiliza la base de datos local de Room para almacenar los juegos favoritos del usuario, guardando todos los detalles de cada juego añadido. Esto garantiza un acceso rápido y fiable a la información incluso cuando no hay conexión a Internet o cuando el usuario quiere iniciar sesión como anónimo. Por otro lado, para gestionar las preferencias del usuario como el modo día/noche o la configuración de la vista inicial, GameHub360 utiliza DataStore, una biblioteca de almacenamiento de datos lanzada por Google. DataStore ofrece un almacenamiento seguro y eficiente para las preferencias del usuario, garantizando una experiencia de usuario consistente y personalizable en toda la aplicación.

Tecnologías

Firebase

Firebase es una plataforma de desarrollo de aplicaciones móviles y web, propiedad de Google, que ofrece una amplia gama de servicios esenciales para el desarrollo de aplicaciones. En el caso de GameHub360, nos centraremos en aprovechar la autenticación de usuarios, el almacenamiento en la nube y las bases de datos en tiempo real, los servicios de Firebase que se utilizarán son los siguientes:

❖ *Firebase Authentication*

Firebase Authentication proporciona múltiples opciones de inicio de sesión, incluyendo correo electrónico y contraseña, así como la autenticación mediante proveedores de identidad social como Google, Facebook y Twitter. También ofrece la posibilidad de autenticación anónima, lo que permite a los usuarios acceder a ciertas funciones sin necesidad de crear una cuenta. Este servicio facilita la gestión del flujo de inicio de sesión de los usuarios y permite realizar acciones como enviar correos

electrónicos de verificación y restablecer contraseñas. Además, añade una capa de seguridad adicional al manejar la autenticación y la gestión de tokens de acceso, protegiendo así las credenciales de inicio de sesión de los usuarios y evitando el acceso no autorizado a la aplicación.

❖ *Firebase Storage*

Firebase Storage permite almacenar y recuperar archivos en la nube, como imágenes, videos, archivos de audio, etc. Proporciona una solución segura y sencilla para almacenar y compartir archivos en la nube, eliminando la necesidad de mantener servidores propios para el almacenamiento de archivos. Además, se integra perfectamente con otros servicios de Firebase como Firebase Authentication y Firebase Database.

❖ *Firestore*

Firestore es una base de datos NoSQL que organiza los datos en colecciones y documentos. Cada documento es una unidad de almacenamiento que contiene campos con datos y puede ser consultado, actualizado o eliminado individualmente. Aunque en el caso de GameHub360 prescindiremos de los filtros que Firestore proporciona y realizaremos los filtrados dentro de la

lógica de la aplicación para reducir las solicitudes a Firestore, este servicio sigue siendo fundamental para la aplicación. Además, Firestore ofrece una sincronización en tiempo real entre los clientes y los datos almacenados en la nube, garantizando que los datos estén siempre actualizados y disponibles en todos los dispositivos conectados a la misma base de datos.

❖ *Firebase Realtime Database*

Firebase Realtime Database es una base de datos en tiempo real alojada en la nube que permite almacenar y sincronizar datos entre los usuarios de manera rápida y eficiente. Utiliza el modelo de datos de árbol JSON, lo que facilita la organización y recuperación de datos de forma flexible. La principal característica de Realtime Database es su capacidad para sincronizar datos en tiempo real entre todos los clientes conectados, lo que significa que cualquier cambio realizado en la base de datos se refleja instantáneamente en todos los dispositivos sin necesidad de recargar la aplicación. Esto hace que sea ideal para aplicaciones que requieren actualizaciones en tiempo real, como chats, juegos multijugador y aplicaciones

colaborativas. En GameHub360, Realtime Database se utilizará para gestionar datos en tiempo real, como la información de los usuarios, las interacciones con los juegos y las preferencias de personalización, proporcionando una experiencia de usuario dinámica y colaborativa. Además, su integración con otros servicios de Firebase, como Firebase Authentication y Firebase Storage, garantiza una experiencia de desarrollo fluida y una gestión eficiente de los datos de la aplicación.

Base de datos ROOM

Room es una biblioteca de persistencia de datos en Android que ofrece una capa de abstracción sobre SQLite, lo que facilita el acceso y la gestión de la base de datos local en las aplicaciones. En el desarrollo de GameHub360, hacemos uso de esta poderosa herramienta para garantizar un almacenamiento eficiente y estructurado de los datos de los usuarios.

Principalmente se usa en GameHub360 para poder acceder a los juegos que hemos guardado en favorito sin necesidad de tener internet en nuestro teléfono

Características

1. Simplicidad: Room simplifica la creación y gestión de la base de datos al proporcionar una interfaz intuitiva y fácil de usar para definir la estructura de los datos y las consultas.
2. Abstracción de SQLite: Al ofrecer una capa de abstracción sobre SQLite, Room elimina la complejidad de trabajar directamente con SQL, lo que permite a los desarrolladores centrarse en la lógica de la aplicación en lugar de los detalles de implementación de la base de datos.
3. Componentes clave: Room consta de tres componentes principales: las entidades, que representan las tablas de la base de datos; los DAOs (Data Access Objects), que proporcionan métodos para acceder y manipular los datos en la base de datos; y la base de datos, que sirve como punto de entrada principal para acceder a la base de datos y obtener instancias de los DAOs.
4. Relaciones entre tablas: Room permite establecer relaciones entre las entidades para modelar relaciones complejas entre los datos, lo que facilita la gestión de la integridad referencial y la recuperación de datos relacionados.

Ventajas

1. Eficiencia: Room ofrece un rendimiento optimizado y una gestión eficiente de la memoria, lo que garantiza una experiencia fluida para los usuarios incluso con grandes volúmenes de datos.
2. Facilidad de mantenimiento: La estructura declarativa de Room facilita la comprensión y el mantenimiento del código, lo que permite a los desarrolladores realizar cambios y actualizaciones de forma rápida y sencilla.
3. Seguridad: Room proporciona una capa adicional de seguridad al evitar vulnerabilidades comunes de SQL y prevenir ataques de inyección de SQL.

En resumen, Room es una herramienta invaluable en el desarrollo de GameHub360, proporcionando una solución robusta y eficiente para el almacenamiento y la gestión de datos en la aplicación. Su simplicidad, rendimiento y seguridad hacen que sea la elección ideal para garantizar una experiencia de usuario óptima en la aplicación.

API de RAWG.IO

La API de RAWG.IO es una plataforma líder en la industria de los videojuegos que proporciona una amplia gama de datos y detalles sobre miles de videojuegos. En el desarrollo de GameHub360, hacemos uso de esta API para enriquecer la experiencia de los usuarios al proporcionarles acceso a una extensa biblioteca de videojuegos con todos sus detalles más importantes.

Enlace de su web oficial: <https://rawg.io/>

Características

1. Amplia base de datos: RAWG.IO cuenta con una extensa base de datos que incluye información detallada sobre miles de videojuegos, incluyendo títulos, géneros, plataformas, desarrolladores, fechas de lanzamiento, puntuaciones y más.
2. Acceso a detalles completos: La API permite acceder a detalles completos sobre cada videojuego, incluyendo imágenes, descripciones, clasificaciones, críticas y tráilers, lo que permite a los usuarios obtener una visión completa de cada juego.

3. Fácil integración: RAWG.IO ofrece una API bien documentada y fácil de usar, con endpoints claros y una documentación detallada que facilita su integración en aplicaciones móviles y web.

4. Actualizaciones frecuentes: La base de datos de RAWG.IO se actualiza regularmente con nuevos lanzamientos, actualizaciones de juegos y contenido adicional, lo que garantiza que los datos estén siempre actualizados y precisos.

Ventajas

1. Enriquecimiento de datos: La API de RAWG.IO enriquece la biblioteca de videojuegos de GameHub360 al proporcionar una amplia gama de detalles y características sobre cada juego, lo que mejora la experiencia de los usuarios al explorar y descubrir nuevos juegos.

2. Actualización constante: Al aprovechar la base de datos actualizada de RAWG.IO, GameHub360 garantiza que los usuarios tengan acceso a la información más reciente sobre los videojuegos, incluyendo lanzamientos recientes, clasificaciones y críticas.

3. Facilidad de implementación: La API de RAWG.IO es fácil de implementar en GameHub360 gracias a su documentación clara y endpoints bien definidos, lo que permite a los desarrolladores integrarla rápidamente en la aplicación y comenzar a utilizarla de inmediato.

Data Store

Data Store es una biblioteca de almacenamiento de datos lanzada por Google que ofrece una solución segura y eficiente para gestionar las preferencias del usuario en aplicaciones Android. En el desarrollo de GameHub360, hacemos uso de esta herramienta para proporcionar una experiencia personalizada y consistente a los usuarios al permitirles configurar preferencias como el modo día/noche o la vista inicial de la aplicación.

Características

1. Almacenamiento seguro: Data Store ofrece un almacenamiento seguro para las preferencias del usuario, garantizando que la información se mantenga privada y protegida en el dispositivo del usuario.

2. Eficiencia: Data Store proporciona un almacenamiento eficiente y optimizado que permite acceder rápidamente a las preferencias del usuario sin afectar el rendimiento de la aplicación.

3. Facilidad de uso: La API de Data Store es fácil de usar y ofrece una interfaz intuitiva para gestionar y acceder a las preferencias del usuario, lo que facilita su integración en aplicaciones Android.

4. Sincronización de datos: Data Store ofrece la capacidad de sincronizar automáticamente las preferencias del usuario entre dispositivos, garantizando una experiencia consistente en todos los dispositivos conectados.

Ventajas

1. Personalización: Data Store permite a los usuarios personalizar su experiencia en GameHub360 configurando preferencias como el modo día/noche, la configuración de la vista inicial y otras opciones de personalización, lo que mejora la experiencia del usuario y aumenta la satisfacción.

2. Consistencia: Al sincronizar automáticamente las preferencias del usuario entre dispositivos, Data Store garantiza una experiencia consistente en todos los dispositivos conectados, lo que facilita la transición entre diferentes dispositivos sin perder la configuración personalizada.

3. Eficiencia: Data Store ofrece un almacenamiento eficiente y optimizado que garantiza un acceso rápido y fiable a las preferencias del usuario, sin afectar el rendimiento general de la aplicación.

Arquitectura de GameHub360

MVVM Model-View-ViewModel

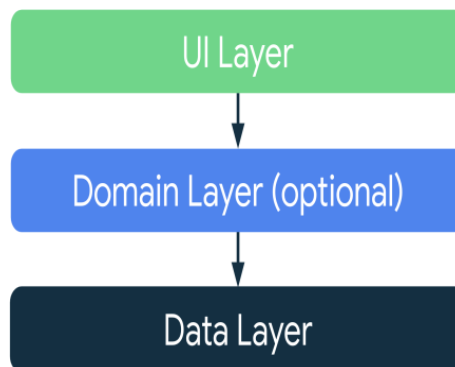
Capa de la Interfaz de Usuario UI Layer

La capa de la Interfaz de Usuario es la parte más cercana al usuario en la arquitectura de la aplicación. Su función principal es presentar la información de manera clara y comprensible para el usuario final. Aquí se definen y diseñan los elementos visuales de la aplicación, como las pantallas, ventanas, botones y otros elementos de la interfaz gráfica. Además, esta capa

incluye la lógica necesaria para gestionar la interacción del usuario con la aplicación, como la validación de entradas, la navegación entre pantallas y la respuesta a eventos del usuario. En resumen, la Capa de la Interfaz de Usuario se encarga de crear una experiencia de usuario intuitiva y atractiva, facilitando la interacción del usuario con la aplicación.

Capa de Dominio Domain Layer

La Capa de Dominio es el núcleo de la aplicación, donde reside la lógica de negocio y las reglas fundamentales que gobiernan el comportamiento de la aplicación. Esta capa define y coordina las operaciones y procesos clave de la aplicación, como la gestión de datos, la lógica de negocio, las reglas de validación y las operaciones de cálculo. Aquí se encuentran las clases y métodos que representan los conceptos y entidades del dominio de la aplicación, como usuarios, productos, pedidos, etc. La Capa de Dominio encapsula la lógica de la aplicación de manera que sea independiente de la interfaz de usuario y del almacenamiento de datos, lo que facilita la reutilización y la mantenibilidad del código.



Capa de Datos Data Layer

La Capa de Datos es responsable de gestionar el almacenamiento y recuperación de la información de la aplicación. Aquí se encuentra la lógica necesaria para interactuar con las fuentes de datos, como bases de datos, servicios web o sistemas de archivos. Los repositorios actúan como intermediarios entre las fuentes de datos y la Capa de Dominio, proporcionando una interfaz de acceso unificada y abstrayendo

los detalles de implementación de las operaciones de lectura y escritura de datos. Además, esta capa puede incluir la lógica necesaria para sincronizar datos entre diferentes fuentes o dispositivos, garantizando la integridad y consistencia de la información almacenada. En resumen, la Capa de Datos garantiza que la aplicación pueda acceder y manipular los datos de manera eficiente y segura.

MVVM aplicado a GameHub360

Capa de Interfaz de Usuario

- **Interfaz:** Definimos la estructura de la interfaz de usuario mediante archivos XML, mientras que las clases ViewModel gestionan la presentación de datos y la interacción con el usuario.
- **ViewModel:** Las clases ViewModel se encargan de gestionar los datos de la interfaz, asegurando su persistencia y facilitando la comunicación con la capa de dominio. Utilizamos flujos de estado para mejorar la comunicación entre la interfaz y el ViewModel.

Capa de Dominio

- En GameHub360, la Capa de Dominio se utiliza para simplificar las interacciones entre la Capa de Interfaz de Usuario y la Capa de Datos. En resumen, la capa de dominio se encargará de coordinar la obtención de los datos necesarios de la capa de datos y transformarlos en un formato que pueda ser utilizado por la Capa de Interfaz de Usuario, sin que esta tenga que preocuparse por la forma en que se han obtenido los datos ni por su estructura.

Capa de Datos

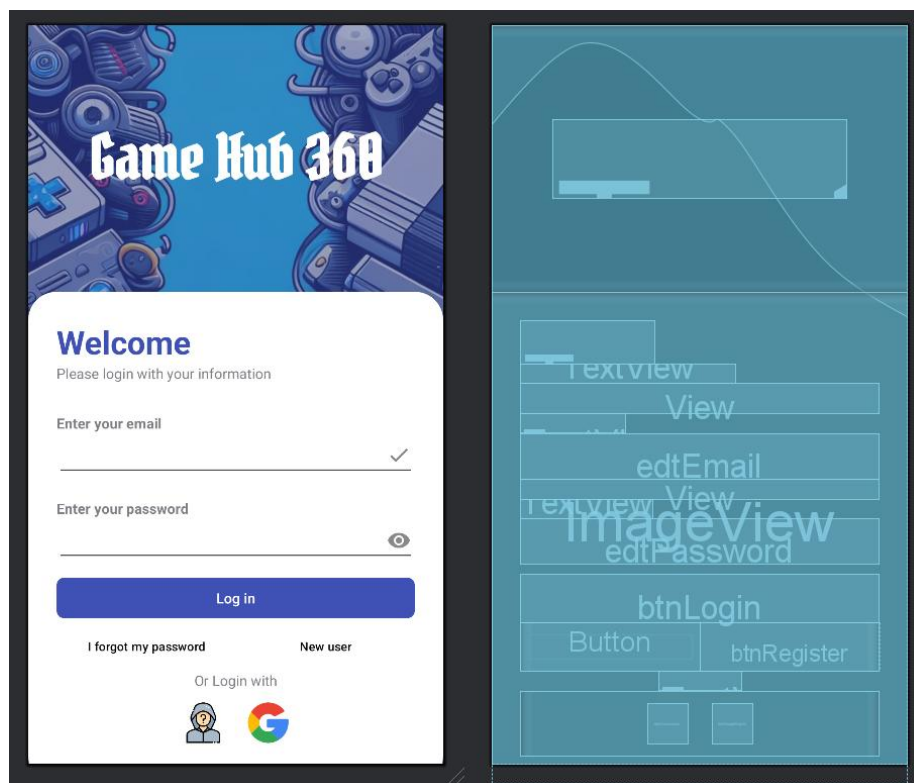
- **Repositorios:** Los repositorios actúan como intermediarios entre las fuentes de datos (API de RAWG.IO y almacenamiento local) y la Capa de Dominio. Gestionan la obtención y almacenamiento de datos, así como su transformación para adaptarlos a las necesidades de la aplicación.
- **Fuentes de Datos:** Utilizamos interfaces y clases para establecer la conexión con la API de RAWG.IO y el almacenamiento local (usando Room). Estas clases son responsables de interactuar con los servicios y gestionar el acceso a los datos.

Interfaces gráficas

Inicio de Sesión con correo o Google

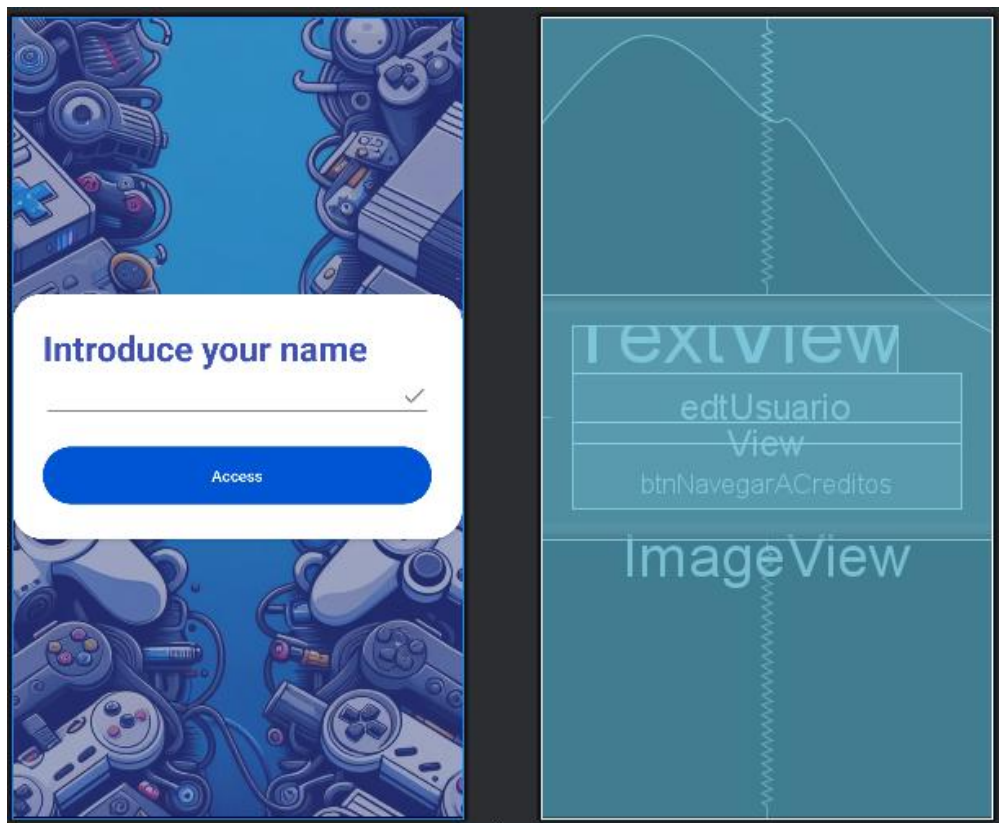
Esta es la primera interfaz que el usuario ve nada más entrar a la aplicación, sus funcionalidades son las siguientes:

- ➔ Ingreso de correo electrónico y contraseña: El usuario puede ingresar su correo electrónico y contraseña en los campos de texto correspondientes. Estos campos están diseñados para aceptar y ocultar la entrada de texto respectivamente.
- ➔ Iniciar sesión: Una vez que el usuario ha ingresado su correo electrónico y contraseña, puede iniciar sesión en la aplicación haciendo clic en el botón correspondiente.
- ➔ Registro de nuevo usuario: Si el usuario no tiene una cuenta, puede navegar a la pantalla de registro
- ➔ Restablecimiento de contraseña: Si el usuario ha olvidado su contraseña, puede navegar a la pantalla de restablecimiento de contraseña
- ➔ Inicio de sesión con Google o de forma anónima: Además de iniciar sesión con un correo electrónico y una contraseña, el usuario también tiene la opción de iniciar sesión con Google o de forma anónima.



Inicio de Sesión como anónimo

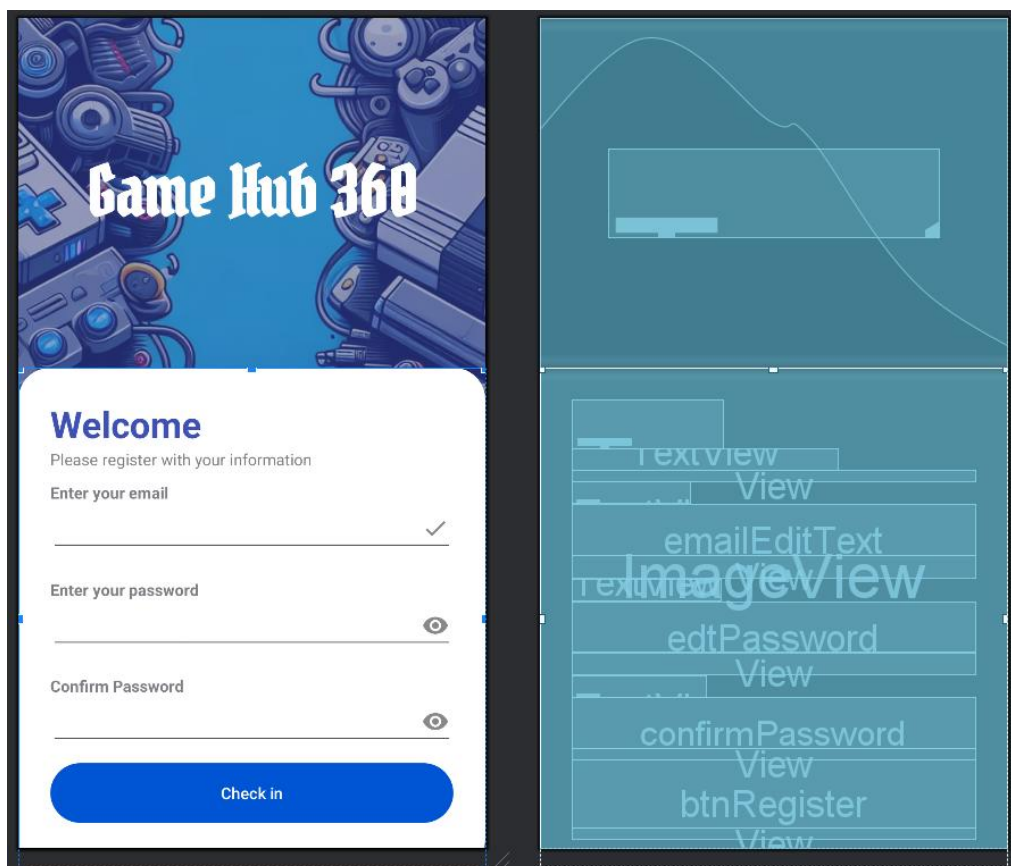
En esta interfaz el usuario pueda añadir su nombre para continuar como anónimo



Registro

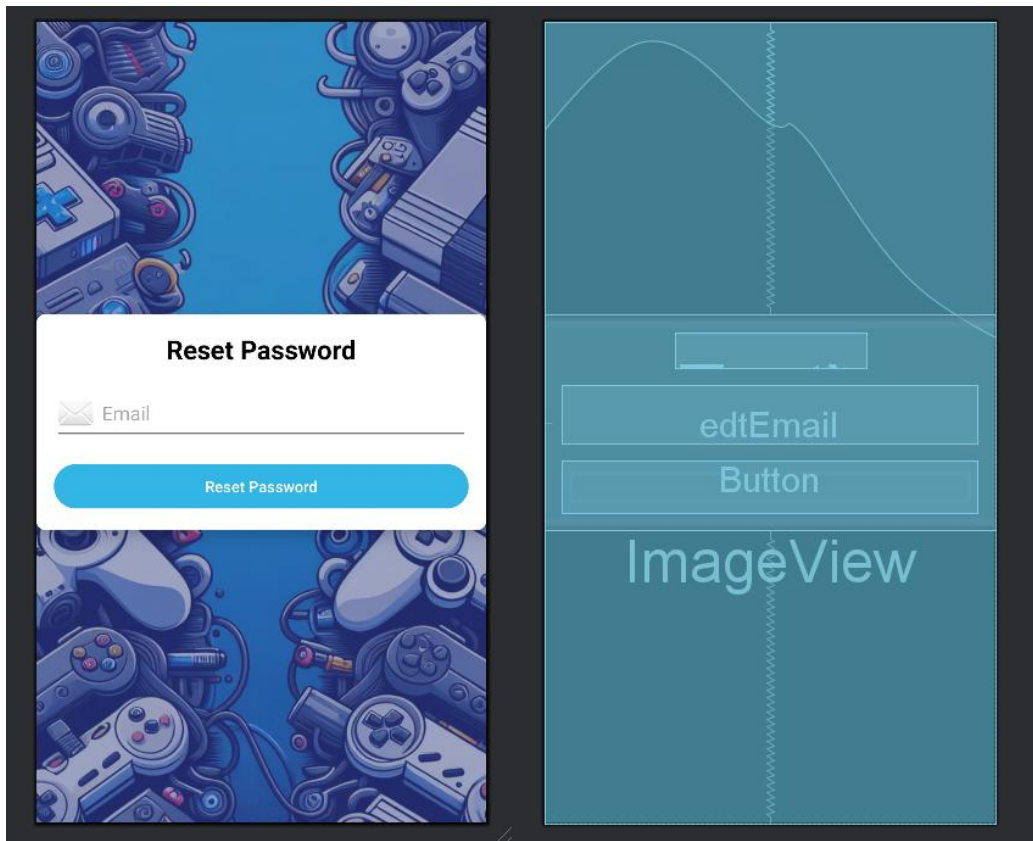
Esta interfaz está hecha para que el usuario se registre, su funcionalidad es muy simple y clara

- ➔ El usuario debe introducir su correo y contraseña, esta última dos veces para confirmar que es correcta
- ➔ Tiene un botón abajo para crear la cuenta una vez se ha validado todos los datos que le usuario ha introducido



Olvido de contraseña

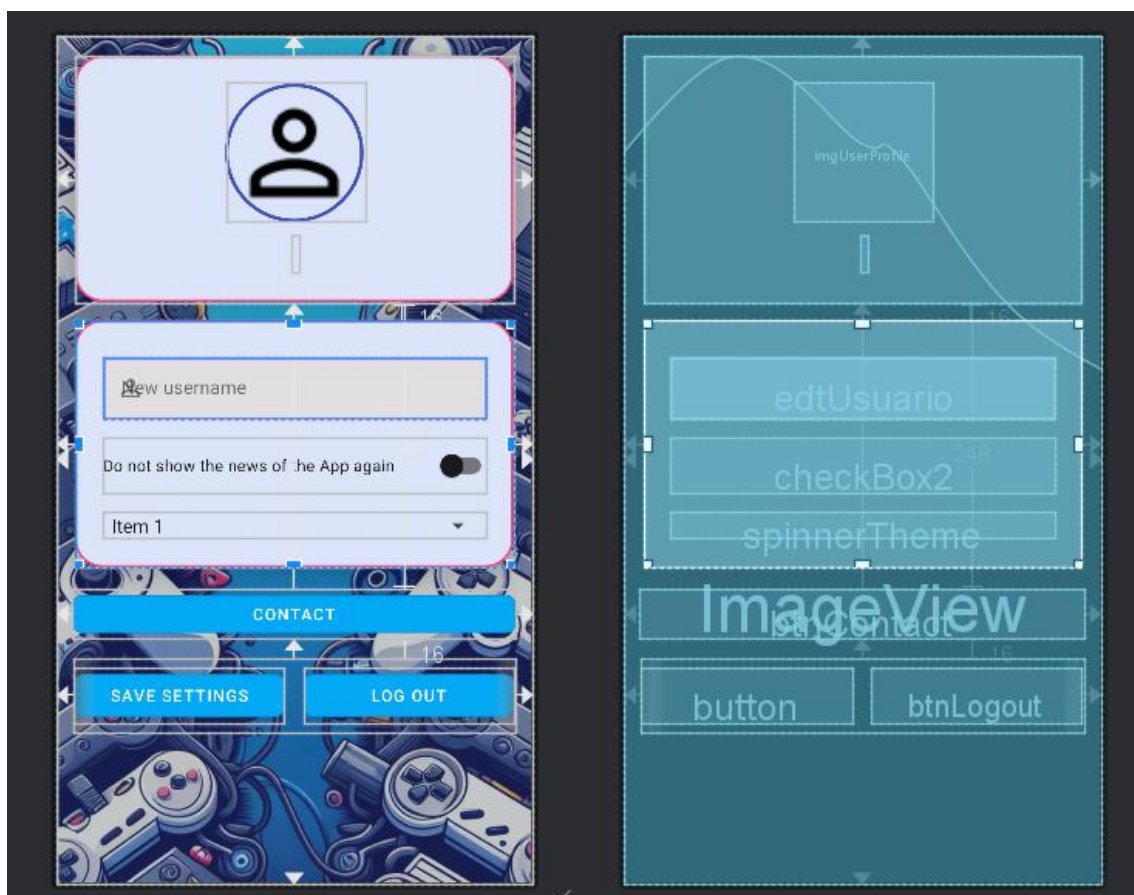
Esta interfaz permite al usuario restablecer su contraseña en caso de la haya olvidado, introduciendo su correo electrónico en el campo correspondiente



Configuración de la aplicación

Esta interfaz es la que usará el usuario para poder configurar su aplicación y ver sus datos, su interfaz está clara, estas son algunas de sus funcionalidades

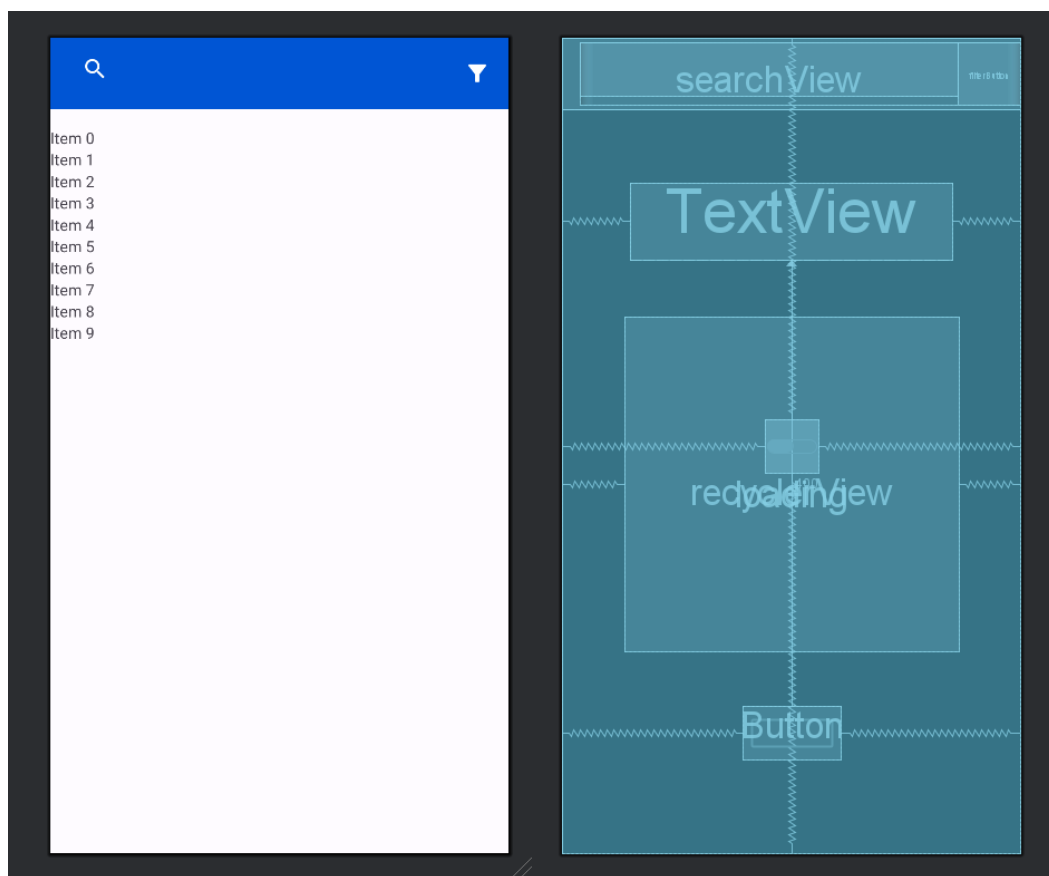
- ➔ Visualización de la imagen de perfil y nombre: El usuario puede ver su imagen de perfil, que se muestra en un círculo en la parte superior del fragmento, además de su nombre que aparecerá justo debajo
- ➔ Edición del nombre de usuario: El usuario puede editar su nombre de usuario ingresando un nuevo nombre en el campo de texto correspondiente.
- ➔ Configuración de las preferencias de la aplicación: El usuario puede configurar las preferencias de la aplicación, como si desea recibir novedades de la aplicación y el tema de la aplicación, utilizando el interruptor y el selector de opciones correspondientes.
- ➔ Contacto con el soporte: El usuario puede contactar con el soporte de la aplicación
- ➔ Guardar cambios: El usuario puede guardar los cambios realizados en sus preferencias y nombre de usuario
- ➔ Cerrar sesión: El usuario puede cerrar sesión en la aplicación



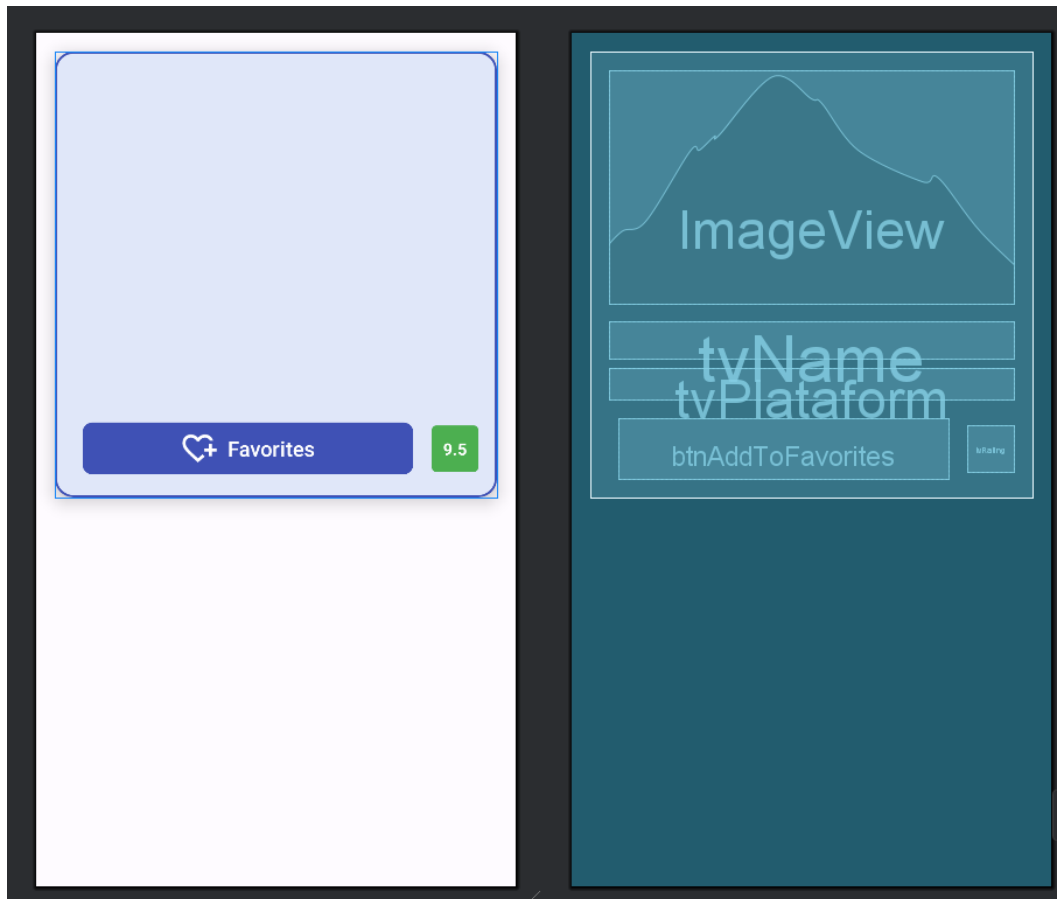
Lista de juegos

Esta es la interfaz principal de la aplicación en la cual el usuario puede ver la lista de juegos, además:

- ➔ Se puede buscar juegos en concreto
- ➔ Se puede buscar una lista de juegos por los filtros seleccionados
- ➔ Esta interfaz viene acompañada de animaciones que están oculta en el código, la cual será mostrada cuando sea necesario

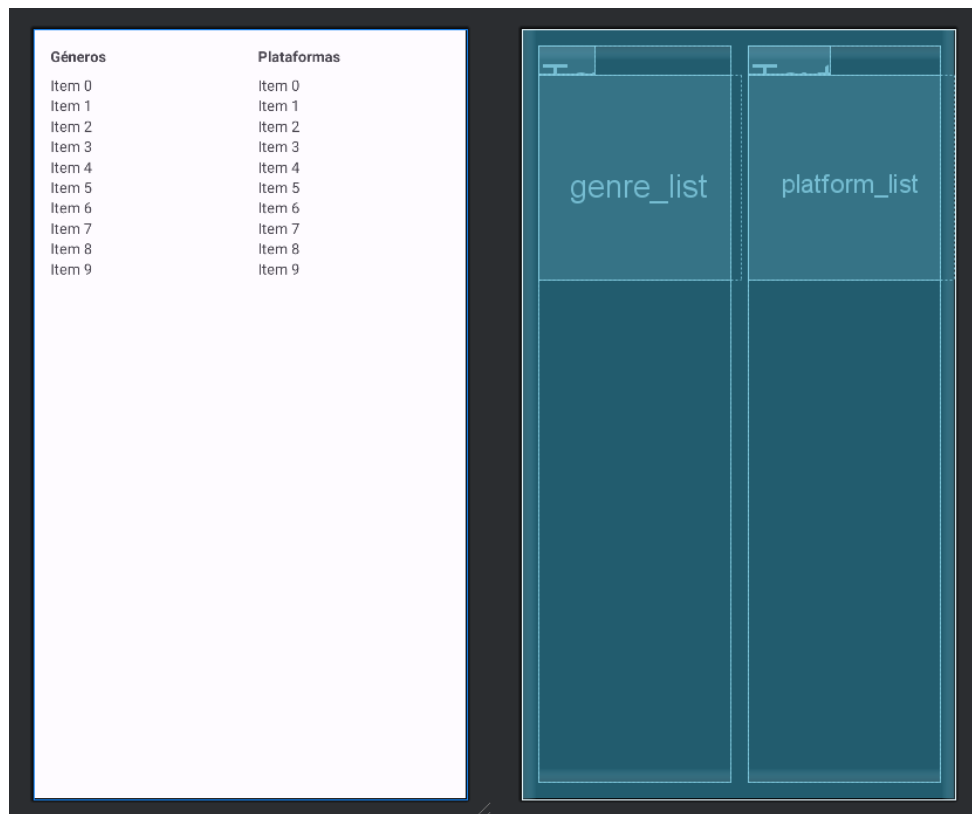


La interfaz anterior va acompañada de cada Item que representa cada juego, su diseño contiene información básica del juego además del botón para añadir a favoritos



Buscador de juegos por filtro

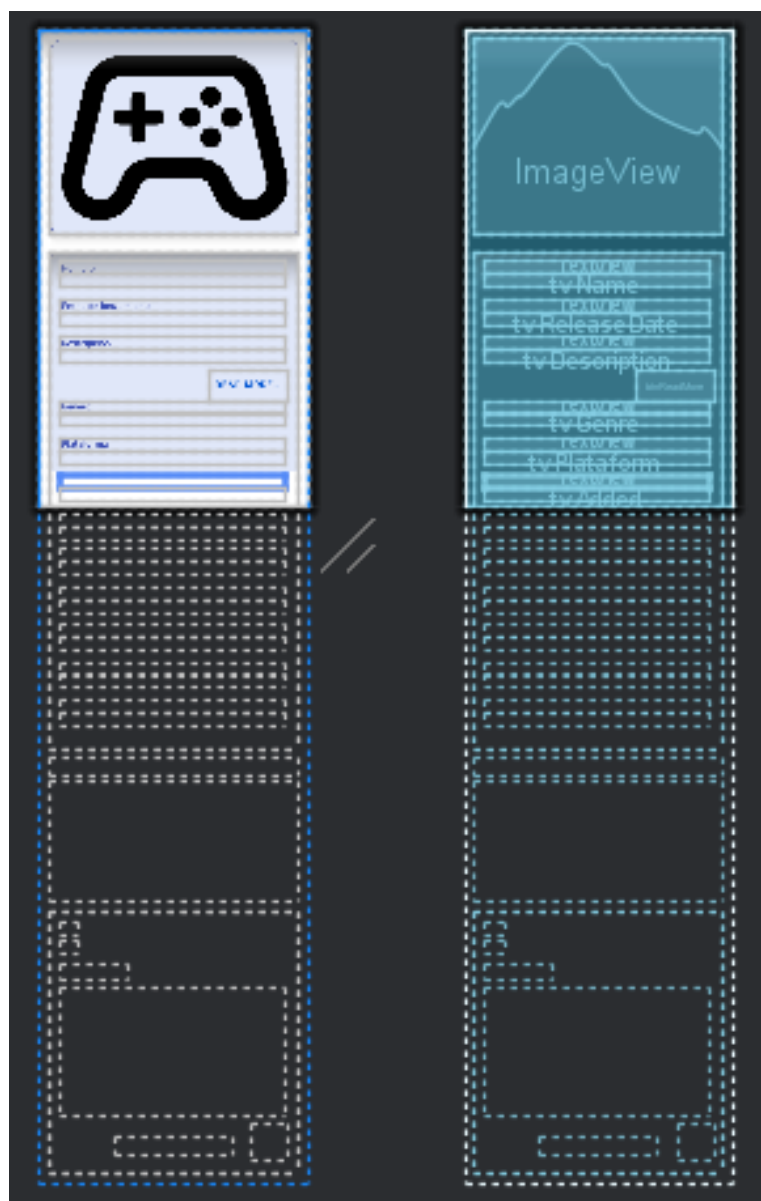
Esta interfaz será la encargada de buscar los juegos según los filtros que el usuario seleccione en la pantalla



Detalles del juego

Esta es la interfaz más currada y larga de la aplicación, muestra todos los detalles del juego, en resumen, en esta interfaz el usuario encuentra:

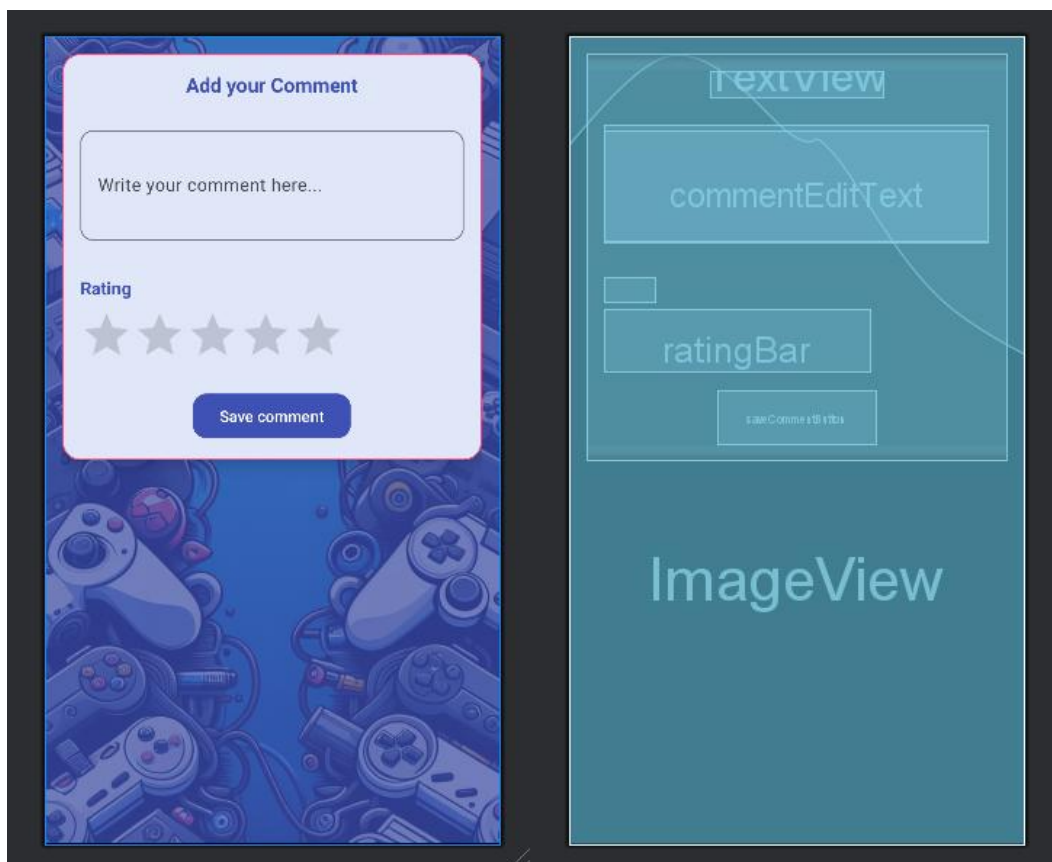
- ➔ Toda la información del juego
- ➔ Imágenes del juego
- ➔ Comentarios del juego de todos los usuarios
- ➔ Un botón para añadir un comentario al juego



Añadir comentario público

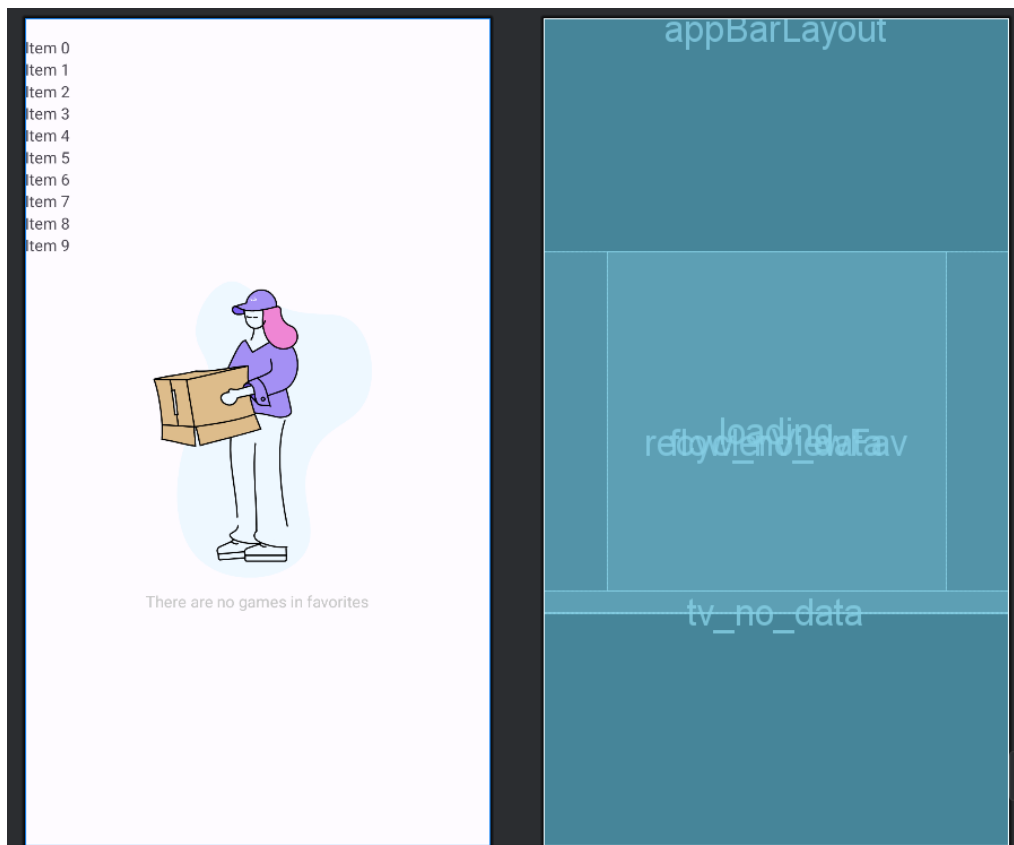
En esta interfaz el usuario añade el comentario al juego, este comentario es público, en la interfaz se recoge los datos del comentario como, por ejemplo:

- ➔ Descripción del comentario
- ➔ Valoración con estrellas, gracias a un rating
- ➔ Un botón para poder añadir el comentario



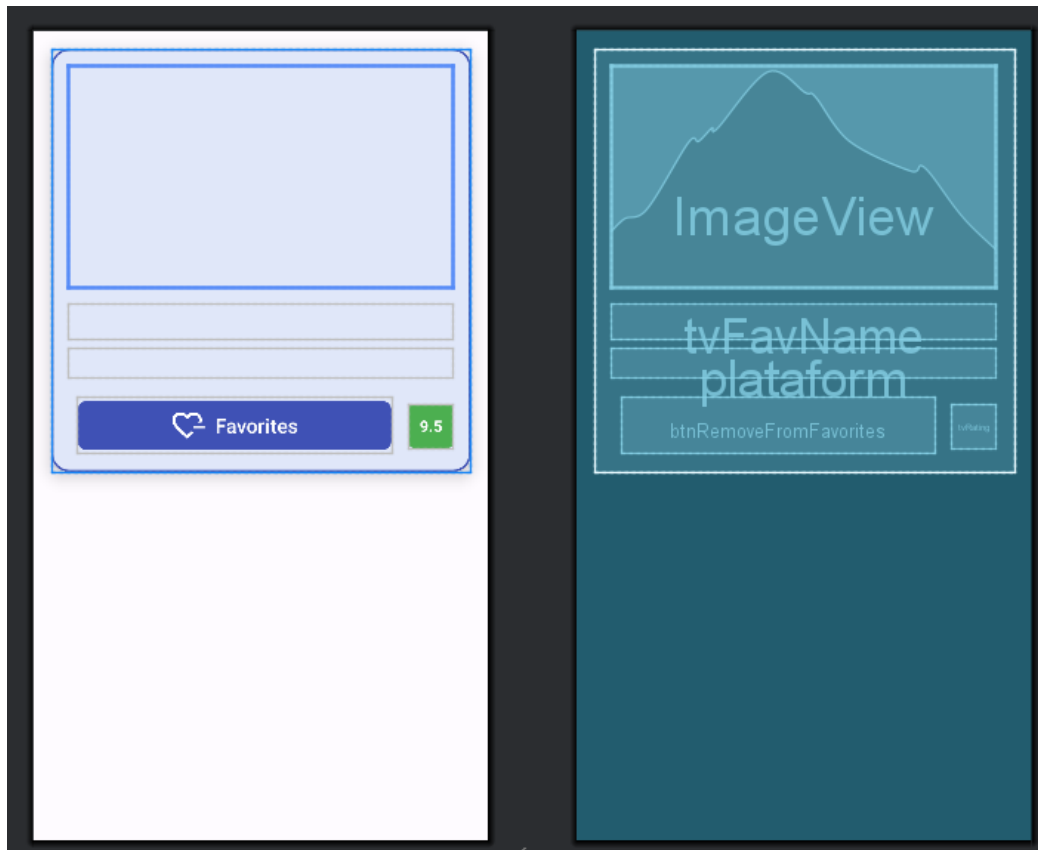
Lista de juegos favoritos

Esta es la interfaz encargada de mostrar la lista de juegos favoritos, además de contiene animaciones para el caso de que no haya juegos en favoritos



La interfaz anterior viene acompañada por este Ítem que representa cada juego, muestra al usuario:

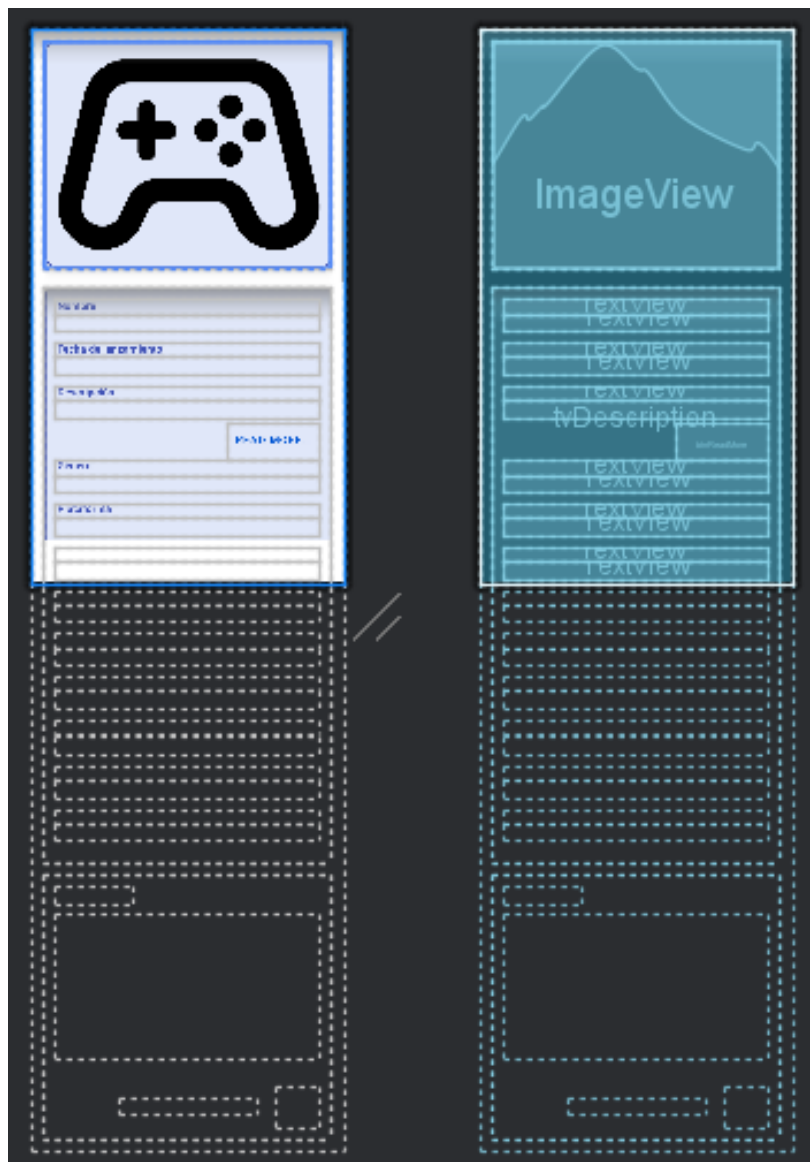
- ➔ La información principal del juego
- ➔ Un botón para eliminar el juego de favoritos



Detalles de un juego favorito

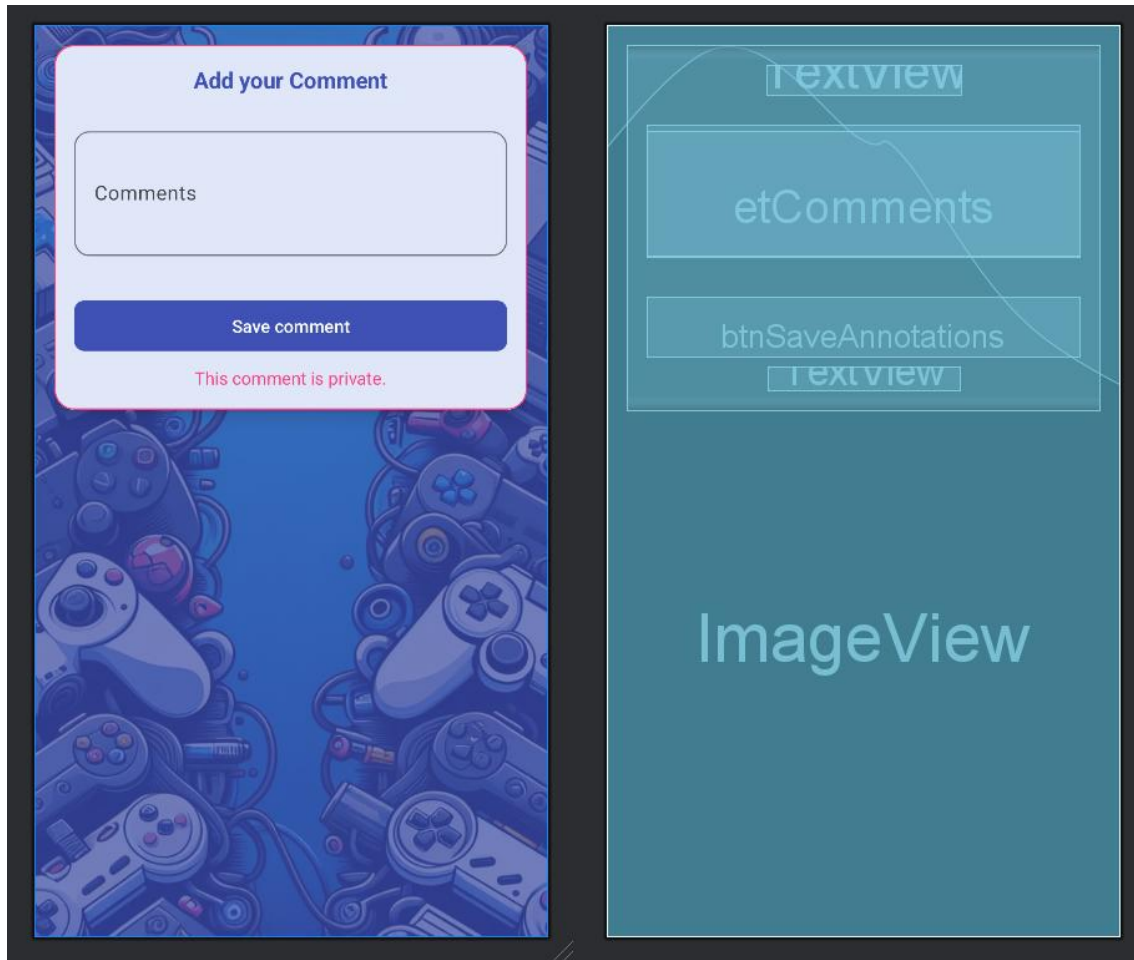
En esta interfaz el usuario podrá visualizar todos los detalles del juego favorito, suyo, la interfaz es idéntica a la de detalle de un juego cualquiera:

- ➔ Contiene toda la información del juego
- ➔ Añadir comentarios privados o anotaciones donde el usuario podrá ver y añadir sus anotaciones personales del juego



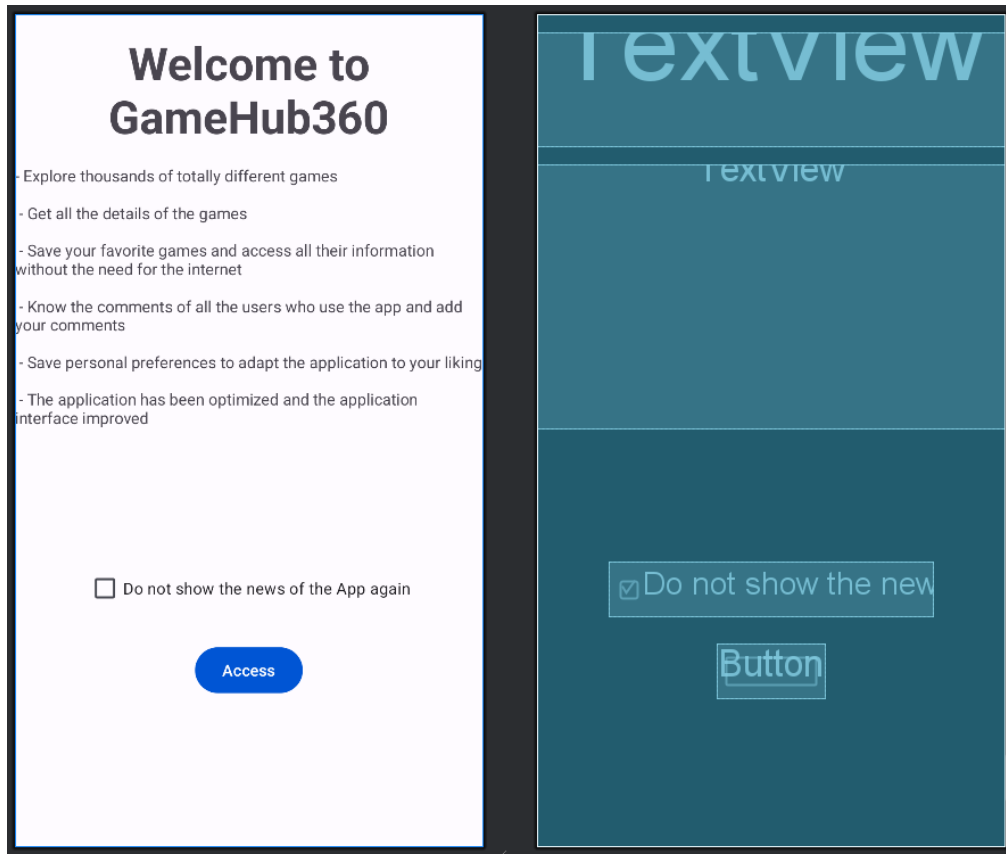
Añadir comentario privado

Esta interfaz es la encargada de que el usuario añada su comentario privado en su juego favorito, su interfaz es muy simple de entender y rápida



Novedades de la App

Esta interfaz se mostrará cuando se entre a la app, al menos que el usuario cambie esto en la configuración, en esta interfaz se muestra las novedades de la aplicación, para mantener informado al usuario



No Internet

Cuando no disponemos de internet, no saltará este fragmento el cual es el encargado de indicar que no disponemos de internet, además de tener un botón para poder volver a intentar la conectividad

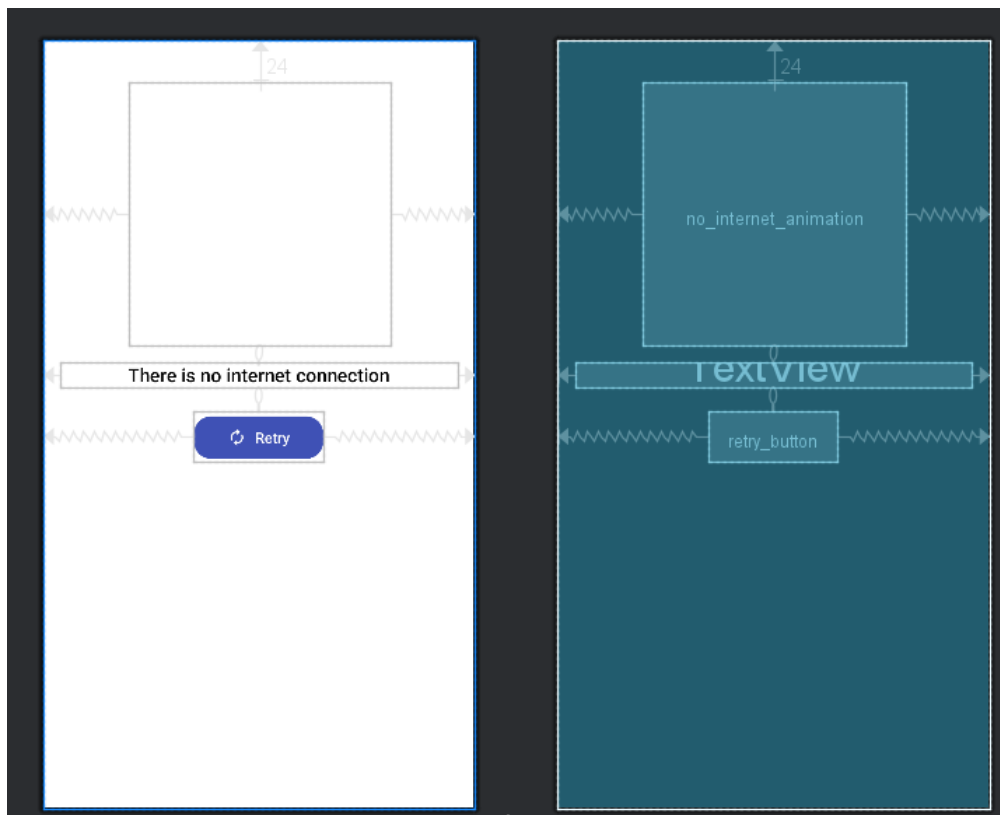


Diagrama de clases

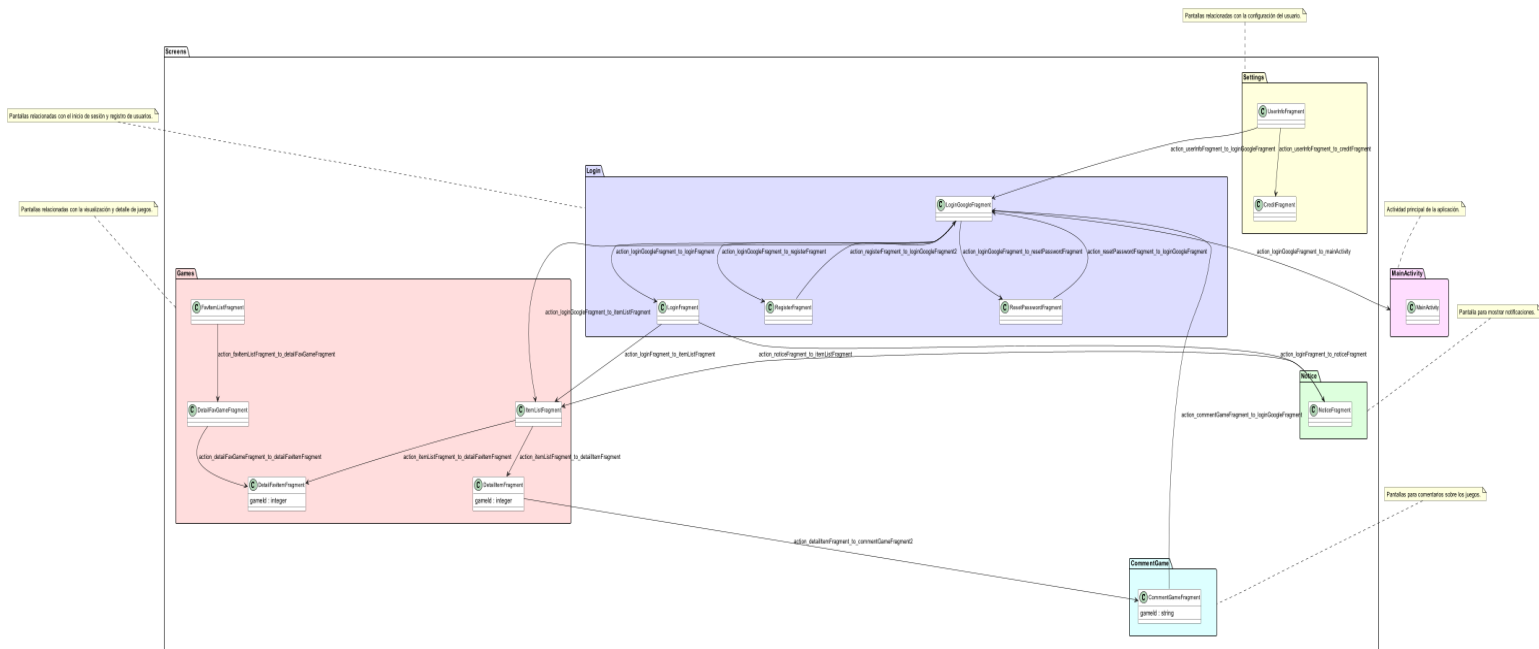


Diagrama de secuencia

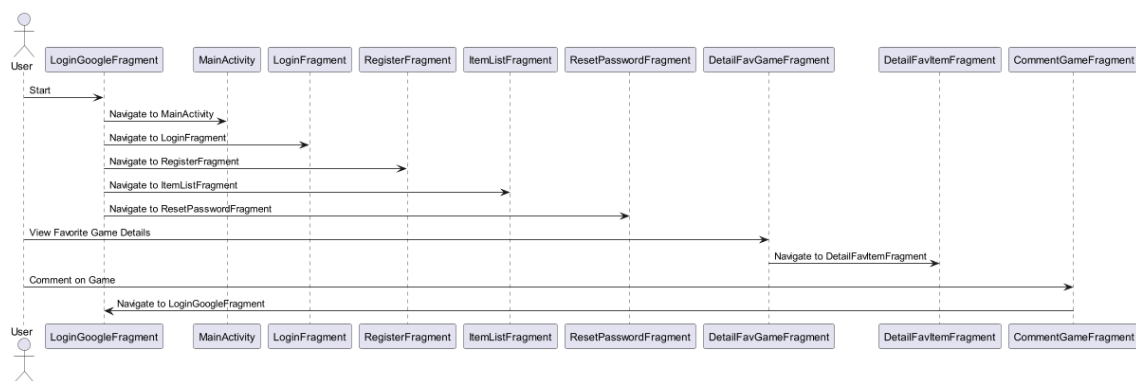
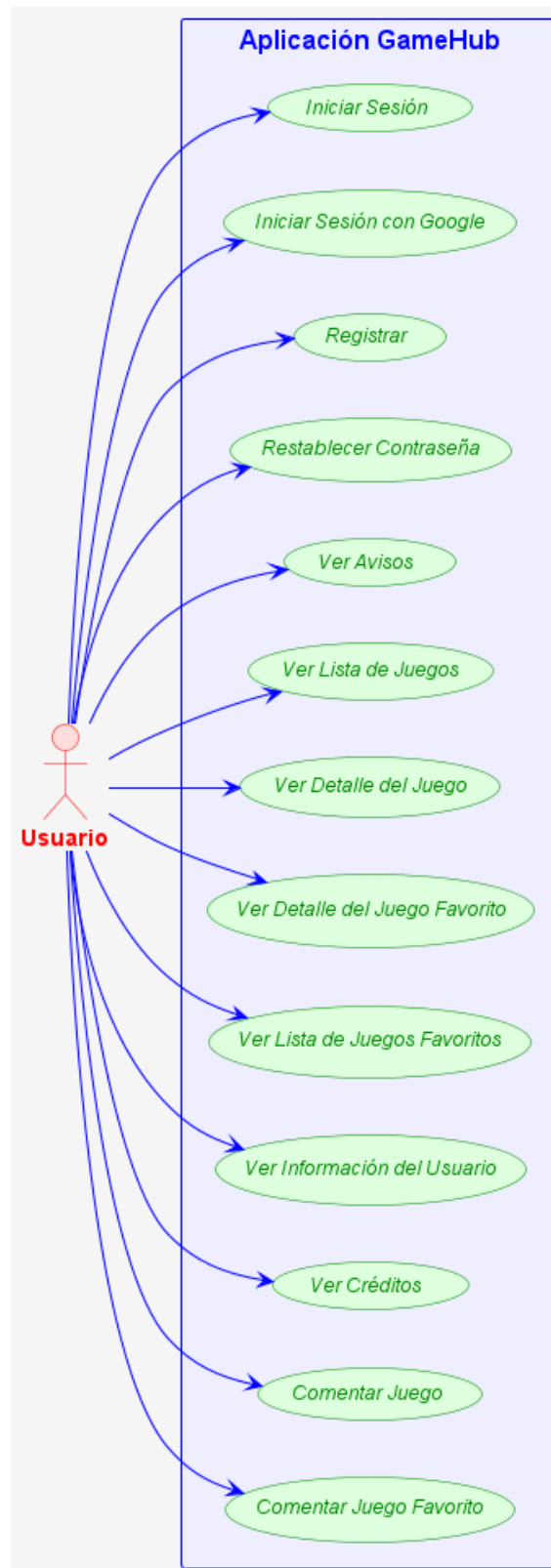
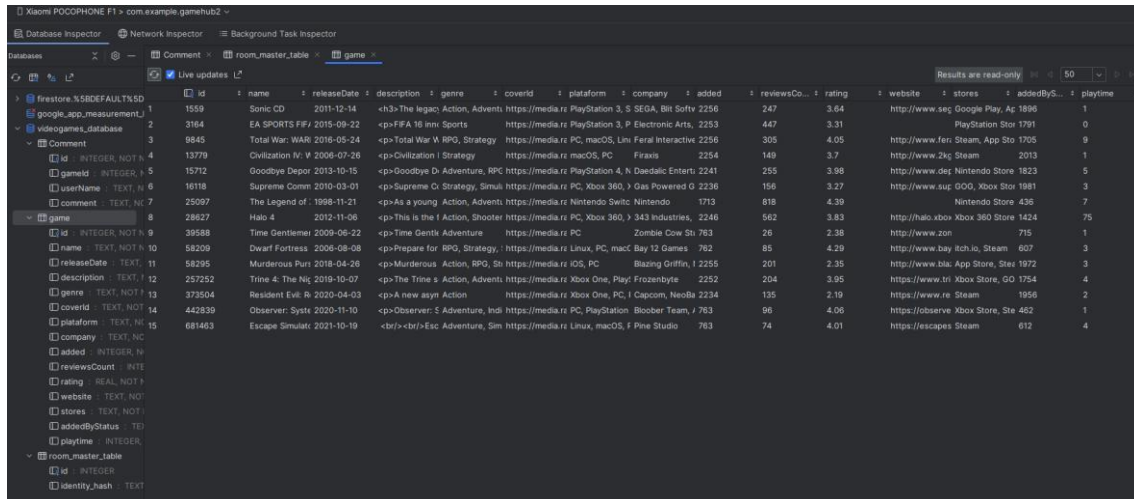


Diagrama de casos de uso



Modelo de datos

Base de datos ROOM



The screenshot shows the Database Inspector tool with the 'game' table selected. The table structure is as follows:

id	name	releaseDate	description	genre	coverId	platform	company	added	reviewsCount	rating	website	stores	addedByStatus	playtime
1	Sonic CD	2011-12-14	<h>The legac Action, Advent	https://media.re PlayStation 3, S SEGA, Bit Softw	2256	247	3.64	http://www.seg Google Play, Ap	1896	1				
2	EA SPORTS FFF	2015-09-22	<p>FFFA 16 im: Sports	https://media.re PlayStation 3, P Electronic Arts, 2253	447	3.31	PlayStation Stor	1791	0					
3	Total War: WAR	2016-05-24	<p>Total War V. RPO, Strategy	https://media.re PC, macOS, Lin Feral Interactive	2256	305	4.05	http://www.feri Steam, App Sto	1705	9				
4	Civilization IV: V	2006-07-26	<p>Civilization I Strategy	https://media.re macOS, PC	2254	149	3.7	http://www.2kg Steam	2013	1				
5	Goodbye Depr	2013-10-15	<p>Goodbye D. Adventure, RPC	https://media.re PlayStation 4, N Daedalic Entert	2241	255	3.88	http://www.ded Nintendo Store	1823	5				
6	Supreme Comm	2010-03-01	<p>Supreme O. Strategy, Simul	https://media.re PC, Xbox 360, > Gas Powered G	2236	156	3.27	http://www.sug GOG, Xbox Stor	1981	3				
7	The Legend of	1998-11-21	<p>As a young Action, Advent	https://media.re Nintendo Switch	1713	818	4.39	Nintendo Store	438	7				
8	Halo 4	2012-11-06	<p>This is the f Action, Shooter	https://media.re Nintendo Switch	1713	562	3.83	http://halo.xbox Xbox 360 Store	1424	75				
9	Time Gentlem	2009-06-22	<p>Time Gentle Adventure	https://media.re PC	Zombie Cow Shi	763	26	2.38	http://www.zon	715	1			
10	Dwarf Fortress	2009-06-08	<p>Prepare for RPO, Strategy	https://media.re Linux, PC, macOS	Bay 12 Games	762	85	4.29	http://www.bay itch.io, Steam	607	3			
11	Murderous Puri	2018-04-26	<p>Murderous Action, RPO, Sh	https://media.re iOS, PC	Blazing Griffin, I	2255	201	2.35	http://www.bla App Store, Stei	1972	3			
12	Trine 4: The Nig	2019-10-07	<p>The Trine s Action, Advent	https://media.re Xbox One, Play	Frozenbyte	2252	204	3.95	https://www.tri Xbox Store, GO	1754	4			
13	Resident Evil: R	2020-04-03	<p>A new asym Action	https://media.re Xbox One, PC, I	Capcom, NeoBa	2234	135	2.19	https://www.re Steam	1956	2			
14	Observer: Syst	2020-11-10	<p>Observer: S Adventure, Ind	https://media.re PC, PlayStation	Bloober Team, J	763	96	4.06	https://observe Xbox Store, Ste	462	1			
15	Escape Simulat	2021-10-19	 Esc Adventure, Sim	https://media.re Linux, macOS, F	Pine Studio	763	74	4.01	https://escapes Steam	612	4			

Tenemos una tabla Comment, con la siguiente estructura:

- id: INTEGER, NOT NULL - gameId: INTEGER, NOT NULL - userName: TEXT, NOT NULL
- comment: TEXT, NOT NULL

Tenemos una tabla game, con la siguiente estructura:

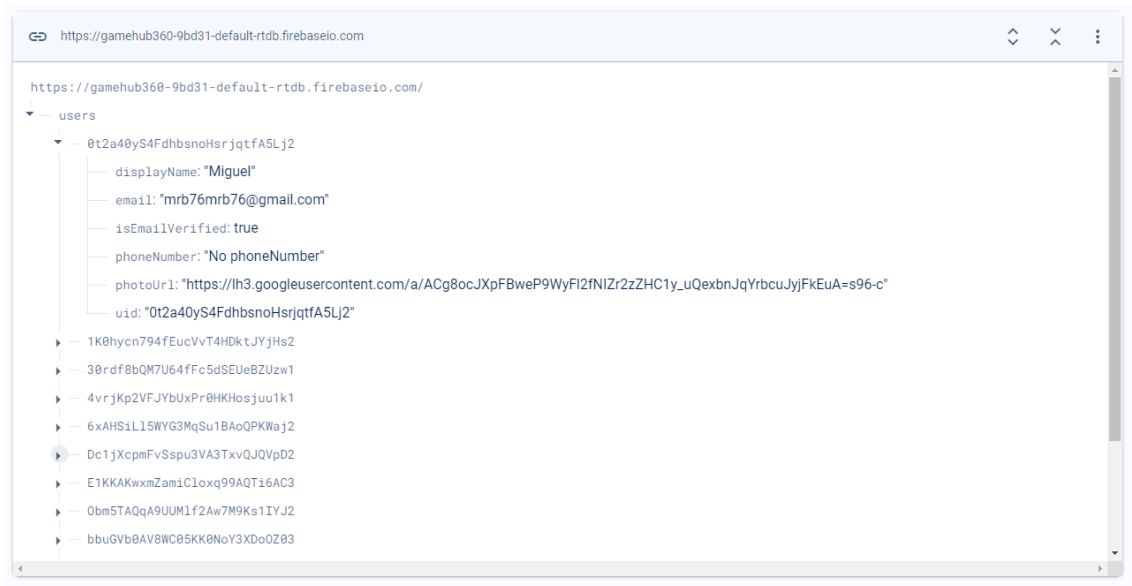
- id: INTEGER, NOT NULL, PRIMARY KEY - name: TEXT, NOT NULL - releaseDate: TEXT, NOT NULL - description: TEXT, NOT NULL - genre: TEXT, NOT NULL - coverId: TEXT, NOT NULL - platform: TEXT, NOT NULL - company: TEXT, NOT NULL - added: INTEGER, NOT NULL - reviewsCount: INTEGER, NOT NULL - rating: REAL, NOT NULL - website: TEXT, NOT NULL - stores: TEXT, NOT NULL - addedByStatus: INTEGER, NOT NULL - playtime: INTEGER, NOT NULL

- La tabla Comment parece almacenar los comentarios de los usuarios sobre los videojuegos, con una referencia al gameId para identificar a qué juego pertenece cada comentario.

- La tabla game contiene información detallada sobre los videojuegos, incluyendo nombre, fecha de lanzamiento, descripción, género, plataforma, compañía, número de reseñas, calificación, sitio web, tiendas donde está disponible, el estado de agregado y el tiempo de juego.

- La tabla room_master_table es una tabla de sistema utilizada por ROOM para el manejo de la base de datos, en particular para la gestión de identidades de hash.

Base de datos Realtime Database



Para almacenar los usuarios se usa la base de datos NoSql de Google, los usuarios y sus datos tienen la siguiente estructura:

1. Raíz de la base de datos:

- La URL base de la base de datos es `https://gamehub360-9bd31-default-rtdb.firebaseio.com/``.

2. Nodo principal "users":

- Todos los usuarios se almacenan bajo el nodo principal denominado ``users``.

3. Nodos individuales de usuario:

- Cada usuario tiene un identificador único (``uid``) generado por Firebase. Estos identificadores sirven como las claves de los nodos hijos bajo el nodo ``users``.

Ejemplos de ``uid`` son:

- ``0t2a40yS4FdhsnoHsrjqtfA5Lj2``

4. Datos del usuario:

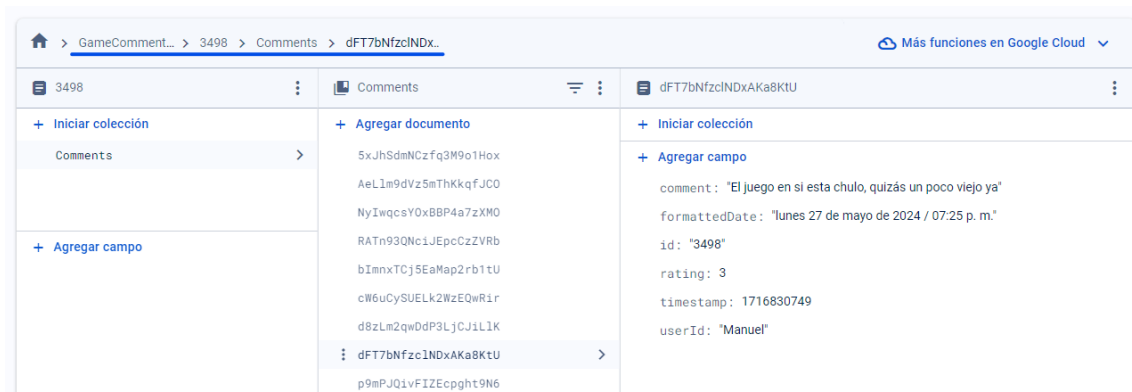
- Cada nodo de usuario contiene varios pares clave-valor que almacenan la información del usuario. Los datos específicos mostrados en la imagen para el usuario con ``uid`` ``0t2a40yS4FdhsnoHsrjqtfA5Lj2`` son:

- `displayName` - `email` - `isEmailVerified` - `phoneNumber` - `photoUrl` - `uid`

Base de datos Cloud Firestore

En esta base de datos se almacena tanto los juegos como los comentarios, pero cada uno en una colección diferente.

Colección comentarios



1. Raíz de la base de datos:

- La estructura comienza con la colección principal GameComments.

2. Documento de juego:

- Dentro de la colección GameComments, hay documentos que representan cada juego. En este caso, estamos viendo el documento con el ID `3498`.

3. Colección de comentarios:

- Dentro del documento del juego (`3498`), hay una subcolección denominada Comments. Esta subcolección almacena los comentarios relacionados con ese juego específico.

4. Documento de comentario:

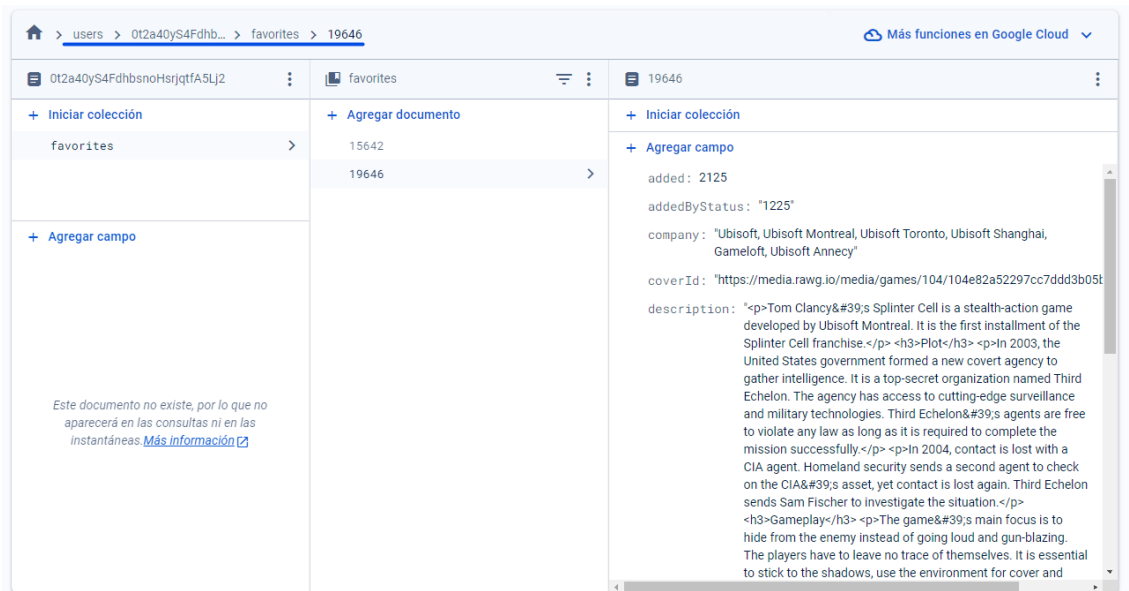
- Cada comentario es almacenado como un documento dentro de la subcolección Comments. En este ejemplo, el documento de comentario tiene el ID dFT7bNfzclNDxAKa8KtU

5. Campos del documento de comentario:

- Cada documento de comentario contiene varios campos que almacenan la información del comentario. Los datos específicos mostrados en la imagen para el comentario con el ID dFT7bNfzclNDxAKa8KtU son:

- comment - formattedDate - id - rating - timestamp - userId

Colección juegos



1. Raíz de la base de datos:

- La estructura comienza con la colección principal `users`.

2. Documento de usuario:

- Dentro de la colección `users`, hay documentos que representan a cada usuario. Gracias al id del usuario

3. Colección de favoritos:

- Dentro del documento del usuario (`0t2a40yS4FdhsnoHsrjqtfa5Lj2`), hay una subcolección denominada `favorites`. Esta subcolección almacena los juegos favoritos del usuario específico.

4. Documento de favorito:

- Cada juego favorito es almacenado como un documento dentro de la subcolección `favorites`. En este ejemplo, el documento de favorito tiene el ID `19646`.

5. Campos del documento de favorito:

- Cada documento de favorito contiene varios campos que almacenan la información sobre el juego favorito. Los datos específicos mostrados en la imagen para el documento con el ID `19646` son:

- `added` - `addedByStatus` - `company` - `coverId` - `description` - `genre` - `id` - `name` - `platform` - `playtime` - `releaseDate` - `reviewsCount` - `stores` - `website`

Conclusiones

Este proyecto me ha proporcionado un aprendizaje significativo. La parte del módulo que más me apasiona es el desarrollo de aplicaciones móviles, por lo que no dudé en elegir el proyecto final enfocado en este ámbito.

Al finalizar el curso, adquirimos un buen nivel de conocimiento en Kotlin y Android Studio, pero este proyecto me ha permitido seguir desarrollando mis habilidades y crear una aplicación completamente funcional y realista.

He dedicado muchas horas, especialmente a la investigación sobre interfaces y las tecnologías utilizadas. El tiempo limitado disponible ha sido un reto, ya que el tiempo ha pasado muy rápidamente, pero he disfrutado cada momento al ver los avances en la aplicación.

Afortunadamente, gestioné bien mi tiempo. Pensé que me costaría más, pero antes de comenzar a escribir código, realicé una investigación exhaustiva sobre todas las tecnologías que integraría en mi aplicación. Una vez que tuve clara la lógica, empecé a codificar. Este enfoque me facilitó enormemente el desarrollo, minimizando los errores y permitiéndome avanzar de manera fluida.

En resumen, he disfrutado enormemente de este proyecto. Las primeras semanas las dediqué exclusivamente a la investigación, y una vez que tuve todo claro, comencé a programar. Esta metodología me ayudó a asegurarme de que tenía una comprensión completa de lo que debía hacer, resultando en una aplicación interesante y detallada, finalizada dentro del tiempo previsto.

Dificultades encontradas

Este proyecto me ha proporcionado un aprendizaje significativo. La parte del módulo que más me apasiona es el desarrollo de aplicaciones móviles, por lo que no dudé en elegir el proyecto final enfocado en este ámbito.

Al finalizar el curso, adquirimos un buen nivel de conocimiento en Kotlin y Android Studio, pero este proyecto me ha permitido seguir desarrollando mis habilidades y crear una aplicación completamente funcional y realista.

He dedicado muchas horas, especialmente a la investigación sobre interfaces y las tecnologías utilizadas. El tiempo limitado disponible ha sido un reto, ya que el tiempo ha pasado muy rápidamente, pero he disfrutado cada momento al ver los avances en la aplicación.

Afortunadamente, gestioné bien mi tiempo. Pensé que me costaría más, pero antes de comenzar a escribir código, realicé una investigación exhaustiva sobre todas las tecnologías que integraría en mi aplicación. Una vez que tuve clara la lógica, empecé a codificar. Este enfoque me facilitó enormemente el desarrollo, minimizando los errores y permitiéndome avanzar de manera fluida.

En resumen, he disfrutado enormemente de este proyecto. Las primeras semanas las dediqué exclusivamente a la investigación, y una vez que tuve todo claro, comencé a programar. Esta metodología me ayudó a asegurarme de que tenía una comprensión completa de lo que debía hacer, resultando en una aplicación interesante y detallada, finalizada dentro del tiempo previsto.

Documentación

Lo que más esfuerzo y, en ocasiones, más pereza me causó fue la fase de investigación. Esta fue la parte más desafiante, ya que algunas tecnologías no tienen una documentación clara. Además, la documentación existente tuvo que ser adaptada a mi aplicación, la cual utiliza la arquitectura MVVM en lugar de Compose, que es una tendencia actual.

Desarrollo de un Diseño Profesional

La parte más complicada fue el diseño de la interfaz. Pensar en cómo hacerlo visualmente atractivo, similar a las aplicaciones reales, e implementarlo, resultó ser un desafío considerable. No hay mucha documentación específica para las tecnologías que utilicé, lo que añadió un nivel de dificultad adicional.

Webgrafía

Documentación de Google

<https://firebase.google.com/docs/firestore?hl=es>

Material Design

<https://m3.material.io/>

Youtube Firebase

<https://youtu.be/xjsgRe7FTCU?si=6ddvOlSkDlNlupXT>

API

<https://api.rawg.io/docs/>

Enlace APK de la aplicación

Desde mi repositorio de GitHub puedes descargar el apk

Enlace: <https://github.com/migueelrb/GameHub360-APK.git>