KeysStoreApp

por Víctor Pareja Ramírez

Introducción

KeysStoreApp una aplicación Android que simula ser una tienda en la que se pueden comprar juegos. El proyecto está desarrollado en Kotlin con Jetpack Compose y sigue la arquitectura recomendada por Android.

Descarga

Inspiración

- Idea de negocio
  - Varias webs de compra-venta de claves de videojuegos, como InstantGaming, G2A o Eneba.
- Desarrollo del proyecto en Jetpack Compose
  - Charlas como el DevSummit de Google
  - El proyecto NowInAndroid del equipo de Android oficial
  - El aprendizaje de Compose, el futuro del desarrollo en Android

Arquitectura

El proyecto tiene una arquitectura MVVM, una arquitectura que ayuda a mantener una mayor separación de responsabilidades y facilita el desarrollo de la aplicación. Esta arquitectura cuenta con dos capas principales:

- UI Layer -> se encarga de mostrar los datos en la pantalla
- Data Layer -> contiene la lógica de la aplicación y emite los datos a la capa de la interfaz del usuario

UI Layer

Esta capa se encarga de mostrar los datos en la pantalla del dispositivo. Esta capa se compone de dos elementos:

- Elementos de la UI, que renderizan los datos en la pantalla
- Contenedores de estado o ViewModels. Los ViewModels retienen los datos de la capa de datos, los exponen a la UI mediante StateFlows y controlan la lógica.

Data Layer

Esta capa contiene la lógica de la aplicación. En esta capa se realizan acciones como:

- Llamadas a la base de datos de Firebase, a las preferencias de usuario, a Room Database…
- El parseo de JSON a objeto

Esta capa:

- emite los datos mediante un flujo unidireccional
- es la capa en la que se inyectan los módulos Singleton de la aplicación
- es una capa formada por repositorios. Debe haber un repositorio por cada módulo que exista en la aplicación.

Patrón Singleton

El proyecto cuenta con el patrón Singleton, patrón que consiste en que solo haya una única fuente de información para un objeto en concreto además de garantizar que solo haya una instancia del mismo. Este patrón de diseño tiene como ventajas:

- Crear módulos independientes y accesibles desde cualquier parte de la aplicación
- Mejora el rendimiento de la aplicación al evitar las múltiples instancias

Aunque también tiene desventajas

- Es una herramienta muy útil pero el uso excesivo de Singletons puede llevar a una aplicación difícil de mantener debido a que haya objetos estáticos siempre instanciados.
- No es recomendable utilizarlo para implementar funcionalidades complejas ya que puede resultar difícil en un futuro mantener esa dependencia.

Jetpack Compose

Jetpack Compose es una biblioteca de diseño para interfaces de usuario declarativa, es decir, las vistas se pintan en pantalla a medida que se va ejecutando.

- Evita la sobrecarga de vistas, ya que estas aparecen y desaparecen en ejecución
- Menos código y facilidad para reutilizarlo
- Al no haber actividades hace la aplicación mucho más ligera
- Código más fácil de leer
- El futuro del desarrollo Android

Aplicación

Demostración de la app

Futuras versiones

- Verificación de email de usuario
- Test unitarios
- Mejorar la navegación de la aplicación
- Añadir opciones de filtrado
- Idioma nuevo y ajuste para cambiarlo
- Comentarios dentro de la ficha
- Cambios de diseño

El código de este proyecto estará público en Github a principios de año.