Aplicación Android con Room y Navegación Avanzada

David Gómez Rivas

18 de enero de 2024

Índice

1. Introducción

Este proyecto es una aplicación Android que utiliza Room para la gestión de datos y una arquitectura basada en ViewModel y Fragmentos para una experiencia de usuario fluida y eficiente. La aplicación maneja entidades denominadas ‘Elementos‘, permitiendo su creación, visualización y búsqueda.

2. Análisis del Problema

La aplicación debe gestionar eficientemente la interacción del usuario, la navegación entre diferentes vistas y el manejo de datos persistente. Los desafíos incluyen: - Crear una interfaz de usuario intuitiva con navegación basada en fragmentos. - Gestionar el ciclo de vida de los datos y la comunicación entre componentes de la UI y la base de datos. - Proporcionar funcionalidades de búsqueda y visualización de datos.

3. Solución Propuesta

La solución implementa varias clases: - ‘Elemento.java‘: Define la entidad Elemento para la base de datos Room. - ‘ElementosBaseDeDatos.java‘: Configura la base de datos Room para almacenar Elementos. - ‘ElementosRepositorio.java‘: Actúa como intermediario entre la base de datos y la UI. - ‘ElementosViewModel.java‘: Provee datos a la UI y sobrevive a cambios de configuración. - ‘MainActivity.java‘: Gestiona la navegación y la interacción global de la aplicación. - ‘MostrarElementoFragment‘, ‘NuevoElementoFragment‘, ‘RecyclerBuscarFragment‘, ‘RecyclerElementosFragment‘, ‘RecyclerValoracionFragment‘: Diferentes Fragmentos para crear, mostrar y buscar Elementos.

4. Análisis, Diseño, Pseudocódigo y Diagrama

4.1. Análisis

4.1.1. Definición del Problema

Desarrollar una aplicación Android robusta y eficiente que gestione 'Elementos' con operaciones CRUD, manteniendo una experiencia de usuario fluida y una arquitectura limpia.

4.1.2. Datos de Entrada

Interacciones del usuario con la aplicación, como navegación, búsqueda y entrada de datos en formularios.

4.1.3. Información de Salida

Visualización de Elementos, navegación entre diferentes vistas y retroalimentación de operaciones de usuario.

4.1.4. Variables

Objetos 'Elemento', ViewModel, Base de Datos Room, y diferentes Fragmentos para manejar las vistas.

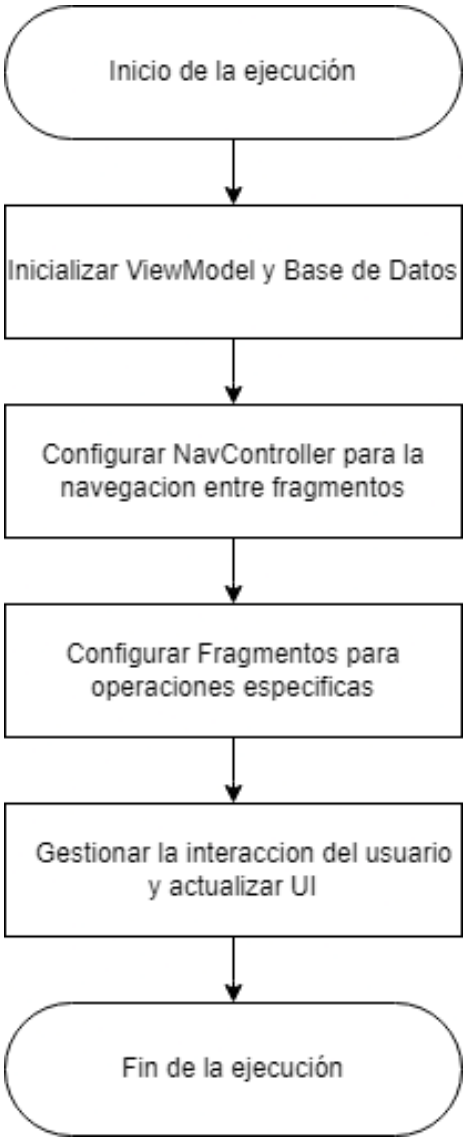
4.2. Diseño

El diseño utiliza un enfoque modular con ViewModel y Room, facilitando la separación de la lógica de negocio de la UI y una gestión eficiente de la base de datos.

4.3. Pseudocódigo

```
Proceso MainActivity
    Inicializar ViewModel y Base de Datos;
    Configurar NavController para la navegacion entre fragmentos;
    Configurar Fragmentos para operaciones especificas;
    Gestionar la interaccion del usuario y actualizar UI;
FinProceso
```

4.4. Diagrama de Flujo



5. Conclusión

La arquitectura y la implementación de la aplicación permiten una gestión eficiente de ‘Elementos‘, con una interfaz de usuario intuitiva y una sólida gestión de datos. La aplicación es escalable y mantiene una separación clara de

preocupaciones, lo que facilita futuras expansiones o modificaciones.