



Prueba técnica Android

Mercado Libre – v1.0.0

<https://github.com/andresmpuerto/mercadolibre>

Andrés Mauricio Martínez Puerto  
Ingeniero Informático / Android Developer  
andresmpuerto@gmail.com  
Año 2020

## Contenido

1. Análisis .....	3
2. Diseño de Arquitectura.....	3
3. Implementación.....	4
3.1. Librerías.....	4
3.2. UI.....	5
3.3. Testing.....	6
3.4. Integración Continúa.....	6
4. Referencias.....	6

## 1. Análisis

Teniendo en cuenta los requerimientos relacionados en la prueba para el desarrollo de la aplicación, me base en los siguientes aspectos para generar una app usable, escalable y segura.

- **Arquitectura**  
La estructura base es fundamental para determinar los componentes adecuados que permitan gestionar de manera eficiente los recursos del dispositivo, sea mantenible y permita la integración de nuevos módulos.
- **Seguridad**  
La información es un punto clave dentro cada aplicación, aunque el alcance de la prueba contenga información pública, se debe garantizar los parámetros básicos de seguridad que brinde confianza al usuario y mitigue los riesgos ante posibles amenazas.
- **Usabilidad**  
Las aplicaciones son elaboradas con el fin de que empatice y sea de utilidad para el usuario sin tener que ser muy compleja y bajo su experiencia de uso generar una cadena que la aplicación se pueda compartir.

Asimismo, es necesario diseñar e implementar una solución para la visualización de una gran cantidad de productos que deben ser listados y cargados mientras el usuario esta navegando. Con lo anterior, se decide crear una aplicación que incluya:

1. Selección del país para consulta de productos de cada uno
2. Búsqueda de producto y resultados.
3. Mostrar más resultados al hacer scroll
4. Detalle con información relevante de un producto
5. Listado de las últimas consultas
6. Configuración de la app

## 2. Diseño de Arquitectura

La Figura 1 ilustra la relación de los componentes en la arquitectura construida para el desarrollo de la aplicación de prueba.

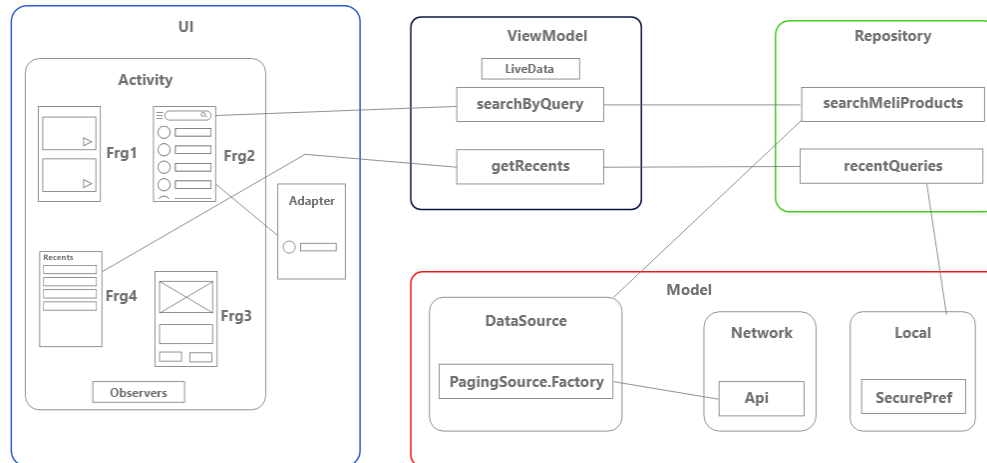


Figura 1. Arquitectura planteada

De acuerdo con la ilustración anterior se determino el uso de componentes bajo el patrón de diseño MVVM que permite el desacople de la lógica de la vista y las reglas de negocio y también que se logre la actualización constante de la información de una manera continua sin agregar complejidad a la implementación y siendo transparente para el usuario.

Para este fin, se plantea utilizar componentes que proporciona Android, componentes de arquitectura de Jetpack.

Componentes JetPack a utilizar:

- Paging Library
- Lifecycle Library (ViewModel, LiveData)
- DataBinding
- Navigation
- DataStore

### 3. Implementación

#### 3.1. Librerías

##### AndroidX Libraries

Componentes de Android que brinda soporte para todas las versiones disponibles.

**Jeckpack - Navigation**

Librería que permite realizar un flujo visual entre las diferentes pantallas

**Material Design**

Paquete que contiene componentes para construir las vistas con la mejor UI/UX para el usuario

**Test Libraries**

Paquetes para realizar pruebas unitarias, de integración de ui

**Dagger Hilt**

Librería para implementación de Inyección de dependencias que mitiga la reutilización de código

**Timber**

Librería para logging de app

**Retrofit**

Librería para implementación de REST Api

**Coroutines**

Librería que permite la ejecución de tareas en segundo plano

**Jetpack - Paging Library**

Librería muy completa para la gestión de listados de información en nuestros RecyclerViews

**Jetpack -DataStore**

Un nuevo actor que nos ayuda para almacenar los datos locales, la mejora a las preferencias.

**Glide**

Librería que nos ayuda con la gestión de las imágenes

## 3.2. UI

Para las interfaces de usuario se utiliza la librería de Material Design, implementando las recomendaciones para fuentes, tamaños de letra, estilos de cada componente gráfico de acuerdo con la diagramación adecuada para el ejercicio actual, temas que incluye el cambio a modo oscuro y la configuración de los colores.

### 3.3. Testing

### 3.4. Integración Continúa

Para esta implementación me apoye de la herramienta GitHub Actions que permite la automatización de las acciones realizadas dentro del repositorio (push, pull request), permitiendo ejecutar los pasos necesarios para testear, compilar y generar el artefacto (apk). De igual manera se debe hacer uso de GitFlow para tener control del ciclo de vida del versionamiento.

## 4. Referencias

<https://developer.android.com/topic/libraries/architecture/lifecycle.html#lc-bp>

<https://developer.android.com/jetpack/guide?hl=es-419>