**Documentación práctica – Desarollador Móvil Trainee | SPS**

**Por: Osvaldo Guerra Lanuza**

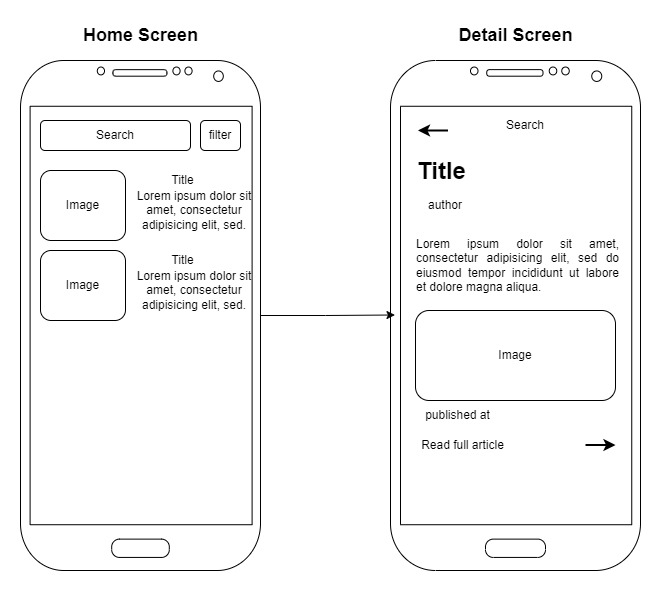
**Tecnologia**

La practica se desarrollo para Android nativo con kotlin.

Herramientas utilizadas:

* Jetpack Compose: es un kit de herramientas diseñado para el desarrollo de IU. Combina un modelo de programación reactivo con la facilidad de uso del lenguaje kotlin
* Retrofit2: Es un cliente REST para consumir servicios web con mayor facilidad
* Coil: Sirve para cargar y mostrar fotos de internet con Compose

**Diseño inicial**

****

Siguiendo el diseño presentado en las especificaciones de la practica se me ocurrio agregar dos componentes más que creo aportan funcionalidad a la aplicación, los cuales son un buscardor y su filtro. Leyendo la documentacion de la Api news me di cuenta que tiene paremetros de consulta como: language, sortBy, to, from; ademas de pageSize y page que ayudan a implementar una paginacion para no traer toda la informacion de una.

**Arquictectura**

La arquitectura que implemente es una arquitectura de 3 capas: data, domain, UI.

Texto

Descripción generada automáticamente con confianza baja

Capa Data

Pantalla de un celular con letras

Descripción generada automáticamente

En la capa de data tenemos el datasource que seria nuestra fuente de información, en este caso es remota (Servicio web). Dentro de la carperta remote se encuentra los model que vienen de la API y NewsService son las peticiones. Tambien tenemos la implementacion del repository.

Capa Domain

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En la capa domain esta el model que va mostrar la UI con ello un mapper. Tambien se encuentra la interfaz de repository. Esta capa más que nada sirve para separar de alguna manera la capa de data con la de UI

Capa UI

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En esta capa se encuentra todo lo relacionado a la interfaz de usuario, lo divido en dos carpetas screen y viewmodel. El viewmodel es el encargado de gestionar el estado de la UI por ello la clase UiState que representa el estado y la interfaz NewsEvent son los eventos que tiene la UI.