

Arquitectura de la App

Carlos Octavio Rodríguez del Toro

La capa de datos debería exponer los datos de la aplicación mediante un repositorio.

Descripción:

La capa de interfaz, elementos componibles, actividades o ViewModels no deben interactuar directamente con las fuentes de datos. En su lugar, se debe acceder a los datos mediante un repositorio como capa intermedia entre la capa de datos y la interfaz de usuario.

Decisión:

Sí

Justificación:

Al usar un repositorio, se podrán hacer modificaciones a la capa de datos sin afectar directamente a los componentes implementados en la interfaz de usuario.

Los ViewModels deben ser independientes del ciclo de vida de Android.

Descripción:

Los ViewModel no deben depender de ningún ciclo de vida de Android como Activity, Fragment, Context o Resources ya que están diseñados para gestionar y mantener los datos de la interfaz de usuario.

Decisión:

Sí

Justificación:

En caso de querer o necesitar modificar la arquitectura de la aplicación, esto se podrá hacer sin afectar a la lógica del ViewModel. Evitar estas dependencias proporciona flexibilidad para la adaptación y mejora la claridad en el diseño de la aplicación.

No anules los métodos de ciclo de vida de objetos Activity ni Fragment.

Descripción:

No se debe anular directamente un ciclo de vida como onResume en fragmentos o actividades. Se recomienda utilizar LifecycleObserver o la API repeatOnLifecycle.

Decisión:

Sí

Justificación:

Al emplear el método LifecycleObserver, se mejora la modularidad y se facilita la comprensión del código. Además, se evita el riesgo de olvidar llamar al método de la superclase ya que no es necesario con este método respecto a onResume.

Usa Hilt.

Descripción:

Hilt se utiliza para la gestión de dependencias. Hilt es una biblioteca de inyección de dependencias que simplifica y mejora la inyección de dependencias en aplicaciones Android.

Decisión:

No

Justificación:

El proyecto no es lo suficientemente complejo como para estar aplicando esta biblioteca. Aunque sea muy útil y recomendada, implicaría añadir una capa de complejidad innecesaria cuando se puede aplicar la inserción manual de dependencias. Hilt además tiene una curva de aprendizaje bastante grande y dada la inexperiencia de los desarrolladores y el poco tiempo para desarrollar el proyecto, no es la opción más viable.

Usa Jetpack Compose.

Descripción:

Usar la librería Jetpack Compose para construir la aplicación ya que simplifica el desarrollo de interfaces en Android.

Decisión:

Sí

Justificación:

Emplea un enfoque declarativo que simplifica la lógica y proporciona una sintaxis más simple y legible. Se podrá emplear mayoritariamente para la navegación por las pantallas de la aplicación aunque permitirá también gestionar los ciclos de vida y la persistencia de datos.

Bibliografía:

<https://developer.android.com/topic/architecture/recommendations?hl=es-419>