

**Universidad Nacional Politécnica, UNP**

Programación Móvil

Investigación II: Room

Fecha: 22 de junio de 2024

Profesor: Luis Guido Calderon

Ingeniería en computación V año

Joselin Raquel Morales Sotelo

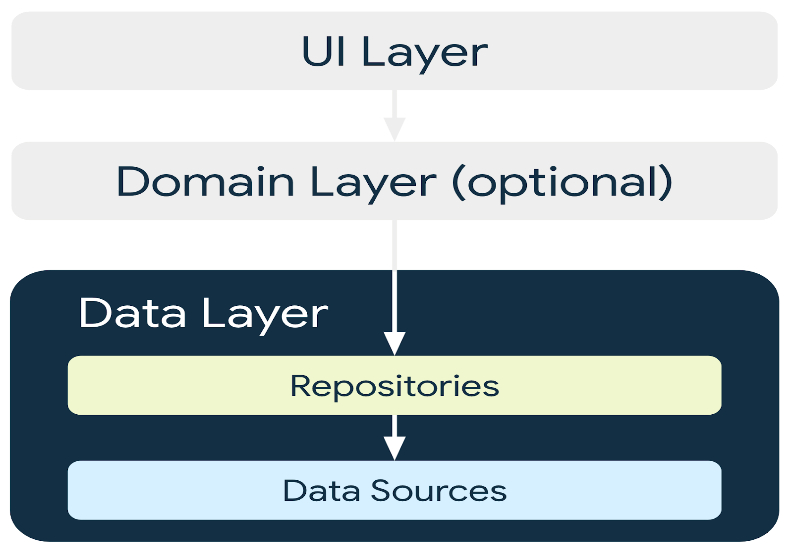
**Introducción**

La mayoría de las apps de calidad de producción tienen datos que deben guardarse. Por ejemplo, una app podría almacenar una lista de reproducción de canciones, elementos de una lista de tareas pendientes, registros de gastos e ingresos, un catálogo de constelaciones o un historial de datos personales. En la mayoría de estos casos, se usa una base de datos para almacenar esos datos persistentes.

Room es una biblioteca de persistencia en Android que ofrece una capa de abstracción sobre SQLite para permitir un acceso más robusto y fluido a la base de datos local en las aplicaciones Android. Fue presentada por Google como parte de la arquitectura de componentes de Android Jetpack, proporcionando una forma más sencilla y eficiente de manejar la persistencia de datos.

En este informe se detalla cada una de sus características y funciones, así también como es utilizada.

Room  es una biblioteca de persistencias que forma parte de Android Jetpack. Es una capa de abstracción que se ubica sobre una base de datos SQLite. SQLite usa un lenguaje especializado (SQL) para realizar operaciones de bases de datos. En lugar de usar SQLite directamente, Room simplifica las tareas de configuración de la base de datos, así como las interacciones con la app. Room también proporciona verificaciones en tiempo de compilación de las instrucciones de SQLite.

Una *capa de abstracción* es un conjunto de funciones que ocultan la implementación o la complejidad subyacente. Proporciona una interfaz para un conjunto existente de funciones, como SQLite en este caso.

En la siguiente imagen, se puede apreciar el modo en que Room, como fuente de datos, se adapta a la arquitectura general recomendada en este curso. Room es una fuente de datos.

**Características Principales**

1. **Objetivos Principales:**
   * Simplificar y optimizar el manejo de bases de datos SQLite en aplicaciones Android.
   * Promover prácticas de código limpio y mantenible mediante el uso de arquitectura de componentes.
2. **Componentes Clave:**
   * **Entity**: Representa una tabla dentro de la base de datos.
   * **DAO (Data Access Object)**: Define métodos que acceden a la base de datos.
   * **Database**: Contiene la base de datos y es el punto de acceso principal.
3. **Beneficios:**
   * **Menos Código Boilerplate**: Room reduce significativamente la cantidad de código necesario para interactuar con SQLite.
   * **Verificación en Tiempo de Compilación**: Ofrece verificación de consultas SQL en tiempo de compilación, evitando errores comunes de sintaxis.
   * **Integración con LiveData y Kotlin Coroutines**: Facilita la integración con componentes como LiveData para actualizaciones automáticas de la interfaz de usuario y Kotlin Coroutines para operaciones asincrónicas.
4. **Uso de Annotations:**
   * **@Entity**: Define una tabla de base de datos.
   * **@PrimaryKey**: Define la clave primaria de una entidad.
   * **@Dao**: Define métodos de acceso a datos.
   * **@Database**: Define la base de datos y especifica las entidades que contiene.
5. **Relaciones entre Tablas:**
   * Room permite establecer relaciones entre entidades usando anotaciones como **@Relation** y consultas SQL personalizadas.
6. **Migraciones de Base de Datos:**
   * Facilita la gestión de versiones de la base de datos mediante la migración de esquemas utilizando clases de migración.
7. **Soporte para Pruebas Unitarias:**
   * Room está diseñado para facilitar las pruebas unitarias mediante la inyección de dependencias y el aislamiento de la base de datos en pruebas de integración.

**Conclusión**

En conclusión, Room Persistence Library ha demostrado ser una herramienta invaluable para los desarrolladores de Android al simplificar el manejo de la persistencia de datos a través de SQLite. Su capacidad para reducir la complejidad del código, mejorar el rendimiento de las operaciones de base de datos y su integración fluida con otros componentes de Jetpack hacen que sea una opción ideal para aplicaciones que requieren una gestión eficiente y segura de datos locales. Al aprovechar Room, los desarrolladores pueden centrarse más en la lógica de la aplicación y menos en los detalles de implementación de la base de datos, mejorando así la productividad y la calidad del software desarrollado para la plataforma Android.