

SQLITE

SQLite en **Android Studio** es una base de datos ligera de código libre que permite el almacenamiento estructurado de datos directamente en la memoria del dispositivo móvil. Es parte de la API de Android, lo que significa que no necesitas instalar nada extra para usarla. SQL Lite es una base de datos relacional que utiliza lenguaje SQL (Structured Query Language), aunque es una versión simplificada, y es ideal para aplicaciones que necesitan almacenar y manipular datos estructurados sin requerir una conexión a internet, ya que los datos se guardan en el dispositivo. Crea un archivo para el esquema completo de una base de datos, lo que permite ahorrarse preocupaciones de seguridad, ya que los datos de las aplicaciones Android no pueden ser accedidos por contextos externos. Puedes encontrar más información en <https://www.sqlite.org/>

¿Cuándo se suele utilizar SQLite en Android?

SQLite es una buena opción cuando:

1. **La aplicación necesita almacenar datos de manera persistente en el dispositivo**, como listas de tareas, datos de usuario, preferencias o cualquier información que deba mantenerse entre sesiones.
2. **Los datos no son muy grandes o complejos**, pues SQLite no es ideal para bases de datos muy grandes.
3. **No es necesario un servidor externo**: se usa en aplicaciones que funcionan offline o en aplicaciones que solo necesitan sincronizar datos ocasionalmente.

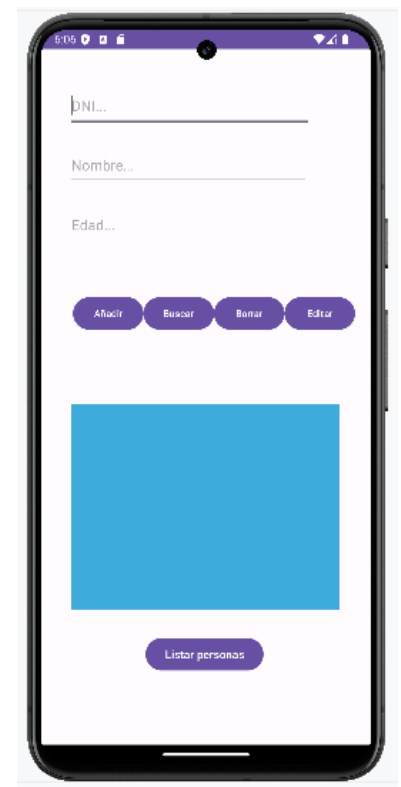
Ejemplo de uso de SQLite en Android Studio

Para comprender los diferentes usos, crearemos un proyecto sobre el que explicaremos diferentes aspectos:

1. Crea un proyecto: File→New Project → Empty Views Activity , lo llamaremos, por ejemplo, **SQLiteEjemplo**.

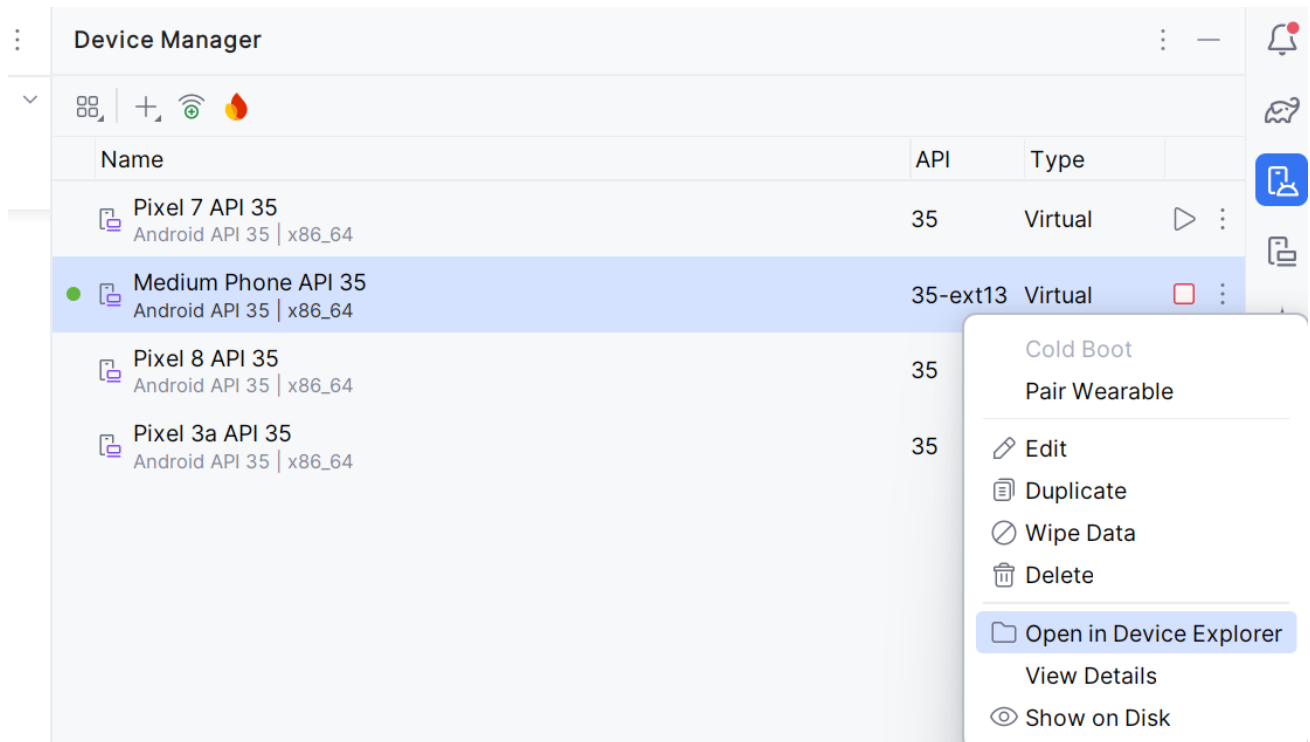
Construye una app que implemente las operaciones CRUD para gestionar personas.

- La clase **AdminSQLiteConexion**, la hacemos nosotros heredando de **SQLiteOpenHelper** y se sobrescriben los métodos: **onCreate** (se lanza cuando se ejecuta la primera vez, para crear la estructura de la base de datos, en nuestro caso crearemos la tabla persona) y **onUpgrade** (si la base de datos ya existe, se ejecuta este método, aquí no será necesario hacer nada)
- La clase **Persona** es nuestro modelo en este caso. Usa data class para representar los datos almacenados en la base de datos (dni, nombre y edad).
- La clase **PersonaConexiónHelper** es la que utiliza las librerías de SQLite. Pero esa clase la implementamos nosotros (insertarPersona, buscarPersonaByDNI, deletePersonaByDNI, updatePersonaByDNI, selectAllPersonas) usando los métodos de [SQLiteDatabase](#).

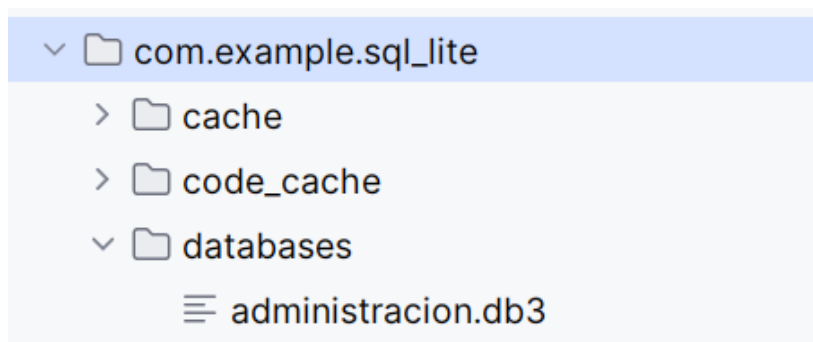


- El archivo con la base de datos **.db3** se encuentra en: *Device Explorer* → *data* → *data* → *com.example.sql_lite* → *databases*

Para acceder a Device Explorer, se puede habilitar desde el Device Manager, pulsamos en los 3 puntos y en el menú que se despliega, seleccionamos Open in Device Explorer



Desde Device Explorer, estamos accediendo al explorador de archivos del dispositivo en el que estamos probando nuestra aplicación. La primera vez que ejecutemos el código, se creará la base de datos en la ruta *data* → *data* → *com.example.sql_lite* → *databases*



Si se pulsa botón derecho sobre la base de datos, se puede descargar, ver la ruta completa. En un móvil real, tienes que tener activa la opción de desarrolladores.

En este enlace: <https://sqlitebrowser.org/dl/> se puede descargar un cliente llamado **SQL Lite Browser** para manejar base de datos en SQL Lite. Una vez hayas obtenido el archivo lo puedes abrir desde SQL Lite Browser.