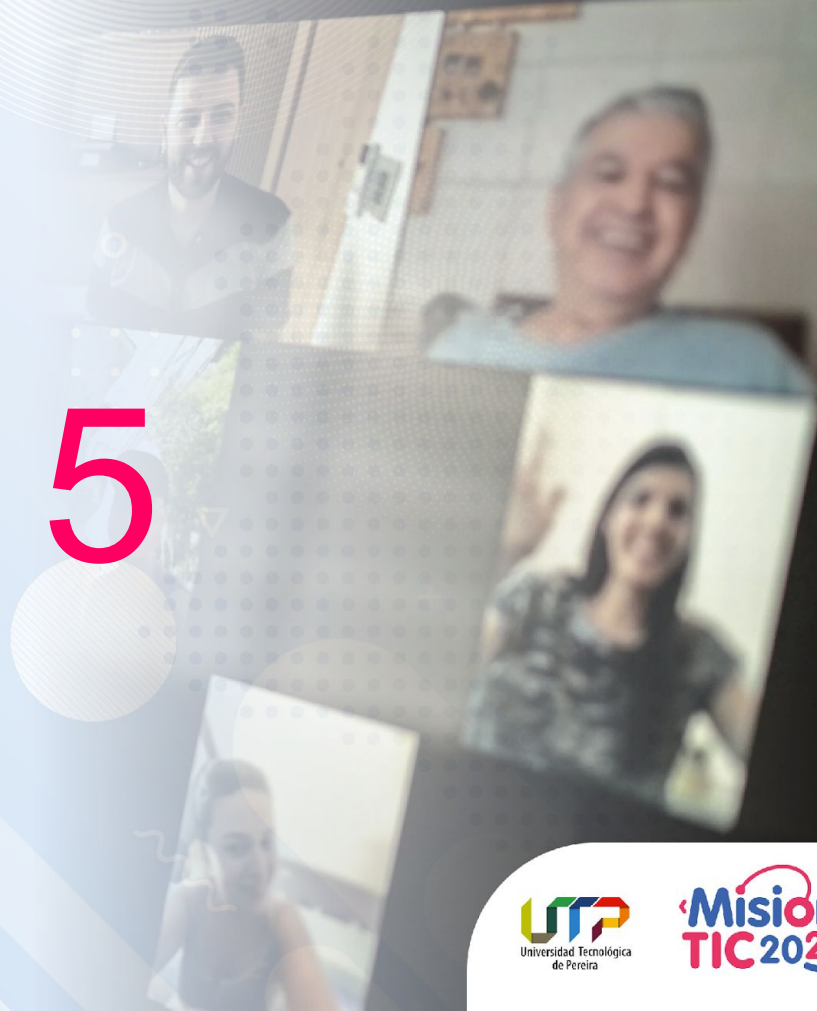




El futuro digital
es de todos

MinTIC

Unidad 5





El futuro digital
es de todos

MinTIC

Tema 3:

Almacenamiento de datos - SQLite en Android: Creación y Acceso Base de Datos



SQLite: Creación y Acceso Base de Datos



Dentro de los distintos sistemas de bases de datos tanto privativos como libres/open source (Oracle, SQLServer, MySQL, etc) existe uno que se adapta perfectamente a las aplicaciones móviles: **SQLite**. El principal motivo es que **SQLite no requiere más que un simple fichero para almacenar los datos**, ya que la lógica de funcionamiento debe ser implementada por la plataforma que desee interactuar con los datos.

En nuestro caso concreto el SDK
Android incluye soporte completo
para SQLite.





SQLite: Creación y Acceso Base de Datos

La mayoría de las aplicaciones móviles (tanto en Android como en otros sistemas) incluyen bases de datos SQLite, bien sea para la gestión total de los datos, o bien para al menos gestionar los datos almacenados localmente cuando las aplicaciones son parte de infraestructuras mayores que incluyen bases de datos centralizadas online o servicios Web para la obtención de los datos.





Creación de la base de datos SQLite

El procedimiento recomendado para crear una nueva base de datos *SQLite* es, de forma esquemática:

Paso 1

Crear una **clase** que **extienda** de ***SQLiteOpenHelper***

Paso 2

Sobreescribir en ella el método `onCreate()`, donde se ejecutará un comando de *SQLite* para crear las tablas de la base de datos.

Paso 3

También habría que **sobreescribir** el método `onUpgrade()`, el cual se ejecutará cada vez que cambiamos la versión de la base de datos y se usará para migrar los datos de la base de datos anterior a la nueva versión.

Paso 4

Se **creará** también el **constructor** de la clase que no será necesario en principio modificarlo.



Práctica

Paso 1

A continuación, vamos a crear una base de datos sencilla usando una clase derivada de *SQLiteOpenHelper* y que la llamaremos *MySQLiteHelper*.

Paso 2

En esta clase sobreescribimos el método `onCreate()` para crear una **tabla** que llamaremos **comments** y que estará compuesta por dos **campos**:

Paso 3

- Un entero (clave primaria) para identificar el dato y que se autoincrementa.
- Un campo de tipo texto.

Paso 4

El constructor recibirá sólo el contexto de la aplicación pero será necesario más datos para hacer el constructor de la superclase (`super(context, DB_NAME, null, DB_VERSION)`), que son:

Vamos a ver como quedaría esta clase: <https://bit.ly/3xoHuGc>



Vemos que hemos creado un String con la sentencia SQL a ejecutar para crear una tabla en nuestra base de datos:

```
private static final String  
COMMENTS_TABLE_CREATE = "CREATE  
TABLE comments(_id INTEGER PRIMARY  
KEY AUTOINCREMENT, user TEXT,  
comment TEXT)";
```

Además, hemos creado otro String con el nombre que tendrá nuestra base de datos (por convención, se usa la extensión *.sqlite* para las bases de datos *SQLite*, aunque no es obligatorio):

```
private static final String DB_NAME =  
"comments.sqlite";
```

Y un entero que indicará la versión de nuestra base de datos:

```
private static final int DB_VERSION = 1;
```

Si este valor se incrementa posteriormente de haber creado la base de datos, se llamará al método `onUpgrade()`.

En el método `onCreate()` simplemente ejecutaremos la sentencia que crea nuestra tabla:

```
db.execSQL(COMMENTS_TABLE_CREATE);
```



Acceso a la Base de Datos

Una vez tenemos nuestra clase derivada de SQLiteOpenHelper creada, **vamos a ver cómo accedemos a la base de datos desde nuestra aplicación.**

a.

Lo primero que tenemos que hacer es **crear un objeto de la clase *MySQLiteHelper*** al que le pasamos el contexto (en este caso la referencia de la Activity actual).

b.

A continuación, obtendremos una **referencia a la base de datos** en sí, y para ello podemos usar dos métodos, `getReadableDatabase()` o `getWritableDatabase()` si lo que queremos es sólo consultar datos o, a parte de esto, también modificarlos.

Nuestra Activity quedaría así: <https://bit.ly/2STD8aW>