

Miguel Ángel

Rodríguez Dalí

Práctica 5 Android



Creación de la base de datos SQLite

Para crear una base de datos con SQLite tenemos que:

- Crear una clase que extienda de SQLiteOpenHelper.
- Sobreescibir el método onCreate().
- Sobreescibir el método onUpgrade().
- Y crear el constructor de la clase.

Después crearemos una clase sencilla derivada de SQLiteOpenHelper que llamaremos MySQLiteHelper, donde sobreescibiremos onCreate() para crear una tabla que llamaremos comments compuesta por dos campos:

- Un int como id.
- Y un String.

También le haremos un constructor que reciba el contexto de la aplicación pero serán necesarios más datos para el constructor de la superclase:

- Contexto.
- Nombre de la BD.
- Objeto CursorFactory que normalice.
- La versión de la BD.

Esto traducido a código quedaría:

```
1 package com.academiaandroid.demodb;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7 public class MyOpenHelper extends SQLiteOpenHelper {
8     private static final String COMMENTS_TABLE_CREATE = "CREATE TABLE comments(_id INTEGER PRIMARY KEY AUTOINCREMENT, user TEXT, comment TEXT)";
9     private static final String DB_NAME = "comments.sqlite";
10    private static final int DB_VERSION = 1;
11    public MyOpenHelper(Context context) {
12        super(context, DB_NAME, null, DB_VERSION);
13    }
14    @Override
15    public void onCreate(SQLiteDatabase db) {
16        db.execSQL(COMMENTS_TABLE_CREATE);
17    }
18    @Override
19    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
20
21    }
22 }
23
```

Acceso a la BD

Una vez hecho lo anterior, vamos a ver cómo acceder a la BD desde la app:

- Primero creamos un objeto de la clase MySQLiteHelper al que le pasamos el contexto.
- Después obtenemos una referencia de la BD pudiendo usar los métodos

getReadableDatabase() o getWritableDatabase(); dependiendo de si queremos consultar o modificar datos.

Quedando el resultado así:

```
1 package com.academiaandroid.demodb;
2
3 import android.app.Activity;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.os.Bundle;
6
7 public class MainActivity extends Activity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13        MyOpenHelper dbHelper = new MyOpenHelper(this);
14        SQLiteDatabase db = dbHelper.getWritableDatabase();
15        if (db != null) {
16            // Hacer las operaciones que queramos sobre la base de datos
17        }
18    }
19 }
20 }
```

Inserción de datos

Ahora tras todo lo anterior vendría la inserción de datos:

- Usando el método `execSQL()` pasándole como parámetro un String con la sentencia deseada insertará los datos.
- Usando el método `insert()` pasando 3 parámetros:
 - Nombre de la tabla.
 - Nombre de la columna.
 - Objeto del tipo `ContentValues` que contiene los datos a insertar en clave-valor.

Usando el método `execSQL()` quedaría:

```
1 db.execSQL("INSERT INTO comments (user, comment) VALUES ('Digital Learning','Esto es un comentario insertado usando el método execSQL()');");
```

Y con `insert()`:

```
1 ContentValues cv = new ContentValues();
2 cv.put("user", "Academia Android");
3 cv.put("comment", "Esto es un comentario insertado usando el método insert()");
4 db.insert("comments", null, cv);
```

No siendo necesario darle valor al id ya que lo hemos declarado autoincremental. Añadiendo este código en el método `onCreate` de `MainActivity` quedaría:

```
1 package com.academiaandroid.demodb;
2
3 import android.app.Activity;
4 import android.content.ContentValues;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.os.Bundle;
7
8 public class MainActivity extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         MyOpenHelper dbHelper = new MyOpenHelper(this);
15         SQLiteDatabase db = dbHelper.getWritableDatabase();
16         if (db != null) {
17             // Insert con execSQL
18             db.execSQL("INSERT INTO comments (user, comment) VALUES ('Digital Learning','Esto es un comentario insertado usando el método execSQL()');");
19
20             // Insert con ContentValues
21             ContentValues cv = new ContentValues();
22             cv.put("user", "Academia Android");
23             cv.put("comment", "Esto es un comentario insertado usando el método insert()");
24             db.insert("comments", null, cv);
25         }
26     }
27
28 }
29 }
```

MediaPlayer

Ahora referida a la parte de cómo reproducir sonidos en Android, empezamos en cómo sería con MediaPlayer usando Kotlin en este caso:

- Creamos un proyecto en Android en blanco, llamado MainActivity; colocando en activity_main.xml 3 botones:



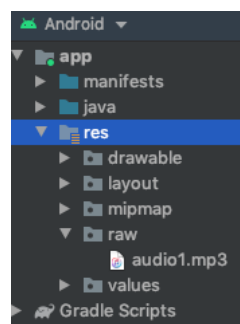
Luego su código XML sería:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3  xmlns:tools="http://schemas.android.com/tools"
4  android:layout_width="match_parent"
5  android:layout_height="match_parent"
6  android:orientation="horizontal"
7  tools:context=".MainActivity">
8
9  <Button
10     android:id="@+id/buttonPlay"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:layout_weight="0.3"
14     android:layout_gravity="center_vertical"
15     android:text="Play"
16     tools:layout_editor_absoluteX="63dp"
17     tools:layout_editor_absoluteY="272dp" />
18
19  <Button
20     android:id="@+id/buttonPause"
21     android:layout_width="wrap_content"
22     android:layout_height="wrap_content"
23     android:layout_weight="0.3"
24     android:layout_gravity="center_vertical"
25     android:text="Pause"
26     tools:layout_editor_absoluteX="176dp"
27     tools:layout_editor_absoluteY="275dp" />
28
29  <Button
30     android:id="@+id/buttonStop"
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content"
33     android:layout_weight="0.3"
34     android:layout_gravity="center_vertical"
35     android:text="Stop"
36     tools:layout_editor_absoluteX="286dp"
37     tools:layout_editor_absoluteY="275dp" />
38 </LinearLayout>

```

Añadimos un archivo de audio en el nuevo directorio creado en res llamado raw:



Vamos a ir a darle funcionalidad al main enlazando primero los botones con la vista y luego inicializando el objeto MediaPlayer, dándole lógica por último a los botones para cuando alguien haga click y terminamos:

```
1 class MainActivity : AppCompatActivity() {
2
3     private lateinit var buttonPlay: Button
4     private lateinit var buttonPause: Button
5     private lateinit var buttonStop: Button
6
7     private lateinit var mediaPlayer: MediaPlayer
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12
13         buttonPlay = findViewById(R.id.buttonPlay)
14         buttonPause = findViewById(R.id.buttonPause)
15         buttonStop = findViewById(R.id.buttonStop)
16
17         mediaPlayer = MediaPlayer.create(this, R.raw.audio1)
18
19         setOnClickListeners(this)
20     }
21
22     private fun setOnClickListeners(context: Context) {
23         buttonPlay.setOnClickListener {
24             mediaPlayer.start()
25             Toast.makeText(context, "Reproduciendo...", Toast.LENGTH_SHORT).show()
26         }
27
28         buttonPause.setOnClickListener {
29             mediaPlayer.pause()
30             Toast.makeText(context, "En pausa...", Toast.LENGTH_SHORT).show()
31         }
32
33         buttonStop.setOnClickListener {
34             mediaPlayer.stop()
35             mediaPlayer = MediaPlayer.create(context, R.raw.audio1)
36             Toast.makeText(context, "Parando...", Toast.LENGTH_SHORT).show()
37         }
38     }
39 }
```