En esta actividad,  creará una aplicación en la cual podrá implementar un simple CRUD sobre una base de datos SQLite.

**Desarrollo de la aplicación**

En el enlace encontrar un documento en formato PDF el cual contiene los pasos para realizar la actividad...

Luego de realizar la actividad deberá:

1- Contestar algunas preguntas sobre el desarrollo

2- Modificar la estética de la app (al menos una característica)

3- subir capturas o video de su desarrollo

Link al documento:

<https://acceso.ispc.edu.ar/pluginfile.php/124766/mod_lesson/page_contents/5229/Android%20Studio-Sqlite-ABM-Sencillo.pdf>

En esta actividad,  creará una aplicación en la cual podrá implementar un simple CRUD sobre una base de datos SQLite.

Verdadero. Android incorpora un conjunto de herramientas que facilitan la creación y gestión de bases de datos SQLite. Puedes utilizar la clase **SQLiteOpenHelper** para crear y gestionar la base de datos, y Android proporciona una API para ejecutar consultas SQL, insertar, actualizar y eliminar registros de la base de datos. Además, puedes usar la clase **Cursor** para recuperar datos de la base de datos SQLite de manera eficiente. En resumen, Android ofrece las herramientas necesarias para trabajar con bases de datos SQLite de manera integrada en el desarrollo de aplicaciones Android.

En Android, para crear, actualizar, y conectar con una base de datos SQLite será a través de una clase auxiliar llamada **SQLiteOpenHelper**, y deberemos:

Personalizar y adaptar a las necesidades concretas de nuestra aplicación.

No existe la clase SQLiteOpenHelper

Crear una clase propia que derive de ella.

Para crear, actualizar y conectar con una base de datos SQLite en Android, debes personalizar y adaptar la clase **SQLiteOpenHelper** a las necesidades concretas de tu aplicación. Por lo tanto, la opción correcta es:

* Personalizar y adaptar a las necesidades concretas de nuestra aplicación.

La opción "No existe la clase SQLiteOpenHelper" es incorrecta, ya que **SQLiteOpenHelper** es una clase fundamental en Android para gestionar bases de datos SQLite. También puedes crear una clase propia que derive de **SQLiteOpenHelper** para implementar la lógica específica de tu base de datos.

Todas las bases de datos SQLite creadas por aplicaciones Android utilizando este método (clase **SQLiteOpenHelper**) se almacenan en la memoria del teléfono en un fichero con el mismo nombre de la base de datos situado en una ruta que sigue el siguiente patrón:

/data/data/paquete.java.de.la.aplicacion/databases/nombre\_base\_datos



Falso



Verdadero

Verdadero. En Android, todas las bases de datos SQLite creadas por aplicaciones utilizando la clase **SQLiteOpenHelper** se almacenan en la memoria del teléfono en un fichero con el mismo nombre de la base de datos, y siguen la siguiente ruta:

**/data/data/paquete.java.de.la.aplicacion/databases/nombre\_base\_datos**

Esta es la ubicación predeterminada para las bases de datos SQLite de aplicaciones Android. La parte "paquete.java.de.la.aplicación" se refiere al paquete Java de tu aplicación, y "nombre\_base\_datos" es el nombre de la base de datos que le hayas dado.

Con respecto al método **execSQL**() , diremos:



No es un método para ejecutar código SQL que le pasemos como parámetro.



Es el método más sencillo de los disponibles en la API de SQLite proporcionada por Android.



Este método se limita a ejecutar directamente el código SQL que le pasemos como parámetro.

Con  
respecto al método **execSQL**() , diremos:

*Su respuesta* :

|  |
| --- |
| Es el método más sencillo de los disponibles en la API de SQLite proporcionada por Android. |
|  |
| Este método se limita a ejecutar directamente el código SQL que le pasemos como parámetro. |

El método **execSQL**() de la clase **SQLiteDatabase** permite ejecutar cualquier sentencia SQL sobre la base de datos, siempre que ésta no devuelva resultados.



Falso



Verdadero

Verdadero. El método execSQL() de la clase SQLiteDatabase permite ejecutar cualquier sentencia SQL sobre la base de datos, siempre que ésta no devuelva resultados. Es útil para sentencias SQL como INSERT, DELETE, UPDATE, entre otras. Sin embargo, no se debe usar para consultas que devuelvan datos, como SELECT.

**Enhorabuena, ha llegado al final de la lección**

Su puntuación es 5 (sobre 5).

Su calificación actual es 100,0 sobre 100

Ha alcanzado el 100% de esta lección

CODIGO DE UsuariosSQLiteHelper.java

import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteDatabase.CursorFactory;  
import android.database.sqlite.SQLiteOpenHelper;  
public class UsuariosSQLiteHelper extends SQLiteOpenHelper {  
 //Sentencia SQL para crear la tabla de Usuarios  
 String sqlCreate = "CREATE TABLE Usuarios (codigo INTEGER, nombre TEXT)";  
 public UsuariosSQLiteHelper(Context contexto, String nombre,  
 CursorFactory factory, int version) {  
 super(contexto, nombre, factory, version);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
//Se ejecuta la sentencia SQL de creación de la tabla  
 db.execSQL(sqlCreate);  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int versionAnterior,  
 int versionNueva) {  
//NOTA: Por simplicidad del ejemplo aquí utilizamos directamente  
// la opción de eliminar la tabla anterior y crearla de nuevo  
// vacía con el nuevo formato.  
// Sin embargo lo normal será que haya que migrar datos de la  
// tabla antigua a la nueva, por lo que este método debería  
// ser más elaborado.  
//Se elimina la versión anterior de la tabla  
 db.execSQL("DROP TABLE IF EXISTS Usuarios");  
//Se crea la nueva versión de la tabla  
 db.execSQL(sqlCreate);  
 }  
}

CODIGO ORIGINAL MainActivity.java

package com.example.bdsqlite\_crud\_simple;  
  
  
import android.os.Bundle;  
import android.app.Activity;  
import android.content.ContentValues;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.view.Menu;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class MainActivity extends Activity {  
  
 private EditText txtCodigo;  
 private EditText txtNombre;  
 private TextView txtResultado;  
  
 private Button btnInsertar;  
 private Button btnActualizar;  
 private Button btnEliminar;  
 private Button btnConsultar;  
  
 private SQLiteDatabase db;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 //Obtenemos las referencias a los controles  
 txtCodigo = (EditText)findViewById(R.id.*txtReg*);  
 txtNombre = (EditText)findViewById(R.id.*txtVal*);  
 txtResultado = (TextView)findViewById(R.id.*txtResultado*);  
  
 btnInsertar = (Button)findViewById(R.id.*btnInsertar*);  
 btnActualizar = (Button)findViewById(R.id.*btnActualizar*);  
 btnEliminar = (Button)findViewById(R.id.*btnEliminar*);  
 btnConsultar = (Button)findViewById(R.id.*btnConsultar*);  
  
 //Abrimos la base de datos 'DBUsuarios' en modo escritura  
 UsuariosSQLiteHelper usdbh =  
 new UsuariosSQLiteHelper(this, "DBUsuarios", null, 1);  
  
 db = usdbh.getWritableDatabase();  
  
 btnInsertar.setOnClickListener(new OnClickListener() {  
 public void onClick(View v) {  
  
 //Recuperamos los valores de los campos de texto  
 String cod = txtCodigo.getText().toString();  
 String nom = txtNombre.getText().toString();  
  
 //Alternativa 1: método sqlExec()  
 // String sql = "INSERT INTO Usuarios (codigo,nombre) VALUES ('" + cod + "','" + nom + "') ";  
 // db.execSQL(sql);  
  
 //Alternativa 2: método insert()  
 ContentValues nuevoRegistro = new ContentValues();  
 nuevoRegistro.put("codigo", cod);  
 nuevoRegistro.put("nombre", nom);  
 db.insert("Usuarios", null, nuevoRegistro);  
 }  
 });  
  
 btnActualizar.setOnClickListener(new OnClickListener() {  
 public void onClick(View v) {  
  
 //Recuperamos los valores de los campos de texto  
 String cod = txtCodigo.getText().toString();  
 String nom = txtNombre.getText().toString();  
  
 //Alternativa 1: método sqlExec()  
 //String sql = "UPDATE Usuarios SET nombre='" + nom + "' WHERE codigo=" + cod;  
 //db.execSQL(sql);  
  
 //Alternativa 2: método update()  
 ContentValues valores = new ContentValues();  
 valores.put("nombre", nom);  
 db.update("Usuarios", valores, "codigo=" + cod, null);  
 }  
 });  
  
 btnEliminar.setOnClickListener(new OnClickListener() {  
 public void onClick(View v) {  
  
 //Recuperamos los valores de los campos de texto  
 String cod = txtCodigo.getText().toString();  
  
 //Alternativa 1: método sqlExec()  
 //String sql = "DELETE FROM Usuarios WHERE codigo=" + cod;  
 //db.execSQL(sql);  
  
 //Alternativa 2: método delete()  
 db.delete("Usuarios", "codigo=" + cod, null);  
 }  
 });  
  
 btnConsultar.setOnClickListener(new OnClickListener() {  
 public void onClick(View v) {  
  
 //Alternativa 1: método rawQuery()  
 Cursor c = db.rawQuery("SELECT codigo, nombre FROM Usuarios", null);  
  
 //Alternativa 2: método delete()  
 //String[] campos = new String[] {"codigo", "nombre"};  
 //Cursor c = db.query("Usuarios", campos, null, null, null, null, null);  
  
 //Recorremos los resultados para mostrarlos en pantalla  
 txtResultado.setText("");  
 if (c.moveToFirst()) {  
 //Recorremos el cursor hasta que no haya más registros  
 do {  
 String cod = c.getString(0);  
 String nom = c.getString(1);  
  
 txtResultado.append(" " + cod + " - " + nom + "\n");  
 } while(c.moveToNext());  
 }  
 }  
 });  
 }  
  
  
  
}

Aquí pueden modificar la clase MainActivity con el siguiente código

import android.app.Activity;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

public class MainActivity extends Activity

{

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Abrimos la base de datos ‘DBUsuarios’ en modo escritura

UsuariosSQLiteHelper usdbh =

new UsuariosSQLiteHelper(this, "DBUsuarios", null, 1);

SQLiteDatabase db = usdbh.getWritableDatabase();

//Si hemos abierto correctamente la base de datos

if(db != null)

{

//Insertamos 5 usuarios de ejemplo

for(int i=1; i<=5; i++)

{

//Generamos los datos

int codigo = i;

String nombre = "Usuario" + i;

//Insertamos los datos en la tabla Usuarios

db.execSQL("INSERT INTO Usuarios (codigo, nombre) " +

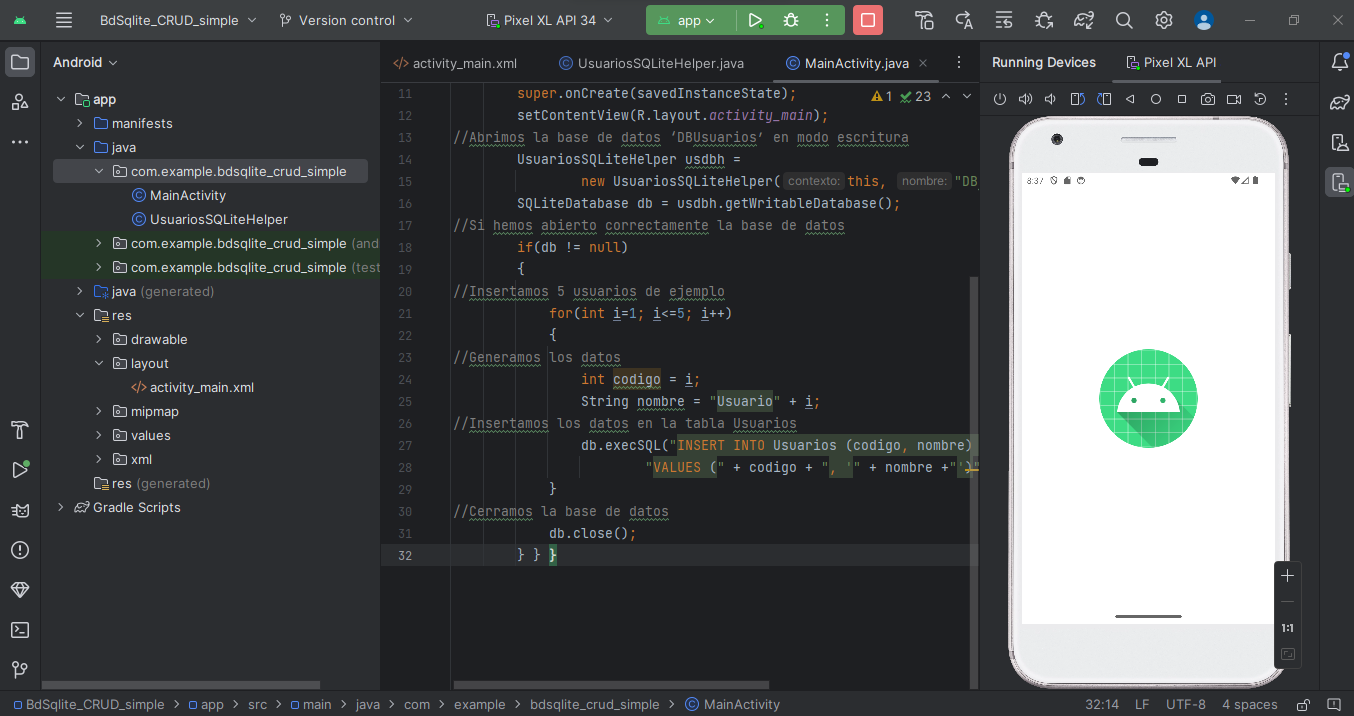
"VALUES (" + codigo + ", '" + nombre +"')");

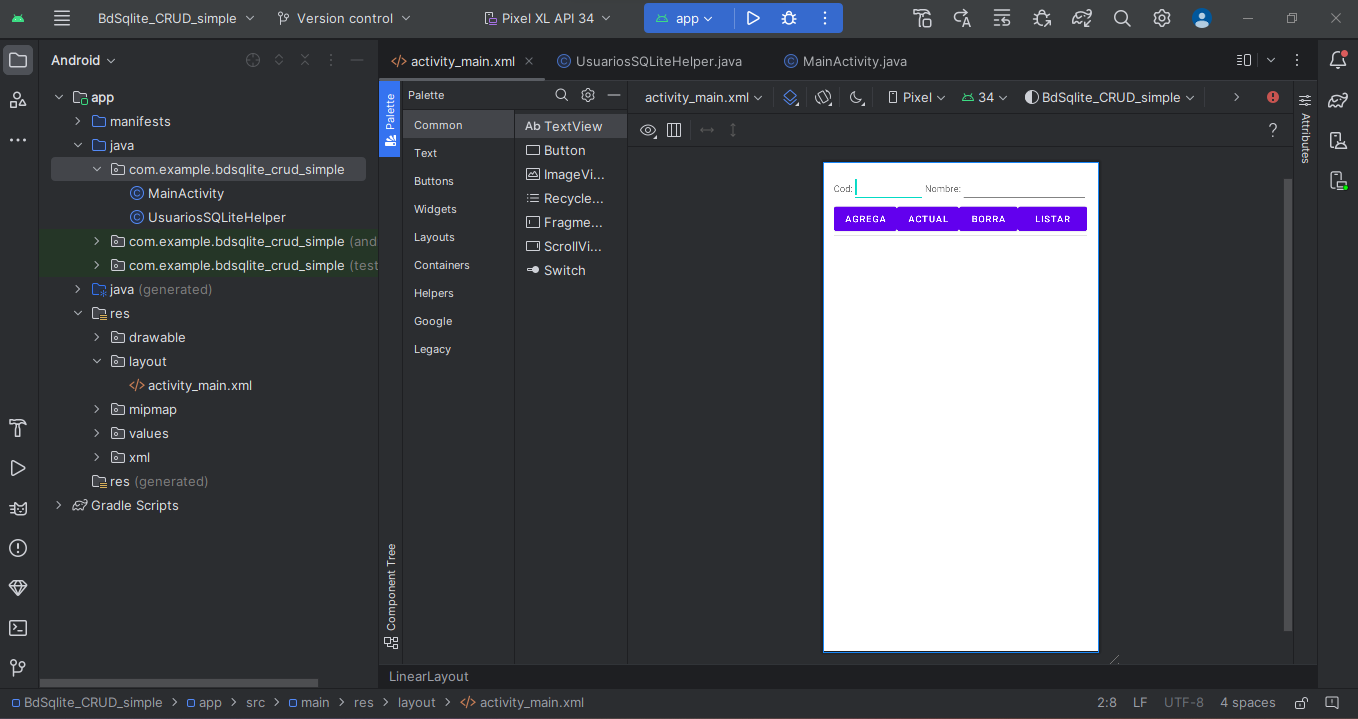
}

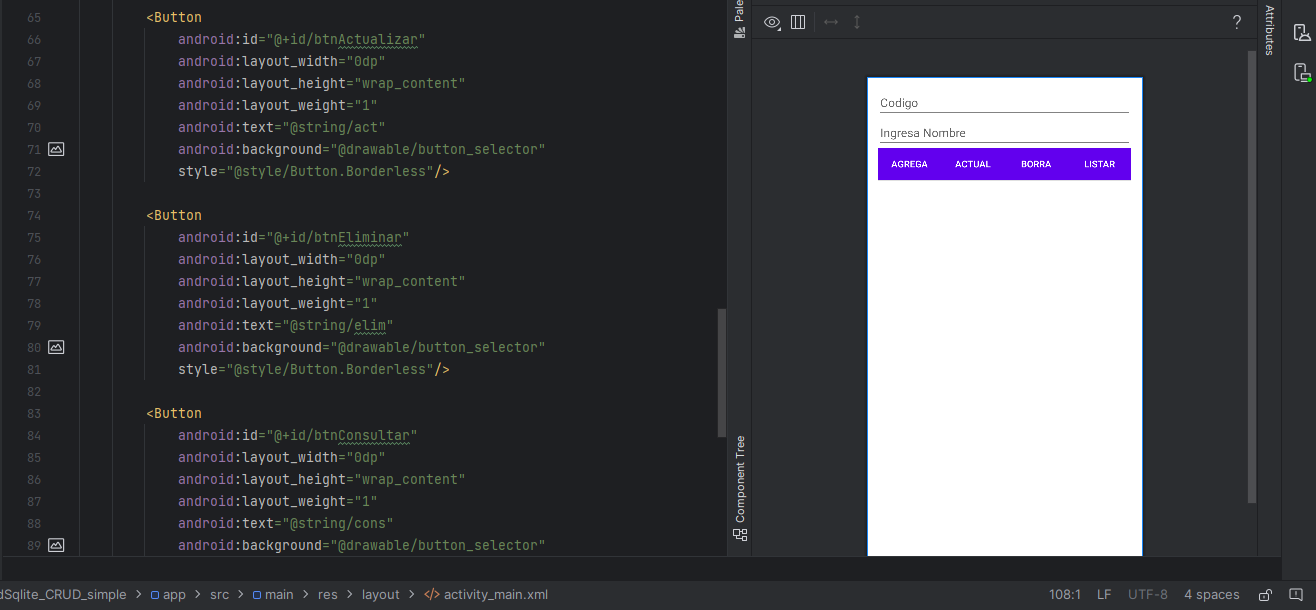
//Cerramos la base de datos

db.close();

} } }







<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/LinearLayout1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context=".MainActivity" >  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/nreg" />  
  
 <EditText  
 android:id="@+id/txtReg"  
 android:hint="Código"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="5"  
 android:ems="10"  
 android:inputType="number">  
  
 <requestFocus />  
 </>  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/val" />  
  
 <EditText  
 android:id="@+id/txtVal"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:hint="Ingresa tu nombre"  
 android:inputType="text" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" >  
  
 <Button  
 android:id="@+id/btnInsertar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/ins" />  
  
 <Button  
 android:id="@+id/btnActualizar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/act" />  
  
 <Button  
 android:id="@+id/btnEliminar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/elim" />  
  
 <Button  
 android:id="@+id/btnConsultar"  
 android:layout\_width="104dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/cons" />  
  
 </LinearLayout>  
  
 <View  
 android:id="@+id/divider"  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="?android:attr/listDivider" />  
  
 <TextView  
 android:id="@+id/txtResultado"  
 android:layout\_width="390dp"  
 android:layout\_height="462dp"  
 android:text="" />  
  
</LinearLayout>