

La Base de Datos SQLite

CONCEPTOS

SQLite es una base de datos SQL de código abierto, que almacena los datos a un archivo de texto en un dispositivo. Android viene interconstruido en la aplicación de base de datos SQLite.

SQLite es compatible con todas las características de bases de datos relacionales.

Para poder acceder a esta base de datos, no es necesario establecer ningún tipo de conexiones con ella. El paquete principal es android.database.sqlite que contiene las clases para gestionar bases de datos propias.

Se necesita crear un método para invocar a la base de datos openOrCreateDatabase, con el nombre de base de datos y el modo como parámetros, y que devuelve una instancia de la base de datos SQLite, que tiene que recibir el objeto.

Sus métodos principales:

```
openDatabase (String ruta, SQLiteDatabase.CursorFactory cf, int flags, DatabaseErrorHandler deh)

Abre la base de datos actual con el modo flag que puede ser OPEN_READWRITE OPEN_READONLY.

openDatabase (String ruta, SQLiteDatabase.CursorFactory cf, int flags)
```

Abre la base de datos actual pero no define un control de errores para la base de datos.

openOrCreateDatabase(String ruta, SQLiteDatabase.CursorFactory cf)

Abre pero crea la base de datos si ésta no existe. Es similar al método openDatabase. openOrCreateDatabase (File file, SQLiteDatabase.CursorFactory factory)

Toma el objeto File como una ruta en lugar de una cadena. Es similar a file.getPath().execSQL(String sql, Object[] bindArgs)

Sólo inserta datos, pero también se usa para actualizar o modificar datos ya existentes usando argumentos de enlace.

Otros métodos útiles son:

```
mydatabase.execSQL("CREATE TABLE IF NOT EXISTS TutorialsPoint(User VARCHAR, Pwd VARCHAR);");
mydatabase.execSQL("INSERT INTO TutorialsPoint VALUES('admin', 'admin');");
Cursor resultSet = mydatbase.rawQuery("Select * from MiTabla", null);
resultSet.moveToFirst();
String username = resultSet.getString(1);
String password = resultSet.getString(2);
```

Métodos

Descripción

```
getColumnIndex(String colName) Regresa el número total de columnas de la tabla.

getColumnIndex(String colName) Regresa el índice de la columna, especificando el nombre de la columna.

getColumnName(int columnIndex) Regresa el nombre de la columna, especificando el índice de la columna.

getColumnNames() Regresa un arreglo con todos los nombre de las columnas de la tablas.

getCount() Regresa el número total de filas en el cursor.

getPosition() Regresa la posición actual del cursor en la tabla.

isClosed() Regresa true si el cursor está cerrado y regresa false en cualquier otro estado.
```

Ejemplo del uso de la clase SQLiteOpenHelper:

```
public class DBHelper extends SQLiteOpenHelper {
   public DBHelper() {
      super(context, DATABASE_NAME, null, 1);
   }
   public void onCreate(SQLiteDatabase db) {}
   public void onUpgrade(SQLiteDatabase database, int oldVersion, int newVersion) {}
}
```

Pasos para ver Archivo de Base de Datos en Android Studio:

1. Digitar el icono de Android Device Monitor ADM, en el menú principal.

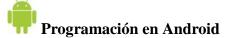
- 2. Digitar el icono de DDMS.
- 3. Clic en: data -> data -> com.example.escom.basededatos1
- 4. Clic en databases: Seleccionar DBContactos
- 5. Digitar el icono superior derecho: Pull a file from the device.

DESARROLLO

EJEMPLO 1.

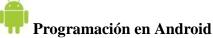
Paso 1. Crear un proyecto DbmsSQLite1. En la carpeta java/com.example.mipaquete, abrir y modificar el archivo MainActivity.java con el siguiente código:

```
import android.os.Bundle;
import android.app.Activity;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;
public class MainActivity extends Activity {
   Edit.Text.
                    jetI, jetN;
   Button
                    jbnA, jbnL;
    TextView
                    jtvL;
    SQLiteDatabase sqld;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity main);
        jetI = (EditText) findViewById(R.id.xetI);
        jetN = (EditText) findViewById(R.id.xetN);
        jbnA = (Button) findViewById(R.id.xbnA);
        jbnL = (Button)
                           findViewById(R.id.xbnL);
        jtvL = (TextView) findViewById(R.id.xtvL);
        DbmsSQLiteHelper dsqlh = new DbmsSQLiteHelper(this, "DBContactos", null, 1);
        sqld = dsqlh.getWritableDatabase();
        jbnA.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                                id = jetI.getText().toString();
                String
                String
                                nombre = jetN.getText().toString();
                ContentValues
                                cv = new ContentValues();
                cv.put("id", id);
                cv.put("nombre", nombre);
                sqld.insert("Contactos", null, cv);
                jetI.setText(""); jetN.setText("");
            }
        });
        jbnL.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                String id, nombre;
                Cursor c = sqld.rawQuery("SELECT id,nombre FROM Contactos", null);
                jtvL.setText("");
                if (c.moveToFirst()) {
                    do {
                        id = c.qetString(0);
                        nombre = c.getString(1);
                        jtvL.append(" " + id + "\t" + nombre + "\n");
                    } while (c.moveToNext());
                }
        });
    }
```



Paso 2. En la carpeta java/com.examples.mipaquete, crear el archivo DbmsSQLiteHelper.java y agregar el siguiente
código:
import android.content.Context;
import android.database.sqlite.*;
import android.database.sqlite.sQLiteDatabase.CursorFactory;
public class DbmsSQLiteHelper extends SQLiteOpenHelper {
 String sqlCreate = "CREATE TABLE Contactos (id INTEGER, nombre TEXT)";
 public DbmsSQLiteHelper(Context c, String s, CursorFactory cf, int v) {
 super(c, s, cf, v);
 }
 @Override
 public void onCreate(SQLiteDatabase db) {
 db.execSQL(sqlCreate);
 }
 @Override
 public void onUpgrade(SQLiteDatabase sqld, int ov, int nv) {
 sqld.execSQL("DROP TABLE IF EXISTS Contactos");
 sqld.execSQL(sqlCreate);
 }
}

```
Paso 3. En la carpeta res/values, abrir el archivo activity_main.xml para modificarlo con el siguiente código:
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout</pre>
    xmlns: android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/xll1"
    android:layout width="match parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/xtvI"
        android:layout width="wrap content"
        android: layout height="wrap content"
        android:text="@string/sid" />
    <EditText
        android:id="@+id/xetI"
        android:layout width="wrap content"
        android:layout_height="wrap_content"
        android:inputType="number" />
    <TextView
        android:id="@+id/xtvN"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:text="@string/snom" />
    <EditText
        android:id="@+id/xetN"
        android:layout width="match parent"
        android: layout height="wrap content"
        android:inputType="text"/>
    <Button
        android:id="@+id/xbnA"
        android:layout width="match parent"
        android:layout_height="wrap_content"
        android:text="@string/ins" />
    <Button
        android:id="@+id/xbnL"
        android:layout width="match parent"
        android: layout height="wrap content"
        android:text="@string/cons" />
```



```
<TextView
    android:id="@+id/xtvL"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="" />
</LinearLayout>
```

Paso 5. Por último, ejecutar la aplicación. Al inicio, se muestra solamente el contenido vacío del ID y del Nombre. Agregar un ID y un nombre y enseguida digitar el botón de Alta. Después se pueden agregar más datos para posteriormente, o en cualquier momento, mostrar la lista de elementos digitando el botón Lista, como se muestra en las siguientes imágenes:







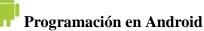




Ejercicio 1.

Diseñar una aplicación móvil que permita las acciones de Altas, Bajas, Cambios y Consultas en una base de datos SQLite. Utilizar los siguientes programas DbmsActivity2.java, activity_dbms2.xml y strings.xml.

```
// Bases de datos con SQLite
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;
public class DbmsActivity2 extends Activity implements OnClickListener{
    EditText
                    jetI, jetN, jetC;
                    jbnA, jbnB, jbnC, jbnV, jbnL, jbnI;
    Button
    SOLiteDatabase db;
    @Override
    public void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity dbms2);
        jetI=(EditText)findViewById(R.id.xetI);
        jetN=(EditText)findViewById(R.id.xetN);
        jetC=(EditText)findViewById(R.id.xetC);
        jbnA=(Button)findViewById(R.id.xbnA);
                                                 jbnA.setOnClickListener(this);
        jbnB=(Button)findViewById(R.id.xbnB);
                                                 jbnB.setOnClickListener(this);
        jbnC=(Button)findViewById(R.id.xbnC);
                                                 jbnC.setOnClickListener(this);
        jbnV=(Button)findViewById(R.id.xbnV);
                                                 jbnV.setOnClickListener(this);
        jbnL=(Button)findViewById(R.id.xbnL);
                                                 jbnL.setOnClickListener(this);
                                                 jbnI.setOnClickListener(this);
        jbnI=(Button) findViewById(R.id.xbnI);
        db=openOrCreateDatabase("DBEstudiantes", Context.MODE PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS estudiantes (ID VARCHAR, nombre VARCHAR,
calificacion VARCHAR);");
```



```
public void onClick(View v) {
        if(v==jbnA){
            if( jetI.getText().toString().trim().length()==0 ||
jetN.getText().toString().trim().length()==0 ||
jetC.getText().toString().trim().length()==0) {
                mensaje("Error", "Ingresar todos los datos");
                return;
            db.execSQL("INSERT INTO estudiantes VALUES('" + jetI.getText() + "','" +
jetN.getText() + "','" + jetC.getText() + "');");
            mensaje("Alta", "Registro agregado");
            limpiar();
        if(v==jbnB){
            if(jetI.getText().toString().trim().length()==0){
                mensaje("Error", "Ingresar el ID");
                return;
            Cursor c=db.rawQuery("SELECT * FROM estudiantes WHERE ID="" + jetI.getText() +
"'", null);
            if(c.moveToFirst()) {
                db.execSQL("DELETE FROM estudiantes WHERE ID='" + jetI.getText() + "'");
                mensaje("Baja", "Registro eliminado");
                mensaje("Error", "ID inválido");
            limpiar();
        if(v==jbnC){
            if(jetI.getText().toString().trim().length()==0) {
                mensaje("Error", "Ingresar el ID");
                return;
            Cursor c=db.rawQuery("SELECT * FROM estudiantes WHERE ID='" + jetI.getText() +
"'", null);
            if(c.moveToFirst()) {
                db.execSQL("UPDATE estudiantes SET nombre='" + jetN.getText() + "',
calificacion='" + jetC.getText() + "' WHERE ID='" + jetI.getText() + "'");
                mensaje("Cambio", "Registro modificado");
            } else {
                mensaje("Error", "ID inválido");
            limpiar();
        if(v==jbnV) {
            if(jetI.getText().toString().trim().length()==0) {
                mensaje("Error", "Ingresar el ID");
                return;
            Cursor c=db.rawQuery("SELECT * FROM estudiantes WHERE ID='" + jetI.getText() +
"'", null);
            if(c.moveToFirst()) {
                jetN.setText(c.getString(1));
                jetC.setText(c.getString(2));
            else {
```



```
mensaje ("Error", "ID inválido");
                limpiar();
            }
        if(v==jbnL) {
            Cursor c=db.rawQuery("SELECT * FROM estudiantes", null);
            if(c.getCount()==0){
                mensaje("Error", "No hay registros");
            StringBuffer buffer=new StringBuffer();
            while(c.moveToNext())
                buffer.append("ID: " + c.getString(0)+"\n");
                buffer.append("Nombre: " + c.getString(1)+"\n");
                buffer.append("Calificacion: " + c.getString(2)+"\n\n");
            mensaje("Lista", buffer.toString());
        if(v==jbnI) {
            mensaje("Control de Calificaciones", "ESCOM");
    public void mensaje(String s, String m)
        Builder b=new Builder(this);
        b.setCancelable(true);
        b.setTitle(s);
        b.setMessage(m);
        b.show();
    public void limpiar()
        jetI.setText("");
        jetN.setText("");
        jetC.setText("");
        jetI.requestFocus();
Archivo activity_dbms2.xml
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    android:id="@+id/myLayout"
    android:layout height="fill parent"
    android:layout width="fill parent"
    android:stretchColumns="0"
    xmlns:android="http://schemas.android.com/apk/res/android">
                android:layout height="wrap content" android:layout width="wrap content"
    <TextView
                android:layout y="10dp" android:layout x="110dp"
                android:text="@string/titulo"/>
    <TextView
                android:layout height="wrap content" android:layout width="wrap content"
                android:layout y="50dp" android:layout x="30dp" android:text="@string/id"/>
    <EditText
                android:layout height="40dp" android:layout width="150dp"
android:id="@+id/xetI"
                android:layout y="50dp" android:layout x="150dp"
android:inputType="number"/>
                android:layout height="wrap content" android:layout width="wrap content"
                android:layout y="100dp" android:layout x="30dp"
android:text="@string/nombre"/>
```



```
android:layout height="40dp" android:layout width="150dp"
    <EditText
android:id="@+id/xetN"
                android:layout y="100dp" android:layout x="150dp" android:inputType="text"/>
    <TextView
                android:layout height="wrap content" android:layout width="wrap content"
                android:layout y="150dp" android:layout x="30dp"
android:text="@string/calificacion"/>
              android:layout height="40dp" android:layout width="150dp"
    <EditText
android:id="@+id/xetC"
                android:layout y="150dp" android:layout x="150dp"
android:inputType="number"/>
    <Button android:layout height="40dp" android:layout width="100dp" android:id="@+id/xbnA"
            android:layout y="200dp" android:layout x="30dp" android:text="@string/alta"/>
    <Button android:layout height="40dp" android:layout width="100dp" android:id="@+id/xbnB"
            android:layout y="200dp" android:layout x="150dp" android:text="@string/baja"/>
    <Button android:layout height="40dp" android:layout width="100dp" android:id="@+id/xbnC"
            android:layout y="250dp" android:layout x="30dp" android:text="@string/cambio"/>
    <Button android:layout height="40dp" android:layout width="100dp" android:id="@+id/xbnV"</pre>
            android:layout y="250dp" android:layout x="150dp" android:text="@string/ver"/>
    <Button android:layout height="40dp" android:layout width="100dp" android:id="@+id/xbnL"</pre>
            android:layout y="300dp" android:layout x="30dp" android:text="@string/lista"/>
    <Button android:layout height="40dp" android:layout width="100dp" android:id="@+id/xbnI"
            android:layout y="300dp" android:layout x="150dp" android:text="@string/info"/>
</AbsoluteLayout>
```

Archivo strings.xml

NOTA. Generar un reporte, como se indica en la rúbrica y enviarlo al sitio indicado por el profesor.