Reporte de Practica en SQLite

Wendy Caribel Nuño Perez

Sistemas Operativos Linux
Ciateq, Campus Guadalajara
w3ndy.carib3l@gmail.com

22 de septiembre de 2017

0.1. Introducción

0.1.1. Descripción de la Practica

Se trata básicamente de una aplicación que hace altas, bajas y modificación de registros, además de su consulta, utilizando SQLite.

SQLite es una herramienta de software libre, escrita en lenguaje c, que permite almacenar información en dispositivos embebidos de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware y/o recursos , como puede ser un teléfono celular ya que es un sistema de gestión de bases de datos que no requiere configuración ni un servidor como tal.

SQLite implementa el estándar SQL92 (tercera revisión del lenguaje SQL) lo que permite el uso transparente en aplicaciones móviles o de escritorio.

Características

Estas son algunas de las características principales de SQLite:

• La base de datos completa se encuentra en un solo archivo.

• Puede funcionar enteramente en memoria, lo que la hace muy rápida.

• Tiene un footprint menor a 230KB.

• Es totalmente auto contenida (sin dependencias externas).

• Cuenta con librerías de acceso para muchos lenguajes de programación.

• Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.

• Soporta funciones SQL definidas por el usuario (UDF).

• El código fuente es de dominio público y se encuentra muy bien documentado.

La aplicación contiene 2 clases en java que se explican a continuación:

AdminSqLiteOpenHelper

Es la clase que se encarga de crear y administrar las versiones de la base de datos y esta a su vez contiene dos subclases onCreate(SQLiteDatabase) la cual se llama para crear la base de datos por primera vez y onUpgrade(SQLiteDatabase, int, int) la cual permite realizar actualizaciones sobre la base de datos cuando se le llama.

MainActivity

Es la clase principal en la cual están definidos los métodos de alta, consulta, baja y modificación, mismos que son mandados llamar con los botones de alta, baja, modificación y consulta. Y a su vez contiene el método onCreate de la actividad y almacena en 4 variables privadas de la clase (et1, et2, et3, et4) los valores que se registren en las cajas de texto.

Para utilizar SQLite se utilizan las siguientes librerías:

android.database.sqlite.SQLiteDatabase: Permite utilizar los métodos de gestión de bases de datos.

android.database.sqlite.SQLiteDatabase.CursorFactory :Se utiliza para devolver la subclase del Cursor.

android.database.sqlite.SQLiteOpenHelper: Se utiliza para mandar llamar la clase indicar la clase auxiliar AdminSqLOpenHelper la cual crea la base de datos requerida o la actualiza a versiones.

0.2. Codigo activitymain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
    res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    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="com.example.wendyriega.database.MainActivity"
        >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Identificador"
```

```
        android:id="@+id/textView"
        android:layout_alignParentStart="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText1"
        android:layout_below="@+id/textView"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nombre y Apellidos"
        android:id="@+id/textView3"
        android:layout_below="@+id/editText1"
        android:layout_alignParentStart="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText2"
        android:layout_below="@+id/textView3"
        android:layout_alignParentStart="true"
        android:layout_alignEnd="@+id/editText1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ciudad"
        android:id="@+id/textView4"
        android:layout_below="@+id/editText2"
        android:layout_alignParentStart="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText3"
        android:layout_below="@+id/textView4"
        android:layout_alignEnd="@+id/editText2"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Numero de Contacto"
        android:id="@+id/textView5"
        android:layout_below="@+id/editText3"
```

```
        android:layout_alignParentStart="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText4"
        android:layout_below="@+id/textView5"
        android:layout_alignParentStart="true"
        android:layout_alignEnd="@+id/editText3"
        android:layout_alignParentEnd="false" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alta"
        android:id="@+id/button"
        android:layout_below="@+id/editText4"
        android:layout_alignParentStart="true"
        android:onClick="alta" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Consulta"
        android:id="@+id/button2"
        android:onClick="consulta"
        android:layout_below="@+id/editText4"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Baja"
        android:id="@+id/button3"
        android:onClick="baja"
        android:layout_below="@+id/editText4"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Modificar datos"
        android:id="@+id/button4"
        android:onClick="modificacion"
        android:layout_below="@+id/button"
        android:layout_toEndOf="@+id/button"
        android:layout_marginTop="55dp" />
```

```
</RelativeLayout>
```

0.3. Código Androidmanifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/
    android"
    package="com.example.wendyriega.database">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN
                    " />

                <category android:name="android.intent.category.
                    LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

0.4. Códigos de Java

Clase MainActivity.java

```
/**
 * Created by Wendyriega on 21/09/17.
 */
package com.example.wendyriega.database;

import android.widget.Toast;
import android.app.Activity;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;


class MainActivity extends Activity {

    private EditText et1, et2, et3, et4;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        // proviene del layout, son los campos de texto
        et1 = (EditText) findViewById(R.id.editText1); et2 = (
            EditText) findViewById(R.id.editText2);
        et3 = (EditText) findViewById(R.id.editText3); et4 = (
            EditText) findViewById(R.id.editText4);

    }

    // Damos de alta los usuarios en nuestra aplicacion
    public void alta(View v) {

        AdminSqLiteOpenHelper admin = new AdminSqLiteOpenHelper(
            this,

                "administracion", null, 1);

        SQLiteDatabase bd = admin.getWritableDatabase();

        String dni = et1.getText().toString();
        String nombre = et2.getText().toString();
        String ciudad = et3.getText().toString();
        String numero = et4.getText().toString();

        ContentValues registro = new ContentValues();

        registro.put("dni", dni);
        registro.put("nombre", nombre);
        registro.put("ciudad", ciudad);
        registro.put("numero", numero);

        // los inserto en la base de datos
        bd.insert("usuario", null, registro);
```

```
        bd.close();

        // ponemos los campos a vacio para insertar el siguiente
            usuario
        et1.setText(""); et2.setText(""); et3.setText(""); et4.
            setText("");

        Toast.makeText(this, "Datos del usuario guardados",
            Toast.LENGTH_SHORT).show();

    }

    // Hacemos busqueda de usuario por DNI
    public void consulta(View v) {

        AdminSqLiteOpenHelper admin = new AdminSqLiteOpenHelper(
            this,

                "administracion", null, 1);

        SQLiteDatabase bd = admin.getWritableDatabase();

        String dni = et1.getText().toString();

        Cursor fila = bd.rawQuery(

                "select nombre, ciudad, numero from usuario
                    where dni=" + dni, null);

        if (fila.moveToFirst()) {

            et2.setText(fila.getString(0));
            et3.setText(fila.getString(1));
            et4.setText(fila.getString(2));

        } else


            Toast.makeText(this, "No existe ningun usuario con
                ese dni",

                    Toast.LENGTH_SHORT).show();

        bd.close();

    }
```

```
/* Metodo para dar de baja al usuario insertado*/
public void baja(View v) {

    AdminSqLiteOpenHelper admin = new AdminSqLiteOpenHelper(
        this,

            "administracion", null, 1);

    SQLiteDatabase bd = admin.getWritableDatabase();

    String dni = et1.getText().toString();

    // aqui borro la base de datos del usuario por el dni
    int cant = bd.delete("usuario", "dni=" + dni, null);

    bd.close();

    et1.setText(""); et2.setText(""); et3.setText(""); et4.
        setText("");

    if (cant == 1)

        Toast.makeText(this, "Usuario eliminado",

                Toast.LENGTH_SHORT).show();

    else

        Toast.makeText(this, "No existe ese usuario",

                Toast.LENGTH_SHORT).show();
}


// Metodo para modificar la informacion del usuario
public void modificacion(View v) {

    AdminSqLiteOpenHelper admin = new AdminSqLiteOpenHelper(
        this,

            "administracion", null, 1);

    SQLiteDatabase bd = admin.getWritableDatabase();

    String dni = et1.getText().toString();
    String nombre = et2.getText().toString();
    String ciudad = et3.getText().toString();
    String numero = et4.getText().toString();
```

```
        ContentValues registro = new ContentValues();

        // actualizamos con los nuevos datos, la informacion
            cambiada
        registro.put("nombre", nombre);
        registro.put("ciudad", ciudad);
        registro.put("numero", numero);

        int cant = bd.update("usuario", registro, "dni=" + dni,
            null);

        bd.close();

        if (cant == 1)

            Toast.makeText(this, "Datos modificados con exito",
                Toast.LENGTH_SHORT)

                    .show();

        else

            Toast.makeText(this, "No existe usuario",

                    Toast.LENGTH_SHORT).show();

    }

/* fin del programa */

}
```

Clase AdminSqLiteOpenHelper.java

```
/**
 * Created by Wendyriega on 21/09/17.
 */
package com.example.wendyriega.database;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

class AdminSqLiteOpenHelper extends SQLiteOpenHelper {

    public AdminSqLiteOpenHelper(Context context, String nombre,
        SQLiteDatabase.CursorFactory factory, int version) {
```

```
        super(context, nombre, factory, version);

    }

    @Override

    public void onCreate(SQLiteDatabase db) {

        //aqui creamos la tabla de usuario (dni, nombre, ciudad,
            numero)
        db.execSQL("create table usuario(dni integer primary key
            , nombre text, ciudad text, numero integer)");

    }

    @Override

    public void onUpgrade(SQLiteDatabase db, int version1, int
        version2) {

        db.execSQL("drop table if exists usuario");

        db.execSQL("create table usuario(dni integer primary key
            , nombre text, ciudad text, numero integer)");

    }
}
```

0.5. Código de Recursos

Codigo en colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

Codigo en dimen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="activity_horizontal_margin">16dp</dimen>
</resources>
```

Codigo en refs.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources></resources>
```

Codigo en strings.xml

```
<resources>
    <string name="app_name">DataBase</string>
</resources>
```

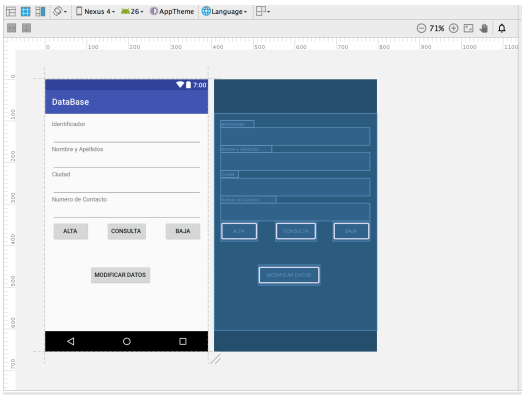
Codigo en styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.
        DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</
            item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

0.6. ActivityMain.xml en Modo Diseño



Bibliografía

[1] ANDROID.DATABASE.SQLITE, *DeveloperAndroid.com* https://developer.android.com/reference/android/database/sqlite/package-summary.html

ANDREA, ARDIONS, *Android Studio FAQS* https://androidstudiofaqs.com/tutoriales/usar-sqlite-en-android-studio

SALVADOR GOMEZ OLIVER, *SGOLIVER.NET* http://www.sgoliver.net/blog/bases-de-datos-en-android-i-primeros-pasos/