**Instituto Politécnico Nacional**

**Escuela Superior de Computo**

Nombre del Alumno:

Emiliano González López

Asignatura: Application Development For Mobile Devices

Profesor: Cifuentes Álvarez Alejandro Sigfrido

SQLite

Fecha de entrega: 4 de junio de 2020

Grupo: 3CV2

**Índice**

**1.Objetivo……………………………………………………….3**

**2.Conceptos previos………………………………………….3**

**3.Desarrollo…………………………………………………….4**

**4.Conclusiones……………………………………..………....11**

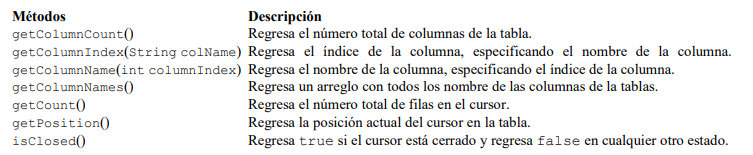
**5.Bibliografia…………………………………………….….....12**

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

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



Desarrollo

Ejercicio 1

El primer ejercicio consta de crear una sencilla aplicación donde se dan de alta 2 diferentes elementos en cada tupla, siendo uno el ID y otro el nombre.

Esta aplicación fue realmente sencilla de implementar, y tiene un gran parecido con lo que hacemos normalmente en java pero de una manera más sencilla.

Código de implementación

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 {

    EditText 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) {

                String id = jetI.getText().toString();

                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.getString(0);

                        nombre = c.getString(1);

                        jtvL.append(" " + id + "\t" + nombre + "\n");

                    } while(c.moveToNext());

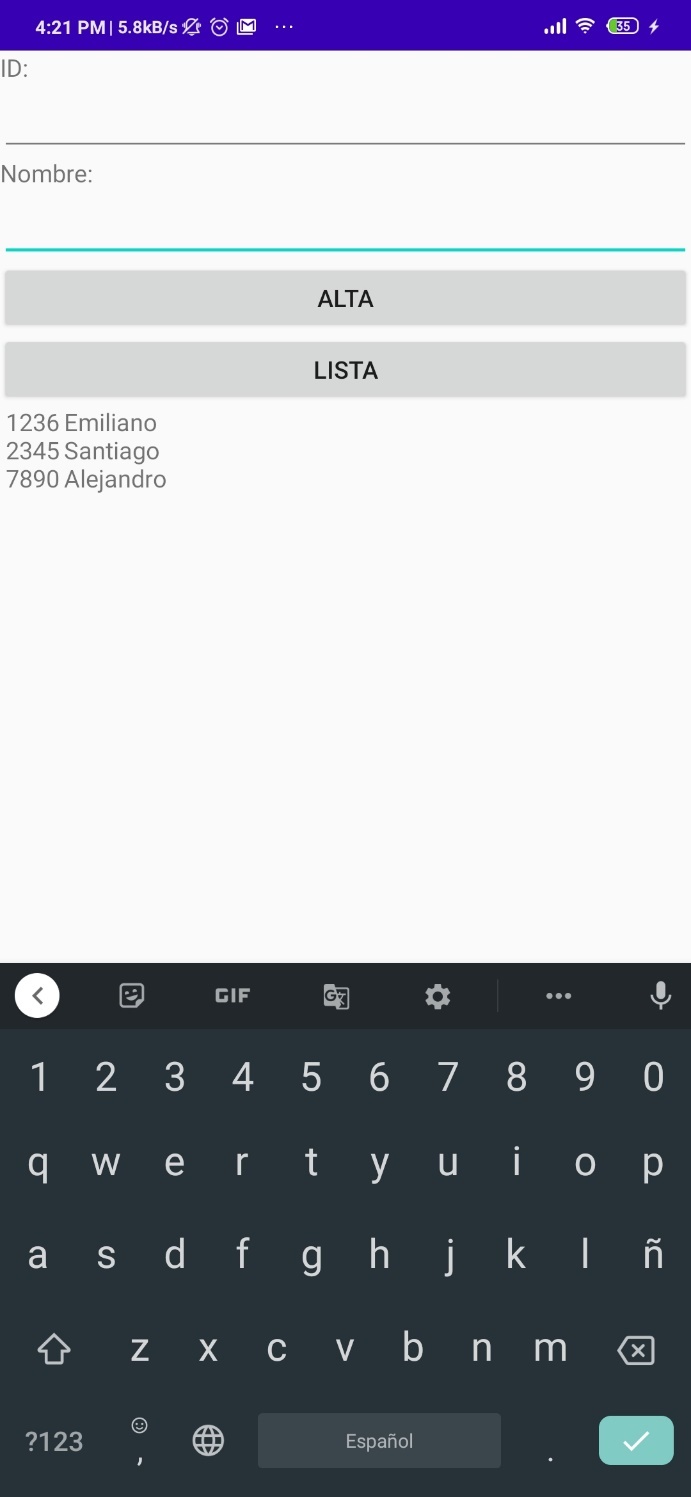
                }

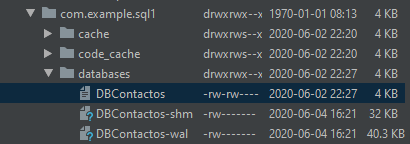
            }

        });

    }

}





Ejercicio 2

Este ejercicio igualmente fue sencillo, solo aumentaba la complejidad de es un CRUD completo, es decir, Create, Read, Update, Delete.

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 MainActivity extends Activity implements OnClickListener{

    EditText        jetI, jetN, jetC;

    Button          jbnA, jbnB, jbnC, jbnV, jbnL, jbnI;

    SQLiteDatabase  db;

    @Override

    public void onCreate(Bundle b){

        super.onCreate(b);

        setContentView(R.layout.activity\_main);

        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=(Button)findViewById(R.id.xbnI);   jbnI.setOnClickListener(this);

        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");

            } else {

                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");

                return;

            }

            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();

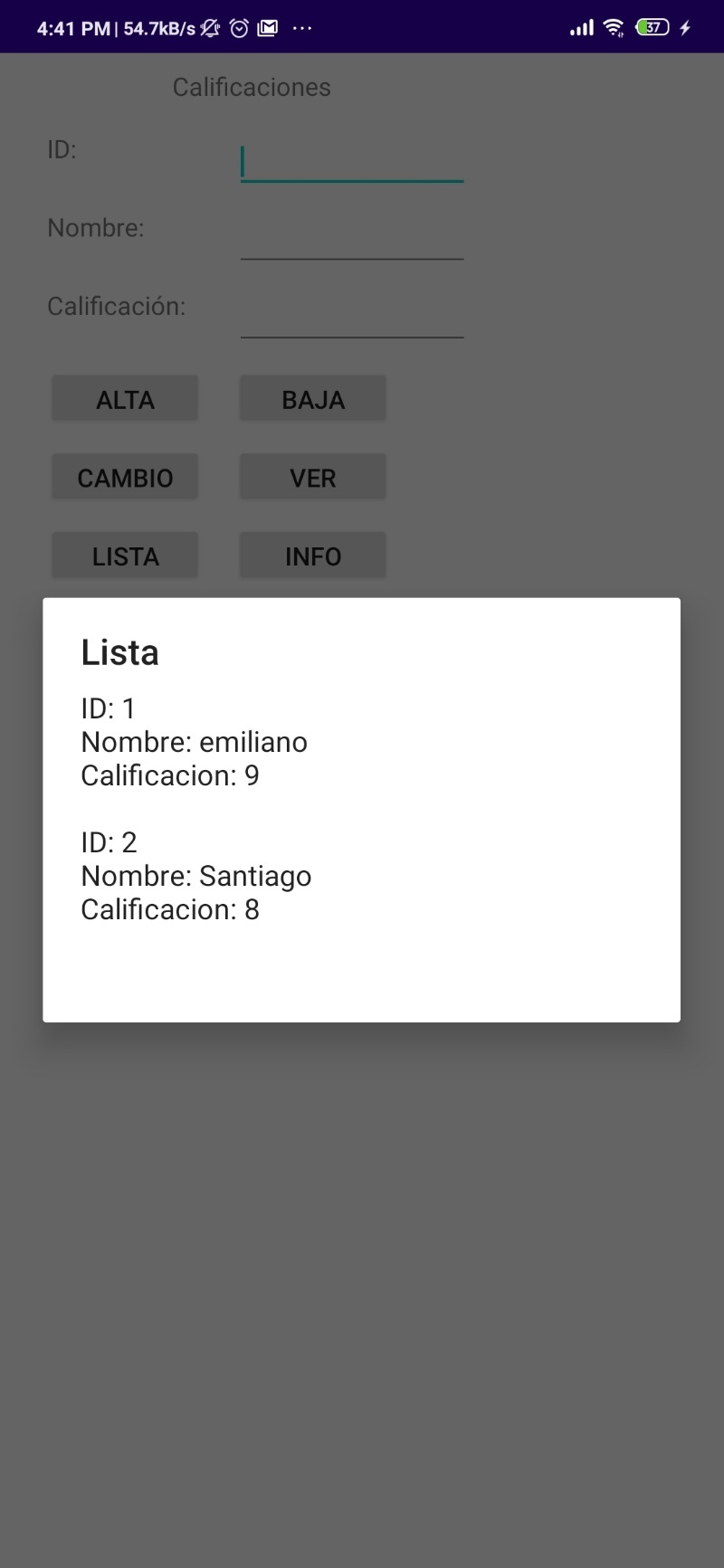
    }

}

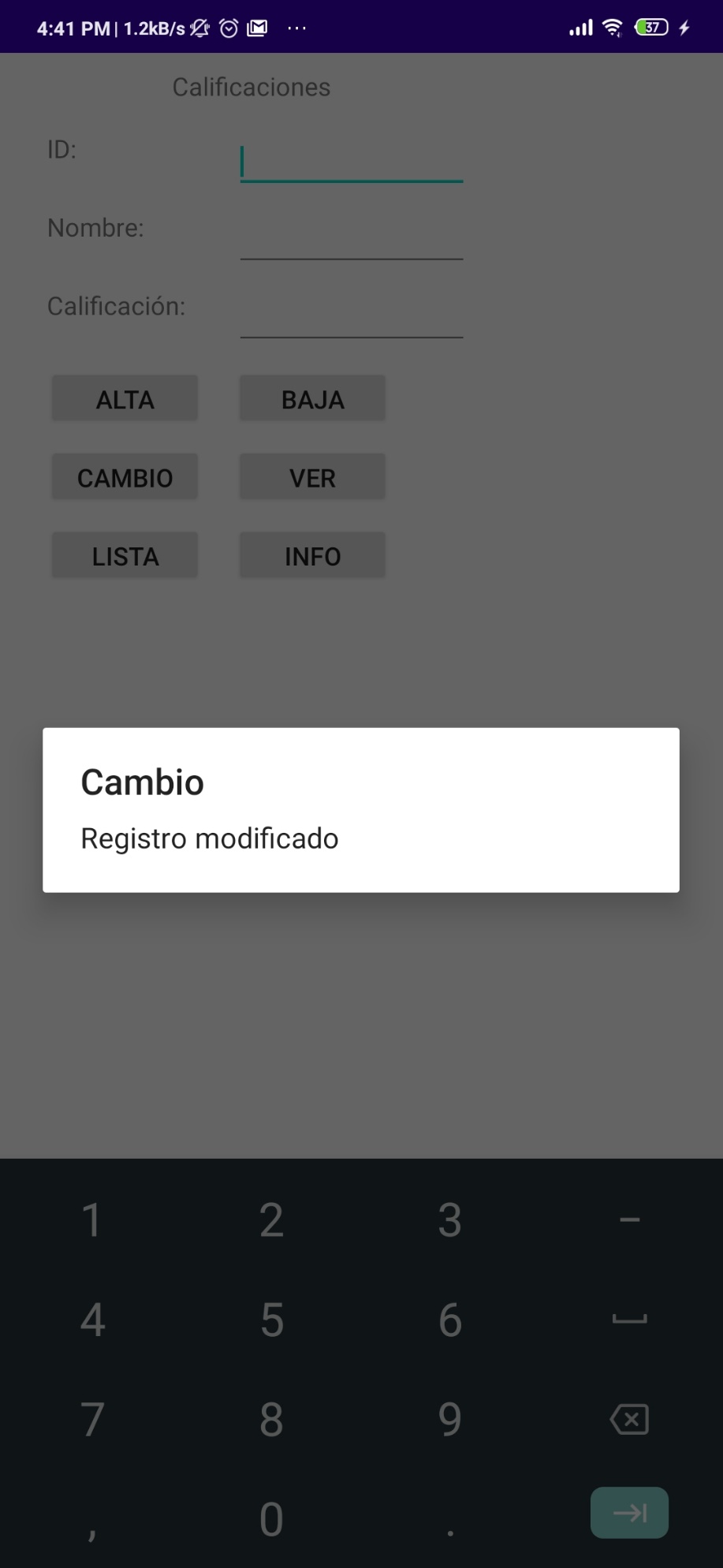
Ejemplo de la aplicación corriendo.



Dos registros agregados, y se muestran.



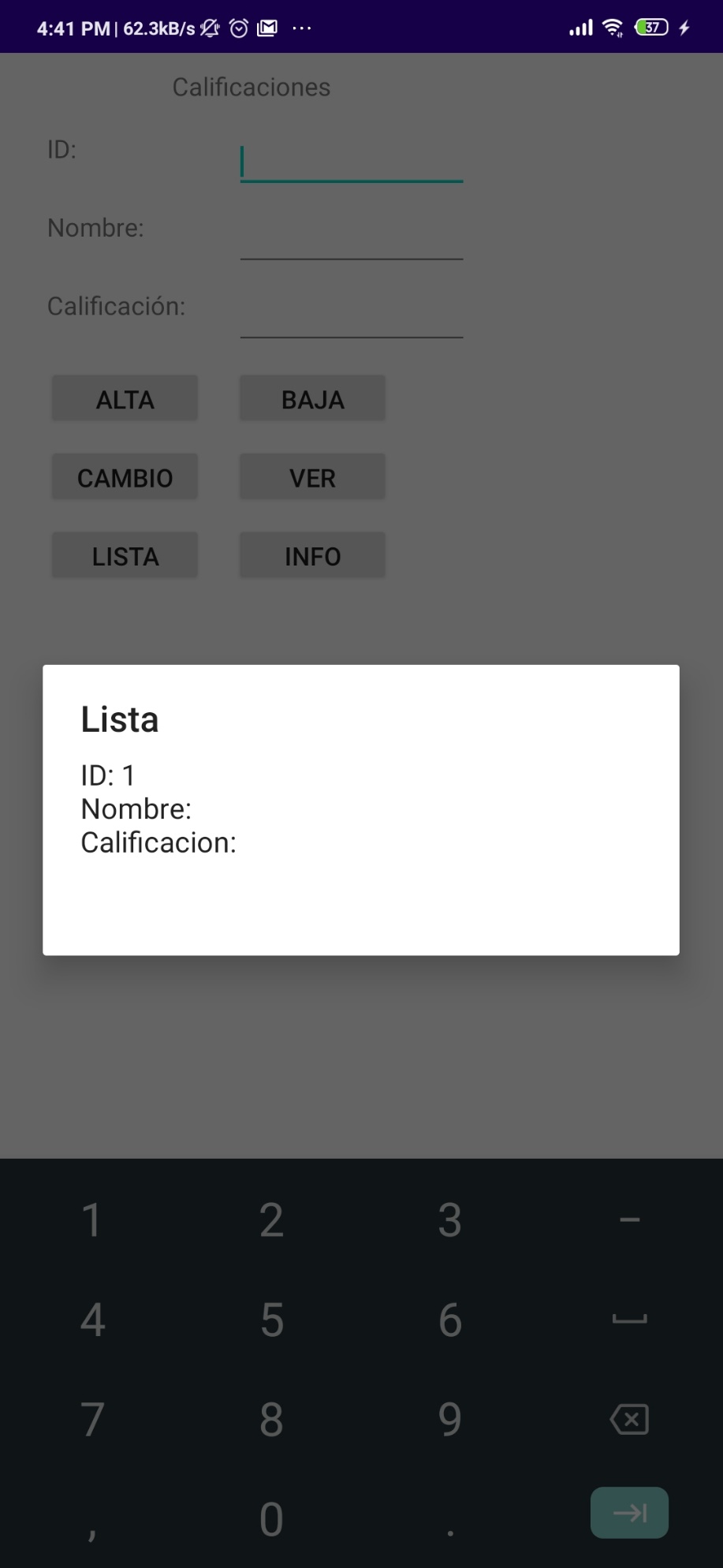
Se modifica el registro con id 1, y se quitan los datos



Se elimina registro con id = 2



Resultado final



Conclusiones

La realización de esta practica fue realmente sencilla, ya que al ser muy parecido a lo que hemos visto de SQL en java, fue fácil adaptarme al comportamiento de sql, además que por proyectos personales ya había trabajado con SQLite.

Referencias

<https://www.tutorialspoint.com/android/android_sqlite_database.htm>

<https://stackoverflow.com/questions/12015731/android-sqlite-example>