



Asignatura: Desarrollo de aplicaciones III

Profesor: Rojas López Arturo

Trabajo: Proyecto final

Alumno: Luis Alberto Lara Ortuño

Grupo: 5º D

Cuatrimestre Enero – Abril 2016

PROYECTO

AGENDA DE MIEMBROS DE UN CLUB DE FANS

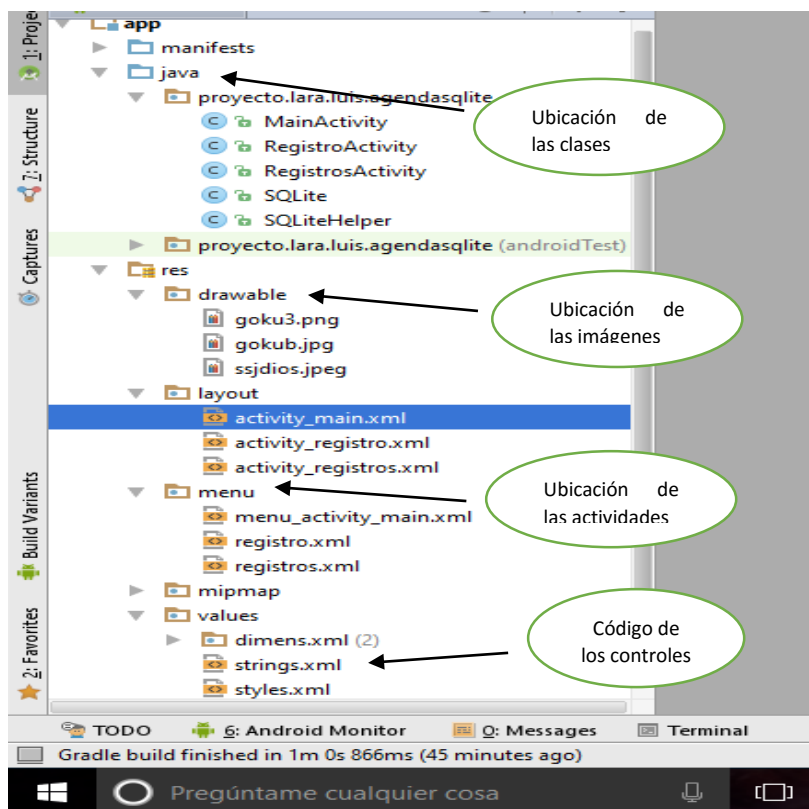
Sobre la aplicación:

Esta aplicación sirve para almacenar a los miembros de un club de fans de dragón ball z, el administrador la usara para saber lo siguiente:

- Id de miembro
- Nombre
- Personaje favorito
- Sexo
- Fecha de ingreso al club
- Recibir las últimas actualizaciones

Este proyecto es una aplicación desarrollada en Android studio con acceso a datos usando SQLite.

Estructura:



En primer lugar explicaremos la creación de la base de datos

Clase SQLiteHelper

```
package proyecto.lara.luis.agendasqlite;

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

public class SQLiteHelper extends SQLiteOpenHelper{
    //nombre de la base de datos
    private static final String __DATABASE = "dbZmiembros";
    //versión de la base de datos
    private static final int __VERSION = 3;
    //nombre tabla y campos de tabla
    public final String __Miembros__ = "Miembros";
    public final String __campo_id = "id";
    public final String __campo_nombre = "Nombre";
    public final String __campo_fechaing = "FechaIngreso";
    public final String __campo_personaje = "Personaje";
    public final String __campo_sexo = "Sexo";
    public final String __campo_actualizaciones = "Actualizaciones";
    //Instrucción SQL para crear las tablas
    /*
    * )
    * */
    private final String sql = "CREATE TABLE " + __Miembros__ + " ( " +
        __campo_id + " INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, " +
        __campo_nombre + " TEXT NULL, " + __campo_fechaing + " TEXT, " + __campo_personaje + " TEXT NULL, " +
        __campo_sexo + " TEXT NULL, " + __campo_actualizaciones + " TEXT NULL " +
        " )";

    public SQLiteHelper(Context context) { super( context, __DATABASE, null, __VERSION ); }

    @Override
    public void onCreate(SQLiteDatabase db) { db.execSQL( sql ); }

    @Override
    public void onUpgrade( SQLiteDatabase db, int oldVersion, int newVersion ) {
        if ( newVersion > oldVersion )
        {
            //elimina tabla
            db.execSQL( "DROP TABLE IF EXISTS " + __Miembros__ );
            //y luego creamos la nueva tabla
            db.execSQL( sql );
        }
    }
}
```

Librerías

Clase y extensión de la clase

Declaración de los campos que llevara la base de datos

Creación de la tabla

Cuando se crea la base de datos por primera vez.

Se llama cuando la base de datos necesita ser mejorada

Y pasamos a la clase SQLite

```

package proyecto.lara.luis.agendasqlite;

import java.util.ArrayList;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

public class SQLite {
    private SQLiteHelper sqliteHelper;
    private SQLiteDatabase db;
    /** Constructor de clase */
    public SQLite(Context context) { sqliteHelper = new SQLiteHelper( context ); }

    /** Abre conexion a base de datos */
    public void abrir(){
        Log.i("SQLite", "Se abre conexion a la base de datos " + sqliteHelper.getDatabaseName() );
        db = sqliteHelper.getWritableDatabase();
    }

    /** Cierra conexion a la base de datos */
    public void cerrar()
    {
        Log.i("SQLite", "Se cierra conexion a la base de datos " + sqliteHelper.getDatabaseName() );
        sqliteHelper.close();
    }
}

```

Librerías

Clase

Crear un objeto de ayuda para crear, abrir, y / o gestionar una base de datos.

```

public boolean addRegistro( String nombre, String fechaing, String personaje, String sexo, String actualizaciones )
{
    if( nombre.length() > 0 )
    {
        ContentValues contentValues = new ContentValues();
        contentValues.put( sqliteHelper.__campo_nombre, nombre);
        contentValues.put( sqliteHelper.__campo_fechaing, fechaing );
        contentValues.put( sqliteHelper.__campo_personaje, personaje);
        contentValues.put( sqliteHelper.__campo_sexo, sexo);
        contentValues.put( sqliteHelper.__campo_actualizaciones, actualizaciones);
        Log.i("SQLite", "Nuevo registro " );
        return ( db.insert( sqliteHelper.__Miembros__, null, contentValues ) != -1 )?true:false;
    }
    else
        return false;
}

```

Método de inserción a la base

```

public int getUltimoID()
{
    int id = -1;

    Cursor cursor = db.query( sqliteHelper.__Miembros__,
        new String[]{ sqliteHelper.__campo_id },
        null, null, null, null,
        sqliteHelper.__campo_id + " DESC ", "1");
    if( cursor.moveToFirst() )
    {
        do
        {
            id = cursor.getInt(0);
        } while ( cursor.moveToNext() );
    }
    return id;
}

```

Se guarda el id en un arreglo y va almacenando los identificadores en forma descendente en la base

```

public Cursor getRegistros()
{
    return db.query( sqliteHelper.__Miembros__ ,
        new String[]{
            sqliteHelper.__campo_id ,
            sqliteHelper.__campo_nombre ,
            sqliteHelper.__campo_fechaing ,
            sqliteHelper.__campo_personaje ,
            sqliteHelper.__campo_sexo ,
            sqliteHelper.__campo_actualizaciones
        },
        null, null, null, null, null);
}

public Cursor getRegistro( int id )
{
    return db.query( sqliteHelper.__Miembros__
        new String[]{
            sqliteHelper.__campo_id ,
            sqliteHelper.__campo_nombre ,
            sqliteHelper.__campo_fechaing ,
            sqliteHelper.__campo_personaje ,
            sqliteHelper.__campo_sexo ,
            sqliteHelper.__campo_actualizaciones
        },
        sqliteHelper.__campo_id + " = " + id ,
        null, null, null, null);
}

public ArrayList<String> getFormatListUniv( Cursor cursor )
{
    ArrayList<String> listData = new ArrayList<>();
    String item = "";
    if( cursor.moveToFirst() )
    {
        do
        {
            item += "ID fan: [" + cursor.getInt(0) + "]\r\n";
            item += "Nombre: " + cursor.getString(1) + "\r\n";
            item += "Ingreso: " + cursor.getString(2) + "\r\n";
            item += "Personaje Favorito: " + cursor.getString(3) + "\r\n";
            item += "Sexo: " + cursor.getString(4) + "\r\n";
            item += "Actualizaciones?: " + cursor.getString(5) + "";
            listData.add( item );
            item="";
        } while ( cursor.moveToNext() );
    }
    return listData;
}

```

Método que obtiene los datos de los miembros

Método que obtiene los datos de un miembro recibiendo el id como parámetro

Los datos de los miembros son almacenados en este arreglo en forma de lista con los datos que se muestran los cuales son los mismos de la tabla

Este proyecto cuenta con tres actividades

1.-actividad Main Activity

Esta es la clase inicial del programa la cual se encarga de registrar a los miembros del club.

Interfaz:



Controles:

Control	Funcion
EditText	En él se escribe el nombre del miembro
TextView	En él se escoge la fecha
Spinner	En él se almacenan los nombres de los personajes con un arreglo
RadioGroup	En él están los RadioButton
RadioButton1	Se decide el sexo hombre
RadioButton2	Se decide el sexo Mujer
CheckBox	En él se decide si se enviaran actualizaciones
Registra fan (Button)	En él se da de alta al miembro y te manda a la actividad de registro
Ver miembros (Button)	En él te manda a la actividad Registros
Cerrar aplicación (Button)	En el cierra la actividad

Codigo xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:paddingLeft="@dimen/activity_horizontal_margin"
  android:orientation="vertical"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin"
  tools:context="proyecto.lara.luis.agendasqlite.MainActivity"
  android:background="@drawable/ssjdios">
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
      android:id="@+id/lbName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/strTitle"
      android:textStyle="bold"
      android:textColor="#fcfafa" />
  </LinearLayout>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
      android:id="@+id/lbTitle"
      android:layout_width="0dip"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:text="@string/strName"
      android:textColor="#fcfafa" />

    <EditText
      android:id="@+id/txtName"
      android:layout_width="0dip"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:ems="10"
      android:inputType="textPersonName"
      android:textColor="#fcfafa" />
  </LinearLayout>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
      android:id="@+id/lbFechaNac"
      android:layout_width="0dip"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:text="@string/strFechaNac"
```

```

        android:textColor="#fcfafa" />

        <TextView
            android:id="@+id/txtFechaNac"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/strfecha"

android:textAppearance="?android:attr/textAppearanceLarge"
            android:textColor="#fcfafa" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/lbPais"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/strPais"
            android:textColor="#fcfafa" />

        <Spinner
            android:id="@+id/sPaíses"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/lbPai"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/strSexo"
            android:textColor="#fcfafa" />

        <RadioGroup
            android:id="@+id/gpSexo"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1" >

            <RadioButton
                android:id="@+id/rMan"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:checked="true"
                android:text="@string/strMan"
                android:textColor="#fcfafa" />

            <RadioButton
                android:id="@+id/rWoman"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

```



```

        android:layout_weight="1"
        android:text="@string/strWoman"
        android:textColor="#fcfafa" />
    </RadioGroup>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/lbSpeak"
        android:layout_width="0dip"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/strSpeak"
        android:textColor="#fcfafa" />

    <CheckBox
        android:id="@+id/chIdioma"
        android:layout_width="0dip"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/strSi"
        android:textColor="#fcfafa" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/btnRegistrar"
        android:layout_width="0dip"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/strReg"
        android:textColor="#fcfafa" />

    <Button
        android:id="@+id/btnRegistros"
        android:layout_width="0dip"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/strRegistros"
        android:textColor="#fcfafa" />

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

</LinearLayout>

<Button
    android:id="@+id/btnCancelar"
    android:layout_width="349dp"
    android:layout_height="66dp"
    android:text="@string/strCerrar"
    android:textColor="#fcfafa" />

</LinearLayout>

```

Clase de la actividad

```
package proyecto.tad.tad.agendaSQLite;
//
import java.util.Calendar;
//
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
```

Librerías de la clase

```
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements OnClickListener{
    //Variables para fecha
    private int mYear;
    private int mMonth;
    private int mDay;
    //
    private TextView txtFechaNac;
    private EditText txtName;
    private Spinner personajes;
    private Button btnRegistrar;
    private Button btnCancelar;
    private Button btnRegistros;
    private CheckBox chIdioma;
    private RadioGroup radioGroup;

    //
    private SQLite sqLite;
    //DatePickerDialog
    private DatePickerDialog.OnDateSetListener mDateSetListener =
        (view, year, monthOfYear, dayOfMonth) -> {
            mYear = year;
            mMonth = monthOfYear;
            mDay = dayOfMonth;
            txtFechaNac.setText( ((mDay<10)?"0"+mDay:mDay) + "/" + ((mMonth<10)?"0"+mMonth:mMonth) + "/" + mYear );
        };
}
```

Nombre de la clase, extensión y métodos de implementación

Declaración de variables

Modificación de variables para la fecha

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //
    chIdioma = (CheckBox) findViewById( R.id.chIdioma );
    txtName = (EditText) findViewById( R.id.txtName );
    btnRegistrar = (Button) findViewById(R.id.btnRegistrar );
    btnRegistrar.setOnClickListener( this );
    btnCancelar = (Button) findViewById(R.id.btnCancelar );
    btnCancelar.setOnClickListener( this );
    btnRegistros = (Button) findViewById(R.id.btnRegistros );
    btnRegistros.setOnClickListener( this );
    radioGroup = (RadioGroup) findViewById( R.id.gpSexo );
    txtFechaNac = (TextView) findViewById(R.id.txtFechaNac );
    txtFechaNac.setOnClickListener( this );

    //Se llena el Spinner con los nombres de los personajes
    String[] paises = getResources().getStringArray(R.array.personajes);
    personajes = (Spinner) findViewById(R.id.sPaises );
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, paises );
    personajes.setAdapter(adapter);

    //Obtiene fecha actual y coloca en el textview
    Calendar c = Calendar.getInstance();
    mYear = c.get(Calendar.YEAR);
    mMonth = c.get(Calendar.MONTH);
    mDay = c.get(Calendar.DAY_OF_MONTH);

```

Evento para crear
los controles

```

//base de datos
sqlite = new SQLite( this );
sqlite.abrir();
//
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_activity_main, menu);
    return true;
}

```

Evento para crear
el menú de la
actividad

```

@Override
public void onClick(View v) {
    switch ( v.getId() )
    {
        case R.id.btnRegistrar:
            int radioButtonID = radioButton.getCheckedRadioButtonId();
            View radioButton = radioButtonGroup.findViewById( radioButtonID );
            int index = radioButtonGroup.indexOfChild( radioButton );
            //Registra en la base de datos
            if ( sqlite.addRegistro( txtName.getText().toString(),
                txtFechaNac.getText().toString(),
                personajes.getSelectedItemAt().toString(),
                ( index == 0 )?"Hombre":"Mujer" ,
                ( chIdioma.isChecked() )?"Si":"No" ) )
            {
                //recupera ID de ultimo registro y pasa como parametro
                int id = sqlite.getUltimoID();
                Bundle bundle = new Bundle();
                bundle.putInt("id", id);
                Intent intent = new Intent( MainActivity.this, RegistroActivity.class );
                intent.putExtras( bundle );
                startActivity( intent );
            }
            else
            {
                Toast.makeText(getApplicationContext(), "Error: Ingresa un nombre" ,Toast.LENGTH_SHORT).show();
            }
            break;
        case R.id.btnCancelar: sqlite.cerrar(); finish(); break;
        case R.id.btnRegistros:
            Intent iRegs = new Intent( MainActivity.this, RegistrosActivity.class );
            startActivity( iRegs );
    }
}

```

Evento para cuando el usuario presione algún botón de la actividad

Mensaje de que no ha ingresado ningún nombre

```

        break;
        case R.id.txtFechaNac: verDatePicker(); break;
    }
}

/**
 * Metodo para mostrar un DatePickerDialog
 */
public void verDatePicker()
{
    DatePickerDialog d = new DatePickerDialog( this , mDateSetListener, mYear, mMonth, mDay );
    d.show();
}
}

```

Actividad 2 Activity Registro

Esta actividad es la que se encarga de mostrar al miembro registrado.

Interfaz:



Controles:

Control	Funcion
TextView	En él se mostrara al miembro registrado o seleccionado
Regresar Al registro (Button)	Con este botón regresas al registro de miembros
Eliminar fan (Button)	Con este botón das de baja a un miembro
Lista de miembros (Button)	Con este botón regresas a la lista de miembros
Cerrar aplicacion (Button)	Con este botón se Cierra la actividad

Codigo xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:id="@+id/LinearLayout2"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
android:paddingRight="@dimen/activity_horizontal_margin"
android:orientation="vertical"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context="proyecto.lara.luis.agendasqlite.RegistroActivity"
android:background="@drawable/gokub">
```

```
<TextView
    android:id="@+id/txtTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/strTitle"
    android:textStyle="bold"
    android:textColor="#fcfafa" />
```

```
<TextView
    android:id="@+id/txtResultado"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/strContenido"
    android:textColor="#f0f8f7" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
```

```
<Button
    android:id="@+id/btnNuevo"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1.12"
    android:text="@string/strNuevo"
    android:textColor="#fcfafa" />
```

```
<Button
    android:id="@+id/btnDelete"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="@string/strDelete"
    android:textColor="#fcfafa" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
```

```
<Button
    android:id="@+id/btnLista"
    android:layout_width="0dip"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="@string/strLista"
    android:textColor="#fcfafa" />
```

```
</LinearLayout>
```

```
<Button
    android:layout_width="340dp"
```

```
        android:layout_height="wrap_content"
        android:text="@string/strCierraRegistro"
        android:id="@+id/CierraRegistro"
        android:textColor="#fcfafa" />

</LinearLayout>
```

Clase de la actividad

```
package proyecto.lara.luis.agendasqlite;

import java.util.ArrayList;
import proyecto.lara.luis.agendasqlite.SQLite;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.support.v7.app.ActionBarActivity;

public class RegistroActivity extends ActionBarActivity implements OnClic
```

Librerías de
la actividad

```

public class RegistroActivity extends ActionBarActivity implements OnClickListener{
    //
    private TextView textView;
    private Button btnNuevo;
    private Button btnDelete;
    private Button btnLista;
    private Button btnCerrar;
    private SQLite sqlite;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registro);
        //
        textView = (TextView) findViewById( R.id.txtResultado );
        btnNuevo = (Button) findViewById( R.id.btnNuevo );
        btnDelete = (Button) findViewById( R.id.btnDelete );
        btnCerrar = (Button) findViewById(R.id.CierraRegistro);
        btnLista = (Button) findViewById( R.id.btnLista );
        btnNuevo.setOnClickListener( this );
        btnLista.setOnClickListener( this );
        btnDelete.setOnClickListener( this );
        btnCerrar.setOnClickListener(this);
        textView.setTextSize(28);
        //Recupera parametro ID de registro
        textView.setText( "" );
        Intent i = getIntent();
        Bundle bundle = i.getExtras();
        if ( bundle != null ) {
            int id = bundle.getInt("id");
            //base de datos
            sqlite = new SQLite( this );
            sqlite.abrir();

```

Clase, extensión y Como se implementara

Declaración de las variables

Método que crea los controles

Aquí se recupera el id de registro

Aquí se abre la base de datos

```

        Cursor cursor = sqlite.getRegistro(id);
        ArrayList<String> reg = sqlite.getFormatListUniv(cursor);
        textView.setText( reg.get(0) );
    }
    //
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.registro, menu);
        return true;
    }

    @Override
    public void onClick(View v) {
        switch ( v.getId() )
        {
            case R.id.btnNuevo:
                Intent iMain = new Intent( RegistroActivity.this, MainActivity.class );
                startActivity( iMain );
                break;
            case R.id.btnDelete:
                //Muestra una ventana de dialo para confirmar
                new AlertDialog.Builder(this)
                    .setIcon(android.R.drawable.ic_dialog_alert)
                    .setTitle("Eliminar")
                    .setMessage("En verdad desea eliminar este registro?")
                    .setPositiveButton("Si", (dialog, which) -> {
                        //Se extrae el ID = [X]
                        int posicionInicial = textView.getText().toString().indexOf("[") + 1;
                        int posicionFinal = textView.getText().toString().indexOf("]",posicionInicial);
                        String resultado = textView.getText().toString().substring(posicionInicial, posicionFinal);

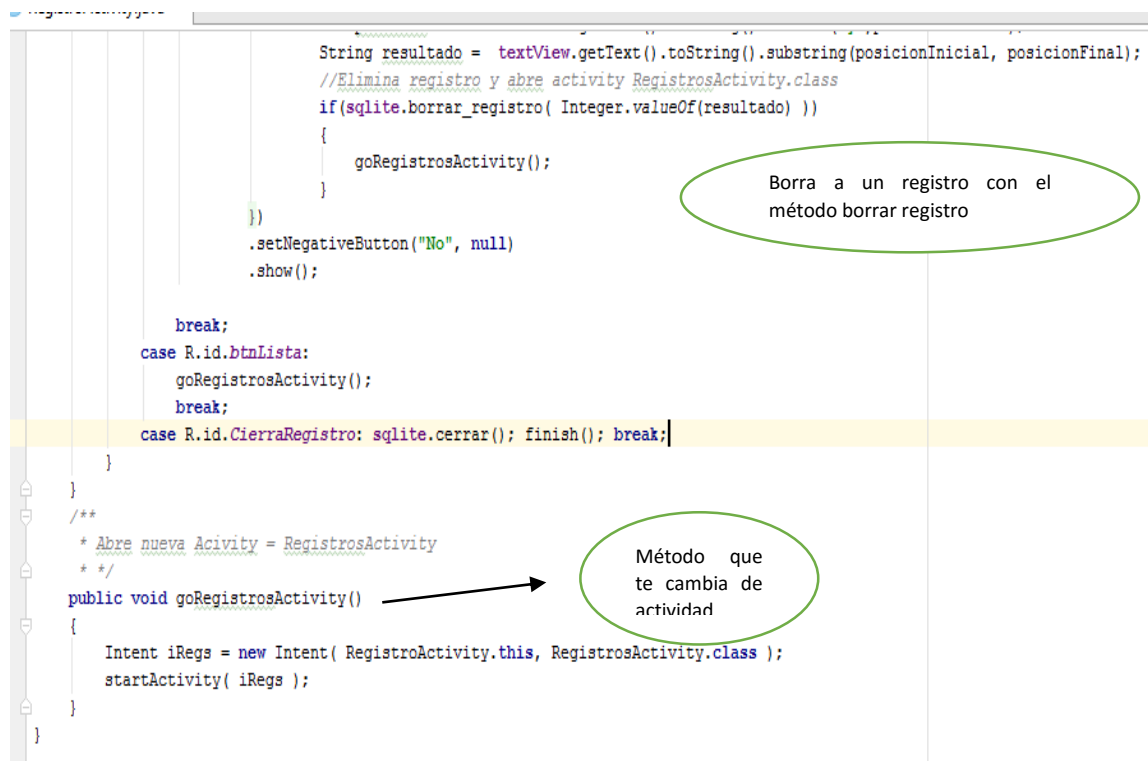
```

Aquí se ordenan en forma de lista los miembros del arreglo en orden del id

Se crea el menú de la actividad

Evento click

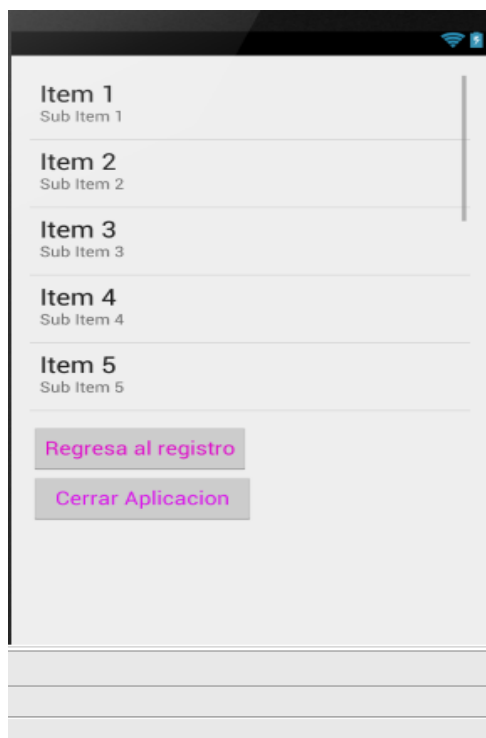
Borra a un registro con el método borrar registro



3. - Actividad de Registros

Se encarga de mostrar a los miembros del club de fans.

Interfaz:



Controles:

Control	Funcion
ListView	En él se muestran los miembros ya registrados
Regresa al registro (Button)	En él te manda al registro de miembros
Cerrar Aplicación (Button)	En él te cierra la actividad actual

Codigo xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:id="@+id/LinearLayout1"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:orientation="vertical"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="proyecto.lara.luis.agendasqlite.RegistrosActivity"
    android:baselineAligned="true">

    <ListView
        android:id="@+id/lstRegistros"
        android:layout_width="match_parent"
        android:layout_height="337dp"
        android:cacheColorHint="#faf4f4"
        >

    </ListView>

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:id="@+id/Regresarreg"
            android:text="@string/strRegresarreg"

            android:textColor="#df12ca" />

    </LinearLayout>

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```

        <Button
            android:layout_width="180dp"
            android:layout_height="match_parent"
            android:id="@+id/Cierrala"
            android:text="@string/strCierrala"

            android:textColor="#d918ea" />

    </LinearLayout>

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></LinearLayout>

</LinearLayout>

```

Clase de la actividad

```

package proyecto.lara.luis.agendasqlite;

import java.util.ArrayList;
import proyecto.lara.luis.agendasqlite.SQLite;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

public class RegistrosActivity extends Activity implements OnItemClickListener, View.OnClickListener {
    //
    private ListView listView;
    private ArrayAdapter<String> adaptador;
    private SQLite sqlite;
    Button boton;
    Button boton2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registros);
        boton = (Button) findViewById(R.id.Regresarreg);
    }
}

```

Diagram illustrating the structure of the `RegistrosActivity` class:

- Librerías** (Libraries): A bracket groups the import statements at the top of the file.
- Clase, extensión e implementaciones** (Class, extension, and implementations): An arrow points to the class declaration `public class RegistrosActivity extends Activity implements OnItemClickListener, View.OnClickListener`.
- Declaración de variables** (Declaration of variables): An arrow points to the private variables `listView`, `adaptador`, `sqlite`, `boton`, and `boton2`.
- Método para crear los controles** (Method to create the controls): An arrow points to the `onCreate` method, specifically to the line `setContentView(R.layout.activity_registros);`.

```

    boton.setOnClickListener(this);
    boton2 = (Button) findViewById(R.id.Cierrala);
    boton2.setOnClickListener(this);

    //
    listView = (ListView) findViewById(R.id.lstRegistros);
    //Abre conexion a sqlite
    sqlite = new SQLite(this);
    sqlite.abrir();
    //obtiene registros e imprimir en el listView
    Cursor cursor = sqlite.getRegistros();
    ArrayList<String> listData = sqlite.getFormatListUniv(cursor);
    adaptador = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, listData);
    listView.setAdapter(adaptador);
    listView.setOnItemClickListener(this);

    if (listData.size() == 0) {
        Toast.makeText(getBaseContext(), "No existen registros", Toast.LENGTH_SHORT).show();
    }
    //
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.registros, menu);
    return true;
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.registros, menu);
    return true;
}

@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    //
    Object object = listView.getItemAtPosition(position);
    //Se extrae el ID = [X]
    int posicionInicial = object.toString().indexOf("[");
    int posicionFinal = object.toString().indexOf("]");
    String resultado = object.toString().substring(posicionInicial, posicionFinal);
    //ejecuta nueva actividad
    Bundle b = new Bundle();
    b.putInt("id", Integer.valueOf(resultado));
    Intent iRegs = new Intent(RegistrosActivity.this, RegistroActivity.class);
    iRegs.putExtras(b);
    startActivity(iRegs);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.Regresarreg:
            Intent intent = new Intent(this, MainActivity.class);
            startActivity(intent);
            break;
        case R.id.Cierrala: sqlite.cerrar(); finish(); break;
    }
}

```

Aquí se obtienen los registros y lo imprime en el listView

Método para crear menú en la actividad

Método de devolución de llamada que se invoca cuando un artículo en este AdapterView ha hecho click.

Método cuando se llama por un click a determinado control

String

```

<resources>
    <string name="app_name">AgendaSQLite</string>
    <string name="action_settings">Settings</string>
    <string name="title_activity_registros">Registros</string>
    <string name="title_activity_registro">Registro</string>

```

```

<string name="title_activity_main">MainActivity</string>
<string name="strTitle">Registro de fans Dragon Ball Z</string>
<string name="strName">Nombre del fan :</string>
<string name="strFechaNac">Fecha de union al club :</string>
<string name="strfecha">00/00/0000</string>
<string name="strPais">Cual es su Personaje Favorito? :</string>
<string name="strSexo">Sexo</string>
<string name="strMan">Hombre</string>
<string name="strWoman">Mujer</string>
<string name="strSpeak">Rec.Actualizaciones? :</string>
<string name="strSi">Si</string>
<string name="strReg">Registrar fan!</string>
<string name="strRegistros">Ver Miembros</string>
<string name="strCerrar">Cerrar Aplicacion</string>
<string name="strContenido">Cerrar</string>
<string name="strNuevo">Regresar al registro</string>
<string name="strDelete">Eliminar Fan</string>
<string name="strLista">lista de miembros</string>
<string name="strRegresarreg">Regresa al registro</string>
<string name="strCierrala">Cerrar Aplicacion</string>
<string name="strCierraRegistro">Cerrar Aplicacion</string>

```

```

<string-array name="personajes">

```

```

    <item>Androide N°13</item>
    <item>Androide N°14</item>
    <item>Androide N°15</item>
    <item>Androide N°16</item>
    <item>Androide N°17</item>
    <item>Androide N°18</item>
    <item>Androide N°19</item>
    <item>Androide N°20</item>
    <item>Babidi</item>
    <item>Bardock</item>
    <item>Bibidi</item>
    <item>Bills</item>
    <item>Bojac</item>
    <item>Broly</item>
    <item>Bulma</item>
    <item>Bura</item>
    <item>Cell</item>
    <item>Champa</item>
    <item>Chaozu</item>
    <item>Cooler</item>
    <item>Dabura</item>
    <item>Dende</item>
    <item>Dodoria</item>
    <item>Freezer</item>
    <item>Gohan</item>
    <item>Goku</item>
    <item>Goten</item>
    <item>Hildegan</item>
    <item>Janemba</item>
    <item>Kaiohim</item>
    <item>Krillin</item>
    <item>Maestro Karin</item>
    <item>Maestro Rosshi</item>
    <item>Majin Boo</item>
    <item>Milk</item>
    <item>Mr Satan</item>
    <item>Mr Popo</item>

```

Arreglo donde se
almacenan los nombres
de los personajes

```
<item>Nappa</item>
<item>Oolong</item>
<item>Pan</item>
<item>Paragas</item>
<item>Piccolo</item>
<item>Puar</item>
<item>Raditz</item>
<item>Slug</item>
<item>Tapion</item>
<item>Ten Shin Han</item>
<item>Trunks</item>
<item>Turtles</item>
<item>Uranai Baba</item>
<item>Vados</item>
<item>Vegeta</item>
<item>Videl</item>
<item>Wiss</item>
<item>Yamcha</item>
<item>Zaabon</item>

</string-array>
<string name="re">SQLite Extreme - Registro</string>
<string name="hello_world">Hello world!</string>
<string name="regi">SQLite Extreme - Registros</string>

</resources>
```



Arreglo donde se guardan
el id de los miembros

Y con esto concluimos con la documentación del Proyecto.