

**DESARROLLO DEL APLICATIVO MÓVIL SOON PARA LA UNIVERSIDAD
COOPERATIVA DE COLOMBIA SEDE BOGOTÁ**

DOCUMENTACION DE CODIGO

Jonathan Alexander Hernández Ruiz

Sergio Alejandro Higuera Rojas

UNIVERSIDAD COOPERATIVA DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA TECNOLOGÍA EN SISTEMAS

BOGOTÁ D.C.

2017

**DESARROLLO DEL APLICATIVO MÓVIL SOON PARA LA UNIVERSIDAD
COOPERATIVA DE COLOMBIA SEDE BOGOTÁ**

**Trabajo de grado en Modalidad de Análisis Sistemático de literatura
Para optar por el título de
Tecnólogo en Sistemas**

Director

Ing. Jaime Páez.

**Línea de investigación
Desarrollo de Software**

**UNIVERSIDAD COOPERATIVA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA TECNOLOGÍA EN SISTEMAS
BOGOTÁ D.C.**

2017

TABLA DE CONTENIDO

| | |
|------------------------------|----|
| 1. MainActivity..... | 4 |
| 2. MapsActivity..... | 5 |
| 3. Adaptador..... | 7 |
| 4. Adaptador2..... | 9 |
| 5. Ayuda..... | 10 |
| 6. Calculadora_creditos..... | 11 |
| 7. Carreras..... | 13 |
| 8. Datos..... | 15 |
| 9. DbHelper..... | 17 |
| 10. DetalleActivity..... | 19 |
| 11. Emergencia..... | 22 |
| 12. InfoClase..... | 25 |
| 13. Lista..... | 26 |
| 14. Lista_Carreras..... | 38 |

1. MainActivity

```
package com.example.jarblack.uccv01;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
/*
    * Created by Jonathan on 21/11/2017.
    * Esta clase es una actividad diseñada para ejecutar el layout de
    la pantalla principal
    */

public class MainActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private Button ingresar,carreras,emergencia;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        toolbar=(Toolbar) findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);

        ingresar = (Button) findViewById(R.id.ingresar);
        carreras=(Button) findViewById(R.id.creditos);
        emergencia=(Button) findViewById(R.id.emergencia);

        ingresar.setOnClickListener(new View.OnClickListener() {

            public void onClick(View arg0) {
                ejecutar_lista(null);
            }

        });
        carreras.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(MainActivity.this,Lista_Carreras.class);
                startActivity(intent);
            }
        });

        emergencia.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
```

```

Intent (MainActivity.this, Emergencia.class);
        startActivity(intent);
    }
});

}

@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_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem opcion_menu) {

    int id=opcion_menu.getItemId();

    if(id==R.id.info){
        ejecutar_info(null);
        return true;
    }

    return super.onOptionsItemSelected(opcion_menu);
}

public void ejecutar_info(View vista){

    Intent i=new Intent(MainActivity.this, InfoClase.class);
    startActivity(i);
}

public void ejecutar_lista(View vista){

    Intent i=new Intent(MainActivity.this, Lista.class);
    startActivity(i);
}

public void salir(View vista){

    System.exit(0);
}

}

```

2. MapsActivity

```
package com.example.jarblack.uccv01;
```

```

import android.app.Activity;
import android.app.Dialog;
import android.support.annotation.Nullable;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.UiSettings;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
/*
    * Created by Jonathan on 21/11/2017.
    * esta clase es una actividad que junto con lospermisos de la Api
de google ejecutan
    * el layout de mapa para que pueda ser posible visualizar la
ubicacion del bloque seleccionado
    */
public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {

    private GoogleMap mMap;
    @Override
    protected void onCreate( Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        // Obtain the SupportMapFragment and get notified when the map is
ready to be used.

        int status=
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getApplicationContex
t());

        if(status== ConnectionResult.SUCCESS) {

            SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
            mapFragment.getMapAsync(this);
        }
        else{
            Dialog
dialog=GooglePlayServicesUtil.getErrorDialog(status, (Activity) getApplikat
ionContext(), 10);
            dialog.show();

```

```

    }
}

/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move
the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user
will be prompted to install
 * it inside the SupportMapFragment. This method will only be
triggered once the user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

    UiSettings uiSettings=mMap.getUiSettings();
    uiSettings.setZoomControlsEnabled(true);

    // Add a marker in Sydney and move the camera
    Bundle datos = this.getIntent().getExtras();
    double num1=datos.getDouble("valor1");
    double num2=datos.getDouble("valor2");
    LatLng ucc = new LatLng(num1,num2);
    mMap.addMarker(new
MarkerOptions().position(ucc).title("Universidad Cooperativa de
Colombia"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(ucc));

    float zoomlevel=16;

    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(ucc,zoomlevel));
}
}

```

3. Adaptador

```

package com.example.jarblack.uccv01;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.List;

```

```

/**
 * Created by Jonathan on 18/11/2017.
 * Esta clase esta diseñada para crear una lista de objetos con
 determinada informacion para crear una lista en
 * la aplicacion con un diseño predeterminado, en este caso la Lista de
 bloques.
 */

public class Adaptador extends BaseAdapter{

    Context contexto;
    List<Datos> ListaObjetos;

    public Adaptador(Context contexto, List<Datos> listaObjetos) {
        this.contexto = contexto;
        ListaObjetos = listaObjetos;
    }

    @Override
    public int getCount() {
        return ListaObjetos.size(); //retorna la cantidad de elementos de
la lista
    }

    @Override
    public Object getItem(int position) {
        return ListaObjetos.get(position);
    }

    @Override
    public long getItemId(int position) {
        return ListaObjetos.get(position).getId();
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        View vista=convertView;
        LayoutInflater inflate=LayoutInflater.from(contexto);
        vista=inflate.inflate(R.layout.itemlistview,null);

        ImageView imagen=(ImageView)vista.findViewById(R.id.imagen);
        TextView titulo=(TextView)vista.findViewById(R.id.titulo);
        TextView detalle=(TextView)vista.findViewById(R.id.detalle);

        titulo.setText(ListaObjetos.get(position).getTitulo().toString());

        detalle.setText(ListaObjetos.get(position).getCaracteristicas().toString(
));

        int imagenes[]=ListaObjetos.get(position).getImagenes();
        imagen.setImageResource(imagenes[0]);

        return vista;
    }
}

```



```
}  
}
```

4. Adaptador 2

```
package com.example.jarblack.uccv01;  
  
import android.content.Context;  
import android.graphics.drawable.Drawable;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import java.util.List;  
  
/**  
 * Created by Jonathan on 18/11/2017.  
 * Esta clase esta diseñada para crear una lista de objetos con  
 * determinada informacion para crear una lista en  
 * la aplicacion con un diseño predeterminado, en este caso la Lista de  
 * programas.  
 */  
  
public class Adaptador2 extends BaseAdapter{  
  
    Context context;  
    List<Carreras> ListaObjetos;  
  
    public Adaptador2(Context contexto, List<Carreras> listaObjetos) {  
        this.context = contexto;  
        ListaObjetos = listaObjetos;  
    }  
  
    @Override  
    public int getCount() {  
        return ListaObjetos.size(); //retorna la cantidad de elementos de  
la lista  
    }  
  
    @Override  
    public Object getItem(int position) {  
        return ListaObjetos.get(position);  
    }  
  
    @Override  
    public long getItemId(int position) {  
        return ListaObjetos.get(position).getId();  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent)
```

```

{
    View vista=convertView;
    LayoutInflater inflate=LayoutInflater.from(context);
    vista=inflate.inflate(R.layout.items_lista_carreras,null);

    ImageView imagen=(ImageView)vista.findViewById(R.id.img);
    TextView titulo=(TextView)vista.findViewById(R.id.title);
    int img=R.drawable.mini_logo;

    titulo.setText(ListaObjetos.get(position).getCarrera().toString());
    imagen.setImageResource(img);

    return vista;
}
}

```

5. Ayuda

```

package com.example.jarblack.uccv01;

import android.Manifest;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

/**
 * Created by Jonathan on 23/11/2017.
 * Esta clase es una actividad esta diseñada para ejecutar el layout de
ayuda
 */

public class Ayuda extends AppCompatActivity {

    Button opcion1;
    private Toolbar toolbar;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ayuda);
    }
}

```

```
}  
  
}
```

6. Calculadora_creditos

```
package com.example.jarblack.uccv01;  
  
import android.content.Intent;  
import android.icu.text.DecimalFormat;  
import android.os.Build;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.support.annotation.RequiresApi;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
/**  
 * Created by Jonathan on 18/11/2017.  
 *Esta clase es una actividad diseñada para ejecutar el layout de  
 calculadora de credits con todas sus funcionalidades  
 */  
  
public class Calculadora_Credits extends AppCompatActivity {  
  
    TextView  
carrera, credits, valor_credito, cantidad_semestres, ciudad, snies, total;  
    EditText num;  
    Button calcular;  
  
    private Toolbar toolbar;  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.calculadora_creditos);  
  
        toolbar = (Toolbar) findViewById(R.id.tool_bar);  
        setSupportActionBar(toolbar);  
    }  
}
```

```

        carrera=(TextView)findViewById(R.id.txtCarrera);
        creditos=(TextView)findViewById(R.id.txtCreditos);
        valor_credito=(TextView)findViewById(R.id.txtValor);
        cantidad_semestres=(TextView)findViewById(R.id.txtSemestres);
        ciudad=(TextView)findViewById(R.id.txtCiudad);
        snies=(TextView)findViewById(R.id.txtSnies);
        total=(TextView)findViewById(R.id.txtTotal);

        num=(EditText)findViewById(R.id.num);

        calcular=(Button)findViewById(R.id.calcular);

        Carreras
obj=(Carreras) getIntent().getExtras().getSerializable("objeto");

        carrera.setText(obj.getCarrera().toString());
        creditos.setText(Integer.toString(obj.getCreditos()));
        valor_credito.setText(Double.toString(obj.getValor_credito()));
        cantidad_semestres.setText(Integer.toString(obj.getSemestres()));
        ciudad.setText(obj.getCiudad().toString());
        snies.setText(Integer.toString(obj.getSnies()));
        double valor=obj.getValor_credito();

        calcular.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Carreras
obj=(Carreras) getIntent().getExtras().getSerializable("objeto");
                double valor=obj.getValor_credito();
                double
totalx=Double.parseDouble(num.getText().toString())*(valor);
                total.setText(String.format("%.2f",totalx));
            }
        });

    }

    @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_creditos, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem opcion_menu) {

        int id = opcion_menu.getItemId();

```

```

        if (id == R.id.info) {
            ejecutar_info(null);
            return true;
        }

        return super.onOptionsItemSelected(opcion_menu);
    }

    public void ejecutar_info(View vista) {

        Intent i = new Intent(Calculadora_Creditos.this,
InfoClase.class);
        startActivity(i);
    }

}

```

7. Carreras

```

package com.example.jarblack.uccv01;

import java.io.Serializable;

/**
 * Created by Jonathan on 23/11/2017.
 * Esta clase se diseñó para almacenar la información de los programas
academicos o carreras
 */

public class Carreras implements Serializable {

    private int id,creditos,semestres,snies;
    private double valor_credito;
    private String carrera,ciudad;

    public Carreras(int id, String carrera, int creditos, double
valor_credito, int semestres, String ciudad,int snies) {
        this.id = id;
        this.creditos = creditos;
        this.snies = snies;
        this.valor_credito = valor_credito;
        this.carrera = carrera;
        this.ciudad = ciudad;
        this.semestres=semestres;
    }
}

```

```

    public int getSemestres() {
        return semestres;
    }

    public void setSemestres(int semestres) {
        this.semestres = semestres;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getCreditos() {
        return creditos;
    }

    public void setCreditos(int creditos) {
        this.creditos = creditos;
    }

    public int getSnies() {
        return snies;
    }

    public void setSnies(int snies) {
        this.snies = snies;
    }

    public double getValor_credito() {
        return valor_credito;
    }

    public void setValor_credito(double valor_credito) {
        this.valor_credito = valor_credito;
    }

    public String getCarrera() {
        return carrera;
    }

    public void setCarrera(String carrera) {
        this.carrera = carrera;
    }

    public String getCiudad() {
        return ciudad;
    }

    public void setCiudad(String ciudad) {
        this.ciudad = ciudad;
    }
}

```

8. Datos

```
package com.example.jarblack.uccv01;

import java.io.Serializable;

/**
 * Created by Jonathan on 18/11/2017.
 * Esta clase se diseño para almacenar los datos de los bloques con sus
 * respectivas imagenes
 */

public class Datos implements Serializable {

    private int id;
    private String
    titulo,descripcion,direccion,transporte,caracteristicas,ubicacion;
    private int []imagenes;
    private double longi,lat;//Coordenadas del mapa

    public Datos(int id, String titulo,String descripcion, String
    direccion, String transporte, String caracteristicas, String ubicacion,
    int[] imagenes, double longi, double lat) {
        this.id = id;
        this.titulo = titulo;
        this.descripcion=descripcion;
        this.direccion = direccion;
        this.transporte = transporte;
        this.caracteristicas = caracteristicas;
        this.ubicacion = ubicacion;
        this.imagenes = imagenes;
        this.longi = longi;
        this.lat = lat;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getTitulo() {
        return titulo;
    }

    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
```

```

        this.direccion = direccion;
    }

    public String getTransporte() {
        return transporte;
    }

    public void setTransporte(String transporte) {
        this.transporte = transporte;
    }

    public String getCaracteristicas() {
        return caracteristicas;
    }

    public void setCaracteristicas(String caracteristicas) {
        this.caracteristicas = caracteristicas;
    }

    public String getUbicacion() {
        return ubicacion;
    }

    public void setUbicacion(String ubicacion) {
        this.ubicacion = ubicacion;
    }

    public int[] getImagenes() {
        return imagenes;
    }

    public void setImagenes(int[] imagenes) {
        this.imagenes = imagenes;
    }

    public double getLongi() {
        return longi;
    }

    public void setLongi(double longi) {
        this.longi = longi;
    }

    public double getLat() {
        return lat;
    }

    public void setLat(double lat) {
        this.lat = lat;
    }

    public String getDescripcion() {
        return descripcion;
    }

```



```

        public void setDescription(String descripcion) {
            this.descripcion = descripcion;
        }
    }
}

```

9. DbHelper

```

package com.example.jarblack.uccv01;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Build;

import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;

/**
 * Created by Jonathan on 22/11/2017.
 * Esta clase fue diseñada para conectar la aplicacion a la base de datos
 * SQLite.
 */

public class DbHelper extends SQLiteOpenHelper {

    private static String DB_PATH="";
    private static String DB_NAME="CarrerasUcc.db";
    private SQLiteDatabase mDataBase;
    private Context mContext=null;

    public DbHelper(Context context) {
        super(context, DB_NAME, null, 1);
        if (Build.VERSION.SDK_INT >=17)
            DB_PATH=context.getApplicationInfo().dataDir+"/databases/";
        else
            DB_PATH="/data/data/"+context.getPackageName()+"/databases";

        mContext=context;
    }

    @Override
    public synchronized void close() {
        if (mDataBase != null)
            mDataBase.close();
        super.close();
    }

    private boolean checkDataBase() {
        SQLiteDatabase tempDB=null;
        try{
            String path=DB_PATH+DB_NAME;
            tempDB=SQLiteDatabase.openDatabase(path,null,SQLiteDatabase.OPEN_READWRIT

```

```

E) ;

        }catch(Exception ex){}
        if(tempDB !=null)
            tempDB.close();
        return tempDB!=null?true:false;

    }

    public void copyDataBase() {
        try{
            InputStream myInput=mContext.getAssets().open(DB_NAME);
            String outputFileName=DB_PATH+DB_NAME;
            OutputStream myOutput=new FileOutputStream(outputFileName);

            byte[] buffer=new byte[1024];
            int length;
            while((length= myInput.read(buffer))>0) {
                myOutput.write(buffer,0,length);
            }

            myOutput.flush();
            myOutput.close();
            myInput.close();

        }catch (Exception ex){}

    }

    public void openDataBase() {
        String path=DB_PATH+DB_NAME;

mDataBase=SQLiteDatabase.openDatabase(path,null,SQLiteDatabase.OPEN_READW
RITE);

    }

    public void createDatabase() {
        boolean isDBExist=checkDataBase();
        if(isDBExist){

        }
        else{
            this.getReadableDatabase();
            try{
                copyDataBase();
            }catch(Exception ex){

            }

        }

    }

    //Select all Data

```

```

public List<Carreras> getAllUsers() {
    List<Carreras> temp=new ArrayList<Carreras>();
    SQLiteDatabase db=this.getWritableDatabase();
    Cursor c;
    try{
        c=db.rawQuery("SELECT * FROM Carreras",null);
        if(c==null)return null;

        c.moveToFirst();

        do{
            Carreras carrer=new
Carreras(c.getInt(c.getColumnIndex("Id")),c.getString(c.getColumnIndex("C
arrera")),c.getInt(c.getColumnIndex("Creditos")),c.getDouble(c.getColumnIndex("Valor_Credito")),c.getInt(c.getColumnIndex("Cantidad_Semestres")),c
.getString(c.getColumnIndex("Ciudad")),c.getInt(c.getColumnIndex("SNIES"
)));
            temp.add(carrer);
        }while(c.moveToNext());
        c.close();

    }catch(Exception e){

    }
    db.close();
    return temp;
}

@Override
public void onCreate(SQLiteDatabase db) {

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

}

}

```

10. DetalleActivity

```

package com.example.jarblack.uccv01;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

```

```

/**
 * Created by Jonathan on 18/11/2017.
 * Esta clase es una actividad diseñada para ejecutar el layout de
 * informacion de bloque con su respectiva informacion
 */

public class DetalleActivity extends AppCompatActivity {

    ImageView
    foto1,foto2,foto3,foto4,foto5,foto6,foto7,foto8,foto9,foto10;
    TextView
    titulo,descripcion,direccion,transporte,caracteristicas,ubicacion;

    private Toolbar toolbar;
    private Button ubicarme;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.bloque1);

        foto1=(ImageView) findViewById(R.id.imgFoto1);
        foto2=(ImageView) findViewById(R.id.imgFoto2);
        foto3=(ImageView) findViewById(R.id.imgFoto3);
        foto4=(ImageView) findViewById(R.id.imgFoto4);
        foto5=(ImageView) findViewById(R.id.imgFoto5);
        foto6=(ImageView) findViewById(R.id.imgFoto6);
        foto7=(ImageView) findViewById(R.id.imgFoto7);
        foto8=(ImageView) findViewById(R.id.imgFoto8);
        foto9=(ImageView) findViewById(R.id.imgFoto9);
        foto10=(ImageView) findViewById(R.id.imgFoto10);

        titulo=(TextView) findViewById(R.id.txtTitulo);
        descripcion=(TextView) findViewById(R.id.txtDescripcion);
        direccion=(TextView) findViewById(R.id.txtDireccion);
        transporte=(TextView) findViewById(R.id.txtTransporte);
        caracteristicas=(TextView) findViewById(R.id.txtCaracteristicas);
        ubicacion=(TextView) findViewById(R.id.txtUbicacion);

        Datos
        obj=(Datos) getIntent().getExtras().getSerializable("objeto");

        int img[]=obj.getImagenes();

        titulo.setText(obj.getTitulo());
        descripcion.setText(obj.getDescripcion());
        direccion.setText(obj.getDireccion());
        transporte.setText(obj.getTransporte());
        caracteristicas.setText(obj.getCaracteristicas());
    }
}

```

```

        ubicacion.setText(obj.getUbicacion());
        foto1.setImageResource(img[0]);
        foto2.setImageResource(img[1]);
        foto3.setImageResource(img[2]);
        foto4.setImageResource(img[3]);
        foto5.setImageResource(img[4]);
        foto6.setImageResource(img[5]);
        foto7.setImageResource(img[6]);
        foto8.setImageResource(img[7]);
        foto9.setImageResource(img[8]);
        foto10.setImageResource(img[9]);

        toolbar = (Toolbar) findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);

        ubicarme = (Button) findViewById(R.id.ubicarse);

        ubicarme.setOnClickListener(new View.OnClickListener() {

            public void onClick(View arg0) {
                ejecutar_mapa(null);
            }

        });

    }

    @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_bloques, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem opcion_menu) {

        int id = opcion_menu.getItemId();

        if (id == R.id.mapa) {
            ejecutar_mapa(null);
            return true;
        }
        if (id == R.id.info) {
            ejecutar_info(null);
            return true;
        }
        if (id == R.id.ayuda) {
            ejecutar_ayuda(null);
            return true;
        }
    }

```

```

        return super.onOptionsItemSelected(opcion_menu);
    }

    public void ejecutar_info(View vista) {

        Intent i = new Intent(DetalleActivity.this, InfoClase.class);
        startActivity(i);
    }

    public void ejecutar_mapa(View vista){

        Datos
obj=(Datos)getIntent().getExtras().getSerializable("objeto");
        Intent intent = new Intent(DetalleActivity.this,
MapsActivity.class);
        intent.putExtra("valor1",obj.getLongi());
        intent.putExtra("valor2",obj.getLat());
        startActivity(intent);
    }

    public void ejecutar_ayuda (View vista){

        Intent i = new Intent(DetalleActivity.this, Ayuda.class);
        startActivity(i);
    }

    public void salirApp(View vista) {
        finish();
    }
}

```

11. Emergencia

```

package com.example.jarblack.uccv01;

import android.Manifest;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

```

```

import cn.pedant.SweetAlert.SweetAlertDialog;

/**
 * Created by Jonathan on 23/11/2017.
 * Esta clase es una actividad diseñada para ejecutar el layout de
 emergencia con su respectiva informacion
 */

public class Emergencia extends AppCompatActivity {
    public static final int MI_PERMISO_CALL_PHONE = 1;
    Button opcion1;
    private Toolbar toolbar;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.emergencia);

        toolbar = (Toolbar) findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);

        opcion1 = (Button) findViewById(R.id.opc1);

        opcion1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                AlertDialog.Builder dialogo = new
AlertDialog.Builder(Emergencia.this);

                dialogo.setTitle("Seguro que desea llamar?");
                dialogo.setMessage("Esta opcion puede generar costos al
operador de telefonía ¿Deseas continuar?");

                dialogo.setPositiveButton("Sí", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
which) {

                        Intent i = new Intent(Intent.ACTION_CALL,
Uri.parse("tel:0315878027"));
                        if (ActivityCompat.checkSelfPermission(Emergencia.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {

                            if
(ActivityCompat.shouldShowRequestPermissionRationale(Emergencia.this,
Manifest.permission.CALL_PHONE)) {

                                new SweetAlertDialog(Emergencia.this,
SweetAlertDialog.WARNING_TYPE)
                                    .setTitleText("Atencion!")
                                    .setContentText("Debes otorgar permisos

```

```

para realizar la llamada")
        .setConfirmText("Solicitar permiso")
        .setCancelText("Cancelar")
        .setCancelClickListener(new
SweetAlertDialog.OnSweetClickListener() {
            @Override
            public void onClick(SweetAlertDialog
sweetAlertDialog) {
                sweetAlertDialog.cancel();
            }
        })
        .setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
            @Override
            public void onClick(SweetAlertDialog
sweetAlertDialog) {
                sweetAlertDialog.cancel();
            }
        })
        .show();
    } else {
        ActivityCompat.requestPermissions(Emergencia.this, new
String[]{Manifest.permission.CALL_PHONE},
MI_PERMISO_CALL_PHONE);
    }
}
} else {
    ActivityCompat.requestPermissions(Emergencia.this, new
String[]{Manifest.permission.CALL_PHONE},
MI_PERMISO_CALL_PHONE);
    }
} else {
    startActivity(i);
}
}
});
    dialogo.setNegativeButton("No", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int
which) {
            dialog.cancel();
        }
    });
    dialogo.create();
    dialogo.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.menu_listas, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem opcion_menu) {

    int id = opcion_menu.getItemId();

    if (id == R.id.info) {
        ejecutar_info(null);
        return true;
    }

    return super.onOptionsItemSelected(opcion_menu);
}

public void ejecutar_info(View vista) {

    Intent i = new Intent(Emergencia.this, InfoClase.class);
    startActivity(i);
}
}

```

12. InfoClase

```

package com.example.jarblack.uccv01;

import android.app.Activity;
import android.os.Bundle;

/**
 * Created by Jonathan on 7/11/2017.
 * Esta clase es una actividad diseñada para ejecutar el layout de
    informacion
 */

public class InfoClase extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.info);
    }
}

```

13. Lista

```

package com.example.jarblack.uccv01;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.io.Serializable;
import java.util.ArrayList;

/**
 * Created by Jonathan on 9/11/2017.
 * Esta clase es una actividad diseñada para ejecutar el layout de lista
 * bloques con su respectiva informacion y organizacion
 */

public class Lista extends AppCompatActivity{
    private Toolbar toolbar;

    ListView listaDatos;
    ArrayList<Datos> Lista;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.lista_bloques);

        toolbar=(Toolbar) findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);

        int
img_bloque1[]={R.drawable.bloque1,R.drawable.bloque1_2,R.drawable.bloque1_3,R.drawable.bloque1_4,R.drawable.bloque1_5,R.drawable.bloque1_6,R.drawable.bloque1_7,R.drawable.bloque1_8,R.drawable.bloque1_9,R.drawable.bloque1_10};

        int
img_bloque2[]={R.drawable.bloque2,R.drawable.bloque2_2,R.drawable.bloque2_3,R.drawable.bloque2_4,R.drawable.bloque2_5,R.drawable.bloque2_6,R.drawable.bloque2_7,R.drawable.bloque2_8,R.drawable.bloque2_9,R.drawable.bloque2_10};

        int
img_bloque3[]={R.drawable.bloque3,R.drawable.bloque3_2,R.drawable.bloque3_3,R.drawable.bloque3_4,R.drawable.bloque3_5,R.drawable.bloque3_6,R.drawable.bloque3_7,R.drawable.bloque3_8,R.drawable.bloque3_9,R.drawable.bloque3_10};

        int
img_bloque4[]={R.drawable.bloque4,R.drawable.bloque4_2,R.drawable.bloque4

```

```

_3,R.drawable.bloque4_4,R.drawable.bloque4_5,R.drawable.bloque4_6,R.drawa
ble.bloque4_7,R.drawable.bloque4_8,R.drawable.bloque4_9,R.drawable.bloque
4_10};

    int
img_bloque5[]={R.drawable.bloque5,R.drawable.bloque5_2,R.drawable.bloque5
_3,R.drawable.bloque5_4,R.drawable.bloque5_5,R.drawable.bloque5_6,R.drawa
ble.bloque5_7,R.drawable.bloque5_8,R.drawable.bloque5_9,R.drawable.bloque
5_10};

    int
img_bloque6[]={R.drawable.bloque6,R.drawable.bloque6_2,R.drawable.bloque6
_3,R.drawable.bloque6_4,R.drawable.bloque6_5,R.drawable.bloque6_2,R.drawa
ble.bloque6_3,R.drawable.bloque6_4,R.drawable.bloque6_5,R.drawable.bloque
6_2};

    int
img_bloque8[]={R.drawable.bloque8,R.drawable.bloque8_2,R.drawable.bloque8
_3,R.drawable.bloque8_4,R.drawable.bloque8_5,R.drawable.bloque8_6,R.drawa
ble.bloque8_7,R.drawable.bloque8_8,R.drawable.bloque8_9,R.drawable.bloque
8_10};

    int
img_bloque9[]={R.drawable.bloque9,R.drawable.bloque9_2,R.drawable.bloque9
_3,R.drawable.bloque9_4,R.drawable.bloque9_5,R.drawable.bloque9_6,R.drawa
ble.bloque9_7,R.drawable.bloque9_8,R.drawable.bloque9_9,R.drawable.bloque
9_2};

    int
img_bloque10[]={R.drawable.bloque10,R.drawable.bloque10_2,R.drawable.bloq
ue10_3,R.drawable.bloque10_4,R.drawable.bloque10_5,R.drawable.bloque10_6,
R.drawable.bloque10_7,R.drawable.bloque10_8,R.drawable.bloque10_9,R.drawa
ble.bloque10_10};

    int
img_bloque12[]={R.drawable.bloque12,R.drawable.bloque12_2,R.drawable.bloq
ue12_3,R.drawable.bloque12_4,R.drawable.bloque12_5,R.drawable.bloque12_6,
R.drawable.bloque12_7,R.drawable.bloque12_8,R.drawable.bloque12_9,R.drawa
ble.bloque12_10};

    int
img_bloque13[]={R.drawable.bloque13,R.drawable.bloque13_2,R.drawable.bloq
ue13_3,R.drawable.bloque13_4,R.drawable.bloque13_5,R.drawable.bloque13_6,
R.drawable.bloque13_7,R.drawable.bloque13_2,R.drawable.bloque13_3,R.drawa
ble.bloque13_4};

    int
img_bloque14[]={R.drawable.bloque14,R.drawable.bloque14_2,R.drawable.bloq
ue14_3,R.drawable.bloque14_4,R.drawable.bloque14_5,R.drawable.bloque14_6,
R.drawable.bloque14_7,R.drawable.bloque14_8,R.drawable.bloque14_9,R.drawa
ble.bloque14_10};

    int
img_bloque15[]={R.drawable.bloque15,R.drawable.bloque15_2,R.drawable.bloq
ue15_3,R.drawable.bloque15_4,R.drawable.bloque15_5,R.drawable.bloque15_6,
R.drawable.bloque15_7,R.drawable.bloque15_8,R.drawable.bloque15_9,R.drawa
ble.bloque15_10};

    int
img_bloque16[]={R.drawable.bloque16,R.drawable.bloque16_2,R.drawable.bloq
ue16_3,R.drawable.bloque16_4,R.drawable.bloque16_5,R.drawable.bloque16_6,
R.drawable.bloque16_7,R.drawable.bloque16_8,R.drawable.bloque16_2,R.drawa
ble.bloque16_3};

    int
img_bloque17[]={R.drawable.bloque17,R.drawable.bloque17_2,R.drawable.bloq
ue17_3,R.drawable.bloque17_4,R.drawable.bloque17_2,R.drawable.bloque17_3,

```

```

R.drawable.bloque17_4,R.drawable.bloque17_2,R.drawable.bloque17_3,R.drawa
ble.bloque17_4};
    int
img_bloque18[]={R.drawable.bloque18,R.drawable.bloque18_2,R.drawable.bloq
ue18_3,R.drawable.bloque18_4,R.drawable.bloque18_5,R.drawable.bloque18_2,
R.drawable.bloque18_3,R.drawable.bloque18_4,R.drawable.bloque18_5,R.drawa
ble.bloque18_2};
    int
img_bloque19[]={R.drawable.bloque19,R.drawable.bloque19_2,R.drawable.bloq
ue19_3,R.drawable.bloque19_4,R.drawable.bloque19_5,R.drawable.bloque19_6,
R.drawable.bloque19_2,R.drawable.bloque19_3,R.drawable.bloque19_4,R.drawa
ble.bloque19_5};
    int
img_bloque23[]={R.drawable.bloque23,R.drawable.bloque23_2,R.drawable.bloq
ue23_3,R.drawable.bloque23_4,R.drawable.bloque23_5,R.drawable.bloque23_6,
R.drawable.bloque23_7,R.drawable.bloque23_8,R.drawable.bloque23_9,R.drawa
ble.bloque23_10};
    int
img_bloque24[]={R.drawable.bloque24,R.drawable.bloque24_2,R.drawable.bloq
ue24_3,R.drawable.bloque24_4,R.drawable.bloque24_5,R.drawable.bloque24_6,
R.drawable.bloque24_7,R.drawable.bloque24_8,R.drawable.bloque24_2,R.drawa
ble.bloque24_3};
    int
img_bloque25[]={R.drawable.bloque25,R.drawable.bloque25_2,R.drawable.bloq
ue25_3,R.drawable.bloque25_4,R.drawable.bloque25_5,R.drawable.bloque25_6,
R.drawable.bloque25_7,R.drawable.bloque25_8,R.drawable.bloque25_9,R.drawa
ble.bloque25_2};
    int
img_bloque26[]={R.drawable.bloque26,R.drawable.bloque26_2,R.drawable.bloq
ue26_3,R.drawable.bloque26_4,R.drawable.bloque26_5,R.drawable.bloque26_6,
R.drawable.bloque26_7,R.drawable.bloque26_8,R.drawable.bloque26_9,R.drawa
ble.bloque26_10};
    int
img_bloque27[]={R.drawable.bloque27,R.drawable.bloque27_2,R.drawable.bloq
ue27_3,R.drawable.bloque27_4,R.drawable.bloque27_2,R.drawable.bloque27_3,
R.drawable.bloque27_4,R.drawable.bloque27_2,R.drawable.bloque27_3,R.drawa
ble.bloque27_4};
    int
img_bloque28[]={R.drawable.bloque28,R.drawable.bloque28_2,R.drawable.bloq
ue28_3,R.drawable.bloque28_4,R.drawable.bloque28_5,R.drawable.bloque28_6,
R.drawable.bloque28_7,R.drawable.bloque28_8,R.drawable.bloque28_9,R.drawa
ble.bloque28_10};
    int
img_bloque30[]={R.drawable.bloque30,R.drawable.bloque30_2,R.drawable.bloq
ue30_3,R.drawable.bloque30_4,R.drawable.bloque30_5,R.drawable.bloque30_6,
R.drawable.bloque30_7,R.drawable.bloque30_8,R.drawable.bloque30_9,R.drawa
ble.bloque30_10};
    int
img_bloque31[]={R.drawable.bloque31,R.drawable.bloque31_2,R.drawable.bloq
ue31_3,R.drawable.bloque31_4,R.drawable.bloque31_5,R.drawable.bloque31_6,
R.drawable.bloque31_7,R.drawable.bloque31_2,R.drawable.bloque31_3,R.drawa
ble.bloque31_4};
    int
img_bloque32[]={R.drawable.bloque32,R.drawable.bloque32_2,R.drawable.bloq
ue32_3,R.drawable.bloque32_4,R.drawable.bloque32_5,R.drawable.bloque32_2,
R.drawable.bloque32_3,R.drawable.bloque32_4,R.drawable.bloque32_5,R.drawa

```

```

ble.bloque32_2};
    int
img_bloque34[]={R.drawable.bloque34,R.drawable.bloque34_2,R.drawable.bloque34_3,R.drawable.bloque34_4,R.drawable.bloque34_5,R.drawable.bloque34_6,R.drawable.bloque34_2,R.drawable.bloque34_3,R.drawable.bloque34_4,R.drawable.bloque34_5};

    listaDatos=(ListView)findViewById(R.id.lstDatos);

    Lista=new ArrayList<Datos>();

//id,titulo,descripcion,direccion,transporte,caracteristicas,ubicacion,
imagenes, longi,lat
    Lista.add(new Datos(
        /*id*/ 1,
        /*titulo*/ "Bloque 1",
        /*descripcion*/ "Ingenierías",
        /*direccion*/ "Av Caracas # 37-63",
        /*Transporte*/ "La estación más cercana de Transmilenio es la Av39 ubicada en la caracas, es el medio de transporte más utilizado por los estudiantes de esta sede.",
        /*caracteristicas*/ "Biblioteca-Correspondencia-Carnetizacion-CIPIIS-Emisora",
        /*ubicacion*/ "Esta sede queda ubicada enfrente de la estación de la Av39 donde se encuentran sitios aledaños, como el edificio de RCN Radio, un Subway y los bloques 3 y 15.",
        /*imagenes*/ img_bloque1,
        /*longi*/ 4.626147,
        /*lat*/ -74.069265
    ));

    Lista.add(new Datos(
        /*id*/ 2,
        /*titulo*/ "Bloque 2",
        /*descripcion*/ "Departamento de comunicaciones",
        /*direccion*/ "Carrera 15 #38-04",
        /*Transporte*/ "La estación más cercana de Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/ "Capilla-oficinas-Gestión Humana",
        /*ubicacion*/ "Para llegar hay que pasar la fotocopiadora y seguir el camino, una vez encuentre la oficina de la Analista de Gestión Humana estará ubicado en el bloque 2.",
        /*imagenes*/ img_bloque2,
        /*longi y lat*/ 4.626464, -74.069623
    ));

    Lista.add(new Datos(
        /*id*/ 3,
        /*titulo*/ "Bloque 3",
        /*descripcion*/ "Facultad de Derecho",
        /*direccion*/ "Av Caracas # 37-43",
        /*Transporte*/ "La estación más cercana de Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/ "Servicio al estudiante-Cajero BBVA",

```

```

        /*ubicacion*/      "Este bloque se encuentra ubicado
entre el bloque 4 y el bloque 1 enfrente de la estación de la Av 39.",
        /*imagenes*/      img_bloque3,
        /*longi*/          4.625770, -74.069024

    ));

    Lista.add(new Datos(
        /*id*/              4,
        /*titulo*/          "Bloque 4",
        /*descripcion*/      "Facultad de Ingenierías",
        /*direccion*/        "Av Caracas # 37-15",
        /*Transporte*/        "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/  "Sala de profesores-Coordinadores
Ingenierías.",
        /*ubicacion*/        "Se encuentra al lado del bloque 3
enfrente de la estación de la Av39.",
        /*imagenes*/        img_bloque4,
        /*longi*/            4.625779, -74.069231

    ));

    Lista.add(new Datos(
        /*id*/              5,
        /*titulo*/          "Bloque 5",
        /*descripcion*/      "Departamento de Idiomas",
        /*direccion*/        "Calle 37 #14-34",
        /*Transporte*/        "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/  "Salas de informática para realizar
trabajos en la plataforma Rosseta Stone-atención al estudiante niveles
de inglés.-Biblioteca-Casilleros-CIFAD ",
        /*ubicacion*/        "Esta bloque queda ubicado detrás del
bloque 3.",
        /*imagenes*/        img_bloque5,
        /*longi*/            4.625631, -74.069328

    ));

    Lista.add(new Datos(
        /*id*/              6,
        /*titulo*/          "Bloque 6",
        /*descripcion*/      "Facultad de Odontología",
        /*direccion*/        "Carrera 13a #38-22",
        /*Transporte*/        "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/  "Instalaciones para prestar servicios
de odontología.",
        /*ubicacion*/        "Este bloque queda ubicado cerca a la
sede 30 de Bienestar y de una cafetería TOSTAO.",
        /*imagenes*/        img_bloque6,
        /*longi*/            4.626329, -74.068015

    ));

```

```

        Lista.add(new Datos(
            /*id*/ 8,
            /*titulo*/ "Bloque 8",
            /*descripcion*/ "Contaduría pública",
            /*direccion*/ "Carrera 14a #40a-73",
            /*Transporte*/ "La estación más cercana de Transmilenio es la de Calle 45 ubicada en la Caracas.",
            /*caracteristicas*/ "Sala de profesores-oficina del Decano de contabilidad-oficinas-salones.",
            /*ubicacion*/ "Este bloque queda ubicado al lado de la funeraria los olivos de la 45 y al frente de la estación de Transmilenio de Calle 45.",
            /*imagenes*/ img_bloque8,
            /*longi*/ 4.629936, -74.068748

        ));

        Lista.add(new Datos(
            /*id*/ 9,
            /*titulo*/ "Bloque 9",
            /*descripcion*/ "Orientación Psicológica",
            /*direccion*/ "Carrera 14a #40a-65",
            /*Transporte*/ "La estación más cercana de Transmilenio es la de Calle 45 ubicada en la Caracas.",
            /*caracteristicas*/ "Orientación Psicológica\n",
            /*ubicacion*/ "Este bloque queda ubicado al lado del bloque 8 muy cerca de un caño. ",
            /*imagenes*/ img_bloque9,
            /*longi*/ 4.630140, -74.068682

        ));

        Lista.add(new Datos(
            /*id*/ 10,
            /*titulo*/ "Bloque 10",
            /*descripcion*/ "Casa de Extensión",
            /*direccion*/ "Calle 39 #14-18",
            /*Transporte*/ "La estación más cercana de Transmilenio es la Av39 ubicada en la Caracas.",
            /*caracteristicas*/ "Recepción- Mercadeo- Oficina de Egresados- Proyección Social -Internacionalización- INDESCO -Extensión",
            /*ubicacion*/ "Este bloque se encuentra ubicado al lado del bloque 14 y enfrente del 27. ",
            /*imagenes*/ img_bloque10,
            /*longi*/ 4.626820, -74.068998

        ));

        Lista.add(new Datos(
            /*id*/ 12,
            /*titulo*/ "Bloque 12",
            /*descripcion*/ "Centro de cómputo",
            /*direccion*/ "Carrera 15 #37-58",

```

```

        /*Transporte*/      "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/ "Salas de informática-Centro de
computo-institucional.",
        /*ubicacion*/      "Este bloque se encuentra ubicado
enseguida del bloque 2.",
        /*imagenes*/      img_bloque12,
        /*longi*/          4.626190, -74.069842

    ));

    Lista.add(new Datos(
        /*id*/              13,
        /*titulo*/          "Bloque 13",
        /*descripcion*/      "Asesora Rectoría",
        /*direccion*/        "Av Caracas #44-21",
        /*Transporte*/      "La estación más cercana de
Transmilenio es la de Calle 45 ubicada en la Caracas.",
        /*caracteristicas*/ "Asesora Rectoría- Secretaría
General- INDESCO -Nacional Editorial- Contaduría Pública.",
        /*ubicacion*/      "Se encuentra ubicado enfrente de la
estación de la calle 45.",
        /*imagenes*/      img_bloque13,
        /*longi*/          4.632002, -74.067967

    ));

    Lista.add(new Datos(
        /*id*/              14,
        /*titulo*/          "Bloque 14",
        /*descripcion*/      "Tesorería y DARC",
        /*direccion*/        "Av Caracas #38-45",
        /*Transporte*/      "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/ "Tesorería- DARC- Subdirección
Financiera- Contabilidad- Gestión Tecnológica -Subdirección de Desarrollo
Institucional- Infraestructura Física- Planeación Dirección de Sede-
Subdirección Académica- Subdirección de Proyección.",
        /*ubicacion*/      "Este bloque se encuentra ubicado al
lado del bloque 15 y cerca al bloque 10 y 16.",
        /*imagenes*/      img_bloque14,
        /*longi*/          4.626541, -74.068931

    ));

    Lista.add(new Datos(
        /*id*/              15,
        /*titulo*/          "Bloque 15",
        /*descripcion*/      "Audiovisuales",
        /*direccion*/        "Av Caracas #38-27",
        /*Transporte*/      "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/ "Cafetería-Audiovisuales-CISCO-Salas
de informática.",
        /*ubicacion*/      "Este bloque queda ubicado al lado
del bloque 34 y 14.Al lado de la cancha.",

```



```

        /*imagenes*/      img_bloque15,
        /*longi*/          4.626444, -74.068891

    ));

    Lista.add(new Datos(
        /*id*/              16,
        /*titulo*/           "Bloque 16",
        /*descripcion*/       "Facultad de ciencias
administrativas",
        /*direccion*/         "Av Caracas #37-20",
        /*Transporte*/         "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/   "Ciencias Administrativas, Económicas
y Afines.",
        /*ubicacion*/         "Este bloque está ubicado enfrente de
la estación de Transmilenio de la Av39 al lado de RCN Radio.",
        /*imagenes*/          img_bloque16,
        /*longi*/             4.625477, -74.068840

    ));

    Lista.add(new Datos(
        /*id*/              17,
        /*titulo*/           "Bloque 17",
        /*descripcion*/       "Postgrados",
        /*direccion*/         "Carrera 6 #11-51",
        /*Transporte*/         "La estacion mas cercana de
transmilenio es Museo del oro ",
        /*caracteristicas*/   "Postgrados",
        /*ubicacion*/         "Este bloque se encuentra ubicado a
un par de cuadras de la estacion del museo del oro",
        /*imagenes*/          img_bloque17,
        /*longi*/             4.598146, -74.074193

    ));

    Lista.add(new Datos(
        /*id*/              18,
        /*titulo*/           "Bloque 18",
        /*descripcion*/       "Tecnologias",
        /*direccion*/         "Diagonal 42 Sur #23-70",
        /*Transporte*/         "La estacion mas cercana de
transmilenio es la Calle 40 Sur en donde podemos coger el alimentador 7-2
tunal y bajarnos en la primera parada que nos dejara a una cuadra del
bloque.",
        /*caracteristicas*/   "Salones de tecnologia-Ping Pong-
Gimnasio",
        /*ubicacion*/         "",
        /*imagenes*/          img_bloque18,
        /*longi*/             4.578702, -74.123886

    ));

    Lista.add(new Datos(
        /*id*/              19,

```

```

        /*titulo*/          "Bloque 19",
        /*descripcion*/      "Sociología",
        /*direccion*/        "Carrera 15 #42-26",
        /*Transporte*/        "La estación más cercana de
Transmilenio es la de Calle 45 ubicada en la Caracas.",
        /*caracteristicas*/  "Sociología",
        /*ubicacion*/        "Este bloque se encuentra ubicado
cerca al bloque 8. ",
        /*imagenes*/         img_bloque19,
        /*longi*/            4.630478, -74.068994

    ));

    Lista.add(new Datos(
        /*id*/                23,
        /*titulo*/            "Bloque 23",
        /*descripcion*/        "Primeros semestres",
        /*direccion*/          "Av Caracas #45a-55",
        /*Transporte*/          "La estación más cercana de
Transmilenio es la de Calle 45 ubicada en la Caracas.",
        /*caracteristicas*/    "Salones primeros semestres",
        /*ubicacion*/          "Este bloque queda muy cerca al
bloque 13 enfrente de la estación de Transmilenio de la Calle 45.",
        /*imagenes*/          img_bloque23,
        /*longi*/            4.632837, -74.067781

    ));

    Lista.add(new Datos(
        /*id*/                24,
        /*titulo*/            "Bloque 24",
        /*descripcion*/        "Primeros semestres",
        /*direccion*/          "Carrera 15 #37-10",
        /*Transporte*/          "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/    "Salones",
        /*ubicacion*/          "Este bloque queda ubicado al lado
del bloque 12.",
        /*imagenes*/          img_bloque24,
        /*longi*/            4.625819, -74.069918

    ));

    Lista.add(new Datos(
        /*id*/                25,
        /*titulo*/            "Bloque 25",
        /*descripcion*/        "Archivo",
        /*direccion*/          "Diagonal 40a#16-44",
        /*Transporte*/          "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/    "Archivo",
        /*ubicacion*/          "Este bloque queda ubicado por la
cuadra de la Uninpahu al lado del colegio Champagna.",
        /*imagenes*/          img_bloque25,
        /*longi*/            4.629480, -74.070534
    ));

```

```

));

Lista.add(new Datos(
    /*id*/ 26,
    /*titulo*/ "Bloque 26",
    /*descripcion*/ "Institucionales",
    /*direccion*/ "Calle 44 #15-20",
    /*Transporte*/ "La estación más cercana de Transmilenio es la de Calle 45 ubicada en la Caracas.",
    /*caracteristicas*/ "Mercadeo-Administración de empresas-Institucional.",
    /*ubicacion*/ "Este bloque queda ubicado cerca al bloque 19 y de la estación de Transmilenio de la Calle 45.",
    /*imagenes*/ img_bloque26,
    /*longi*/ 4.631978, -74.069014

));

Lista.add(new Datos(
    /*id*/ 27,
    /*titulo*/ "Bloque 27",
    /*descripcion*/ "Consultorio Jurídico",
    /*direccion*/ "Calle 39 #14-39",
    /*Transporte*/ "La estación más cercana de Transmilenio es la Av39 ubicada en la Caracas.",
    /*caracteristicas*/ "Consultorio Jurídico, Centro de conciliación.",
    /*ubicacion*/ "Este bloque se encuentra ubicado al lado del bloque 14 y enfrente del bloque 10.",
    /*imagenes*/ img_bloque27,
    /*longi*/ 4.626844, -74.069181

));

Lista.add(new Datos(
    /*id*/ 28,
    /*titulo*/ "Bloque 28",
    /*descripcion*/ "Odontología\n",
    /*direccion*/ "Carrer 13 Bis #43-23",
    /*Transporte*/ "La estación más cercana de Transmilenio es la de Calle 45 ubicada en la Caracas. Otro medio de transporte es el SITP que pasa en la Carrera 13.",
    /*caracteristicas*/ "Cafetería-Audiovisuales-Papelería-Odontología.",
    /*ubicacion*/ "Se encuentra ubicada al lado de una funeraria cerca de la Carrera 13.",
    /*imagenes*/ img_bloque28,
    /*longi*/ 4.630934, -74.067182

));

Lista.add(new Datos(
    /*id*/ 30,
    /*titulo*/ "Bloque 30",

```

```

        /*descripcion*/      "Bienestar",
        /*direccion*/        "Calle 38 #13a-79",
        /*Transporte*/        "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/   "Gimnasio-Baile-Música-Ping-Pong-
Eventos.",
        /*ubicacion*/        "Se encuentra cerca al Subway ubicado
enfrente de la estación de Transmilenio de la Av 39 y cerca al bloque
6.",
        /*imagenes*/         img_bloque30,
        /*longi*/             4.626031, -74.068111

    ));

    Lista.add(new Datos(
        /*id*/                 31,
        /*titulo*/             "Bloque 31",
        /*descripcion*/        "Campus-Psicología",
        /*direccion*/          "Carrera 9a #172-90",
        /*Transporte*/          "Portal Norte-Alimentador San
Cristóbal tercera parada.",
        /*caracteristicas*/    "Gimnasio-ping-pong -cancha de
futbol-sauna-cancha de baloncesto",
        /*ubicacion*/          "Se encuentra ubicada en la 170 al
lado del colegio La Salle y de la Universidad San Buenaventura.",
        /*imagenes*/          img_bloque31,
        /*longi*/              4.751034, -74.031033

    ));

    Lista.add(new Datos(
        /*id*/                 32,
        /*titulo*/             "Bloque 32",
        /*descripcion*/        "",
        /*direccion*/          "Diagonal 40 #14-35",
        /*Transporte*/          "Las estaciones más cercanas de
Transmilenio son la de la Calle 45 y la Av39.",
        /*caracteristicas*/    "Salones",
        /*ubicacion*/          "Este bloque se encuentra ubicado en
la Caracas entre las estaciones de Transmilenio de la Calle 45 y la
Av39.Al lado hay un caño.",
        /*imagenes*/          img_bloque32,
        /*longi*/              4.628583, -74.068294

    ));

    Lista.add(new Datos(
        /*id*/                 34,
        /*titulo*/             "Bloque 34",
        /*descripcion*/        "Salones",
        /*direccion*/          "Calle 39 #14-29",
        /*Transporte*/          "La estación más cercana de
Transmilenio es la Av39 ubicada en la Caracas.",
        /*caracteristicas*/    "Salones",
        /*ubicacion*/          "Este bloque se ubica pasando por la

```

```

cafetería del bloque 15 hasta llegar a una fuente en donde encontraremos
al costado derecho unos baños, si sigues este camino lo encontraras
fácilmente. ",
        /*imagenes*/          img_bloque34,
        /*longi*/              4.626834, -74.069096

    ));

    Adaptador miadaptador=new
    Adaptador(getApplicationContext(),Lista);

    listaDatos.setAdapter(miadaptador);

    listaDatos.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
        position, long id) {
            Datos obj=(Datos)parent.getItemAtPosition(position);

            Intent paso=new
            Intent(getApplicationContext(),DetalleActivity.class);
            paso.putExtra("objeto", (Serializable)obj);
            startActivity(paso);
        }
    });

}

@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_listas, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem opcion_menu){

    int id=opcion_menu.getItemId();

    if(id==R.id.info){
        ejecutar_info(null);
        return true;
    }

    return super.onOptionsItemSelected(opcion_menu);
}

public void ejecutar_info(View vista){

    Intent i=new Intent(Lista.this, InfoClase.class);
    startActivity(i);
}

```

```
}
```

14. Lista_Carreras

```
package com.example.jarblack.uccv01;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.TextView;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

/**
 * Created by Jonathan on 21/11/2017.
 * Esta clase es una actividad diseñada para ejecutar el layout de lista
 * de programas con su respectiva informacion y organizacion
 */

public class Lista_Carreras extends AppCompatActivity {

    private Toolbar toolbar;
    ListView listaDatos;

    List<Carreras> lstUser=new ArrayList<Carreras>();
    DbHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.lista_carreras);

        dbHelper=new DbHelper(getApplicationContext());
        dbHelper.createDatabase();

        toolbar=(Toolbar) findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);

        listaDatos=(ListView) findViewById(R.id.lstCarreras);
        lstUser=dbHelper.getAllUsers();

        Adaptador2 miadaptador=new
```

```

Adaptador2(getApplicationContext(),lstUser);

        listaDatos.setAdapter(miadaptador);

        listaDatos.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                Carreras
obj=(Carreras)parent.getItemAtPosition(position);

                Intent paso=new
Intent(getApplicationContext(),Calculadora Creditos.class);
                paso.putExtra("objeto", (Serializable)obj);
                startActivity(paso);
            }
        });

        @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_listas, menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem opcion_menu) {

            int id=opcion_menu.getItemId();

            if(id==R.id.info){
                ejecutar_info(null);
                return true;
            }

            return super.onOptionsItemSelected(opcion_menu);
        }

        public void ejecutar_info(View vista){

            Intent i=new Intent(Lista_Carreras.this, InfoClase.class);
            startActivity(i);
        }

    }
}

```