

## **Anexo VI - Código**

Guía de puntos de interés de la Ciudad de Madrid

# 1. Índice

<b>Anexo VI - Código</b> .....	<b>1</b>
1. Índice .....	2
2. Paquete es.guia.interfaz .....	4
Ayuda.....	4
CalculoRuta .....	4
DescripcionPOI.....	8
Galeria .....	10
InformacionPOI .....	12
ListadoPOIs .....	14
MapaSituacion .....	16
Menus.....	18
POIUsuario.....	22
3. Paquete es.guia.beans.....	30
Bounds .....	30
Distance.....	30
Duration.....	30
End_location .....	31
Legs.....	31
Northeast .....	32
POI .....	32
Routes.....	33
Ruta.....	34
Southwest .....	34
Start_location.....	34
Steps.....	35
TipoPOI.....	36
4. Paquete es.guia.db .....	37
GuiaDatabaseHelper .....	37
5. Paquete es.guia.db.adapters.....	40
ImageAdapter .....	40
POIAdapter .....	41
TipoPOIAdapter.....	46
6. Paquete es.guia.maps.gps .....	49
GeoUpdateHandler.....	49
7. Paquete es.guia.maps.overlay .....	53
PlacesItemizedOverlay .....	53
PlacesOverlay .....	55
UserItemizedOverlay .....	56
UserOverlay .....	57
8. Paquete es.guia.util.....	59
XMLReaderHelper .....	59
9. Layouts .....	61
ayuda .....	61
detalle_poi .....	62
galeria.....	63
gestionar_poi_usuario.....	64
info_previa .....	64
listado_pois.....	65
main .....	65
mapa_situacion .....	66
poi_item .....	66



## 2. Paquete es.guia.interfaz

### Ayuda

---

```
package es.guia.interfaz;

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

public class Ayuda extends Activity {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ayuda);
    }
}
```

### CalculoRuta

---

```
package es.guia.interfaz;

import java.io.*;
import java.net.*;
import java.util.*;

import android.app.Activity;
import android.content.Context;
import android.location.*;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.webkit.WebView;
import android.widget.*;

import com.google.gson.*;
import com.google.gson.stream.*;

import es.guia.beans.*;

public class CalculoRuta extends Activity {

    final String mimetype = "text/html; charset=iso-8859-1";
    final String encoding = "UTF-8";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ruta);
    }
}
```

```

        Log.d("CalculoRuta", "llego al calculo de ruta");

        String longitudDestino=
        (String) (getIntent().getExtras()).get("longitudDestino");
        String latitudDestino=
        (String) (getIntent().getExtras()).get("latitudDestino");

        Log.d("longitudDestino", longitudDestino);
        Log.d("latitudDestino", latitudDestino);

        //TextView texto=(TextView) findViewById(R.id.lecturaJSON);
        WebView texto=(WebView) findViewById(R.id.webview);

        //Calculo posición actual
        LocationManager locationManager = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);

        //Posición actual
        Criteria criteria=new Criteria();
        String
        bestProvider=locationManager.getBestProvider(criteria, true);
        Location
        loc=locationManager.getLastKnownLocation(bestProvider);

        if(loc==null) Log.e("CalculoRuta", "La localizacion actual es
        nula. El proveedor de GPS podría no estar operativo.");

        Log.d("calculoposicion", "finaliza el calculo de posicion");
        Log.d("INFORMACION", "Comienza el proceso de conexion ...");

        try{
            //URL url= new
            URL("http://maps.google.com/maps/api/directions/json?origin=40.4138,-
            3.6903&destination=40.4138,-
            3.692386&sensor=true&mode=walking&language=es");
            Log.d("INFORMACION", "Preparando conexion ...");
            URL url= new
            URL("http://maps.google.com/maps/api/directions/json?origin="
                +loc.getLatitude()+","+loc.getLongitude()

            +"&destination="+latitudDestino+","+longitudDestino
                +"&sensor=true&mode=walking&language=es");

            URLConnection urlCon=url.openConnection();
            Log.d("INFORMACION", "Conexion abierta ...");
            InputStream in=urlCon.getInputStream();

            StringBuffer store=new StringBuffer();
            DataInputStream dataIn=new DataInputStream(in);

            int b=in.read();
            while(b!=-1){
                store.append((char)b);
                b=in.read();
            }

            dataIn.close();

            Toast.makeText(getBaseContext(), "Finalizado calculo de
            ruta" , Toast.LENGTH_LONG).show();
        }
    }
}

```

```

StringBuffer txt=new StringBuffer();
txt.append("<html><head><style>" +
    "div.paso {" +
    "background-color: #95abb7;" +
    "float:center;" +
    "margin-top: 2px;margin-right: 4px;" +
    "color: #ffffff;border-radius: 1em 4em 1em
4em;" +
    "font-family: verdana;" +
    "font-size: 11;" +
    "text-align: left; " +
    "border-radius: 5px;" +
    "-moz-border-radius: 5px;}"+
    "div.cabecera {" +
    "background-color: #0e4b6a;" +
    "float:center;" +
    "margin-top: 2px;margin-right: 4px;" +
    "color: #ffffff;border-radius: 1em 4em 1em
4em;" +
    "font-family: arial;" +
    "font-size: 13;" +
    "text-align: left; " +
    "border-radius: 5px;" +
    "-moz-border-radius: 5px;}"+
    "</style>" +
    "<body bg-color=\"#f2f3f5\">");

//Parseo de la respuesta JSon
Gson gson=new Gson();
Ruta ruta=gson.fromJson(store.toString(), Ruta.class);

Routes route=(Routes) (ruta.getRoutes()).get(0);
List<Legs> legs=route.getLegs();

String stDest=null;

for(int i=0;i<legs.size();i++){
    Legs leg=(Legs)legs.get(i);

    stDest=leg.getEnd_address();
    txt.append("<div class=\"cabecera\">");
    txt.append("De:
<br/><b><i>" +leg.getStart_address()+"</i></b><br/> a:<br/>
<b><i>" +leg.getEnd_address()+"<i></b><br/><br/>");
    txt.append("<hr/>");
    txt.append("<center><b>Distancia del recorrido:</b>
"+leg.getDistance().getText()+"</center><br/></div>");

    List<Steps> pasos=leg.getSteps();

    for(int k=0;k<pasos.size();k++){
        Steps step=(Steps)pasos.get(k);

        //System.out.println(step.getDistance().getText()+"
"+step.getDuration().getText()+" "+step.getHtml_instructions());
        txt.append("<div class=\"paso\">");
        txt.append("<p>");
        txt.append(step.getDistance().getText()+"

```

```

"+step.getDuration().getText()+"
"+step.getHtml_instructions()+"<br/>");

        if(step.getHtml_instructions().indexOf("derecha")!=-1){
            txt.append("<center><img
src=\"file:///android_asset/right-arrow.png\" /></center>");
        }else
if(step.getHtml_instructions().indexOf("izquierda")!=-1){
            txt.append("<center><img
src=\"file:///android_asset/left-arrow.png\" /></center>");
        }else
if(step.getHtml_instructions().indexOf("Cont")!=-1 ||
step.getHtml_instructions().indexOf("recto")!=-1){
            txt.append("<center><img
src=\"file:///android_asset/go-arrow.png\" /></center>");

        }
        txt.append("</p>");
        txt.append("</div>");

    }

}

txt.append("<div class=\"cabecera\">");
txt.append("Ha llegado a su destino: <br/><br/>");
txt.append("<b><i>"+stDest+"<i></b><br/></div>");

//

//Log.d("LECTURA", store.toString());

txt.append("<br/></body></html>");
//texto.loadData(txt.toString(),mimetype,encoding);

final String mimeType = "text/html";
final String encoding = "utf-8";
final String html = "<h1>Header</h1><p>Custom
HTML</p><p><img src=\"file:///android_asset/right-arrow.png\" /></p>";

texto.loadDataWithBaseURL("fake://not/needed",txt.toString() ,
mimeType, encoding, "");

//texto.setTextColor(3);
//texto.setText("hola");

} catch (Exception e){
    Log.e("TENGO ERRORES",e.toString());
}

}

}

```

## DescripcionPOI

---

```
package es.guia.interfaz;

import es.guia.beans.*;
import es.guia.db.adapters.POIAAdapter;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;

public class DescripcionPOI extends Activity implements
OnClickListener {

    private ImageView imgCabecera=null;
    private POI poi=null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.detalles_poi);

        String id=(String) (getIntent().getExtras().get("id"));

        POIAAdapter poiAdapter=new POIAAdapter(getApplicationContext());
        poiAdapter.open();

        poi=poiAdapter.getPOI(id);

        imgCabecera=(ImageView) findViewById(R.id.icon);
        imgCabecera.setImageResource(Menus.hImagenes.get(poi.getImagen()));

        TextView titulo=(TextView) findViewById(R.id.titulo);
        titulo.setText(poi.getNombre());

        TextView cuerpo=(TextView) findViewById(R.id.cuerpo);
        cuerpo.setText(poi.getDesc_larga());

        Toast.makeText(getApplicationContext(), poi.getNombre(),
        Toast.LENGTH_LONG).show();
    }

    public void onClick(View v){

        Log.d("=====", "Llegooooo");
        Log.d("DescripcionPIO", String.valueOf(v.getTag()));
    }
}
```



```

        Log.d("Longitud: ", poi.getLongitud());
        Log.d("Latitud: ", poi.getLatitud());

        Intent intento=new Intent(this, CalculoRuta.class);

        intento.putExtra("longitudDestino", poi.getLongitud());
        intento.putExtra("latitudDestino", poi.getLatitud());

        startActivity(intento);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_descripcion_poi, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        Intent intento=null;

        switch (item.getItemId()) {
            case R.id.desc_llegar:

                intento=new Intent(DescripcionPOI.this,
CalculoRuta.class);
                intento.putExtra("longitudDestino",
poi.getLongitud());
                intento.putExtra("latitudDestino",
poi.getLatitud());

                break;
            case R.id.desc_galeria:
                intento=new
Intent(DescripcionPOI.this, Galeria.class);
                intento.putExtra("id", poi.getId());
                break;
            case R.id.desc_info:
                intento=new
Intent(DescripcionPOI.this, InformacionPOI.class);
                intento.putExtra("id", poi.getId());
                break;
            default:
                return super.onOptionsItemSelected(item);
        }

        startActivity(intento);

        return true;
    }
}

```

## Galeria

---

```
package es.guia.interfaz;

import es.guia.db.adapters.ImageAdapter;
import android.app.Activity;
import android.content.Context;
import android.content.res.TypedArray;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.AdapterView.OnItemClickListener;

import java.util.*;

public class Galeria extends Activity {

    private Gallery gallery;
    private ImageView imgView;

    /* private Integer[] Imgid = {
        Menus.hImagenes.get("fraguavulcano_velazquez_b"),
        R.drawable.hilanderas_velazquez_b,
        R.drawable.meninas_velazquez_b,
        R.drawable.rendicionbreda_velazquez_b,
        R.drawable.venusespejo_velazquez_b,
        R.drawable.esopo_velazquez_b
    };*/

    private Integer[] Imgid=null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.galeria);

        //Recuperando la galeria de imágenes del punto de interes
        String id=(String)(getIntent().getExtras().get("id"));
        Log.i("Guia","Recuperando galeria del punto: "+id);

        ImageAdapter imageAdapter=new
ImageAdapter(getApplicationContext());
        imageAdapter.open();

        Vector<String> galeria=imageAdapter.getGalleryImages(id);
        //for(int i=0;i<galeria.size();i++) Log.i("Guia","Recuperada
"+galeria.elementAt(i));
        Imgid=new Integer[galeria.size()];
        for(int i=0;i<galeria.size();i++)
Imgid[i]=Menus.hImagenes.get(galeria.elementAt(i));

        imgView = (ImageView) findViewById(R.id.ImageView01);
        imgView.setImageResource(Imgid[0]);
    }
}
```

```

        gallery = (Gallery) findViewById(R.id.galeriaobras);
        gallery.setAdapter(new AddImgAdp(this));

        gallery.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(AdapterView parent, View v, int
position, long id) {
                imageView.setImageResource(Imgid[position]);
            }
        });
    }

    public class AddImgAdp extends BaseAdapter {
        int GalItemBg;
        private Context cont;

        public AddImgAdp(Context c) {
            cont = c;
            TypedArray typArray =
obtainStyledAttributes(R.styleable.GalleryTheme);
            GalItemBg =
typArray.getResourceId(R.styleable.GalleryTheme_android_galleryItemBac
kground, 0);
            typArray.recycle();
        }

        public int getCount() {
            return Imgid.length;
        }

        public Object getItem(int position) {
            return position;
        }

        public long getItemId(int position) {
            return position;
        }

        public View getView(int position, View convertView, ViewGroup
parent) {
            ImageView imageView = new ImageView(cont);

            imageView.setImageResource(Imgid[position]);
            imageView.setLayoutParams(new Gallery.LayoutParams(80, 70));
            imageView.setScaleType(ImageView.ScaleType.FIT_XY);
            imageView.setBackgroundResource(GalItemBg);

            return imageView;
        }
    }
}

```

## InformacionPOI

```
package es.guia.interfaz;

import es.guia.beans.*;
import es.guia.db.adapters.POIAdapter;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;

public class InformacionPOI extends Activity implements
OnClickListener {

    private ImageView imgCabecera=null;
    private POI poi=null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.detalles_poi);

        String id=(String) (getIntent().getExtras()).get("id");

        POIAdapter poiAdapter=new POIAdapter(getApplicationContext());
        poiAdapter.open();

        poi=poiAdapter.getPOI(id);

        imgCabecera=(ImageView) findViewById(R.id.icon);
        imgCabecera.setImageResource(Menus.hImagenes.get(poi.getImagen()));

        TextView titulo=(TextView) findViewById(R.id.titulo);
        titulo.setText(poi.getNombre());

        String precio=poi.getPrecio();
        String horario=poi.getHorario();

        if(!precio.equals("No Aplica")) precio=precio+" €";

        String contenido="\n\nPrecio: "+precio+"\n"+"Horario de
apertura: "+horario;

        TextView cuerpo=(TextView) findViewById(R.id.cuerpo);
        cuerpo.setText(contenido);

        Toast.makeText(getApplicationContext(), poi.getNombre(),
Toast.LENGTH_LONG).show();
```

```

    }

    public void onClick(View v) {

        Log.d("=====", "Llegooooo");
        Log.d("InformacionPIO", String.valueOf(v.getTag()));

        Log.d("Longitud: ", poi.getLongitud());
        Log.d("Latitud: ", poi.getLatitud());

        Intent intento=new Intent(this, CalculoRuta.class);

        intento.putExtra("longitudDestino", poi.getLongitud());
        intento.putExtra("latitudDestino", poi.getLatitud());

        startActivity(intento);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_informacion_poi, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        Intent intento=null;

        switch (item.getItemId()) {
            case R.id.desc_llegar:

                intento=new Intent(InformacionPOI.this,
CalculoRuta.class);
                intento.putExtra("longitudDestino",
poi.getLongitud());
                intento.putExtra("latitudDestino",
poi.getLatitud());

                break;
            case R.id.desc_galeria:
                intento=new
Intent(InformacionPOI.this, Galeria.class);
                intento.putExtra("id", poi.getId());
                break;
            case R.id.desc_info:

                break;
            default:
                return super.onOptionsItemSelected(item);
        }

        startActivity(intento);

        return true;
    }
}

```

## ListadoPOIs

---

```
package es.guia.interfaz;

import java.util.*;

import android.os.*;
import android.util.*;
import android.view.*;
import android.widget.*;
import android.content.*;
import android.app.ListActivity;

import es.guia.beans.*;
import es.guia.db.adapters.*;

public class ListadoPOIs extends ListActivity {
    private List<POI> m_locals = null;
    private IconListAdapter m_adapter;
    private POIAdapter adapterPOI;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.listado_pois);

        m_locals=new ArrayList<POI>();

        this.m_adapter = new IconListAdapter(this,
R.layout.poi_item, m_locals);
        setListAdapter(this.m_adapter);

        getPuntosDisponibles();
    }

    @Override
    protected void onItemClick(ListView l, View v, int
position, long id) {
        // Local local = (Local) l.getItemAtPosition(position);

        // Toast.makeText(this, local.getLocalName(),
        // // Toast.LENGTH_LONG).show();
        //Intent intent = new Intent(IconListView.this,
Museos.class);
        //startActivity(intent);

        Intent intento=new
Intent(ListadoPOIs.this,DescripcionPOI.class);
        intento.putExtra("id", (m_locals.get(position)).getId());

        startActivity(intento);

        //Toast.makeText(getApplicationContext(),
(m_locals.get(position)).getNombre(), Toast.LENGTH_LONG).show();
    }
}
```

```

private void getPuntosDisponibles () {

    try {
        adapterPOI=new POIAdapter (getApplicationContext ());

        adapterPOI.open ();
        m_locals= adapterPOI.getAllPOIs ();

    } catch (Exception e) {
        Log.e ("BACKGROUND_PROC", e.getMessage ());
    }

    if (m_locals != null && m_locals.size () > 0) {
        for (int i=0;i<m_locals.size ();i++)
            m_adapter.add (m_locals.get (i));
    }

    m_adapter.notifyDataSetChanged ();

}

public class IconListViewAdapter extends ArrayAdapter<POI> {

    private List<POI> items;

    public IconListViewAdapter (Context context, int
textViewResourceId, List<POI> items) {
        super (context, textViewResourceId, items);
        this.items = items;
    }
    @Override
    public View getView (int position, View convertView,
ViewGroup parent) {
        View v = convertView;
        if (v == null) {
            LayoutInflater vi =
(LayoutInflater) getSystemService (Context.LAYOUT_INFLATER_SERVICE);
            v = vi.inflate (R.layout.poi_item, null);
        }

        //Carga de elementos
        POI o = items.get (position);
        if (o != null) {

            //poblamos la lista de elementos

            TextView tt = (TextView)
v.findViewById (R.id.row_toptext);
            ImageView im = (ImageView)
v.findViewById (R.id.icon);

            if (im!= null) {

                im.setImageResource (Menus.hImagenes.get (o.getImagen ()));
            }
            if (tt != null) {
                tt.setText (o.getNombre ());
            }
        }
    }
}

```

```

        }
        return v;
    }
}

```

## MapaSituacion

```

package es.guia.interfaz;

import java.util.*;

import android.content.Context;
import android.content.Intent;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

import com.google.android.maps.*;

import es.guia.db.*;
import es.guia.db.adapters.*;
import es.guia.beans.*;
import es.guia.interfaz.*;
import es.guia.maps.gps.*;
import es.guia.maps.overlay.*;

public class MapaSituacion extends MapActivity {
    private static MapView mapView;
    private MapController mapController;
    private List <Overlay> mapOverlays;
    private LocationManager locationManager;
    private GeoUpdateHandler miGPS=null;

    public void onCreate(Bundle savedInstanceState) {
        Log.i("Guia", "Creando MapaSituacion");
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapa_situacion);

        mapView=(MapView) findViewById(R.id.mapview);
        mapView.displayZoomControls(true);
        mapOverlays=mapView.getOverlays();
        mapController = mapView.getController();
        mapController.setZoom(17);

        updatePosition();
    }
    @Override

```



```

protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}

public void onResume () {
    super.onResume ();
    updatePosition ();
}

private void updatePosition () {
    miGPS= new
GeoUpdateHandler (this, mapController, mapView, mapOverlays, this);
    locationManager = (LocationManager)
getSystemService (Context.LOCATION_SERVICE);

    if (locationManager.isProviderEnabled (LocationManager.GPS_PROVIDE
R)) {
        locationManager.requestLocationUpdates (LocationManager.GPS_PROVI
DER, 5000, 5, miGPS);
    } else {
        Toast.makeText (getBaseContext (), "Proveedor de
servicio GPS no encontrado", Toast.LENGTH_SHORT).show ();
    }

    //Posición actual
    Criteria criteria=new Criteria ();
    String
bestProvider=locationManager.getBestProvider (criteria, true);
    Location
currentPos=locationManager.getLastKnownLocation (bestProvider);

    if (currentPos==null) return;

    Toast.makeText (getBaseContext (), "Localizacion actual:
"+currentPos.getLatitude ()+" "+currentPos.getLongitude (),
Toast.LENGTH_SHORT).show ();
    miGPS.onLocationChanged (currentPos);
}

public void onPause () {
    super.onPause ();
    locationManager.removeUpdates (miGPS);
    Log.i ("Guia", "onPause: Eliminando el listener del GPS");
}

@Override
public boolean onCreateOptionsMenu (Menu menu) {
    MenuInflater inflater = getMenuInflater ();
    inflater.inflate (R.menu.menu_principal, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected (MenuItem item) {
    return true;
}

```

```

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.i("Guia", "onDestroy: Eliminando el listener del GPS");
    //locationManager.removeUpdates(miGPS);
    if(miGPS!=null) miGPS=null;
    System.gc();

    /* final Intent poke = new Intent();
       poke.setClassName("com.android.settings",
       "com.android.settings.widget.SettingsAppWidgetProvider");
    // $NON-NLS-1$ // $NON-NLS-2$
       poke.addCategory(Intent.CATEGORY_ALTERNATIVE);
       poke.setData(Uri.parse("3")); // $NON-NLS-1$
       getApplicationContext().sendBroadcast(poke);*/

    //Coordenadas de prueba iniciales: longitud: -3,6903
    latitud: 40,4138

}
}

```

## Menus

---

```

package es.guia.interfaz;

import java.io.*;
import java.util.*;
import java.lang.reflect.Field;

import org.xmlpull.v1.XmlPullParser;

import com.google.android.maps.MapActivity;

import es.guia.db.*;
import es.guia.beans.*;
import es.guia.db.adapters.*;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.res.AssetManager;
import android.content.res.Resources;
import android.content.res.XmlResourceParser;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.TextView;

```

```

import es.guia.util.*;

public class Menu extends Activity {
    public static final String APP_BASE_DIR="appBaseDir";
    public static final String FOTOGRAFIA="dirFotografia";
    public static final String BACKUP="backup";
    public static final String XML="xml";

    public static Hashtable <String, Integer> hImagenes = new
Hashtable <String, Integer>();
    public static Hashtable <String,String> hResourcePath = new
Hashtable <String, String>();

    private TextView texto;
    //private ImageView imgLogotipo;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        //texto=(TextView) findViewById(R.id.texto);

        //Carga las Hashtable de imágenes
        directorios();

        try{
            configuracion();
        } catch (Exception e) {
            Log.e("Guia",e.toString());
        }

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_principal, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        //Bundle b=new Bundle();
        //Context contexto=getApplicationContext();

        Intent intento=new Intent(Menu.this, PasoParametro.class);

        switch (item.getItemId()) {
            case R.id.MnuOpcl:
                intento=new Intent(Menu.this, MapaSituacion.class);
                break;
        }
    }
}

```

```

        case R.id.MnuOpc2:
            intento=new Intent (Menus.this, POIUsuario.class);

            break;
        case R.id.MnuOpc5:
            //texto.setText ("Opcion 3 pulsada!");
            //b.putCharSequence ("tipoPOI", "1");
            //intento.putExtra ("tipoPOI", "3");
            intento=new Intent (Menus.this,Ayuda.class);
            break;
        case R.id.MnuOpc4:
            intento=new Intent (Menus.this,ListadoPOIs.class);

            break;
        default:
            return super.onOptionsItemSelected(item);
    }

    startActivity(intento);

    return true;
}

/*
 * Mapea los objetos drawables para poder recuperarlos por nombre.
 */

private void directorios () {
    Class resources = R.drawable.class;
    Field[] fields = resources.getFields ();

    try{
        for (Field field : fields) {

            hImagenes.put (field.getName (), field.getInt (field));
        }
    } catch (Exception e) {
        Log.e ("Guia", e.getMessage ());
    }

    Enumeration<String> en=hImagenes.keys ();
    while (en.hasMoreElements ()) {
        String key=en.nextElement ();
        Log.d (key, String.valueOf (hImagenes.get (key)));
    }
}

/**
 * Recrea la configuración necesaria, a nivel de propiedades y directorios, para que el aplicativo funcione
 *
 * @throws IOException
 */

private void configuracion () throws IOException {
    //Lectura de propiedades

    Resources assetResources=this.getResources ();
    AssetManager assetManager=assetResources.getAssets ();
}

```

```

        Log.i("Guia", "Leyendo propiedades de la aplicación.");
        Properties properties=null;
        try{
            InputStream
in=assetManager.open("propiedades.properties");
            properties=new Properties();
            properties.load(in);
        }catch(Exception e){
            Log.e("Guia", e.toString());
        }

        Log.i("Guia", "Inicialización de propiedades
completada.");
        Log.i("Guia", "Recreando configuración de la guia.");

        String
appBaseDir=Environment.getExternalStorageDirectory()+properties.getPro
perty(APP_BASE_DIR);

        Log.i("Guia", "Directorio base: "+appBaseDir);

        File fAppDir=new File(appBaseDir);

        if(!fAppDir.exists()){
            fAppDir.mkdir();
            Log.d("Guia", "Creado directorio: "+appBaseDir);

            new
File(appBaseDir+properties.getProperty(FOTOGRAFIA)).mkdir();
            new
File(appBaseDir+properties.getProperty(BACKUP)).mkdir();
            new
File(appBaseDir+properties.getProperty(XML)).mkdir();
        }

        hResourcePath=new Hashtable<String, String>();

        hResourcePath.put(APP_BASE_DIR, appBaseDir);
        hResourcePath.put(FOTOGRAFIA,
appBaseDir+properties.getProperty(FOTOGRAFIA));
        hResourcePath.put(BACKUP,
appBaseDir+properties.getProperty(BACKUP));
        hResourcePath.put(XML,
appBaseDir+properties.getProperty(XML));

        //getResources().getXml(id)

        Log.i("Guia", "La configuración se ha recreado
correctamente.");
        Log.d("Guia", "Fotografía:
"+hResourcePath.get(FOTOGRAFIA));
        Log.d("Guia", "XML: "+hResourcePath.get(XML));
    }

    /**
     * Comprobación del estado de la SDCard
     * @return
     */
    private boolean checkExternalMedia(){

```

```

        boolean mExternalStorageAvailable = false;
        boolean mExternalStorageWriteable = false;
        String state = Environment.getExternalStorageState();

        if (Environment.MEDIA_MOUNTED.equals(state)) {
            // We can read and write the media
            mExternalStorageAvailable =
mExternalStorageWriteable = true;
        } else if
(Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
            // We can only read the media
            mExternalStorageAvailable = true;
            mExternalStorageWriteable = false;
        } else {
            // Something else is wrong. It may be one of many
other states, but all we need
            // to know is we can neither read nor write
            Log.i("Guia", "State="+state+" Not good");
            mExternalStorageAvailable =
mExternalStorageWriteable = false;
        }

        Log.i("Guia", "Available="+mExternalStorageAvailable+"Writeable="
+mExternalStorageWriteable+" State"+state);
        return (mExternalStorageAvailable &&
mExternalStorageWriteable);
    }
}

```

## POIUsuario

---

```

package es.guia.interfaz;

import java.io.*;
import java.util.*;

import es.guia.beans.*;
import es.guia.maps.gps.*;
import es.guia.db.adapters.*;

import android.location.Location;
import android.media.MediaScannerConnection;
import android.net.*;
import android.util.Log;
import android.view.*;
import android.widget.*;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.graphics.drawable.BitmapDrawable;

```

```

import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import
android.media.MediaScannerConnection.MediaScannerConnectionClient;

public class POIUsuario extends Activity implements
View.OnFocusChangeListener{
    private static final int TAKE_PICTURE = 1;
    private static final int SELECT_PICTURE = 2;

    boolean fotoTomada=false;
    int posSeleccionada=0;

    String name=null;
    ImageView previsualizar=null;
    Bitmap foto=null;

    POIAdapter poiAdapter=null;
    static String[] SITIOS;
    List<POI> lista=null;

    AutoCompleteTextView nombre=null;
    TextView descripcion=null;

    /*
     * Método de creación de la actividad.
     * @see android.app.Activity#onCreate(android.os.Bundle)
     */

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.gestionar_poi_usuario);

        //Inicialización de posición seleccionada
        posSeleccionada=-1;

        poiAdapter=new POIAdapter(getApplicationContext());
        poiAdapter.open();

        autocomplete();

        //Previsualización de imagen
previsualizar=(ImageView) findViewById(R.id.poi_previsualizar);
        previsualizar.setDrawingCacheEnabled(false);

        //

        if(savedInstanceState!=null){
            name=savedInstanceState.getString("name");
            foto=BitmapFactory.decodeFile(name);
            int w = foto.getWidth();
            int h = foto.getHeight();
            // Matriz contenedora de imagen
            Matrix mtx = new Matrix();
            mtx.postRotate(90);

```

```

        // Rotación de Bitmap
        foto = Bitmap.createBitmap(foto, 0, 0, w, h,
mtx, true);
        BitmapDrawable bmd = new BitmapDrawable(foto);
        previsualizar.setImageDrawable(bmd);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_gestionar_punto_usuario,
menu);
    return true;
}

/*
 * Control de acciones pulsadas.
 * @see
android.app.Activity#onOptionsItemSelected(android.view.MenuItem)
 */
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    Intent intento = null;

    switch (item.getItemId()) {
        case R.id.usuario_poi_camara:
            intento=new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);

            //Configuración de ruta

            name=Environment.getExternalStorageDirectory()+"/test.jpg";

            Uri output = Uri.fromFile(new File(name));

            intento.putExtra(MediaStore.EXTRA_OUTPUT,output);
            startActivityForResult(intento,TAKE_PICTURE);
            return true;

        case R.id.usuario_poi_guardar:

            if(nombre.getText().length()==0){
                Toast.makeText(getBaseContext(), "Debe
introducir un nombre para el punto.", Toast.LENGTH_SHORT).show();
                return true;
            }if(descripcion.getText().length()==0){
                Toast.makeText(getBaseContext(), "Debe
introducir una descripcion para el punto.",
Toast.LENGTH_SHORT).show();
                return true;
            }else if(name==null){
                Toast.makeText(getBaseContext(), "Debe
hacer una fotografia al punto o seleccionar desde la galeria.",
Toast.LENGTH_SHORT).show();
                return true;
            }
    }
}

```



```

    }

    POI poi=new POI ();
    poi.setNombre (nombre.getText ().toString ());

    poi.setDesc_breve (descripcion.getText ().toString ());

    poi.setDesc_larga (descripcion.getText ().toString ());

    Location loc=new
GeoUpdateHandler (this) .getCurrentLocation ();

    poi.setLatitud (String.valueOf (loc.getLatitude ()));

    poi.setLongitud (String.valueOf (loc.getLongitude ()));

    poi.setTipoPOI ("4");

    try{
        poi.setImagen (saveToFile (foto));
        Log.d ("Guia", "Foto insertada
correctamente en directorio fotografico.");
    } catch (IOException io) {
        Log.e ("Guia", "Error copiando foto
seleccionada de galeria:\n"+io.toString ());
    }

    Log.d ("Guia", poi.toString ());

    poiAdapter.open ();
    poiAdapter.insertUserPOI (poi);

    Toast.makeText (getBaseContext (), "El punto ha
lado guardado", Toast.LENGTH_SHORT) .show ();
    onCreate (null); //Llamada a inicializar la
actividad

    break;

    case R.id.usuario_poi_borrar:
        //intento=new
Intent (Menus.this, ListadoPOIs.class);
        if (posSeleccionada != -1) {
            Log.i ("Guia", "Borrando el punto:
"+lista.get (posSeleccionada) .getNombre ());

            poiAdapter.open ();

            poiAdapter.delete (lista.get (posSeleccionada));

            try{

                eraseFile (lista.get (posSeleccionada) .getImagen ());
            } catch (IOException io) {
                Log.e ("Guia", "Error borrando imagen
asociada: "+io.toString ());
            }

            posSeleccionada=-1;
            Toast.makeText (getBaseContext (), "El
punto ha sido borrado", Toast.LENGTH_SHORT) .show ();

```

```

onCreate(null); //Llamada a inicializar
la actividad
    }else
        Toast.makeText(getBaseContext(),
"Introduzca el nombre del punto a borrar en Nombre del Punto",
Toast.LENGTH_SHORT).show();

        break;
    case R.id.usuario_poi_galeria:
        intento = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.INTERNAL_CONTENT_URI);
        //Configuración de ruta

        name=Environment.getExternalStorageDirectory()+"/test.jpg";
        startActivityForResult(intento, SELECT_PICTURE);

    default:
        return super.onOptionsItemSelected(item);
}

return true;
}

@Override
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
    Log.d("POIUsuario---
===== && =====>", "Vuelvo de
la llamada: requestCode =" + requestCode + " resultCode=" + resultCode);
    switch (requestCode) {
        case TAKE_PICTURE:

            fotoTomada=true;
            if (data != null) {
                if (data.hasExtra("data")) {
                    Log.d("Guia", "TAKE PICTURE con
extra");

                    foto=(Bitmap) data.getParcelableExtra("data");
                    int w = foto.getWidth();
                    int h = foto.getHeight();
                    // Setting post rotate to 90
                    Matrix mtx = new Matrix();
                    mtx.postRotate(90);
                    // Rotating Bitmap
                    foto = Bitmap.createBitmap(foto, 0,
0, w, h, mtx, true);

                    BitmapDrawable bmd = new
BitmapDrawable(foto);

                    previsualizar.setImageDrawable(bmd);
                    /*try{
                        saveToFile(foto);
                    }catch(IOException io){
                        Log.e("CameraTrial", "Error
comprimiendo: "+io.toString());
                    }*/
                }
            }
        }else{
            Log.d("Guia", "NO Tiene extra");
            foto=BitmapFactory.decodeFile(name);

```

```

        int w = foto.getWidth();
        int h = foto.getHeight();
        // Setting post rotate to 90
        Matrix mtx = new Matrix();
        mtx.postRotate(90);
        // Rotating Bitmap
        foto = Bitmap.createBitmap(foto, 0, 0, w,
h, mtx, true);
        BitmapDrawable bmd = new
BitmapDrawable(foto);

        previsualizar.setImageDrawable(bmd);

        /*try{
            saveToFile(foto);
        }catch(IOException io){
            Log.e("Guia", "Error comprimiendo:
"+io.toString());
        }*/

        new MediaScannerConnectionClient() {
            private MediaScannerConnection msc =
null; {
                msc = new
MediaScannerConnection(getApplicationContext(), this); msc.connect();
            }
            public void
onMediaScannerConnected() {
                msc.scanFile(name, null);
            }
            public void onScanCompleted(String
path, Uri uri) {
                msc.disconnect();
            }
        };
        break;
        case SELECT_PICTURE:
            Uri selectedImage = data.getData();
            InputStream is;
            try {
                is =
getContentResolver().openInputStream(selectedImage);
                BufferedInputStream bis = new
BufferedInputStream(is);
                foto = BitmapFactory.decodeStream(bis);
                previsualizar.setImageBitmap(foto);
            } catch (FileNotFoundException e) {}

            break;
        default:
    }
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    if(name!=null) outState.putString("name", name);
    poiAdapter.close();
    super.onSaveInstanceState(outState);
}

```

```

    }

    @Override
    protected void onPause () {
        super.onPause ();
    }

    @Override
    protected void onDestroy () {
        poiAdapter.close ();
        super.onDestroy ();
    }

    private String saveToFile (Bitmap b) throws IOException {
        String
        fileName=String.valueOf(Calendar.getInstance().getTimeInMillis())+".jpg";

        Log.i("Guia", "path:
"+Menus.hResourcePath.get (Menus.FOTOGRAFIA));
        OutputStream fOut = null;
        File file = new
File (Menus.hResourcePath.get (Menus.FOTOGRAFIA), fileName);
        fOut = new FileOutputStream (file);

        b.compress (Bitmap.CompressFormat.JPEG, 70, fOut);
        fOut.flush ();
        fOut.close ();

        MediaStore.Images.Media.insertImage (getContentResolver (), file.getAbsolutePath (), file.getName (), file.getName ());

        return fileName;
    }

    private void eraseFile (String name) throws IOException {
        Log.i ("Guia", "Borrando:
"+Menus.hResourcePath.get (Menus.FOTOGRAFIA) + "/" + name);

        File f=new
File (Menus.hResourcePath.get (Menus.FOTOGRAFIA) + "/" + name);
        f.delete ();

        Log.d ("Guia", "Imagen asociada borrada");
    }

    /*
    * Consulta para autocompletar puntos de usuario.
    *
    */
    private void autocomplete () {
        Log.d ("Guia", "Construyendo estructura para
autocompletar");

        lista=poiAdapter.getAllUserPOIs ();

```

```

        SITIOS=new String[lista.size()];
        for(int i=0;i<lista.size();i++) SITIOS[i] =
(lista.get(i)).getNombre();

        // for(int i=0;i<SITIOS.length;i++) Log.d("Guia",
SITIOS[i]);

        //Opciones de autocompletar
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line, SITIOS);
        nombre =
(AutoCompleteTextView) findViewById(R.id.poi_usuario_nombre_auto);
        nombre.setAdapter(adapter);

        nombre.setOnFocusChangeListener(this);
        Log.d("Guia", "Construcción de autocompletar finalizada");

        descripcion=(TextView) findViewById(R.id.poi_usuario_desc);
    }

    public void onFocusChange(View arg0, boolean arg1) {
        Log.d("Guia", "Entro");
        if(lista!=null && SITIOS!=null){
            Log.d("Guia", nombre.getText().toString());
            int i=0;
            for(i=0;i<SITIOS.length;i++){

                if(SITIOS[i].equals(nombre.getText().toString())){

                    descripcion.setText(lista.get(i).getDesc_breve());
                    posSeleccionada=i;

                    //Cargar la foto
                    try{
                        Log.d("Guia", "cargado la
imagen:
"+Menus.hResourcePath.get(Menus.FOTOGRAFIA)+"/"+lista.get(i).getImagen
());
                        FileInputStream in = new
FileInputStream(Menus.hResourcePath.get(Menus.FOTOGRAFIA)+"/"+lista.ge
t(i).getImagen());

                        foto = BitmapFactory.decodeStream(in);
                        previsualizar.setImageBitmap(foto);

                    }catch(Exception io){

                        Log.d("Guia", io.toString());}

                        break;
                    }
                }
            }
        }
    }
}

```

### 3. Paquete es.guia.beans

#### Bounds

---

```
package es.guia.beans;

import java.util.List;

public class Bounds{
    private Northeast northeast;
    private Southwest southwest;

    public Northeast getNortheast(){
        return this.northeast;
    }
    public void setNortheast(Northeast northeast){
        this.northeast = northeast;
    }
    public Southwest getSouthwest(){
        return this.southwest;
    }
    public void setSouthwest(Southwest southwest){
        this.southwest = southwest;
    }
}
```

#### Distance

---

```
package es.guia.beans;

import java.util.List;

public class Distance{
    private String text;
    private Number value;

    public String getText(){
        return this.text;
    }
    public void setText(String text){
        this.text = text;
    }
    public Number getValue(){
        return this.value;
    }
    public void setValue(Number value){
        this.value = value;
    }
}
```

#### Duration

---

```
package es.guia.beans;

import java.util.List;
```

```

public class Duration{
    private String text;
    private Number value;

    public String getText(){
        return this.text;
    }
    public void setText(String text){
        this.text = text;
    }
    public Number getValue(){
        return this.value;
    }
    public void setValue(Number value){
        this.value = value;
    }
}

```

## End\_location

---

```

package es.guia.beans;

import java.util.List;

public class End_location{
    private Number lat;
    private Number lng;

    public Number getLat(){
        return this.lat;
    }
    public void setLat(Number lat){
        this.lat = lat;
    }
    public Number getLng(){
        return this.lng;
    }
    public void setLng(Number lng){
        this.lng = lng;
    }
}

```

## Legs

---

```

package es.guia.beans;

import java.util.List;

public class Legs{
    private Distance distance;
    private Duration duration;
    private String end_address;
    private End_location end_location;
    private String start_address;
    private Start_location start_location;
}

```

```

    private List<Steps> steps;
    // private List<Via_waypoint> via_waypoint;

    public Distance getDistance() {
        return this.distance;
    }
    public void setDistance(Distance distance) {
        this.distance = distance;
    }
    public Duration getDuration() {
        return this.duration;
    }
    public void setDuration(Duration duration) {
        this.duration = duration;
    }
    public String getEnd_address() {
        return this.end_address;
    }
    public void setEnd_address(String end_address) {
        this.end_address = end_address;
    }
    public End_location getEnd_location() {
        return this.end_location;
    }
    public void setEnd_location(End_location end_location) {
        this.end_location = end_location;
    }
    public String getStart_address() {
        return this.start_address;
    }
    public void setStart_address(String start_address) {
        this.start_address = start_address;
    }
    public Start_location getStart_location() {
        return this.start_location;
    }
    public void setStart_location(Start_location start_location) {
        this.start_location = start_location;
    }
    public List<Steps> getSteps() {
        return this.steps;
    }
    public void setSteps(List<Steps> steps) {
        this.steps = steps;
    }
    /* public List<Via_waypoint> getVia_waypoint() {
        return this.via_waypoint;
    }
    public void setVia_waypoint(List<Via_waypoint> via_waypoint) {
        this.via_waypoint = via_waypoint;
    }*/
}

```

## Northeast

---

## POI

---



---

## Routes

---

```
package es.guia.beans;

import java.util.List;

public class Routes{
    private Bounds bounds;
    private String copyrights;
    private List<Legs> legs;
    // private Overview_polyline overview_polyline;
    private String summary;
    // private List<Warnings> warnings;
    // private List<Waypoint_order> waypoint_order;

    public Bounds getBounds() {
        return this.bounds;
    }
    public void setBounds(Bounds bounds) {
        this.bounds = bounds;
    }
    public String getCopyrights() {
        return this.copyrights;
    }
    public void setCopyrights(String copyrights) {
        this.copyrights = copyrights;
    }
    public List<Legs> getLegs() {
        return this.legs;
    }
    public void setLegs(List<Legs> legs) {
        this.legs = legs;
    }
    /* public Overview_polyline getOverview_polyline() {
        return this.overview_polyline;
    }
    public void setOverview_polyline(Overview_polyline
overview_polyline) {
        this.overview_polyline = overview_polyline;
    }*/
    public String getSummary() {
        return this.summary;
    }
    public void setSummary(String summary) {
        this.summary = summary;
    }
    /* public List<Warnings> getWarnings() {
        return this.warnings;
    }
    public void setWarnings(List<Warnings> warnings) {
        this.warnings = warnings;
    }*/
    /* public List<Waypoint_order> getWaypoint_order() {
        return this.waypoint_order;
    }
    public void setWaypoint_order(List<Waypoint_order>
waypoint_order) {
```

```
        this.waypoint_order = waypoint_order;
    }*/
}
```

## Ruta

---

```
package es.guia.beans;

import java.util.List;

public class Ruta{
    private List<Routes> routes;
    private String status;

    public List<Routes> getRoutes(){
        return this.routes;
    }
    public void setRoutes(List<Routes> routes){
        this.routes = routes;
    }
    public String getStatus(){
        return this.status;
    }
    public void setStatus(String status){
        this.status = status;
    }
}
```

## Southwest

---

```
package es.guia.beans;

import java.util.List;

public class Southwest{
    private Number lat;
    private Number lng;

    public Number getLat(){
        return this.lat;
    }
    public void setLat(Number lat){
        this.lat = lat;
    }
    public Number getLng(){
        return this.lng;
    }
    public void setLng(Number lng){
        this.lng = lng;
    }
}
```

## Start\_location

---

```
package es.guia.beans;
```

```

import java.util.List;

public class Start_location{
    private Number lat;
    private Number lng;

    public Number getLat(){
        return this.lat;
    }
    public void setLat(Number lat){
        this.lat = lat;
    }
    public Number getLng(){
        return this.lng;
    }
    public void setLng(Number lng){
        this.lng = lng;
    }
}

```

## Steps

```

package es.guia.beans;

import java.util.List;

public class Steps{
    private Distance distance;
    private Duration duration;
    private End_location end_location;
    private String html_instructions;
    // private Polyline polyline;
    private Start_location start_location;
    private String travel_mode;

    public Distance getDistance(){
        return this.distance;
    }
    public void setDistance(Distance distance){
        this.distance = distance;
    }
    public Duration getDuration(){
        return this.duration;
    }
    public void setDuration(Duration duration){
        this.duration = duration;
    }
    public End_location getEnd_location(){
        return this.end_location;
    }
    public void setEnd_location(End_location end_location){
        this.end_location = end_location;
    }
    public String getHtml_instructions(){
        return this.html_instructions;
    }
    public void setHtml_instructions(String html_instructions){
        this.html_instructions = html_instructions;
    }
}

```

```

/*  public Polyline getPolyline(){
        return this.polyline;
    }
    public void setPolyline(Polyline polyline){
        this.polyline = polyline;
    }*/
    public Start_location getStart_location(){
        return this.start_location;
    }
    public void setStart_location(Start_location start_location){
        this.start_location = start_location;
    }
    public String getTravel_mode(){
        return this.travel_mode;
    }
    public void setTravel_mode(String travel_mode){
        this.travel_mode = travel_mode;
    }
}

```

## TipoPOI

```

package es.guia.beans;

public class TipoPOI {

    private String id;
    private String nombre;
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}

```

## 4. Paquete es.guia.db

### GuiaDatabaseHelper

```
package es.guia.db;

import java.util.*;

import es.guia.interfaz.Menus;
import es.guia.interfaz.R;
import android.content.Context;
import android.content.res.AssetManager;
import android.content.res.Resources;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import es.guia.util.*;
import es.guia.beans.*;

public class GuiaDatabaseHelper extends SQLiteOpenHelper {
    public static final String TIPO_POI="TIPO_POI"; //Tipología del
    punto de interés
    public static final String POI="POI"; //Puntos de interés de la
    aplicación
    public static final String GALERIA="GALERIA"; //Galeria
    relacionada con el punto de interés
    public static final String DATABASE_NAME="guia";
    public static final int DATABASE_VERSION=3;

    private Context context=null;

    public GuiaDatabaseHelper(Context context, String databaseName,
    CursorFactory factory, int version) {
        super(context, databaseName, factory, version);
        this.context=context;

        Log.d("Guia","Constructor GuiaDatabaseHelper");
    }

    public GuiaDatabaseHelper(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.context=context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        Log.d("Guia","Ejecutando el script de creacion");

        //Script completo de creación de BBDD.
        db.execSQL("CREATE TABLE \""+TIPO_POI+"\" (\\"id\" INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL , \\"NOMBRE\" VARCHAR NOT NULL
UNIQUE );");
        db.execSQL("CREATE TABLE \""+POI+"\" (\\"id\" INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL , \\"nombre\" VARCHAR NOT NULL ,
\\"latitud\" DOUBLE, \\"longitud\" DOUBLE, \\"imagen\" VARCHAR,
```

```

\"desc_breve\" VARCHAR, \"desc_larga\" VARCHAR, \"precio\"
VARCHAR, \"horario\" VARCHAR , tipo_poi INTEGER, FOREIGN KEY(tipo_poi)
REFERENCES TIPO_POI (id));");
        db.execSQL("CREATE TABLE \""+GALERIA+"\" (\"ID\" INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL , \"IMAGEN\" VARCHAR NOT NULL,
id_poi INTEGER, FOREIGN KEY(id_poi) REFERENCES ID_POI (id));");

        //Datos pre-almacenados
        db.execSQL("insert into "+TIPO_POI+" (nombre) values
('Museo')");
        db.execSQL("insert into "+TIPO_POI+" (nombre) values
('Monumento')");
        db.execSQL("insert into "+TIPO_POI+" (nombre) values
('Restaurante')");
        db.execSQL("insert into "+TIPO_POI+" (nombre) values
('Usuario')");

        //Puntos de interés catalogados
        if(context.getResources().getXml(R.xml.lista_pois)==null)
Log.i("Guia", "El xmlresources es nulo");
        XMLReaderHelper xmlHelper=new
XMLReaderHelper(context.getResources().getXml(R.xml.lista_pois));
        Vector<POI> listaPOIs=xmlHelper.getPOIs();

        for(int i=0;i<listaPOIs.size();i++){
            POI p=listaPOIs.elementAt(i);
            String sql="insert into "+POI+"
(nombre, longitud, latitud, imagen, desc_breve, desc_larga, tipo_poi, precio,
horario) values "
                +"('"+p.getNombre()+"'
                +",""+p.getLongitud()
                +",""+p.getLatitud()
                +",""+p.getImagen()+"'
                +",""+p.getDesc_breve()+"'
                +",""+p.getDesc_larga()+"'
                +",""+p.getTipoPOI()
                +",""+p.getPrecio()+"'
                +",""+p.getHorario()+"'
                +")";

            db.execSQL(sql);

            //Inserción de imágenes asociadas
            Vector<String> galeria=p.getImagenesGaleria();

            for(int k=0;k<galeria.size();k++){
                String imagen=galeria.elementAt(k);
                sql="insert into "+GALERIA+" (imagen,id_poi)
values "
                    +"('"+imagen+"'
                    +",""+(i+1)+")";

                db.execSQL(sql);
                Log.i("Guia", "insertada la imagen: "+imagen+"
asociada al POI: "+p.getNombre());
            }
        }
    }
}

```

```
        Log.d("Guia", "Se ha recreado el modelo correctamente CON
descripciones largas y sitios personalizados");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int version, int
newVersion) {
        Log.d("Guia", "Existe un cambio de versión en BBDD.
Preparado para actualizar.");

        // TODO Auto-generated method stub
        db.execSQL("DROP TABLE IF EXISTS "+TIPO_POI);
        db.execSQL("DROP TABLE IF EXISTS "+POI);
        db.execSQL("DROP TABLE IF EXISTS "+GALERIA);

        Log.d("DEBUG", "Base de datos borrada");

        onCreate(db);
    }
}
```

## 5. Paquete es.guia.db.adapters

### ImageAdapter

```
package es.guia.db.adapters;

import java.util.*;

import android.content.ContentValues;
import android.content.Context;

import es.guia.db.*;
import es.guia.beans.*;
import android.database.*;
import android.database.sqlite.*;
import android.util.Log;

public class ImageAdapter {

    //Declaración de campos de la entidad TipoPOI

    public static final String ID = "ID";
    public static final String IMAGEN = "IMAGEN";
    public static final String POI = "ID_POI";

    final String DATABASE_TABLE="GALERIA";

    //
    private GuiaDatabaseHelper guiaDatabaseHelper;
    private SQLiteDatabase db;
    private Context context;

    public ImageAdapter(Context context){
        this.context=context;
        open();
        Log.i("Guia", "Conexión lista para ser utilizada");
    }

    public ImageAdapter open() throws SQLException{
        guiaDatabaseHelper=new GuiaDatabaseHelper(context);
        db=guiaDatabaseHelper.getWritableDatabase();

        return this;
    }

    public void close() throws SQLException{
        db.close();
        guiaDatabaseHelper.close();
    }

    /**
     * Consulta la galería de imágenes de un punto dado.
     */
}
```



```

        */

        public Vector<String> getGalleryImages(String idPOI) throws
SQLException{
            Log.d("Guia","Consultando el punto con id: "+idPOI);

            String[] params=new String[]{idPOI};

            Cursor cursor=db.query(DATABASE_TABLE, new
String[]{IMAGEN}, "id_poi=?", params, null, null, null);

            Vector<String> listaImagenes=new Vector<String>();

            if(!cursor.moveToNext()) return listaImagenes;
            do{
                String imagen=cursor.getString(0);
                Log.d("Guia","Recuperada imagen: "+imagen);
                listaImagenes.add(imagen);
            }while(cursor.moveToNext());

            cursor.close();
            close();

            return listaImagenes;
        }

        public long insert(String imagen, String idPOI) throws
SQLException{
            ContentValues content=createContent(imagen,idPOI);
            return db.insert(DATABASE_TABLE, null, content);
        }

        /**
         * Crea un ContentValues para operar con los métodos CRUD de este adaptador.
         */

        private ContentValues createContent(String imagen, String
idPoi){
            ContentValues content = new ContentValues();
            content.put("IMAGEN", imagen);
            content.put("ID_POI", idPoi);

            return content;
        }
    }
}

```

## POIAdapter

```

package es.guia.db.adapters;

import java.util.*;

import android.content.ContentValues;
import android.content.Context;

```

```

import es.guia.db.*;
import es.guia.beans.*;
import android.database.*;
import android.database.sqlite.*;
import android.util.Log;

public class POIAdapter {

    //Declaración de campos de la entidad POI

    final String ID = "ID";
    final String NOMBRE = "NOMBRE";
    final String LONGITUD = "LONGITUD";
    final String LATITUD = "LATITUD";
    final String TIPO_POI = "TIPO_POI";
    final String IMAGEN = "IMAGEN";
    final String DESC_BREVE="DESC_BREVE";
    final String DESC_LARGA="DESC_LARGA";
    final String PRECIO="PRECIO";
    final String HORARIO="HORARIO";

    final String DATABASE_TABLE="POI";

    //
    private GuiaDatabaseHelper guiaDatabaseHelper;
    private SQLiteDatabase db;
    private Context context;

    public POIAdapter(Context context){
        this.context=context;
        open();
        Log.i("Guia", "Conexión lista para ser utilizada");
    }

    public POIAdapter open() throws SQLException{
        guiaDatabaseHelper=new GuiaDatabaseHelper(context);
        db=guiaDatabaseHelper.getWritableDatabase();

        return this;
    }

    public void close() throws SQLException{
        db.close();
        guiaDatabaseHelper.close();
    }

    public long insertUserPOI(POI poi) throws SQLException{
        return
insert(poi.getNombre(),poi.getLongitud(),poi.getLatitude(),poi.getImag
n(),poi.getDesc_breve(),poi.getDesc_breve(),poi.getTipoPOI(),"");
    }

    public long insert(String nombre, String longitud, String
latitud, String imagen, String descBreve, String descLarga,String
tipo,String precio, String horario) throws SQLException{
        ContentValues
content=createContent(nombre,longitud,latitud,imagen,descBreve,descLar

```

```

ga, tipo, precio, horario);
        return db.insert(DATABASE_TABLE, null, content);
    }

    public long delete(POI poi) throws SQLException{
        Log.d("Guia", "Borrando el punto de usuario con id:
"+poi.getId());
        String[] params={poi.getId()};
        return db.delete(DATABASE_TABLE, "id=?", params);
    }

    /*
    * Obtiene una lista de puntos de un tipo determinado.
    *
    * @param type - Tipo a recuperar
    * @return List - Lista de puntos recuperados
    */

    public List<POI> getPOIsByType(String type) throws SQLException{
        Log.i("[GUIA]", "Recuperando los puntos de interés");
        String[] params={type};

        Cursor cursor=db.query(DATABASE_TABLE, new
String[]{ID, NOMBRE, LONGITUD, LATITUD, IMAGEN, DESC_BREVE, TIPO_POI},
"tipo_poi=?", params, null, null, null);
        List<POI> listaPOIs=new ArrayList<POI>();

        if(!cursor.moveToFirst()) return listaPOIs;

        do{
            String id=cursor.getString(0);
            String nombre=cursor.getString(1);
            String longitud=cursor.getString(2);
            String latitud=cursor.getString(3);
            String imagen=cursor.getString(4);
            String desc_breve=cursor.getString(5);
            String tipo=cursor.getString(6);

            POI poi=new POI(id, nombre, longitud, latitud, tipo);
            poi.setImagen(imagen);
            poi.setDesc_breve(desc_breve);

            listaPOIs.add(pei);

            Log.d("POIAdapter", "Añadiendo "+nombre+" a la
lista");

        }while (cursor.moveToNext());

        cursor.close();

        Log.i("[GUIA]", "La lista de puntos de interés se ha
generado correctamente.");

        return listaPOIs;
    }

    public List<POI> getAllPOIs() throws SQLException{

```

```

        Log.i("[GUIA]", "Recuperando los puntos de interés");
        String[] params={"4"};

        Cursor cursor=db.query(DATABASE_TABLE, new
String[]{ID,NOMBRE, LONGITUD, LATITUD, IMAGEN, DESC_BREVE, TIPO_POI},
"tipo_POI<>?", params, null, null, null);
        List<POI> listaPOIs=new ArrayList<POI>();

        if(!cursor.moveToFirst()) return listaPOIs;

        do{
            String id=cursor.getString(0);
            String nombre=cursor.getString(1);
            String longitud=cursor.getString(2);
            String latitud=cursor.getString(3);
            String imagen=cursor.getString(4);
            String desc_breve=cursor.getString(5);
            String tipo=cursor.getString(6);

            POI poi=new POI(id,nombre,longitud,latitud,tipo);
            poi.setImagen(imagen);
            poi.setDesc_breve(desc_breve);

            listaPOIs.add(poi);

            Log.d("POIAdapter", "Añadiendo "+nombre+" a la
lista");

        }while(cursor.moveToNext());

        cursor.close();

        Log.i("[GUIA]", "La lista de puntos de interés se ha
generado correctamente.");

        return listaPOIs;

    }

    /*
    * Devuelve todos los puntos registrados por el usuario.
    */

    public List<POI> getAllUserPOIs() throws SQLException{
        Log.i("[GUIA]", "Recuperando los puntos de interés de
usuario");

        String[] params=new String[]{"4"};

        Cursor cursor=db.query(DATABASE_TABLE, new
String[]{ID,NOMBRE, LONGITUD, LATITUD, IMAGEN, DESC_BREVE, DESC_LARGA, TIPO_
POI}, "tipo_poi=?", params, null, null, null);
        List<POI> listaPOIs=new ArrayList<POI>();

        if(!cursor.moveToFirst()) return listaPOIs;

        do{
            String id=cursor.getString(0);
            String nombre=cursor.getString(1);

```

```

        String longitud=cursor.getString(2);
        String latitud=cursor.getString(3);
        String imagen=cursor.getString(4);
        String desc_breve=cursor.getString(5);
        String desc_larga=cursor.getString(6);
        String tipo=cursor.getString(7);

        POI poi=new POI(id,nombre,longitud,latitud,tipo);
        poi.setImagen(imagen);
        poi.setDesc_breve(desc_breve);
        poi.setDesc_larga(desc_larga);

        listaPOIs.add(pei);

        Log.d("POIAdapter", "Añadiendo punto de usuario
"+nombre+" a la lista");

    }while(cursor.moveToNext());

    cursor.close();

    Log.i("[GUIA]","La lista de puntos de interés se ha
generado correctamente.");

    return listaPOIs;

}

/**
 * Consulta el punto de interés por identificador.
 */

public POI getPOI(String id) throws SQLException{
    Log.d("[POIAdapter]","Consultando el punto con id: "+id);

    String[] params=new String[]{id};

    POI poi=null;
    Cursor cursor=db.query(DATABASE_TABLE, new
String[]{ID,NOMBRE, LONGITUD, LATITUD, IMAGEN, DESC_BREVE, DESC_LARGA, TIPO_
POI,PRECIO,HORARIO}, "id=?", params, null, null, null);

    if(cursor.moveToFirst()){

        String nombre=cursor.getString(1);
        String longitud=cursor.getString(2);
        String latitud=cursor.getString(3);
        String imagen=cursor.getString(4);
        String desc_breve=cursor.getString(5);
        String desc_larga=cursor.getString(6);
        String tipo=cursor.getString(7);
        String precio=cursor.getString(8);
        String horario=cursor.getString(9);

        poi=new POI(id,nombre,longitud,latitud,tipo);
        poi.setImagen(imagen);

```

```

        poi.setDesc_breve(desc_breve);
        poi.setDesc_larga(desc_larga);
        poi.setPrecio(precio);
        poi.setHorario(horario);
    }

    cursor.close();
    close();

    return poi;
}

/**
 * Crea un ContentValues para operar con los métodos CRUD de este adaptador.
 */

private ContentValues createContent(String nombre, String
longitud, String latitud, String imagen, String descBreve, String
descLarga, String tipo, String precio, String horario){
    ContentValues content = new ContentValues();
    content.put("NOMBRE", nombre);
    content.put("LONGITUD", longitud);
    content.put("LATITUD", latitud);
    content.put("IMAGEN", imagen);
    content.put("DESC_BREVE", descBreve);
    content.put("DESC_LARGA", descLarga);
    content.put("TIPO_POI", tipo);
    content.put("PRECIO", precio);
    content.put("HORARIO", horario);

    return content;
}

private ContentValues createContent(String id){
    ContentValues content=new ContentValues();
    content.put("ID", id);

    return content;
}
}

```

## TipoPOIAdapter

---

```

package es.guia.db.adapters;

import java.util.*;
import android.content.Context;

import es.guia.db.*;
import es.guia.beans.*;
import android.database.*;

```

```

import android.database.sqlite.*;
import android.util.Log;

public class TipoPOIAdapter {

    //Declaración de campos de la entidad TipoPOI

    public static final String ID = "ID";
    public static final String NOMBRE = "NOMBRE";

    final String DATABASE_TABLE="TIPO_POI";

    //
    private GuiaDatabaseHelper guiaDatabaseHelper;
    private SQLiteDatabase db;
    private Context context;

    public TipoPOIAdapter(Context context){
        this.context=context;
        open();
        Log.i("Guia", "Conexión lista para ser utilizada");
    }

    public TipoPOIAdapter open() throws SQLException{
        guiaDatabaseHelper=new GuiaDatabaseHelper(context);
        db=guiaDatabaseHelper.getWritableDatabase();

        return this;
    }

    public void close() throws SQLException{
        db.close();
        guiaDatabaseHelper.close();
    }

    /**
     * Obtiene la lista completa de tipos de puntos de interés.
     * @return List<TipoPOI> - Lista de tipos de punto
     */

    public List<TipoPOI> getAllTypes() throws SQLException{
        Log.i("[GUIA]", "Recuperando los tipos punto de interés");

        Cursor cursor=db.query(DATABASE_TABLE, new
String[]{ID,NOMBRE}, null, null, null, null, null);
        List<TipoPOI> listaPOIs=new ArrayList<TipoPOI>();

        if(!cursor.moveToNext()) return listaPOIs;
        do{
            String id=cursor.getString(0);
            String nombre=cursor.getString(1);

```

```

        TipoPOI tipo=new TipoPOI();
        tipo.setId(id);
        tipo.setNombre(nombre);

        listaPOIs.add(tipo);

        Log.d("TipoPOIAdapter", "Añadiendo "+nombre+" a la
lista");
    }while(cursor.moveToNext());

    cursor.close();

    Log.i("[GUIA]", "La lista de puntos de interés se ha
generado correctamente.");

    return listaPOIs;
}

/**
 * Consulta el punto de interés por identificador.
 */

public TipoPOI getTipoPOI(String id) throws SQLException{
    Log.d("[POIAdapter]", "Consultando el punto con id: "+id);

    String[] params=new String[]{id};

    TipoPOI tipo=null;
    Cursor cursor=db.query(DATABASE_TABLE, new
String[]{NOMBRE}, "id=?", params, null, null, null);

    if(cursor.moveToFirst()){

        String nombre=cursor.getString(1);

        tipo=new TipoPOI();
        tipo.setId(id);
        tipo.setNombre(nombre);
    }

    cursor.close();
    close();

    return tipo;
}
}

```



## 6. Paquete es.guia.maps.gps

### GeoUpdateHandler

---

```
package es.guia.maps.gps;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

import es.guia.beans.POI;
import es.guia.db.adapters.POIAdapter;
import es.guia.interfaz.Menus;
import es.guia.interfaz.R;
import android.content.Context;
import android.graphics.drawable.Drawable;
import android.location.*;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;
import android.app.*;

import es.guia.beans.*;
import es.guia.db.*;
import es.guia.db.adapters.*;

import java.util.*;

import es.guia.maps.overlay.*;

public class GeoUpdateHandler implements LocationListener {
    private MapController mapController;
    private GeoPoint position;
    private MapView mapView;
    private List<Overlay> mapOverlays;
    private OverlayItem userItem, placeItem;
    private UserOverlay userPicOverlay;
    private PlacesOverlay markerOverlay;
    private Context mContext;
    private Activity actividad;

    public GeoUpdateHandler(Context mContext, MapController
mapController, MapView mapView, List<Overlay>mapOverlays, Activity
actividad){
        this.mapController = mapController;
        this.mapView = mapView;
        this.mapOverlays=mapOverlays;
        this.mContext=mContext;
        this.actividad=actividad;
    }

    public GeoUpdateHandler(Activity actividad){
        this.actividad=actividad;
    }
}
```

```

public void onLocationChanged(Location location) {
    double lat = location.getLatitude();
    double lng= location.getLongitude();
    position = new GeoPoint((int) (lat*1E6), (int) (lng*1E6));

    mapOverlays=mapView.getOverlays();
    mapOverlays.clear();

    mapController.setCenter(position);
    mapController.setZoom(17);
    mapController.animateTo(position);
    mapView.setBuiltInZoomControls(true);

    //Obtención de dirección postal
    Location loc=getCurrentLocation();
    Geocoder geoCoder=new Geocoder(mContext);
    String calle="No disponible";
    try{
        List<Address>
    direccion=geoCoder.getFromLocation(loc.getLatitude(),
loc.getLongitude(), 1);
        if(direccion.size() !=0)
    calle=direccion.get(0).getAddressLine(0)+" "+direccion.get(0).getAddressLine(1);
        Log.d("Guia", calle);
    }catch(Exception e){Log.e("Guia", e.toString());}

    List<Overlay> mapOverlays = mapView.getOverlays();
    UserItemizedOverlay itemizedoverlay = new
UserItemizedOverlay(mContext.getResources().getDrawable(R.drawable.per
sona2),mContext,calle);

    //Listado de items del mapa

    //Item del usuario
    OverlayItem dondeEstoy=new OverlayItem(position,"Su
posicion:",calle);
    itemizedoverlay.addOverlay(dondeEstoy);

    //Recuperación de puntos de interés.

    TipoPOIAdapter tipoAdapter=new
TipoPOIAdapter(actividad);
    List<TipoPOI> listaTipos=tipoAdapter.getAllTypes();

    Log.d("GeoUpdateHandler","Antes de la consulta a los
tipos de punto");

    POIAdapter poiAdapter=new POIAdapter(actividad);
    poiAdapter.open();
    List<POI> listaPOI=poiAdapter.getAllPOIs();

    Log.d("GeoUpdateHandler","Consulta los tipos de punto:
"+listaTipos.size());

    for(int k=0;k<listaTipos.size();k++){

        Log.d("GeoUpdateHandler","=====");

```

```

        TipoPOI tipoPOI=listaTipos.get(k);

        Log.d("GeoUpdateHandler",tipoPOI.getId());
        Log.d("GeoUpdateHandler",tipoPOI.getNombre());

        poiAdapter=new POIAdapter(actividad);
        listaPOI=poiAdapter.getPOIsByType(tipoPOI.getId());
        Log.d("GeoUpdateHandler","Consultar por tipo:
"+tipoPOI.getId());

        if(listaPOI.size()==0) continue;

        //Generación de items de punto de interés
        String nombreTipo=tipoPOI.getNombre().toLowerCase();
        PlacesItemizedOverlay poiOverlay= new
PlacesItemizedOverlay(mContext.getResources().getDrawable(Menu.hImage
nes.get(nombreTipo)),mContext,actividad);

        for(int i=0;i<listaPOI.size();i++){
            POI p=listaPOI.get(i);
            Log.d("GeoUpdateHandler",p.getNombre());
            PlacesOverlay nuevoPunto=new PlacesOverlay(new
GeoPoint((int)(Double.parseDouble(p.getLatitude())*1E6),(int)(Double.pa
rseDouble(p.getLongitude())*1E6)),p.getTipoPOI(),p.getNombre(),p);
            poiOverlay.addOverlay(nuevoPunto);
        }

        mapOverlays.add(poiOverlay);
    }

    tipoAdapter.close();
    poiAdapter.close();

    //Añadir al mapa las distintas listas de puntos
    mapOverlays.add(itemizedoverlay);
}

public void onProviderDisabled(String arg0) {
    Toast.makeText(mContext,"GPS Desactivado",
Toast.LENGTH_SHORT).show();
}

public void onProviderEnabled(String arg0) {
    Toast.makeText(mContext,"GPS Activado",
Toast.LENGTH_SHORT).show();
}

public void onStatusChanged(String arg0, int arg1, Bundle arg2)
{
    Toast.makeText(mContext,"GPS Cambio de estado",
Toast.LENGTH_SHORT).show();
}

/**
 * Devuelve la localización actual en coordenadas geográficas
 * @return Location
 */

```

```
public Location getCurrentLocation(){
    //Calculo posición actual
    LocationManager locationManager = (LocationManager)
    actividad.getSystemService(Context.LOCATION_SERVICE);

    //Posición actual
    Criteria criteria=new Criteria();
    String
    bestProvider=locationManager.getBestProvider(criteria, true);
    Location
    loc=locationManager.getLastKnownLocation(bestProvider);

    return loc;
}
}
```

## 7. Paquete es.guia.maps.overlay

### PlacesItemizedOverlay

---

```
package es.guia.maps.overlay;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.util.Log;
import android.view.*;
import android.widget.*;

import com.google.android.maps.*;

import es.guia.beans.*;
import es.guia.interfaz.*;

import java.util.*;

public class PlacesItemizedOverlay extends
ItemizedOverlay<PlacesOverlay>{

    private ArrayList<PlacesOverlay> mOverlays = new
ArrayList<PlacesOverlay>();
    private Context mContext=null;
    private Activity actividad=null;

    public PlacesItemizedOverlay(Drawable arg0) {
        super(boundCenterBottom(arg0));
    }

    public PlacesItemizedOverlay(Drawable defaultMarker, Context
context, Activity actividad) {
        super(boundCenterBottom(defaultMarker));
        mContext = context;
        this.actividad= actividad;
    }

    @Override
    protected PlacesOverlay createItem(int i) {
        return (PlacesOverlay)mOverlays.get(i);
    }

    @Override
    public int size() {
        return mOverlays.size();
    }

    public void addOverlay(PlacesOverlay overlay) {
        mOverlays.add(overlay);
        populate();
    }
}
```

```

    }

    @Override
    protected boolean onTap(int index) {
        PlacesOverlay item =(PlacesOverlay) mOverlays.get(index);
        POI poi=item.getPOI();

        AlertDialog.Builder builder;
        AlertDialog alertDialog;

        LayoutInflater inflater = (LayoutInflater)
mContext.getSystemService(Activity.LAYOUT_INFLATER_SERVICE);
        View layout = inflater.inflate(R.layout.info_previa,
            (ViewGroup)
actividad.findViewById(R.id.layout_root));

        TextView text = (TextView) layout.findViewById(R.id.text);
        text.setText(poi.getDesc_breve());

        ImageView image = (ImageView) layout.findViewById(R.id.image);
        image.setImageResource(Menus.hImagenes.get(poi.getImagen()));

        builder = new AlertDialog.Builder(mContext);
        builder.setView(layout);

        //Manejador de opciones

        OpcionesHandler oHandler=new OpcionesHandler(poi,actividad);

        builder.setPositiveButton("Llegar",oHandler);

        builder.setNeutralButton("+Info",oHandler);
        builder.setNegativeButton("Cancelar",oHandler);

        alertDialog = builder.create();

        alertDialog.setTitle(poi.getNombre());
        alertDialog.show();

        return true;
    }

    /**
     * Se encarga de manejar las opciones de los botones de la
     * ventana de dialog.
     * @author Daniel Cortés
     */

    class OpcionesHandler implements
DialogInterface.OnClickListener{

        public static final int LLEGAR=-1;
        public static final int CANCEL=-2;
        public static final int INFO=-3;
    }

```

```

private POI poi;
private Activity actividad;

public OpcionesHandler(POI poi, Activity actividad){
    super();
    this.poi=poi;
    this.actividad=actividad;
}

public void onClick(DialogInterface dialog, int id) {

    Log.d("Id: ",String.valueOf(id));
    Intent intento=null;

    switch(id){
        case LLEGAR:
            Log.d("Longitud: ",poi.getLongitud());
            Log.d("Latitud: ",poi.getLatitud());

            intento=new Intent(actividad,
CalculoRuta.class);

            intento.putExtra("longitudDestino",
poi.getLongitud());
            intento.putExtra("latitudDestino",
poi.getLatitud());

            actividad.startActivity(intento);

            break;
        case INFO:
            Log.d("+Info", "Mas informacion");
            intento=new Intent(actividad,
DescripcionPOI.class);

            intento.putExtra("id",poi.getId());

            actividad.startActivity(intento);
            break;
        default:
            dialog.cancel();
    }
}
}
}
}

```

## PlacesOverlay

```

package es.guia.maps.overlay;

import java.util.ArrayList;

import com.google.android.maps.Geopoint;
import com.google.android.maps.ItemizedOverlay;
import com.google.android.maps.OverlayItem;

```

```

import android.app.AlertDialog;
import android.content.Context;
import android.graphics.drawable.Drawable;

import es.guia.beans.*;

public class PlacesOverlay extends OverlayItem{
    private POI poi;

    public PlacesOverlay(GeoPoint geo, String titulo, String
snippet, POI poi ){
        super(geo,titulo,snippet);
        this.poi=poi;
    }

    public POI getPOI(){
        return poi;
    }
}

```

## UserItemizedOverlay

```

package es.guia.maps.overlay;

import android.app.AlertDialog;
import android.content.Context;
import android.graphics.drawable.Drawable;
import android.widget.Toast;

import com.google.android.maps.*;
import java.util.*;

public class UserItemizedOverlay extends ItemizedOverlay{

    private ArrayList<OverlayItem> mOverlays = new
ArrayList<OverlayItem> ();
    private Context mContext=null;
    private String calle=null;

    public UserItemizedOverlay(Drawable arg0) {
        super(boundCenterBottom(arg0));
    }

    public UserItemizedOverlay(Drawable defaultMarker, Context
context) {
        super(boundCenterBottom(defaultMarker));
        mContext = context;
    }

    public UserItemizedOverlay(Drawable defaultMarker, Context
context, String calle) {
        super(boundCenterBottom(defaultMarker));
        mContext = context;
        this.calle=calle;
    }

    @Override

```



```

protected OverlayItem createItem(int i) {
    return mOverlays.get(i);
}

@Override
public int size() {
    return mOverlays.size();
}

public void addOverlay(OverlayItem overlay) {
    mOverlays.add(overlay);
    populate();
}

@Override
protected boolean onTap(int index) {
    OverlayItem item = mOverlays.get(index);
    AlertDialog.Builder dialog = new
AlertDialog.Builder(mContext);
    dialog.setTitle(item.getTitle());
    dialog.setMessage(calle);
    dialog.setIcon(es.guia.interfaz.R.drawable.icon);
    dialog.show();

    return true;
}
}

```

## UserOverlay

---

```

package es.guia.maps.overlay;

import java.util.ArrayList;

import com.google.android.maps.ItemizedOverlay;
import com.google.android.maps.OverlayItem;

import android.content.Context;
import android.graphics.drawable.Drawable;

public class UserOverlay extends ItemizedOverlay<OverlayItem> {

    private ArrayList<OverlayItem> mOverlays = new
ArrayList<OverlayItem>();
    private Context context;

    public UserOverlay(Drawable defaultMarker, Context context) {
        super(boundCenterBottom(defaultMarker));
        this.context = context;
    }

    public void updateOverlay(OverlayItem overlay) {
        mOverlays.clear();
        mOverlays.add(overlay);
    }
}

```

```
        populate();
    }

    public void addOverlay(OverlayItem overlay) {
        mOverlays.clear();
        mOverlays.add(overlay);
        populate();
    }

    @Override
    protected OverlayItem createItem(int i) {
        return (OverlayItem) mOverlays.get(i);
    }

    @Override
    public int size() {
        // TODO Auto-generated method stub
        return 0;
    }
}
```

## 8. Paquete es.guia.util

### XMLReaderHelper

```
package es.guia.util;

import java.util.Vector;

import org.xmlpull.v1.XmlPullParser;

import android.content.res.Resources;
import android.content.res.XmlResourceParser;
import android.util.Log;
import es.guia.beans.POI;
import es.guia.interfaz.R;

public class XMLReaderHelper {
    private XmlResourceParser xmlFile=null;

    public XMLReaderHelper(XmlResourceParser xmlFile){
        this.xmlFile=xmlFile;
    }

    public Vector<POI> getPOIs() {
        Vector<POI> listaPois=new Vector<POI>();

        //XmlResourceParser xmlFile =
resources.getXml(R.xml.lista_pois);

        try{
            xmlFile.next();//Get next parse event
            int eventType = xmlFile.getEventType(); //Get
current xml event i.e., START_DOCUMENT etc.

            POI poi=null;
            while (eventType != XmlPullParser.END_DOCUMENT) {

                if(eventType == XmlPullParser.START_DOCUMENT) {
                    Log.i("Guia", "Comienza el catalogo de puntos
de interés");
                }else if(eventType == XmlPullParser.START_TAG) {
                    String nameTag=xmlFile.getName();
                    if(nameTag.equals("poi")) {
                        String
nombre=xmlFile.getAttributeValue(0);
                        String
latitud=xmlFile.getAttributeValue(1);
                        String
longitud=xmlFile.getAttributeValue(2);
                        String
tipo=xmlFile.getAttributeValue(3);
                        String
precio=xmlFile.getAttributeValue(4);
                        String
horario=xmlFile.getAttributeValue(5);
```

```

        poi=new POI ();

        poi.setNombre (nombre);
        poi.setLongitud (longitud);
        poi.setLatitud (latitud);
        poi.setTipoPOI (tipo);
        poi.setPrecio (precio);
        poi.setHorario (horario);
    }else if (nameTag.equals ("imagen")) {
        xmlFile.next ();
        String imagen=xmlFile.getText ();
        Log.d ("Guia", "Imagen: "+imagen);
        poi.setImagen (imagen);
    }else if (nameTag.equals ("descBreve")) {
        xmlFile.next ();
        String descBreve=xmlFile.getText ();
        Log.d ("Guia", "DescBreve: "+descBreve);
        poi.setDesc_breve (descBreve);
    }else if (nameTag.equals ("descLarga")) {
        xmlFile.next ();
        String descLarga=xmlFile.getText ();
        Log.d ("Guia", "DescLarga: "+descLarga);
        poi.setDesc_larga (descLarga);
    }else if (nameTag.equals ("imagenGaleria")) {
        Log.i ("Guia", "PASO POR LA GALERIA DE
IMAGENES");

        xmlFile.next ();
        String imagenGaleria=xmlFile.getText ();
        Log.i ("Guia", imagenGaleria);
        poi.addImagenGaleria (imagenGaleria);
    }
    }else if (eventType==XmlPullParser.TEXT) {
        Log.d ("Guia", xmlFile.getText ());
    }else if (eventType==XmlPullParser.END_TAG) {
        if (xmlFile.getName ().equals ("poi")) {
            //Añadir a la lista
            Log.d ("Guia", "Cerrando poi:
"+poi.toString ());

            listaPois.add (poi);
        }
    }

    xmlFile.next ();
    eventType=xmlFile.getEventType ();
}
} catch (Exception e) {
    Log.e ("Guia", e.toString ());
}

Log.i ("Guia", "Resumen de puntos leidos");
Log.i ("Guia", "=====");
for (int i=0; i<listaPois.size (); i++) {
    POI a=listaPois.elementAt (i);
    Log.i ("Guia", a.toString ());
}

return listaPois;
}
}

```

## 9. Layouts

### ayuda

---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <TextView android:id="@+id/textView1"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceLarge"
  android:layout_width="wrap_content" android:text="Tipología de puntos
  de interés"></TextView>

  <LinearLayout
  android:orientation="horizontal"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content">
    <ImageView android:layout_height="wrap_content"
  android:src="@drawable/museo" android:id="@+id/imageView1"
  android:layout_width="wrap_content"></ImageView>
    <TextView android:id="@+id/textView2"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceSmall"
  android:layout_width="wrap_content" android:text="
  Museos"></TextView>

  </LinearLayout>

  <LinearLayout
  android:orientation="horizontal"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content">
    <ImageView android:layout_height="wrap_content"
  android:src="@drawable/monumento" android:id="@+id/imageView3"
  android:layout_width="wrap_content"></ImageView>
    <TextView android:id="@+id/textView3"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceSmall"
  android:layout_width="wrap_content" android:text="
  Monumentos"></TextView>

  </LinearLayout>

  <LinearLayout
  android:orientation="horizontal"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content">
    <ImageView android:layout_height="wrap_content"
  android:src="@drawable/usuario" android:id="@+id/imageView4"
  android:layout_width="wrap_content"></ImageView>
    <TextView android:id="@+id/textView4"
  android:layout_height="wrap_content"
  android:textAppearance="?android:attr/textAppearanceSmall"
  android:layout_width="wrap_content" android:text="      Punto de
  Usuario"></TextView>
```

```

</LinearLayout>

<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <TextView android:id="@+id/textView11"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_width="wrap_content" android:text="Catalogación de
    punto de usuario"></TextView>
    <TextView android:id="@+id/textView42"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:layout_width="wrap_content" android:text="Para catalogar un
    punto de usuario se deberán realizar las siguientes acciones:
    "></TextView>
    <TextView android:id="@+id/textView421"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:layout_width="wrap_content" android:text="
    1.) rellenar nombre y descripción
    "></TextView>
    <TextView android:id="@+id/textView423"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:layout_width="wrap_content" android:text="
    2.) Fotografiar o seleccionar fotografía de galeria
    "></TextView>
    <TextView android:id="@+id/textView424"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:layout_width="wrap_content" android:text="
    3.) Pulsar 'Guardar'"></TextView>
</LinearLayout>

</LinearLayout>

```

## detalle\_poi

```

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

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical"
>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="left">

        <ImageView
            android:id="@+id/icon"
            android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_marginRight="8dip"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/icon"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_centerVertical="true"
        android:textStyle="bold"
        android:textSize="14dip"
        android:id="@+id/titulo"
    />
</LinearLayout>

<ScrollView
    android:scrollbars="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_marginTop="1dip">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:textColor="#ffffff"
        android:isScrollContainer="true"
        android:id="@+id/cuerpo"
    />
</ScrollView>
</LinearLayout>

```

## galeria

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="left"
    >
    <Gallery
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/galeriaobras"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="left"
    />

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center">
        <ImageView android:id="@+id/ImageView01"
            android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    </LinearLayout>

```

```
</LinearLayout>
```

## gestionar\_poi\_usuario

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" android:weightSum="1">

    <TextView
        android:id="@+id/ges_poi_nombre"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Nombre Punto:"
    />
    <AutoCompleteTextView android:layout_height="wrap_content"
        android:id="@+id/poi_usuario_nombre_auto" android:text=""
        android:layout_width="match_parent">
        <requestFocus></requestFocus>
    </AutoCompleteTextView>

    <TextView
        android:id="@+id/ges_poi_descripcion"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Descripcion:"
    />
    <EditText android:layout_height="wrap_content"
        android:layout_width="match_parent" android:id="@+id/poi_usuario_desc"
        android:inputType="textMultiLine"></EditText>

    <TextView
        android:id="@+id/ges_poi_previsualizacion"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Previsualizacion:"
    />
    <ImageView android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:id="@+id/poi_previsualizar"></ImageView>

</LinearLayout>
```

## info\_previa

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

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/layout_root"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:padding="10dp"
    >
    <ImageView android:id="@+id/image"
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
    />
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#FFF"
    />
</LinearLayout>

```

## listado\_pois

```

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

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/settings"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ListView
        android:id="@+id/android:list"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />

</LinearLayout>

```

## main

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<!-- <TextView
    android:id="@+id/texto"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />

<EditText
    android:id="@+id/caja"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    />-->

    <ImageView
        android:id="@+id/logotipo"
        android:src="@drawable/logotipo"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"

```

```
    />  
</LinearLayout>
```

## mapa\_situacion

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  android:id="@+id/mapView"  
  android:orientation="vertical"  
  android:layout_width="fill_parent"  
  android:layout_height="fill_parent" >  
  
  <com.google.android.maps.MapView  
    android:id="@+id/mapview"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:clickable="true"  
    android:apiKey="0-cw0fiGv77pTNEIRPg_3wDQX1MnzZhjgQ8mgFA"  
  />  
</LinearLayout>
```

## poi\_item

```
<RelativeLayout  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  android:layout_width="fill_parent"  
  android:layout_height="?android:attr/listPreferredItemHeight"  
  android:padding="8dip">  
  
  <ImageView  
    android:id="@+id/icon"  
    android:layout_width="wrap_content"  
    android:layout_height="fill_parent"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentBottom="true"  
    android:layout_alignParentRight="true"  
    android:layout_marginRight="8dip"  
  />  
  
  <TextView  
    android:id="@+id/row_toptext"  
  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
  
    android:layout_toRightOf="@id/icon"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentTop="true"  
    android:layout_centerVertical="true"  
  
    android:singleLine="true"  
    android:ellipsize="marquee"/>  
</RelativeLayout>
```

## ruta

---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <WebView
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
  />
</LinearLayout>
```