

Prueba módulo 02

Introducción

El desarrollo de aplicaciones móviles supedita fundamentos de programación en particular si se trata de aplicaciones móviles basadas en Android es necesario tener sólidos conocimientos del lenguaje de programación Java y entendimiento del formato de datos sobre XML.

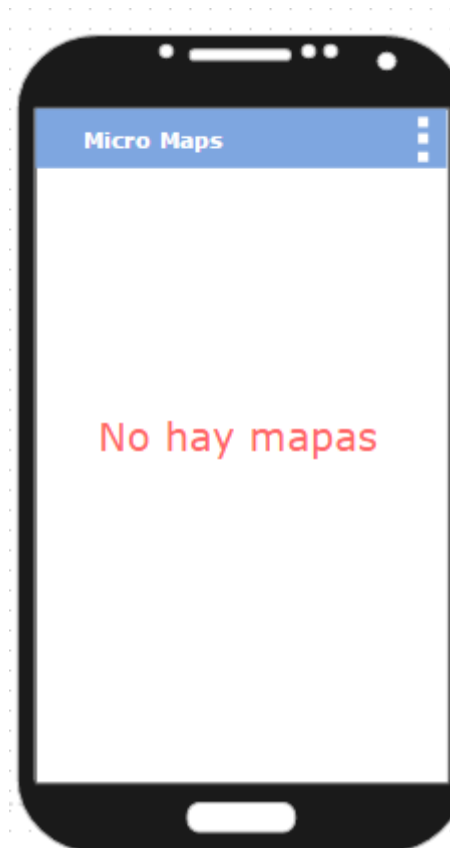
Requerimiento

Se necesita implementar una aplicación móvil llamada “Micro Maps” que tenga las siguientes funcionalidades:



1. Pantalla inicial

Pantalla inicial: Esta pantalla será la pantalla de presentación del aplicativo y deberá redirigir automáticamente a la pantalla número 2 luego de pasado 3 segundos.



2. Pantalla principal (sin mapas)

Pantalla principal: Esta pantalla será la pantalla donde se mostrar un listado de mapas en forma de ítems. Si no hay mapas grabados en el aplicativo mostrar un mensaje que diga **no hay mapas**.



3. Pantalla inicial haciendo click en menú

Pantalla inicial con click: Al hacer click en Agregar mapa se redirigirá a la pantalla 4 para agregar mapa.



4. Pantalla de agregar mapa (cargando)

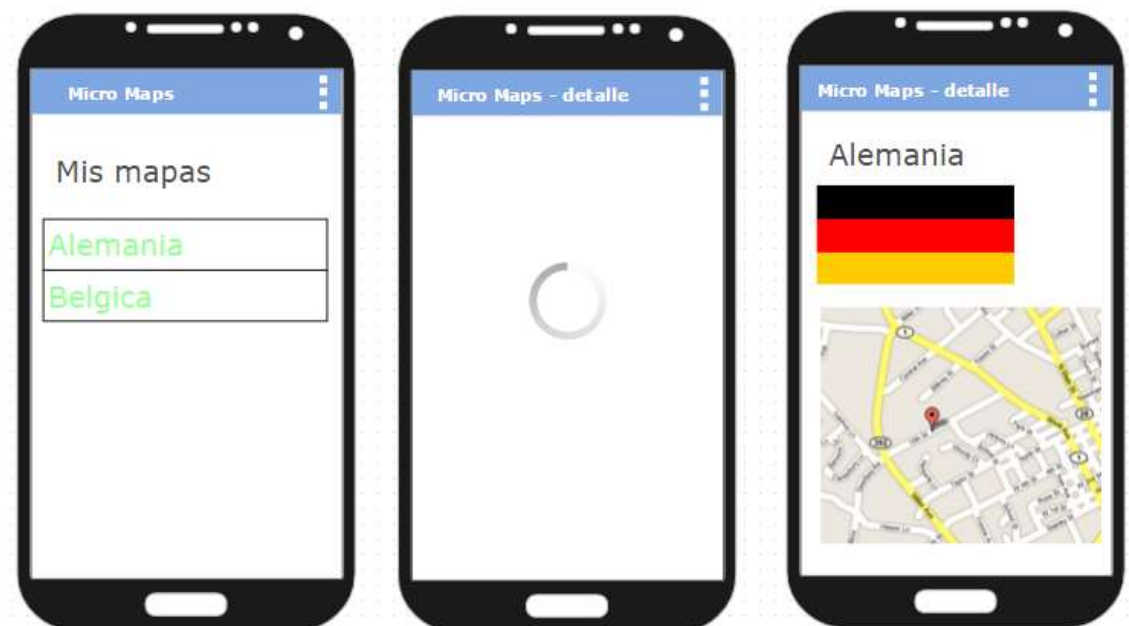
Pantalla de agregar mapa: Esta pantalla cargará un listado de países de la siguiente manera:

- Si no hay países en la base de datos local se deberá de cargar los países del servicio web (S01)
- Al cargar los datos de los países se deberá de guardar en un tabla con SQLite los siguientes campos
 - Nombre
 - Capital
 - Código de país (`altSpellings[0]`)
- Si hay países, se deberá cargar de la base de datos SQLite y ya no consultar el servicio web



5. Pantalla de agregar mapa (después de descargada la información en una base de datos SQLite en el aplicativo)

- Los países se mostraran en una lista
- Los países que se encuentren agregados en el aplicativo se mostraran con color verde
- Para buscar países se deberá crear una barra de búsqueda (revisar el punto de **Recursos de código y dependencias**)
- Para agregar o quitar un país al aplicativo se deberá presionar sobre el ítem y luego mostrar un menú contextual donde estén las opciones de agregar o quitar países localmente para la pantalla principal



6. Al ingresar a la pantalla principal nuevamente se deberán de observar solo los países agregados y poder entrar a ver su detalle como se muestra

- Se deberá mostrar en el detalle un google map con la localización del país
- Se deberá mostrar la bandera de país de manera dinámica
- Recordar usar loaders para ocultar la carga

Indicación de los servicios web:

Servicio (S01)

Para obtener la lista de todos los países:

<https://restcountries.eu/rest/v1/all>

Servicio (S02)

Para obtener la información del país

<http://www.geognos.com/api/en/countries/info/<CODIGO DE PAIS>.json>

Ejemplo:

- <http://www.geognos.com/api/en/countries/info/PE.png> (Si es PE (Perú))
- <http://www.geognos.com/api/en/countries/flag/US.png> (Si es US (USA))

Servicio (S03)

Para obtener la imagen de las banderas

<http://www.geognos.com/api/en/countries/flag/<CODIGO DE PAIS>.png>

Ejemplo:

- <http://www.geognos.com/api/en/countries/flag/PE.png> (Si es PE (Perú))
- <http://www.geognos.com/api/en/countries/flag/US.png> (Si es US (USA))

Mayor información del servicio geognos

<http://www.geognos.com/geo/en/world-countries-API.html>

Iconos e imágenes estáticas

- Imagen de la lupa:
- https://www.iconfinder.com/icons/728952/explore_find_look_magnifier_search_view_zoom_icon#size=32
- Imagen del icono del aplicativo en todos sus tamaños:
- https://www.iconfinder.com/icons/334635/devices_foursquare_fs_function_geo-designed_maps_mobile_network_social_icon

Recursos de código y dependencias

Usar las siguientes dependencias

```
compile 'com.android.support:appcompat-v7:23.0.1'
compile 'com.github.kevinsawicki:http-request:6.0'
compile 'com.google.code.gson:gson:2.3.1'
compile 'com.google.android.gms:play-services:6.5.87'
```

Agregar una imagen de URL en una ImageView

```
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.widget.ImageView;

import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;

public class ImageLoadTask extends AsyncTask<Void, Void, Bitmap> {

    private String url;
    private ImageView imageView;

    public ImageLoadTask(String url, ImageView imageView) {
        this.url = url;
        this.imageView = imageView;
    }

    @Override
    protected Bitmap doInBackground(Void... params) {
        try {
            URL urlConnection = new URL(url);
            HttpURLConnection connection = (HttpURLConnection)
urlConnection
                .openConnection();
            connection.setDoInput(true);
            connection.connect();
            InputStream input = connection.getInputStream();
            Bitmap myBitmap = BitmapFactory.decodeStream(input);
            return myBitmap;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Bitmap result) {
        super.onPostExecute(result);
        imageView.setImageBitmap(result);
    }

}
```

Código para crear una barra de búsqueda

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="wrap_content" >

<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="Enter search key" />

<ImageButton
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:src="@drawable/search"
    android:layout_centerVertical="true"
    android:layout_margin="5dp"
    android:text="Button"/>

</RelativeLayout>
```

Donde src se cambia por la imagen de la lupa

Crear el siguiente layout para el menu contextual (R.menu.menu_context_item)

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

<TextView
    android:id="@+id/LblMensaje"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="Menú contextual (mantenga presionado aquí)" />

</LinearLayout>
```

Crear un menú contextual (Sobre escribir los siguientes métodos en el activity de agregado de paises)

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_context_item, menu);
}
```

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {
        case R.id.CtxLblOpc1:
            lblMensaje.setText("Etiqueta: Opcion 1 pulsada!");
            return true;
        case R.id.CtxLblOpc2:
            lblMensaje.setText("Etiqueta: Opcion 2 pulsada!");
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

Luego usar el siguiente método en el mismo activity para agregar el menu contextual (agregar este método una vez cargado el listado de países desde el web service si es que no hay países en la BD SQLite local)

```
this.registerForContextMenu(elListViewCreado);
```