



**Universidad de El Salvador
Facultad de Ingeniería y Arquitectura
Escuela de Ingeniería en Sistemas**

Programación en Dispositivos Móviles

GUIA DE LABORATORIO N° 5

Conocimiento de librerías especiales Android

Objetivos:

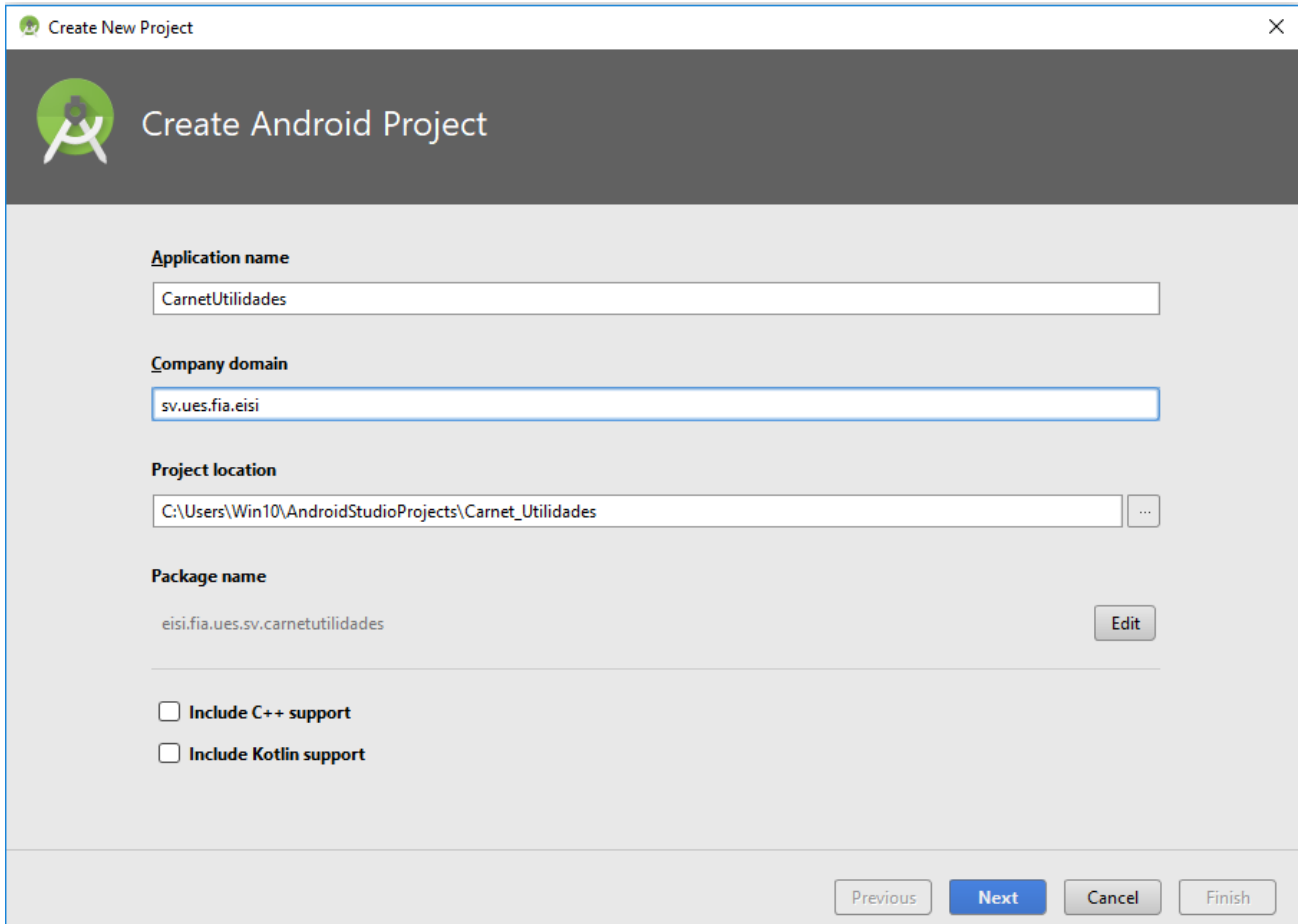
- *Hacer una aplicación que contenga 6 funcionalidades de forma independiente: Mediaplayer, Video, Camara, TextToSpeech(Voz en base a Texto), Speech(Reconocimiento de voz), Location(GPS), para que sirvan de insumo en la creación de la tarea 2 de la asignatura,
- *Probar su aplicación en el dispositivo Smartphone con sistema operativo Android para las funcionalidades que no se permiten ejecutar en el emulador.

Contenido

Creación del Proyecto Utilidades	1
Estructura de Activities y layouts para el proyecto	4
Menú principal	5
Menu (MainActivity)	5
Archivo de Recursos(variables) string.	6
String.xml	6
Audio	6
Layout(xml).....	6
Activity (java)	7
Agregar el recurso de musica(music.mp3).....	8
Video	10
Layout(xml).....	10
Activity (java)	10
Archivo de recursos.....	10
Permisos	11
Cámara	13
Layout(xml).....	13
Activity (java)	13
TextToSpeech.....	15
Layout(xml).....	15
Activity (java)	16
Speech	18
Layout(xml).....	18
Activity (java)	18
GPS.....	20
Layout(xml).....	20
Activity (java)	21
Permisos	23

Creación del Proyecto Utilidades

Crear un proyecto con las siguientes especificaciones.



Create New Project

Create Android Project

Application name
CarnetUtilidades

Company domain
sv.ues.fia.eisi

Project location
C:\Users\Win10\AndroidStudioProjects\Carnet_Utilidades

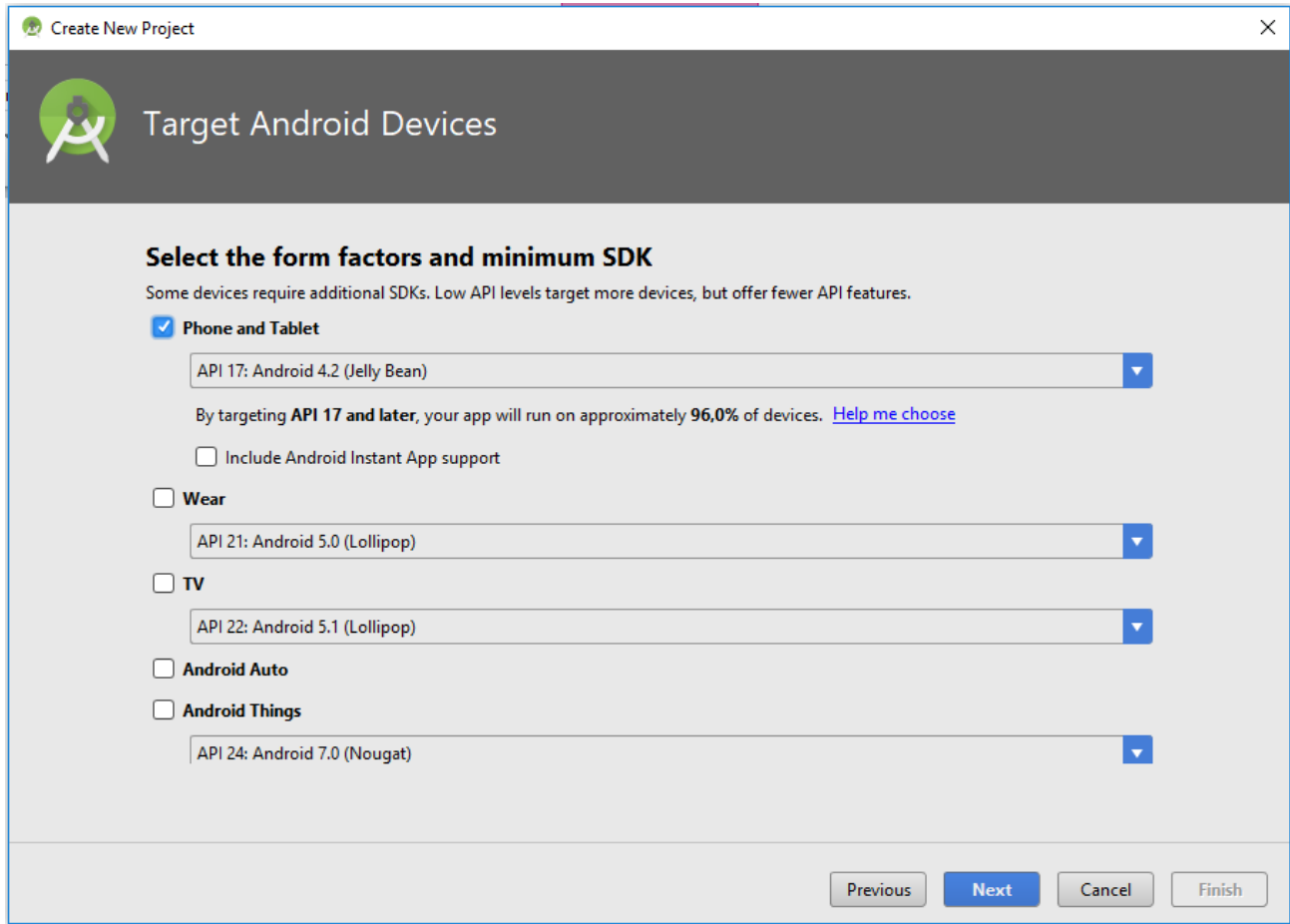
Package name
eisi.fia.ues.sv.carnetutilidades Edit

Include C++ support

Include Kotlin support

Previous Next Cancel Finish

Presione next... y seleccione El API17 como SDK minimo



Create New Project

Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet

API 17: Android 4.2 (Jelly Bean)

By targeting **API 17 and later**, your app will run on approximately **96,0%** of devices. [Help me choose](#)

Include Android Instant App support

Wear

API 21: Android 5.0 (Lollipop)

TV

API 22: Android 5.1 (Lollipop)

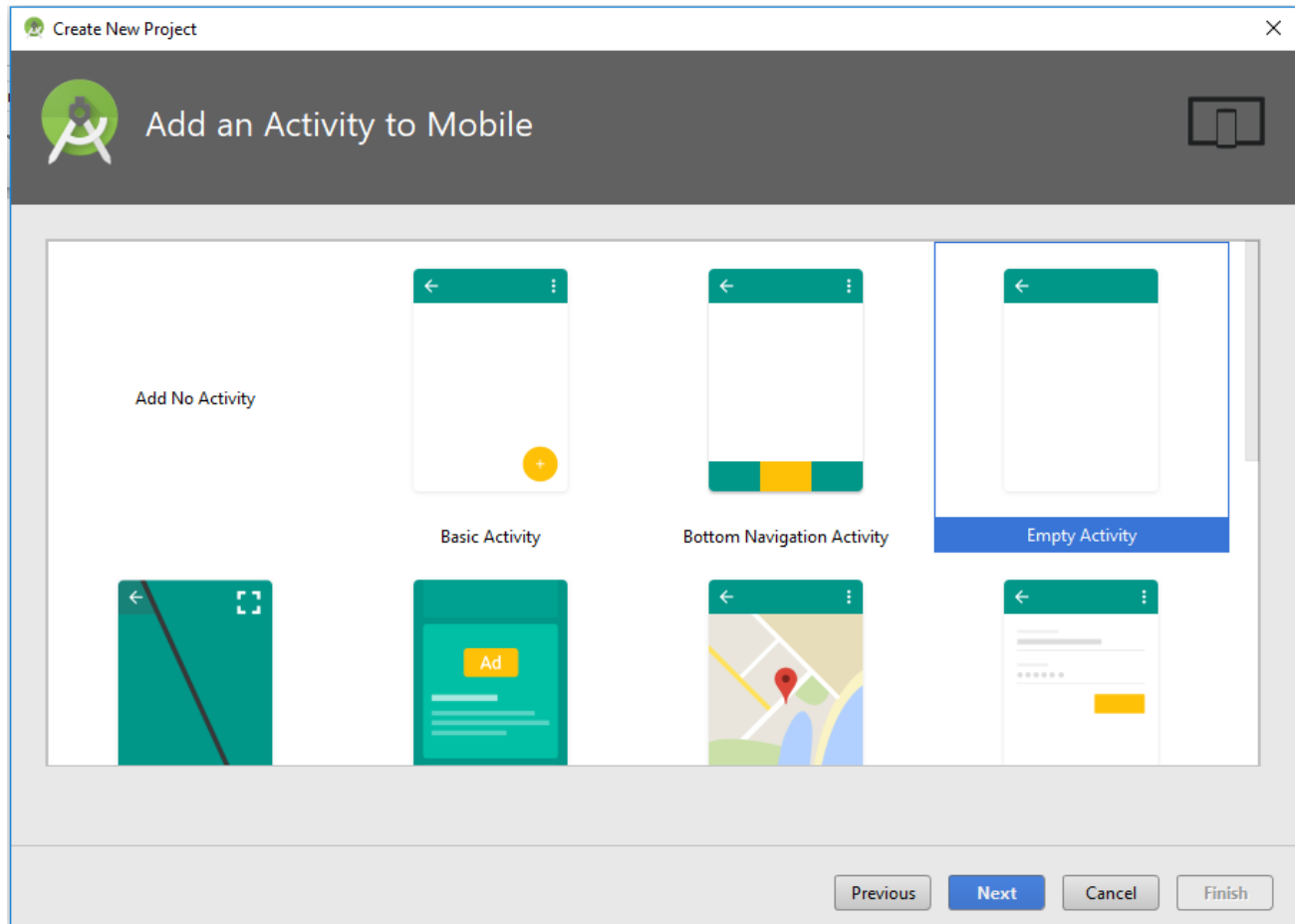
Android Auto

Android Things

API 24: Android 7.0 (Nougat)

Previous Next Cancel Finish

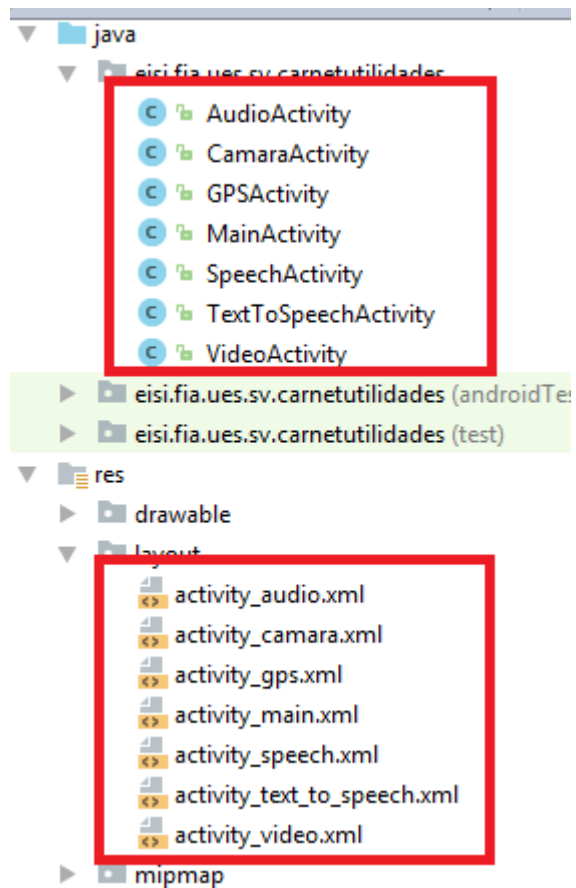
Con lo que se habrá creado un activity en blanco(Empty Activity), luego se dejara el valor del nombre por default(MainActivity)



Presione clic en Next y luego Finish

Estructura de Activities y layouts para el proyecto

Luego debemos crear los activities (y layouts) necesarios para que la estructura del proyecto se observe como la siguiente:



Menú principal

Menu (MainActivity)

```

public class MainActivity extends ListActivity {

    String[] menu={"Audio","Video","Camara","TextToSpeech","Speech","GPS"};
    String[]
activities={"AudioActivity","VideoActivity","CamaraActivity","TextToSpeechActivity","Spe
echActivity","GPSActivity"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ListView listView = getListView();
        listView.setBackgroundColor(Color.rgb(0, 0, 255));

        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1, menu);
        setListAdapter(adapter);

    }
    protected void onItemClick(ListView l,View v,int position,long id){
        super.onItemClick(l, v, position, id);

        String nombreValue=activities[position];

        l.getChildAt(position).setBackgroundColor(Color.rgb(128, 128, 255));

        try{
            Class<?>
clase=Class.forName("eisi.fia.ues.sv.carnetutilidades."+nombreValue);
            Intent inte = new Intent(this,clase);
            this.startActivity(inte);
        }
        catch(ClassNotFoundException e){
            e.printStackTrace();
        }
    }
}

```

Archivo de Recursos(variables) string.

Las variables son útiles para poner los textos en las etiquetas

String.xml

```
<resources>
  <string name="app_name">CarnetUtilidades</string>
  <string name="title_activity_audio">AudioActivity</string>
  <string name="title_activity_camara">CamaraActivity</string>
  <string name="title_activity_gps">GPSActivity</string>
  <string name="title_activity_speech">SpeechActivity</string>
  <string name="title_activity_text_to_speech">TextToSpeechActivity</string>
  <string name="title_activity_video">VideoActivity</string>
  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>
  <string name="btnplay">Play </string>
  <string name="btnstop">stop </string>
  <string name="mensajevideo">Haga Click arriba de este mensaje</string>
  <string name="imagen1">imagen</string>
  <string name="boton1">tomar foto</string>
  <string name="txtAltitud">Altitud </string>
  <string name="txtLatitud">Latitud </string>
  <string name="txtLongitud">Longitud </string>
  <string name="txtDireccion">Direccion </string>
  <string name="btnObtenerDatos">Obtener Datos Posicion </string>
  <string name="edt_tts">Diga algo</string>
  <string name="btn_tts1">Escuchar</string>
  <string name="btn_tts2">Guardar</string>
  <string name="btnTalk">Hable</string>
</resources>
```

Audio

En esta actividad, se reproduce un archivo mp3 ya establecido, la dirección está definida dentro del programa, se pueden realizar modificaciones a esta dirección, pero adelante se especifica para la aplicación cual se utilizara.

Layout(xml)

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="horizontal"
  android:paddingTop="20dp" >

  <Button
    android:id="@+id/play"
    android:layout_width="0dp"
```



```

        android:layout_height="50dp"
        android:layout_weight="1"
        android:text="@string/btnplay"/>

<Button
    android:id="@+id/stop"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_weight="1"
    android:text="@string/btnstop"/>

</LinearLayout>

```

Activity (java)

```

public class AudioActivity extends Activity {
    MediaPlayer Media;
    Button Play;
    Button Stop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_audio);
        Play=(Button) findViewById(R.id.play);
        Stop=(Button) findViewById(R.id.stop);
        Play.setOnClickListener(onClick);
        Stop.setOnClickListener(onClick);
        Media=MediaPlayer.create(getApplicationContext(), R.raw.music);
    }
    View.OnClickListener onClick=new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            if (v.getId()==R.id.play){
                if (Media.isPlaying()){
                    Media.pause();
                    Play.setText("Play");
                }
                else{
                    Media.start();
                    Play.setText("Pause");
                }
            }
            else{
                Media.stop();
                Play.setText("Play");
                try{
                    Media.prepare();
                }
                catch(IllegalStateException e){
                    e.printStackTrace();
                }
                catch(IOException e){
                    e.printStackTrace();
                }
            }
        }
    }
}

```

```

    }
}
};
}

```

Al reemplazar el código, quite los errores con alt+Enter, a excepción de la parte de creación de instancia de mediaplayer en el parámetro R.raw.music. Este parámetro da error porque no está incorporado el recurso de música.

```

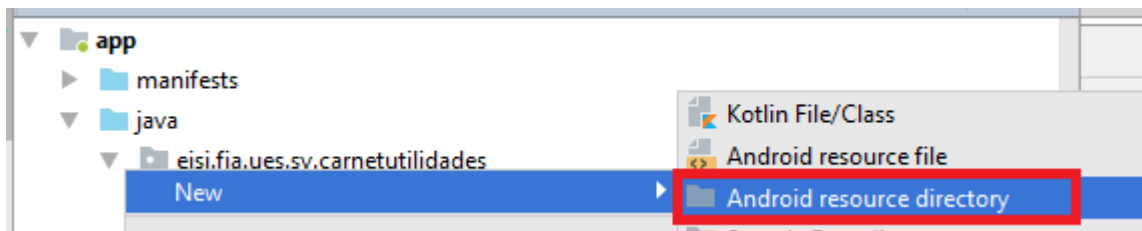
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_audio);
    Play=(Button) findViewById(R.id.play);
    Stop=(Button) findViewById(R.id.stop);
    Play.setOnClickListener(onClick);
    Stop.setOnClickListener(onClick);
    Media=MediaPlayer.create(getApplicationContext(), R.raw.music);
}

```

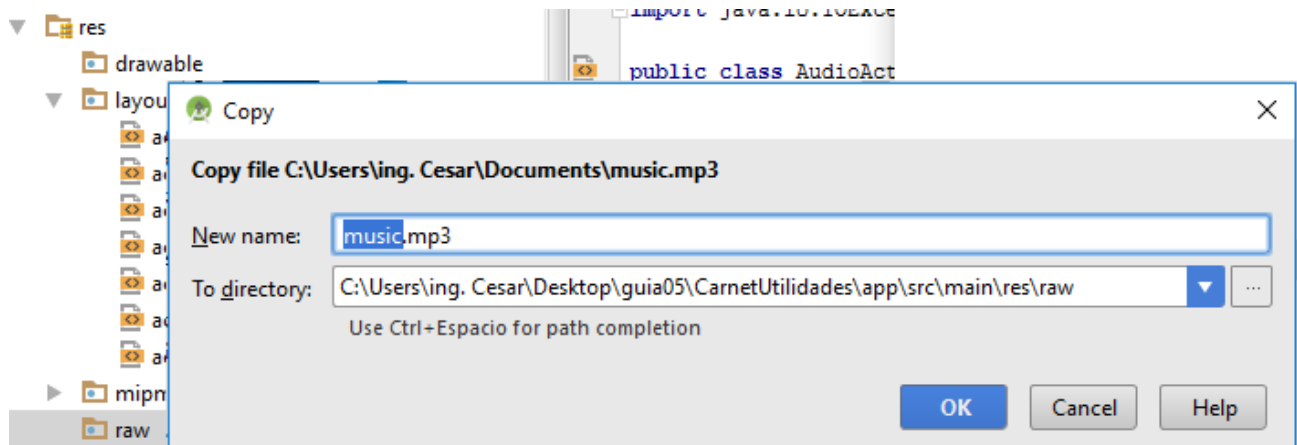
Agregar el recurso de musica(music.mp3)

Para introducir el archivo de sonido que utilizara la aplicación, proceder a crear una carpeta adentro de la carpeta “res”, ya existente en nuestro proyecto, para esto seleccionar la carpeta “res”, hacer clic derecho, y seleccionar opción “New”, después seleccionar la opción de “Android Resource Directory”, como se presenta a continuación.

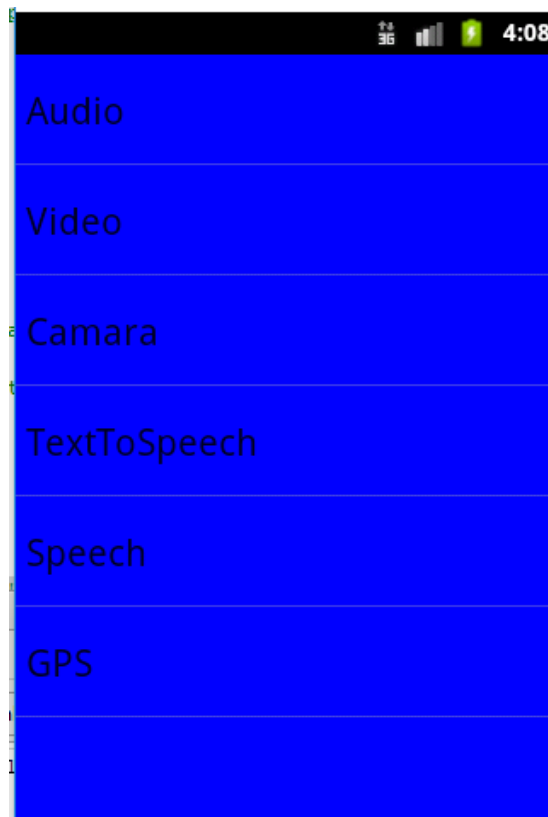
A esta carpeta colocar el nombre de “raw”.



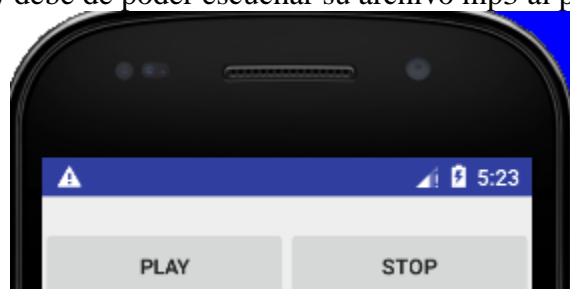
Ya creada la carpeta, se copiara el archivo externo llamado “music” con formato mp3 (busque en los recursos de la asignatura aula.fia.ues.edu.sv/materialpublico/pdm115 y descargue music.rar), copie el archivo en el escritorio, descomprimalo y dentro de la carpeta “raw” anteriormente creada péguelo (clic derecho paste), para que se muestre como a continuación.



Ejecute este programa en su emulador y vera algo como lo siguiente:



Entre a la opción de audio y debe de poder escuchar su archivo mp3 al presionar play



Video

Layout(xml)

```
<?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:orientation="vertical" >

    <VideoView
        android:id="@+id/video"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="5" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/mensajevideo" />
</LinearLayout>
```

Activity (java)

```
public class VideoActivity extends Activity {
    Button Play;
    Button Stop;
    VideoView video;
    MediaController mediacontrol;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_video);
        video=(VideoView) findViewById(R.id.video);
        File f=new File(Environment.getExternalStorageDirectory(),"wildlife.3gp");
        if (f.exists()){
            Uri uri=Uri.fromFile(f);
            video.setVideoURI(uri);
            mediacontrol=new MediaController(this);
            video.setMediaController(mediacontrol);
            mediacontrol.show();

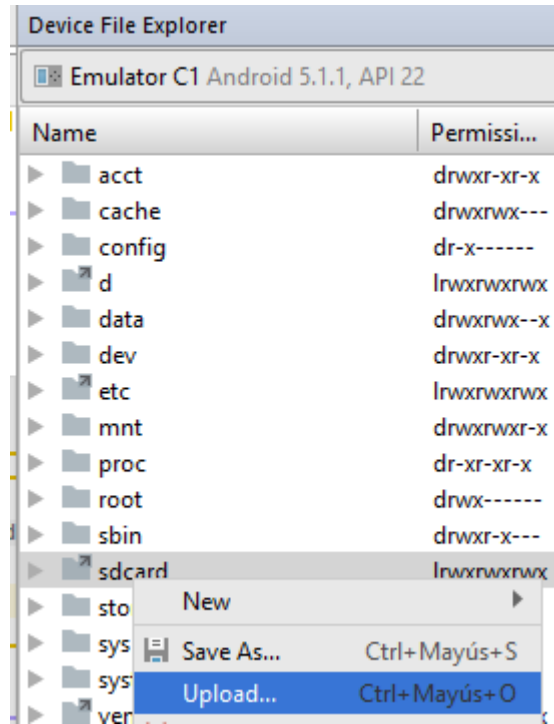
        }
    }
}
```

Archivo de recursos

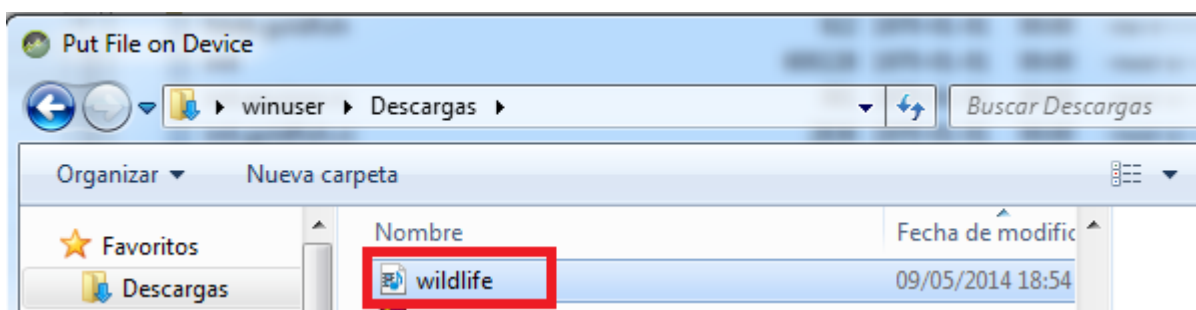
Antes de iniciar, descargue el archivo de video que utilizara(wildlife.rar o video2.rar o video3.rar), estos archivos 3gp se encuentran en aula.fia.ues.edu.sv/materialpublico/pdm115 en formato comprimido. Descargue el que desee y descomprímalo.

Para incorporar el archivo a la sdcard del simulador, primero ejecute el proyecto entre al Monitor de dispositivos Android , busque el file explorer y Luego la SDcard, y clic derecho, y click en upload

Busque su archivo y presione abrir.



Buscamos el archivo wildlife y luego presionamos clic en abrir



Permisos

En el archivo AndroidManifest.xml se deben agregar unos permisos para que la aplicación tenga acceso al gestor de localización del dispositivo, para esto se agregara la sentencia que se presenta a continuación, la cual tiene como especificación los permisos: <users-permission

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="eisi.fia.ues.sv.carnetutilidades">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".AudioActivity"
            android:label="@string/title_activity_audio"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".CamaraActivity"
            android:label="@string/title_activity_camara"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".GPSActivity"
            android:label="@string/title_activity_gps"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".SpeechActivity"
            android:label="@string/title_activity_speech"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".TextToSpeechActivity"
            android:label="@string/title_activity_text_to_speech"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".VideoActivity"
            android:label="@string/title_activity_video"
            android:theme="@style/AppTheme.NoActionBar"></activity>
    </application>
</manifest>

```

Ejecute este programa en su dispositivo

Cámara

Antes de iniciar, descargue el archivo de imagen que utilizara(background_eisi.jpg), este se encuentra en aula.fia.ues.edu.sv/materialpublico/pdm115.

El funcionamiento de la cámara en Android, se observara utilizada en esta actividad, ya que se realizara un cambio de una imagen colocada en un ImageView, para esto se ejecuta una intento de captura de imagen, el cual convoca a la cámara, al capturar la imagen, se observa la sustitución de la imagen. Para comprobar el funcionamiento de esta aplicación utilice un dispositivo físico.

Layout(xml)

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

    <ImageView
        android:id="@+id/mainimage"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="5"
        android:adjustViewBounds="true"
        android:contentDescription="@string/imagen1"
        android:src="@mipmap/ic_launcher" />
    <Button
        android:id="@+id/mainbttomarfoto"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:text="@string/boton1" />

</LinearLayout>
```

Activity (java)

```
public class CamaraActivity extends Activity {
    Button TomarFoto;
    ImageView image;
    final int FOTOGRAFIA = 654;
    Uri file;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_camara);
        TomarFoto = (Button) findViewById(R.id.mainbttomarfoto);
        image = (ImageView) findViewById(R.id.mainimage);
        TomarFoto.setOnClickListener(onClick);
        if (savedInstanceState != null) {
            if (savedInstanceState.getString("Foto") != null) {
                image.setImageURI(Uri.parse(savedInstanceState
                    .getString("Foto")));
                file = Uri.parse(savedInstanceState.getString("Foto"));
            }
        }
    }
}
```

```

    }
    public void onSaveInstanceState(Bundle bundle){
        if (file!=null){
            bundle.putString("Foto", file.toString());
        }
        super.onSaveInstanceState(bundle);
    }
    View.OnClickListener onClick=new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Intent intent=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            File photo =new
File(Environment.getExternalStorageDirectory(),String.valueOf(Calendar.getInstance().get
TimeInMillis())+".jpg");
            file=Uri.fromFile(photo);
            intent.putExtra(MediaStore.EXTRA_OUTPUT, file);
            startActivityForResult(intent,FOTOGRAFIA);

        }
    };
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent intent) {
        if (requestCode==FOTOGRAFIA){
            if(resultCode == RESULT_OK){
                image.setImageURI(file);
            }
            else{
                Toast.makeText(getApplicationContext(),"fotografia No tomada",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

Ejecute este programa en su dispositivo, opción Cámara, luego presione Tomar foto y aparecerá su programa predeterminado para tomar fotos y al tomarla, se sustituirá la imagen del logo de Android por su foto recién tomada



TextToSpeech

La actividad de TextToSpeech, se encarga de convertir un texto introducido en sonido, en este apartado, se realiza el uso de objetos de la clase TextToSpeech, la cual captura el texto introducido por el usuario, y según un lenguaje establecido reproduce la palabra en el lenguaje. Además con el botón guardar, se ejecuta el sonido y después se guarda el archivo en la sdcard(chequee solo en dispositivo smartphone)

Layout(xml)

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

    <EditText
        android:id="@+id/edtText2Speech"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/edt_tts">

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/btnText2SpeechPlay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btn_tts1" android:layout_gravity="right" />
    <Button
        android:id="@+id/btnText2SpeechSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btn_tts2" android:layout_gravity="right" />

</LinearLayout>
```

Activity (java)

```

import java.util.HashMap;
import java.util.Locale;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.app.Activity;
import android.os.Environment;
import android.speech.tts.TextToSpeech;

import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class TextToSpeechActivity extends Activity {

    TextToSpeech tts;
    TextView Textto;
    Button BtnPlay;
    Button BtnSave;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_text_to_speech);
        Textto=(TextView) findViewById(R.id.edtText2Speech);
        BtnPlay = (Button) findViewById(R.id.btnText2SpeechPlay);
        BtnSave = (Button) findViewById(R.id.btnText2SpeechSave);
        tts = new TextToSpeech(this,OnInit);
        BtnPlay.setOnClickListener(onClick);
        BtnSave.setOnClickListener(onClick);
    }
    TextToSpeech.OnInitListener OnInit = new TextToSpeech.OnInitListener() {

        @Override
        public void onInit(int status) {
            // TODO Auto-generated method stub
            if (TextToSpeech.SUCCESS==status){
                tts.setLanguage(new Locale("spa","ESP"));
            }
            else
            {
                Toast.makeText(getApplicationContext(), "TTS no disponible",
Toast.LENGTH_SHORT).show();
            }
        }
    };
    View.OnClickListener onClick=new View.OnClickListener() {

```

```

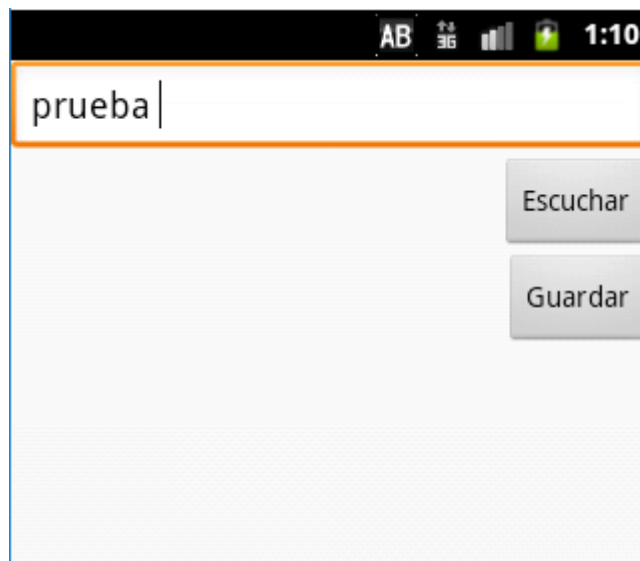
@SuppressLint("SdCardPath")
public void onClick(View v) {
    // TODO Auto-generated method stub
    if (v.getId()==R.id.btnText2SpeechPlay){
        tts.speak(Texto.getText().toString(), TextToSpeech.QUEUE_ADD, null);
    }
    if (v.getId()==R.id.btnText2SpeechSave){
        tts.speak(Texto.getText().toString(), TextToSpeech.QUEUE_ADD, null);

        HashMap<String, String> myHashRender = new HashMap<String, String>();
        String Texto_tts =Texto.getText().toString();
        String destFileName = Environment.getExternalStorageDirectory().getAbsolutePath() +
"/Download/tts1.wav";
        myHashRender.put(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID,Texto_tts);
        // tts.synthesizeToFile()
        //pdte
        // tts.synthesizeToFile(Texto_tts, myHashRender, destFileName);
        int sr = tts.synthesizeToFile(Texto_tts, myHashRender, destFileName);

    }
}
};
public void onDestroy(){
    tts.shutdown();
    super.onDestroy();
}
}
}

```

Ejecute este programa en su emulador, seleccione la opción texttospeech y vera algo como lo siguiente:



Presione escuchar y sonara el sintetizador de audio con el texto que se haya digitado. Si Presiona guardar, sonara el sintetizador de audio con el texto que se haya digitado y además guardara el audio en un archivo .wav.

Speech

Esta actividad, muestra la funcionalidad del micrófono de Andorid, ya que graba por un momento lo que se diga en el micrófono y breves segundos después, comprueba la palabra mencionada, dando opciones de posibles palabras o frases, esta acción la facilita y se realiza con intent de reconocimiento, como se observara en el código posterior. Si desea observar el funcionamiento completo de la aplicación, corroborarlo en un dispositivo físico.

Layout(xml)

```
<?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:orientation="vertical" >

    <Button
        android:id="@+id/bvoice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btnTalk" />

    <ListView
        android:id="@+id/lvVoiceReturn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>
</LinearLayout>
```

Activity (java)

```
import java.util.ArrayList;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.speech.RecognizerIntent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;

public class SpeechActivity extends Activity implements OnClickListener {

    ListView lv;
    static final int check=1111;
    Button Voice;
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_speech);
    Voice=(Button) findViewById(R.id.bvoice);
    lv =(ListView) findViewById(R.id.lvVoiceReturn);
    Voice.setOnClickListener(this);
}

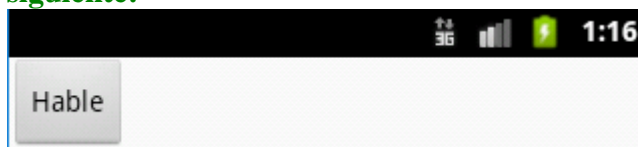
public void onClick(View v) {
    // TODO Auto-generated method stub
    if (v.getId() == R.id.bvoice) {
        // Si entramos a dar clic en el boton
        Intent i = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        i.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        i.putExtra(RecognizerIntent.EXTRA_PROMPT, "Hable ahora ");
        startActivityForResult(i, check);
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    if (requestCode==check && resultCode==RESULT_OK){
        ArrayList<String> results =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        lv.setAdapter(new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,results));
    }
    super.onActivityResult(requestCode, resultCode, data);
}

public void onDestroy(){
    super.onDestroy();
}
}

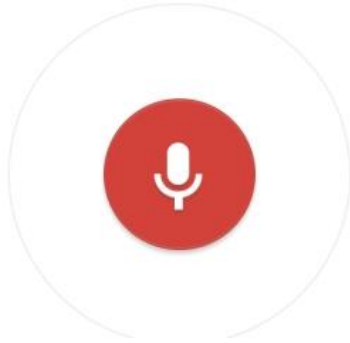
```

Ejecute este programa en su dispositivo, seleccione la opción speech y vera algo como lo siguiente:



Presione Hable y aparecerá el micrófono de google para esperar que usted hable y luego enviar la petición para recuperar la lista de palabras o frases que encuentre coincidentes con el audio.

Habla ahora



GPS

Para realizar el layout y la clase que controla la función del GPS, colocaremos los códigos que se presentan a continuación, los cuales se utilizan las clases que proporcionan la implementación de localización, la cual se llama LocationManager. Se obtendrán datos de localización actual, si desea corroborar la ubicación, compruebe compilando directamente desde un dispositivo real.

Layout(xml)

```
<?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:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/txtLatitud" >
    </TextView>
    <EditText
        android:id="@+id/edtLatitud"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:width="1dp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/txtLongitud" >
    </TextView>

    <EditText
        android:id="@+id/edtLongitud"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:width="1dp" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/txtAltitud" >
```

```

</TextView>

<EditText
    android:id="@+id/edtAltitud"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:width="1dp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/txtDireccion" >
</TextView>
<EditText
    android:id="@+id/edtDireccion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:width="1dp" />

<Button
    android:id="@+id/btnObtenerDatosPos"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btnObtenerDatos" />
</LinearLayout>

```

Activity (java)

```

import java.util.List;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class GPSActivity extends Activity {
    Button obtenerDir;
    EditText edtlatitud;
    EditText edtlongitud;
    EditText edtaltitud;
    TextView edtdireccion;

    LocationManager locationManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_gps);

obtenerDir = (Button) findViewById(R.id.btnObtenerDatosPos);
edtlatitud = (EditText) findViewById(R.id.edtLatitud);
edtlongitud = (EditText) findViewById(R.id.edtLongitud);
edtaltitud = (EditText) findViewById(R.id.edtAltitud);
edtdireccion = (TextView) findViewById(R.id.edtDireccion);
obtenerDir.setOnClickListener(onClickDireccion);
locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
}

OnClickListener onClickDireccion = new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Geocoder g = new Geocoder(getApplicationContext());
        List<Address> ad = null;
        try {
            ad = g.getFromLocation(
                Double.valueOf(edtlatitud.getText().toString()),
                Double.valueOf(edtlongitud.getText().toString()),
1);
            } catch (Exception e) {
                e.printStackTrace();
            }
            if (ad != null && ad.isEmpty() == false) {
                edtdireccion.setText(ad.get(0).getThoroughfare() + ","
                    + ad.get(0).getSubAdminArea() + ","
                    + ad.get(0).getCountryName());
            }
        }
    };

LocationListener locationListener = new LocationListener() {

    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        edtlatitud.setText(String.valueOf(location.getLatitude()));
        edtlongitud.setText(String.valueOf(location.getLongitude()));
        edtaltitud.setText(String.valueOf(location.getAltitude()));

    }

    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub

    }

    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub

    }
}

```



```

        public void onStatusChanged(String provider, int status, Bundle extras) {
            // TODO Auto-generated method stub

        }

    };
    @Override
    public void onPause() {
        super.onPause();
        locationManager.removeUpdates(locationListener);
    }

    @Override
    public void onResume() {
        super.onResume();
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0,
            0, locationListener);
        locationManager.requestLocationUpdates(
            LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
    }
}

```

Permisos

En el archivo AndroidManifest.xml se deben agregar unos permisos para que la aplicación tenga acceso al gestor de localización del dispositivo, para esto se agregara la sentencia que se presenta a continuación, la cual tiene como especificación los permisos: <users-permission

```


<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="carnet.fia.ues.sv.carnetutilidades">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".AudioActivity" />
        <activity android:name=".CamaraActivity" />
        <activity android:name=".GPSActivity" />
        <activity android:name=".SpeechActivity" />
        <activity android:name=".TextToSpeechActivity" />
        <activity android:name=".VideoActivity"></activity>
    </application>
</manifest>

```

Ejecute este programa en su dispositivo, seleccione la opción GPS y vera algo como lo siguiente:



The screenshot shows a mobile application interface with four text input fields labeled "Latitud", "Longitud", "Altitud", and "Direccion". Below these fields is a large grey button with the text "OBTENER DATOS POSICION".

Automáticamente se recuperaran los datos de Latitud, longitud y altitud, pero dirección se recuperara hasta presionar clic en Obtener datos de Posición.

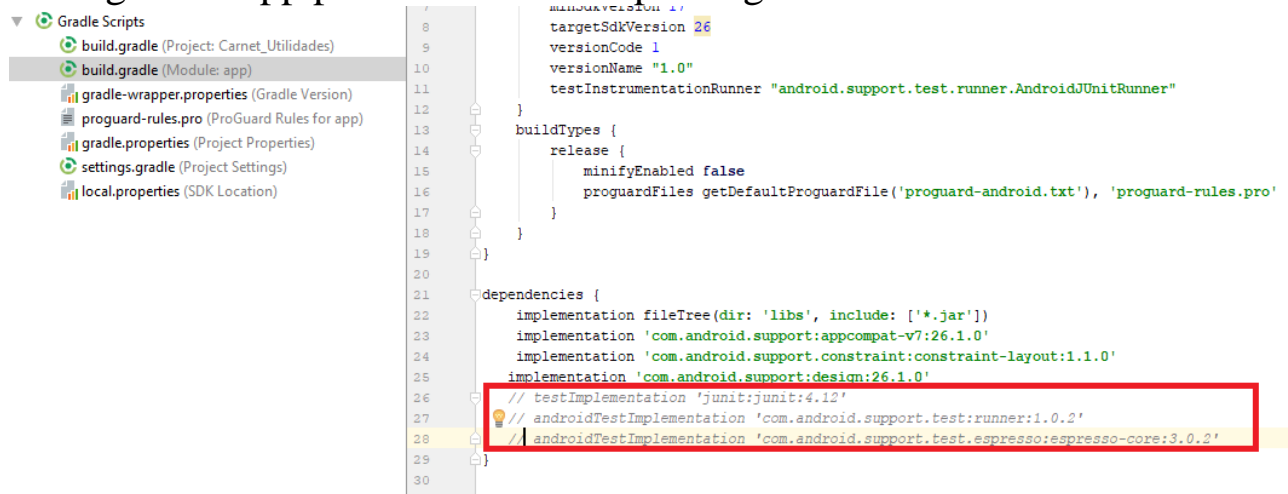
Nota: Se recomienda que se ejecute esta opción en un espacio abierto, no en edificios o parques techados.

Al finalizar comprima su proyecto y súbala en la opción respectiva



Anexo(solo realice esta operación si le da errores desde un inicio)

Configure su app para evitar errores por el gradle



```
8      minSdkVersion 17
9      targetSdkVersion 26
10     versionCode 1
11     versionName "1.0"
12     testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
13 }
14 buildTypes {
15     release {
16         minifyEnabled false
17         proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18     }
19 }
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:26.1.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.0'
25     implementation 'com.android.support:design:26.1.0'
26     // testImplementation 'junit:junit:4.12'
27     // androidTestImplementation 'com.android.support.test:runner:1.0.2'
28     // androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
29 }
30 }
```