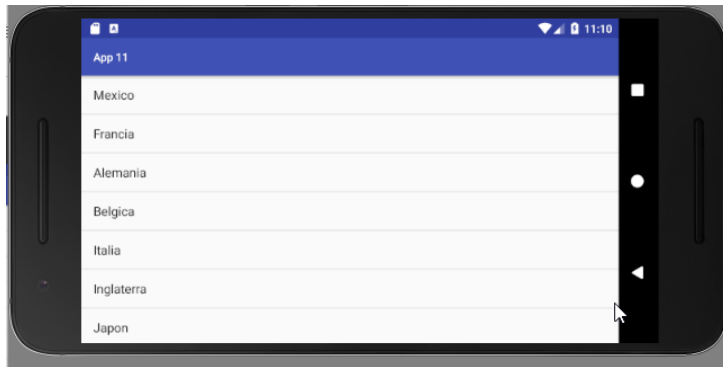
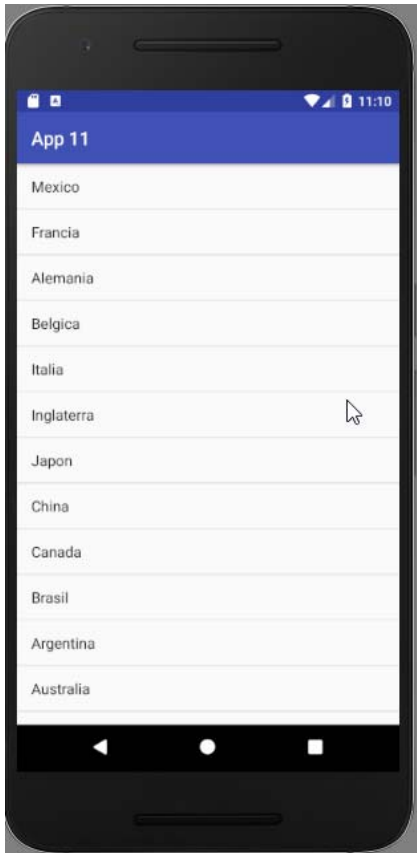


PRACTICA 11.

OBJETIVO GENERAL.

El alumno aplicará la estructura de arreglos definidos en la carpeta de recursos strings y su relación existente con el elemento ListView.



OBJETIVOS ESPECÍFICOS:

- Utilizar LinearLayout como contenedor principal de la aplicación.
- Crear objetos Actividad:
 - Un ListView : Utilizado para desplegar la cita
- Crear un objeto adapter que prepare los datos que serán cargados al ListView.

CREACION DE LA INTERFACE GRAFICA.

Usaremos el código siguiente para la interface gráfica.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.delio.app11.MainActivity">

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

    </ListView>

</LinearLayout>
```

DESARROLLO DE LA LOGICA DE APLICACIÓN.

El código inicial de nuestra actividad principal se verá:

```
package com.example.delio.app11;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Vamos de definir los elementos:

```
ListView lstPaises;
ArrayAdapter<String> adpPaises;
String [] arrPaises;
```

Una forma de manejar el objeto ListView es definir una fuente de datos de donde tomará los datos a mostrar, una característica de este objeto es que contiene un scrollview interno que se adapta y podemos subir o bajar los datos en la pantalla si es que esta (la pantalla) no existe suficiente área para desplegarlos.

Explicaremos la lógica principal, como es habitual relacionamos la parte grafica con la lógica mediante el casting de los elementos (ListView).

```
lstPaises = (ListView) findViewById(R.id.lstPaises);
```

Posteriormente mediante el arreglo definido y los datos definidos en el archivo string.xml hacemos una carga de datos.

```
arrPaises = getResources().getStringArray(R.array.paises);
```

Seguido, creamos la fuente de datos mediante un objeto adaptador, debemos de tener cuidado de especificar el tipo de dato del objeto adapter, en este caso es del tipo string.

```
adpPaises = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_1,  
    arrPaises);
```

Finalmente ligamos el listView con el adapter mediante la instrucción:

```
lstPaises.setAdapter(adpPaises);
```

Nuestro código final es el siguiente:

```
lstPaises = (ListView) findViewById(R.id.lstPaises);  
arrPaises = getResources().getStringArray(R.array.paises);  
adpPaises = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_1,  
    arrPaises);  
lstPaises.setAdapter(adpPaises);
```