



CFP/UAX

Centro de Formación Profesional

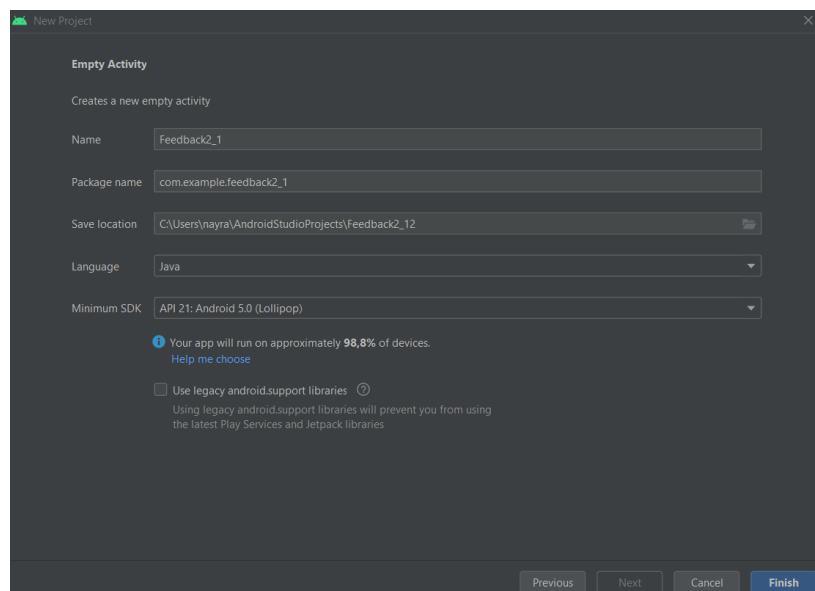
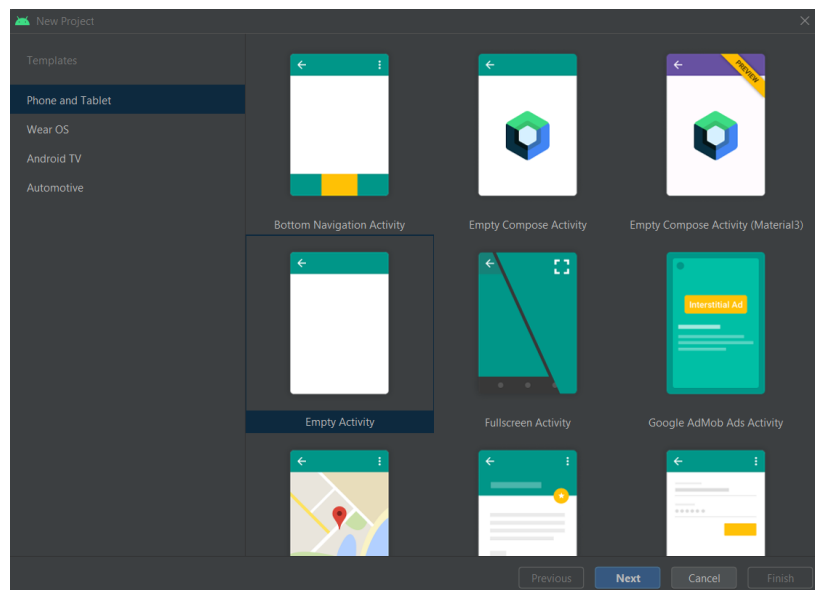
PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

Nayra Infantes Tardón

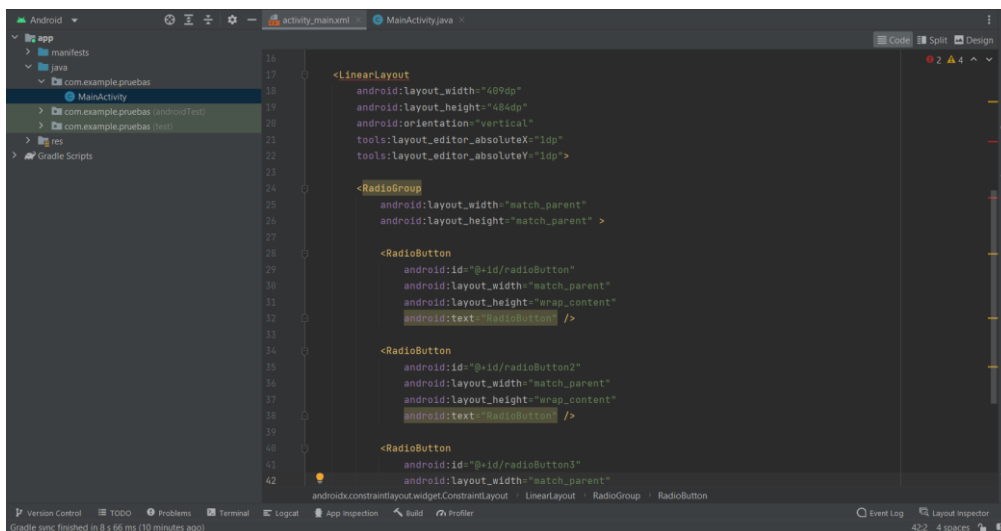
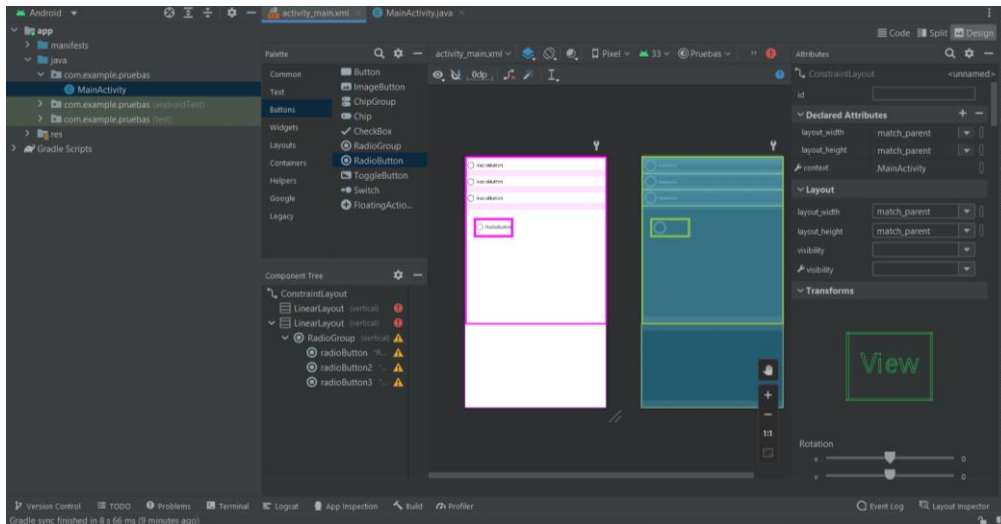
Ejercicio feedback unidad 2

Diseña una aplicación que muestre el nombre de 10 países como opciones en botones tipo radio y que al pulsar en cada una de ellas se muestren diferentes imágenes (la bandera del país seleccionado).

Para poder diseñar nuestra aplicación lo primero que debemos hacer es crear un nuevo proyecto, en nuestro caso se llamará Feedback2_1, partiremos de una actividad vacía para poder tener una base previa, y el lenguaje a utilizar será Java.



Gracias a la interfaz que nos proporciona Android Studio podemos crear nuestros botones de tipo radio desde de una parte gráfica en la que funciona todo de una manera muy intuitiva o directamente a través de código en nuestro archivo `activity_main.xml`.



Es importante comprender para la realización del ejercicio que los botones de selección son mutuamente excluyentes, es decir, al seleccionar uno eliminamos la selección del anterior. Para que esto suceda y nuestro programa funcione correctamente debemos crear un `RadioGroup` que almacene nuestros `RadioButtons`.

Si quisiéramos, por ejemplo, crear una aplicación con preguntas que contengan varias respuestas de tipo `RadioButton`, pero solo quisiéramos que el usuario seleccionase una de ellas, cada juego de respuestas debería estar contenido por un `RadioGroup`.

Nuestros botones de tipo radio tendrán un id que nos ayudará a poder referenciarlos más tarde, ya que deben mostrar imágenes al ser pulsados; width y height, es decir, ancho y alto; un texto con el nombre del país al que corresponda su bandera; y además atributos de texto como el tamaño o el estilo.

Además de todo esto, en nuestro archivo main_activity.xml debemos ubicar el contenedor de las imágenes que queremos mostrar, esta acción se puede realizar con la etiqueta ImageView, que tiene unos atributos similares a nuestros botones, en este caso dispone de un id, y un ancho y alto, aunque también podría tener bordes y muchas otras características si quisiéramos.

Como detalles opcionales, en nuestro caso hemos decidido añadir un fondo al LinearLayout que contiene todo lo demás con "android:background".

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/mapamundiblanco"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <RadioGroup
        android:id="@+id/rg"
        android:layout_width="match_parent"
        android:layout_height="479dp"
        tools:layout_editor_absoluteX="153dp"
        tools:layout_editor_absoluteY="68dp">

        <RadioButton
            android:id="@+id/es"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:buttonTint="#000000"
            android:text="ESPAÑA"
            android:textStyle="bold"
            android:textSize="16sp" />

    </RadioGroup>

</LinearLayout>
```

```
<RadioButton
    android:id="@+id/fr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="FRANCIA"
    android:textStyle="bold"
    android:textSize="16sp" />

<RadioButton
    android:id="@+id/it"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ITALIA"
    android:textStyle="bold"
    android:textSize="16sp" />
</RadioGroup>

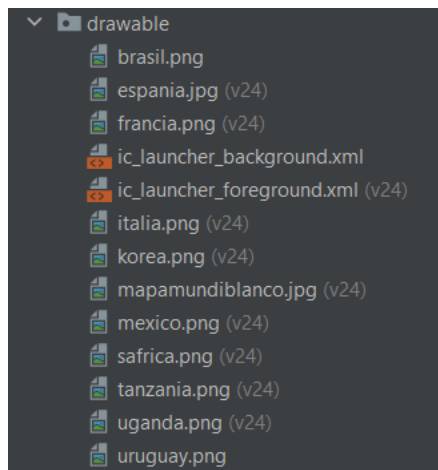
<ImageView
    android:id="@+id/iv"
    android:layout_width="match_parent"
    android:layout_height="185dp" />
</LinearLayout>
```

También hemos modificado algunos valores por defecto, como los colores de la parte de la app que se ubica en la zona superior de nuestra pantalla y el nombre de la app para que muestre la palabra “BANDERAS”, estos cambios se realizan en los archivos strings.xml y colors.xml, ubicados dentro de la carpeta Values que a su vez se encuentra en la carpeta res.

```
<resources>
    <string name="app_name">BANDERAS</string>
</resources>

<resources>
    <color name="purple_200">#2C2C2C</color>
    <color name="purple_500">#686868</color>
    <color name="purple_700">#292929</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
</resources>
```

Cuando tengamos, por así decirlo, la parte más visual terminada, debemos añadir a nuestra carpeta drawable las imágenes que vayamos a utilizar, es el mismo proceso que hemos realizado con nuestro fondo para la app.



Para poder dar la utilidad que necesitan a nuestros botones trabajaremos en nuestro archivo MainActivity.java.

En primer lugar, debemos importar las clases necesarias para poder referenciar nuestros botones y nuestro contenedor para las imágenes.

Gracias a la opción OnCheckedChangeListener en el RadioGroup podemos configurar una escucha de eventos a través de la interfaz.

En este caso, con un switch le indicamos que cada vez que se pulse el RadioButton con el id x nos muestre la imagen y que se encuentra en drawable.

```
package com.example.feedback2_1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ImageView;
import android.widget.RadioGroup;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Ref radiogroup
        RadioGroup RG=findViewById(R.id.rg);

        //Ref imageview

        ImageView IV=findViewById(R.id.iv);

        RG.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup radioGroup, int i) {

                switch (i) {
```



```

case R.id.es:
    IV.setImageResource(R.drawable.espania);
    break;
case R.id.ur:
    IV.setImageResource(R.drawable.uruguay);
    break;
case R.id.ug:
    IV.setImageResource(R.drawable.uganda);
    break;
case R.id.tz:
    IV.setImageResource(R.drawable.tanzania);
    break;
case R.id.mex:
    IV.setImageResource(R.drawable.mexico);
    break;
case R.id.ko:
    IV.setImageResource(R.drawable.korea);
    break;
case R.id.sa:
    IV.setImageResource(R.drawable.safrica);
    break;

```

Nuestra aplicación se mostrará de la siguiente manera:

BANDERAS	BANDERAS
<input type="radio"/> ESPAÑA	<input type="radio"/> ESPAÑA
<input type="radio"/> URUGUAY	<input type="radio"/> URUGUAY
<input type="radio"/> UGANDA	<input type="radio"/> UGANDA
<input type="radio"/> TANZANIA	<input type="radio"/> TANZANIA
<input type="radio"/> MEXICO	<input checked="" type="radio"/> MEXICO
<input type="radio"/> KOREA	<input type="radio"/> KOREA
<input type="radio"/> SUDÁFRICA	<input type="radio"/> SUDÁFRICA
<input type="radio"/> BRASIL	<input type="radio"/> BRASIL
<input type="radio"/> FRANCIA	<input type="radio"/> FRANCIA
<input type="radio"/> ITALIA	<input type="radio"/> ITALIA

Diseña una aplicación que muestre por pantalla una lista de 10 países y debajo un TextView, para que al mantener pulsado un determinado país se muestre en el TextView el país que se ha pulsado y que al pulsar una vez un determinado país se acceda a otra activity con dos TextView, en uno sale el nombre de ese país y en otro el número de habitantes del mismo.

Para realizar este ejercicio necesitaremos dos archivos main_activity.xml, en el primer archivo crearemos las vistas para que nuestra aplicación pueda mostrar tanto la lista con 10 países como el texto que corresponde a su país.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/mapamundiblanco"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/lv"
        android:layout_width="match_parent"
        android:layout_height="510dp"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/tv"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:fontFamily="sans-serif"
        android:paddingTop="10dp"
        android:textAlignment="center"
        android:textSize="20sp"
        android:textStyle="bold" />
</LinearLayout>
```

En el segundo archivo, llamado main_activity2.xml crearemos los TextViews necesarios para poder realizar la segunda parte de nuestro ejercicio, es decir, mostrar tanto el nombre del país como la población.


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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:layout_width="match_parent"
    android:background="@drawable/mapamundiablanca"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/typais"
        android:layout_width="189dp"
        android:layout_height="104dp"
        android:layout_marginStart="108dp"
        android:layout_marginTop="116dp"
        android:paddingTop="40dp"
        android:text="@string/typais"
        android:textAlignment="center"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tyhab"
        android:layout_width="320dp"
        android:layout_height="152dp"
        android:layout_marginStart="52dp"
        android:layout_marginBottom="220dp"
        android:paddingTop="65dp"
        android:text="@string/tyhab" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

También necesitaremos un tercer archivo .xml que haga referencia a la lista de países.

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:textSize="20dp"
    android:layout_height="match_parent" />
```

Dado que disponemos de dos archivos main_activity.xml, necesitaremos un archivo MainActivity.java por cada uno de ellos.

En el primero crearemos dos arrays de Strings, uno para cada dato, es decir, países y número de habitantes.

Debemos hacer uso de la clase ArrayAdapter que permite insertar los datos desde un array de String en nuestro ListView.

Los métodos onItemClick y onItemLongClick nos permiten comprobar como su nombre indica que tipo de acción se está ejecutando sobre nuestros botones y realizar una acción en consecuencia.

En el segundo, con el método onCreate(), ejecutamos la lógica de arranque básica de la aplicación que debe ocurrir una sola vez en toda la vida de la actividad. Por ejemplo, nuestra implementación de onCreate() en este caso vincula datos a listas.

```
package com.example.feedback22;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

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

        ListView lv = findViewById(R.id.lv);
        TextView tv = findViewById(R.id.tv);

        String [] pais = new String[] {"ESPAÑA", "URUGUAY", "UGANDA", "TANZANIA", "MEXICO",
                                       "KOREA", "SUDÁFRICA", "BRASIL", "FRANCIA", "ITALIA"};
        String [] hab = new String[] {"47.435.597", "3.485.152", "42.460.000", "61.498.438", "130.262.220",
                                       "51.628.117", "60.143.000", "213.993.441", "67.800.000", "59.236.213"};

        ArrayAdapter<String> adaptador = new ArrayAdapter<String>(context, this, R.layout.list_item_pais, pais);

        lv.setAdapter(adaptador);

        lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
            @Override
            public boolean onItemLongClick(AdapterView<?> adapterView, View view, int i, long l) {

                tv.setText(lv.getItemAtPosition(i).toString());

                return true;
            }
        });

        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

                Intent in = new Intent( packageContext, MainActivity2.class);

                in.putExtra( name: "data", lv.getItemAtPosition(i).toString());

                in.putExtra( name: "data2", hab[i]);

                startActivity(in);
            }
        });
    }
}
```

```
package com.example.feedback22;

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

public class MainActivity2 extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        TextView tvpais = findViewById(R.id.tvpais);
        TextView tvhab = findViewById(R.id.tvhab);

        Bundle bundle = getIntent().getExtras();

        String nPais = bundle.getString(key: "data");
        String nHab = bundle.getString(key: "data2");

        tvpais.setText(nPais);
        tvhab.setText(nHab);
    }
}
```