

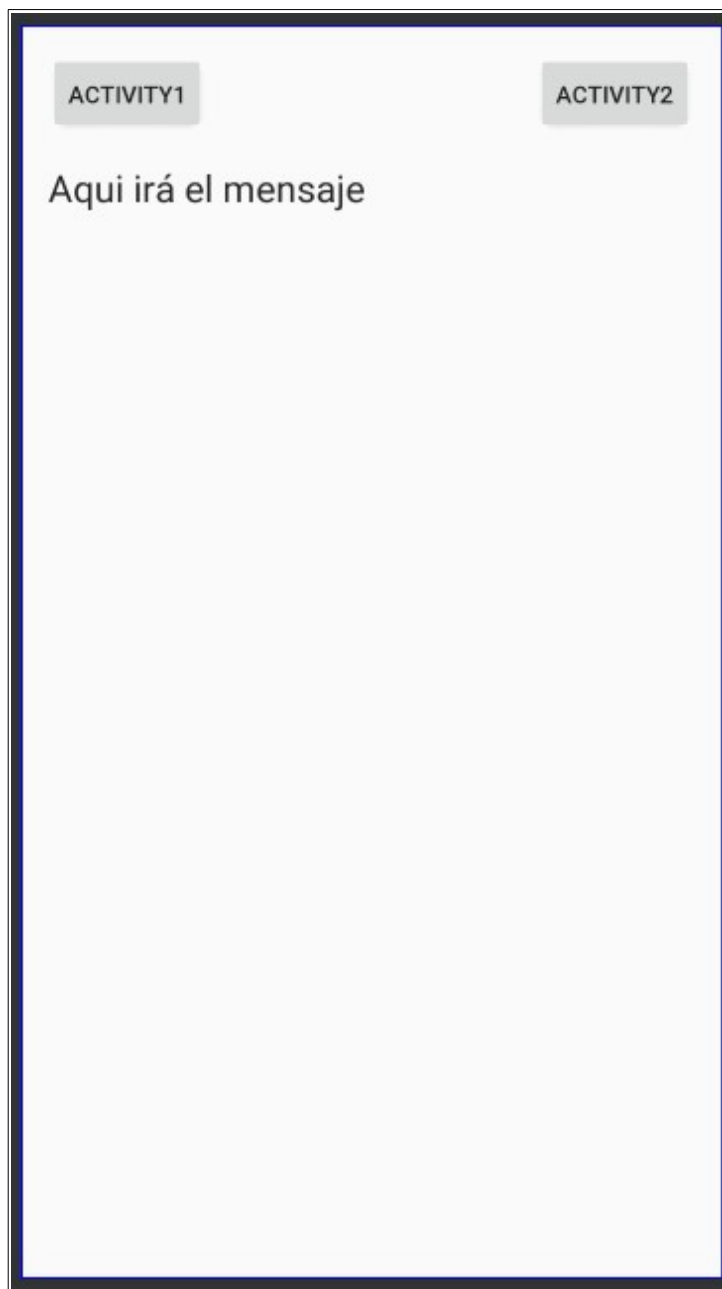
Documentación

Ejercicios propuesto 3 Activitys

1) Definir el activity_main.xml:

Lo primero que voy a hacer para este proyecto es definir el activity_main.xml, que es el activity principal.

Es un layout sencillo con dos botones y un TextView donde colocaremos el mensaje de retorno de las activitys secundarias.



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

<RelativeLayout 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"

    tools:context=".MainActivity"

    android:paddingTop="16dp">

    <Button

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:id="@+id/boton1"

        android:text="ACTIVITY1"

        android:layout_alignParentStart="true"

        android:layout_marginStart="16dp"/>

    <Button

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:id="@+id/boton2"

        android:text="ACTIVITY2"

        android:layout_alignParentEnd="true"

        android:layout_marginEnd="16dp"/>

    <TextView

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:id="@+id/textview"

        android:layout_below="@id/boton1"

        android:layout_marginTop="16dp"

        android:textAppearance="?android:attr/textAppearanceLarge"

        android:layout_marginStart="16dp"

        android:layout_marginEnd="16dp"/>

</RelativeLayout>
```

2) Desarrollo de las Activitys secundarias:

El desarrollo de las activitys secundarias es similar, no es más que un TextView indicando el activity en el que me encuentro y un botón para volver a la activity principal.



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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:id="@+id/textview"
```

```
        android:text="Actividad Uno"
        android:textAppearance="?android:attr/textAppearanceLarge"/>
<Button
    android:id="@+id/boton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textview"
    android:layout_alignStart="@id/textview"
    android:layout_alignEnd="@id/textview"
    android:text="Atrás"/>

</RelativeLayout>
```

3) Implementación de MainActivity.java

Para la implementación del código java, lo que voy a hacer es crear dos variables globales que utilizaré como código de activity, para tener una manera de identificarlas.

Una vez realizado esto, voy a implementar el evento “Click” sobre los botones y dependiendo del botón que sea (utilizando el id), lanzaré una activity u otra con un mensaje indicando cuál es el activity.

Para el envío de mensajes se utiliza un Intent y se le añade un par de clave – valor, que indicará el nombre del dato y el dato en sí; para la extracción de datos se hace el proceso inverso, se indica el nombre del dato y se recibe el dato identificado.

Para lanzar el activity utilizo el método startActivityForResult(...) para que la activity a la que llamo me devuelva una serie de datos que yo luego trataré.

Finalmente, implemento el método que me va a gestionar los datos recibidos de los activities diferenciando a través de la comparación del requestCode y el código de activity que declaré al principio. Esta implementación hace que se muestre un mensaje en el TextView que indica de que activity se vuelve (siempre y el resultCode sea OK).

```
package com.danito.p_3activities;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    int codActivity1 = 1;
    int codActivity2 = 2;

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

```
Button b1 = findViewById(R.id.boton1);
b1.setOnClickListener(this);
Button b2 = findViewById(R.id.boton2);
b2.setOnClickListener(this);
}

@Override
public void onClick(View v){
    switch(v.getId()){
        case R.id.boton1:
            Intent i1 = new Intent(this, Activity1.class);
            i1.putExtra("activity", "Activity 1 iniciada desde la actividad principal");
            startActivityForResult(i1, codActivity1);
            break;
        case R.id.boton2:
            Intent i2 = new Intent(this, Activity2.class);
            i2.putExtra("activity", "Activity 1 iniciada desde la actividad principal");
            startActivityForResult(i2, codActivity2);
            break;
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (
        requestCode == codActivity1 &&
        resultCode == RESULT_OK &&
        data != null) {
        TextView t = findViewById(R.id.textview);
        t.setText("Retorno de activity 1 con mensaje " + data.getStringExtra("mensaje"));
    } else if (
        requestCode == codActivity2 &&
        resultCode == RESULT_OK &&
        data != null) {
        TextView t = findViewById(R.id.textview);
        t.setText("Retorno de activity 2 con mensaje " + data.getStringExtra("mensaje"));
    }
}
```

4) Implementación de los javas correspondientes a los activitys secundarios:

Esta implementación es, al igual que los XML, muy parecida. El único cambio que se realiza es el mensaje que se le devuelve al activity principal.

La lógica es simplemente recibir el mensaje de la activity principal y mostrarlo con un Toast.

También implemento el evento “Click” del botón para que cuando se pulse, se finalice la activity secundaria con el código OK y el evento “onBackPressed” para que se finalice la activity secundaria con el código CANCEL.

Cuando se pulsa el botón se envía el mensaje en forma de datos a través de un intent y con el método `setResult(...)`, pero solamente cuando se pulsa el botón, cuando se pulsa el botón de “Back”, no se envían datos.

```
package com.danito.p_3activities;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Activity1 extends AppCompatActivity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity1);

        Intent datosRecibidos = getIntent();

        Toast.makeText(getApplicationContext(),
            datosRecibidos.getStringExtra("activity"),
            Toast.LENGTH_LONG).show();

        Button boton = findViewById(R.id.boton);
        boton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent datosEnviar = new Intent();
                datosEnviar.putExtra("mensaje", "Finalizada activity 1"); // Esto se cambia en el activity 2
                setResult(RESULT_OK, datosEnviar);
                finish();
            }
        });
    }

    @Override
    public void onBackPressed() {
        setResult(RESULT_CANCELED);
        finish();
    }
}
```