

Actividad - 3 Activitys con regreso.

Documentación realizada por Sergio Ramos Santonja

Esta actividad consiste en crear 3 Activitys que cuando nos desplazemos de una Activity a otra nos muestre un mensaje desde que Activity nos hemos desplazado. Este mensaje no lo mostrara tanto desplazándonos con un botón de la aplicación o mediante el botón de regresar del propio móvil.

1. Código de los Layouts.

- Activity principal

Como vemos a continuación en la imagen del código, en el Activity principal añadimos dos botones para desplazarnos a la dos Activitys secundarias. Además, añadimos un TextView para mostrar los mensajes de retorno.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:id="@+id/activity_main"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/boton_activity1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:ems="10"
        android:text="Activity 1"/>

    <Button
        android:id="@+id/boton_activity2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_toEndOf="@id/boton_activity1"
        android:ems="10"
        android:text="Activity 2"/>

    <TextView
        android:id="@+id/mostrarTexto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/boton_activity1"
        android:textSize="30dp"/>

</RelativeLayout>
```

- Activity Secundaria 1 y 2

El desarrollo de las Activitys secundarias es el siguiente, ambas Activitys tienen diferentes .xml, pero el código es el mismo. La Activity contendrá un TextView con el nombre de la Activity para mostrar en cual nos encontramos y también tendrá un botón con el que volveremos hacia la Activity principal, el cual lo controlaremos desde su propio .java y desde el MainActivity.java.

```
<?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"
    android:background="@color/colorPrimaryDark"
    android:id="@+id/layout_activity1">

    <TextView
        android:id="@+id/tV_activity1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="Activity Uno"
        android:textSize="30dp"
        android:textColor="@android:color/white"/>

    <Button
        android:id="@+id/boton_vueltaActivity1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/tV_activity1"
        android:ems="10"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        android:hint="Volver"/>

</RelativeLayout>
```

2. Códigos Java.

- Código de las Activitys Secundarias.

En el .Java de las Activitys Secundarias hacemos varias cosas, como recoger y enviar mensajes.

En primer lugar, mediante el Toast mostramos el mensaje que recogemos de la Activity principal.

En segundo lugar, cuando le damos al botón de la Activity, mediante el `.setOnClickListener` enviamos hacia el la Activity principal el mensaje de que volvemos de de dicha Activity, el cual sera recogido en el MainActivity.

Y por ultimo controlamos el botón de regreso del móvil mediante el método `onBackPressed()`, el cual tiene el mismo funcionamiento que el botón de la Activity, excepto que el resultado en vez de ser `RESULT_OK`, pondremos `RESULT_CANCELED` para que la Activity devuelva correctamente los datos.

```
package com.example.a3_activitysconretorno;

import ...

public class activity1 extends AppCompatActivity {

    TextView texto;
    Button boton;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity1);
        Toast.makeText( context: this, getIntent().getStringExtra( name: "ENTRADA"), Toast.LENGTH_LONG).show();

        boton = (Button)findViewById(R.id.boton_vueltaActivity1);

        boton.setOnClickListener((v) -> {
            Intent datos = new Intent();
            datos.putExtra( name: "SALIDA", value: "Finalizada Activity 1");
            setResult(RESULT_OK, datos);
            finish();
        });
    }

    @Override
    public void onBackPressed()
    {
        Intent datos = new Intent();
        datos.putExtra( name: "SALIDA", value: "Finalizada Activity 1");
        setResult(RESULT_CANCELED, datos);
        finish();
    }
}
```

- MainActivity.java

En el MainActivity.java inicializamos los botones de todas las Activitys para poder mostrar tanto los Toast en las Activitys Secundarias como para recibir los mensajes de dichas Activitys.

Ademas, tenemos dos métodos:

- Método onClick, en el recogeremos los botones de la Activity principal y mandaremos el mensaje que sera utilizado en los .java de las Activitys Secundarias para mostrar el Toast, dependiendo el botón en el que hagamos click mandaremos un mensaje u otro.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    Button boton1;
    Button boton2;
    Button volver1;
    Button volver2;
    TextView texto;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        boton1 = (Button) findViewById(R.id.boton_activity1);
        boton1.setOnClickListener(this);
        boton2 = (Button) findViewById(R.id.boton_activity2);
        boton2.setOnClickListener(this);

        volver1 = (Button) findViewById(R.id.boton_vueltaActivity1);
        volver2 = (Button) findViewById(R.id.boton_vueltaActivity2);

        texto = (TextView) findViewById(R.id.mostrarTexto);
    }
    @Override
    public void onClick(View v)
    {
        if(v.getId() == R.id.boton_activity1)
        {
            Intent intento = new Intent( packageContext: this, activity1.class);
            intento.putExtra( name: "ENTRADA", value: "De la principal a la Activity 1");
            startActivityForResult(intento, requestCode: 1);
        }
        if (v.getId() == R.id.boton_activity2)
        {
            Intent intento = new Intent( packageContext: this, activity2.class);
            intento.putExtra( name: "ENTRADA", value: "De la principal a la Activity 2");
            startActivityForResult(intento, requestCode: 2);
        }
    }
}
```

- Método `.onActivityResult()`, en este método recogemos el mensaje que nos envían las Activities Secundarias. Sabemos que Activity es cual mediante el `requestCode` que hemos guardado en el método `onClick()`, además cuando ya sabemos que Activity es la que nos manda el mensaje tenemos que saber mediante que botón lo lanza y para ello vemos si el resultado es `RESULT_OK` o `RESULT_CANCEL` y el mensaje es escrito en el `TextView` de la Activity principal.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode)
    {
        case 1:
            if(resultCode == activity1.RESULT_OK)
            {
                texto.setText(data.getStringExtra( name: "SALIDA"));
            }
            if(resultCode == activity1.RESULT_CANCELED)
            {
                texto.setText(data.getStringExtra( name: "SALIDA"));
            }
        case 2:
        {
            if(resultCode == activity1.RESULT_OK)
            {
                texto.setText(data.getStringExtra( name: "SALIDA"));
            }
            if(resultCode == activity1.RESULT_CANCELED)
            {
                texto.setText(data.getStringExtra( name: "SALIDA"));
            }
        }
    }
}
```

3. AndroidManifest.xml

Por ultimo, para que nuestra aplicación pueda mostrar todas las Activitys debemos añadir en el AndroidManifest.xml las 2 Activitys Secundarias.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.a3_activitysconretorno">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="3_ActivitysConRetorno"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="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=".activity1"/>
        <activity android:name=".activity2"/>
    </application>

</manifest>
```