

INDICE

INTRODUCCIÓN	1
1. INTENTS EXPLÍCITOS	2
2. CASO PRÁCTICO 1: LANZAR UNA SEGUNDA ACTIVITY PROPIA DE LA APLICACIÓN	2
2.1. EL XML DEL LAYOUT DE LA ACTIVITY PRINCIPAL	4
2.2. LA CLASE JAVA DE LA ACTIVITY PRINCIPAL	5
2.3. EL XML DE LA ACTIVITY QUE RECIBE LA LLAMADA	7
2.4. LA CLASE JAVA DE LA ACTIVITY QUE RECIBE LA LLAMADA. ACTIVITY SECUNDARIA.	8
3. CONTINUACIÓN CASO PRÁCTICO 1: PASAR DATOS DE LA ACTIVIDAD SECUNDARIA A LA PRINCIPAL	9
3.1. PASAR DATOS DE LA SECUNDARIA A LA PRINCIPAL: LA CLASE JAVA DE LA ACTIVITY PRINCIPAL	11
3.2. PASAR DATOS DE LA SECUNDARIA A LA PRINCIPAL: EL XML DE LA ACTIVITY SECUNDARIA	13
3.3. PASAR DATOS DE LA SECUNDARIA A LA PRINCIPAL: LA CLASE JAVA DE LA ACTIVITY SECUNDARIA	14
4. CONTINUACIÓN CASO PRÁCTICO 1: LLAMAR A UNA ACTIVITY DE OTRA APLICACIÓN: LA CALCULADORA	15
4.1. LLAMAR A LA CALCULADORA: EL XML DE LA ACTIVITY PRINCIPAL	16
4.2. LLAMAR A LA CALCULADORA: LA CLASE JAVA DE LA ACTIVITY PRINCIPAL	17

Introducción

- ✓ Un **Intent** es lo que le permite a una aplicación manifestar la "intención" de que desea hacer algo. Por ejemplo:
 - Abrir una nueva actividad (pantalla), de la misma o distinta aplicación.
 - Pasar datos de una activity a otra, de la misma o distinta aplicación.
 - Interconectar componentes de la misma o distinta aplicación a través de mensajes.
- ✓ Los **Intents** pueden iniciar componentes de 2 formas:
 - **Explícita**: llamando a la clase Java del componente que queremos lanzar/ejecutar.
 - **Implícita**: invocando una **acción** y los datos a aplicar para esa acción. Por ejemplo: acción: ver, datos: url de google. Android informará de las actividades receptoras que pueden procesar esa acción con los datos ajuntados.
- ✓ Cuando se crea una aplicación se crea una Activity *principal* (La pantalla *principal*).
- ✓ Desde esta Activity se puede llamar a otras Activities (pantallas) de la propia aplicación o de otras.
- ✓ Para lanzar una activity desde otra se usa la clase **Intent**.
- ✓ Un Intent permite llamar a otras Activities, pasarles y recoger información, interactuar con servicio del sistema, etc.
- ✓ Referencias:

- <http://developer.android.com/training/basics/firstapp/starting-activity.html>
- <http://developer.android.com/reference/android/content/Intent.html>
- <http://developer.android.com/guide/components/intents-filters.html>
- ✓ También veremos un poco más a fondo el fichero **AndroidManifest.xml** para:
 - Indicar las actividades de las que se compone la aplicación y cuál es la principal.
 - Los recursos (**permisos**) que va usar nuestra aplicación.

1. Intents explícitos

- ✓ Los intents explícitos son la forma que tenemos de 'llamar', ósea 'abrir' otra activity por su nombre.

Su nombre está definido en el archivo **AndroidManifest.xml** pero si estamos en un proyecto y queremos llamar a activities definidas dentro del mismo, también podemos hacer uso del nombre de la clase de la forma: NombreClaseActivity.class (lo veremos en los ejemplos).

- ✓ Vamos a comenzar usando los Intents de forma explícita, primero llamando a una segunda activity que vamos a crear nosotros y luego llamando a la **calculadora**.
- ✓ Finalmente llamaremos a distintas activities, como contactos, navegador web, etc de forma **implícita**.
- ✓ En cualquiera de los casos vamos a pasar información entre las Activities, bien desde la que llama a la llamada, como viceversa.
- ✓ Vamos comenzar creando un Aplicación con 2 Activities: una principal y otra secundaria.
- ✓ Luego lanzaremos la **calculadora** del sistema.

2. Caso Práctico 1: Lanzar una segunda Activity propia de la aplicación

- ✓ Crearemos un nuevo paquete de nombre: **Intents**.
- ✓ Dentro del paquete **Intents** crear una nueva 'Empty Activity' de nombre: **UD06_01_Intents** de tipo Launcher. Modificar el archivo **AndroidManifest.xml** y añadir una label a la activity.

- ✓ Dentro del paquete **Intents** crear una nueva 'Empty Activity' sin ser de tipo Launcher de nombre: **UD06_01_Segunda_Intents no Launcher**. Modificar el archivo **AndroidManifest.xml** y añadir una etiqueta a la activity.

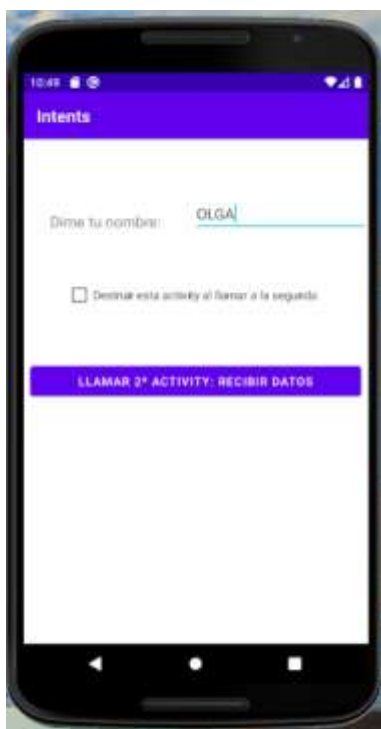
- ✓ Como ya comentamos anteriormente al desmarcar la opción **Launcher** lleva consigo que en el archivo AndroidManifest.xml no se cree la sección 'Intent-filter' como vemos en la siguiente imagen:

```
<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/Theme.Intents">
    <activity android:name=".UD06_01_Segunda_Intents"
        android:label="UD06_01_Segunda_Intents">
    </activity>
    <activity android:name=".UD06_01_A1_Intents">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

- ✓ Esta práctica va a constar de dos activities en la que una de ellas llamará a la otra:

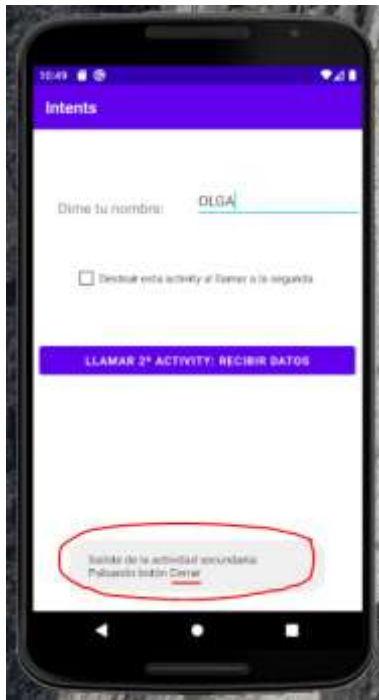
Llamadas entre Activities



Desde la pantalla principal se va introducir un nombre. Cuando se pulsa el botón se llamará a la segunda activity que recibirá el valor del cuadro de texto. Además en

Segunda pantalla que recibe la información de la primera y además tiene un botón para destruir esta segunda Activity ...

esta primera pantalla tenemos la opción de destruir esa activity en el momento de llamar a la segunda. Se aconseja experimentar con esa opción de usar el botón **Back** una vez que se está en la segunda Activity.



... se pulsamos el botón **Cerrar**, se lanza un Toast desde el método **finish()** de la segunda Activity que indica si la segunda Activity fue cerrada desde el botón **Cerrar** o....

... desde el botón **Retroceso/Back** de la botonera.

2.1. El XML del layout de la Activity principal

```
<?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:layout_height="match_parent"
    tools:context=".UD06_01_A1_Intents">

    <TextView
        android:id="@+id/tvNombre_UD06_01_A1_Intent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="8dp"
        android:text="Dime tu nombre:"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.137" />

    <EditText
        android:id="@+id/etNombre_UD06_01_A1_Intents"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="44dp"
        android:ems="10"
        android:hint="Nombre a enviar"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.12" />

<CheckBox
    android:id="@+id/ckDestruir_UD06_01_A1_Intents"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:text="Destruir esta activity al llamar a la segunda"
    android:textSize="14sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.292"
    tools:ignore="UnknownId" />

<Button
    android:id="@+id/btnLanzar_UD06_01_A1_Intent"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:text="Llamar 2ª Activity: Recibir Datos"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/ckDestruir_UD06_01_A1_Intents"
    app:layout_constraintVertical_bias="0.262" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

2.2. La clase Java de la activity principal

```

package com.example.intents;

import androidx.appcompat.app.AppCompatActivity;

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

public class UD06_01_A1_Intents extends AppCompatActivity {

    public final static String NOMBRE = "nombre"; // Es el nombre de la variable

    private void gestionarEventos(){

        Button btnLanzarActivity = findViewById(R.id.btnLanzar_UD06_01_A1_Intent);
        btnLanzarActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CheckBox chkDestruir = findViewById(R.id.chkDestruir_UD06_01_A1_Intents);
            }
        });
    }
}

```

```

        EditText etNombre = findViewById(R.id.etNombre_UD06_01_A1_Intents);

        Intent intent = new Intent(getApplicationContext(), UD06_01_A2_Intents.class);
        intent.putExtra(NOMBRE, etNombre.getText().toString());

        startActivity(intent);

        if (chkDestruir.isChecked()) {
            finish();
        }

    });

}

public void finish() {
    super.finish();
    Toast.makeText(this, "Mataste la actividad principal", Toast.LENGTH_SHORT).show();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_u_d06_01_a1_intents);

    gestionarEventos();
}
}

```

- ✓ **Línea 14:** Definimos una constante de tipo String, llamada NOMBRE. Observar que é pública. Esto lo hacemos para que desde la segunda activity pueda acceder a ella. O objetivo es que el nombre del parámetro que estamos utilizando para enviar el dato, nos aseguremos que sea el mismo.
- ✓ **Línea 27:** Creamos un objeto de tipo **Intent**. El constructor recibe 2 parámetros:
 - El primer parámetro es una referencia al contexto. No podemos poner 'this' ya que estamos dentro de una clase anónima y por tanto this no es una activity. Por eso ponemos getApplicationContext().
 - El segundo parámetro es la clase que el sistema 'intentará' cargar (en nuestro caso el nombre de la clase asociada a la Activity que deseamos cargar).
- ✓ **Línea 28:** Asignamos al intent un par CLAVE-VALOR a través del método **putExtra()**. En este caso un valor String a través de NOMBRE, definido en la línea 14.
 - También podremos no usar la constante: intent.putExtra("nombre", etNome.getText().toString())
- ✓ **Línea 30:** Lanzamos la Activity.
- ✓ **Líneas 32-34:** Si el CheckBox del Layout está marcado entonces destruimos esta Activity:

Fijarse por tanto que a pesar de que llamamos a la segunda Activity, el código se sigue ejecutando. Si no queremos que eso sea así tendremos que emplear **return**; para que salga del método.

- ✓ Realizar pruebas marcando y desmarcando esta marca y pulsando después el botón **Back** de la Segunda Activity.

Nota: Indicar que también podemos lanzar la segunda activity con este código:

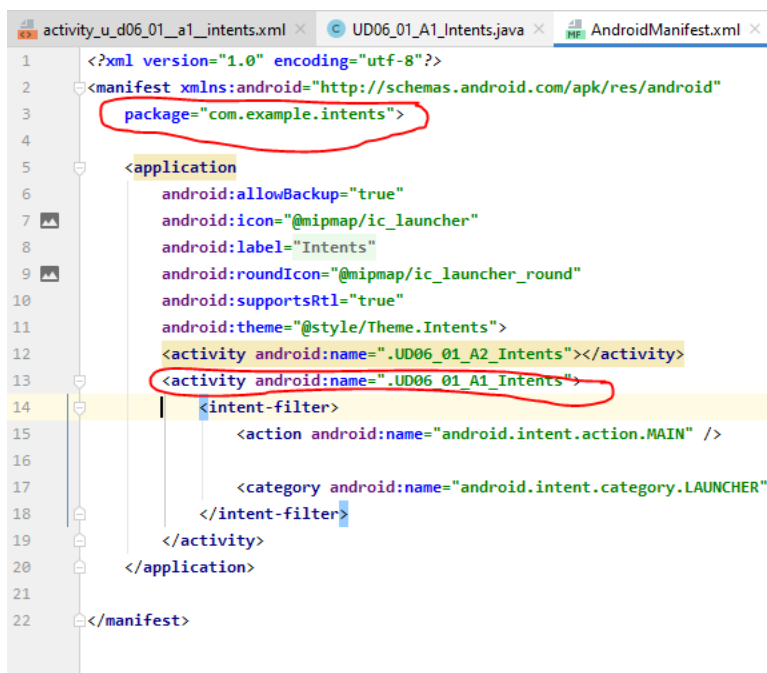
```
Intent intent = new Intent();
intent.setClassName("com.example.intents", "com.example.intents.UD06_01_A2_Intents");
```

Siendo ' com.example.intents' el nombre del paquete definido en el AndroidManifest.xml y ' com.example.intents.UD06_01_A2_Intents' el nombre del paquete junto con el nombre de la Activity también definido en el AndroidManifest.xml.

O con este otro:

```
intent.setClassName(getApplicationContext(), "com.example.intents", "com.example.intents.UD06_01_A2_Intents");
```

- ✓ Los datos anteriores los obtenemos del **AndroidManifest.xml**:



Veremos más adelante, como llamar a otras aplicaciones instaladas en el S.O. Android empleándolo de esta forma.

2.3. El XML de la Activity que recibe la llamada

```
<?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:layout_height="match_parent"
tools:context=".UD06_01_A2_Intents">
```

```

<TextView
    android:id="@+id/tvResultado_UD06_01_A2_Intent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:contentDescription="Nombre enviado"
    android:text="Resultado"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btnCerrar_UD06_01_A2_Intent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:text="Cerrar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.511"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/tvResultado_UD06_01_A2_Intent"
    app:layout_constraintVertical_bias="0.272" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

2.4. La clase Java de la Activity que recibe la llamada. Activity Secundaria.

```

package com.example.intents;

import androidx.appcompat.app.AppCompatActivity;

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

public class UD06_01_A2_Intents extends AppCompatActivity {

    private String formaCerrarActivity =null;

    private void gestionarEventos(){
        findViewById(R.id.btnCerrar_UD06_01_A2_Intent).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formaCerrarActivity = " Pulsando botón Cerrar";

                EditText et_apellido = (EditText) findViewById(R.id.etApellido_UD06_01_A2_Intents);

                finish();
            }
        });
    }

    public void finish() {
        super.finish(); // IMPORTANTE. SIEMPRE HAY QUE LLAMAR A SUPER.XXXX en cualquier

```



```

método del ciclo de vida de la activity
// que sobrecarguemos

    if (formaCerrarActivity == null) { // Quiere decir que hemos pulsado el botón atrás, no el
botón cerrar, ya que si pulsamos
// el botón Cerrar le damos un valor a la variable
        formaCerrarActivity = " Pulsando botonera Retroceso";
    }
    Toast.makeText(this, "Saliste de la actividad secundaria: " + formaCerrarActivity, To-
ast.LENGTH_SHORT).show();

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_u_d06_01_a2_intents);

    // Recuperamos el nombre enviado desde la Activity 1
    TextView tvResultado = (TextView) findViewById(R.id.tvResultado_UD06_01_A2_Intent);
    Intent intent = getIntent();
    tvResultado.setText("Hola " + intent.getExtras().getString(UD06_01_A1_Intents.NOMBRE));

    gestionarEventos();
}
}

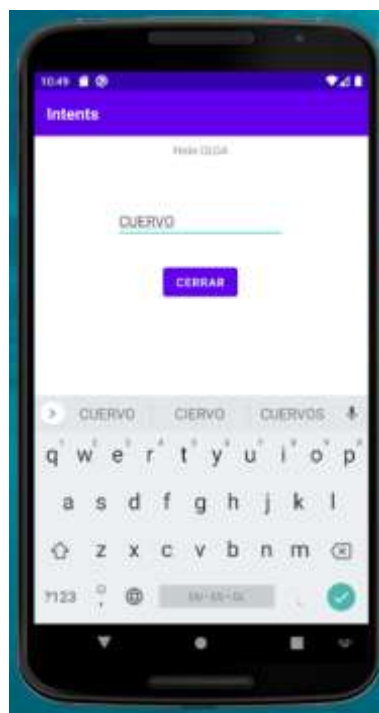
```

- ✓ Para recuperar la información en la segunda Activity hay que realizarlo cuando se está creando la Activity, puede ser en un método aparte llamado desde el onCreate() o en el propio onCreate();
- ✓ **Línea 14:** Variable tipo String para guardar la forma en la que se sale de la segunda actividad: Botón **Cerrar** o tecla **Back**.
- ✓ **Línea 22:** En caso de pulsar el botón 'Cerrar' la variable anterior tiene un texto indicándolo (entonces ya no vale null).
- ✓ **Línea 26:** Llamamos al método finish() para que cierre la activity.
- ✓ **Líneas 35-37:** En caso de que la variable formaCerrarActivity==null quiere decir que estamos destruyendo la activity pero no a través del botón, si no pulsando el botón Back.
- ✓ **Línea 49:** Recuperamos el intent enviado desde la activity principal.
- ✓ **Línea 50:** Cogemos la información que venga en NOMBRE y mostrándola en una etiqueta de texto.

3. Continuación caso Práctico 1: Pasar datos de la actividad secundaria a la principal

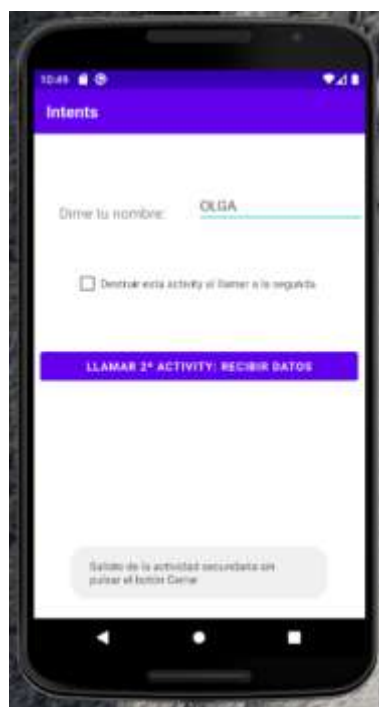
- ✓ Hasta ahora hemos pasado datos de la actividad principal a la secundaria.
- ✓ Vamos a modificar la aplicación anterior, para que en la secundaria se pueda introducir el apellido de una persona y este se pase a la actividad principal.
- ✓ Además vamos a controlar en la Actividad Principal si se vuelve de la actividad secundaria pulsando el botón **Cerrar** o la botonera **Back**.

Paso de valores de la Activity Secundaria a la Principal



Igual que antes desde la pantalla principal se va a introducir un nombre. Pulsar en el botón que lanza la segunda actividad.

Ahora aparte de recibir el Nombre de la Actividad Principal, podemos escribir en la Actividad Secundaria los apellidos de la persona y pulsar en Cerrar.



Observar como la actividad Principal muestra los apellidos introducidos en la Actividad Secundaria.

En esta ocasión se retornó de la actividad secundaria a la Principal pulsando en la botone-
ra **Back**.

✓ A continuación se van a ver los cambios introducidos en el código anterior.

3.1. Pasar datos de la secundaria a la principal: La clase Java de la Activity principal

- ✓ A continuación se va a indicar el código que se añadió o modificó con respecto al anterior.
- ✓ Ahora la actividad secundaria se llamará con el método **startActivityResult(intent, número)**. Este método le indica que esperamos que nos devuelva un resultado cuando esta termine.
- ✓ El número que le pasamos a la actividad secundaria es el que nos va a devolver esta cuando se cierre y así cuando se vuelva a la actividad principal, en esta, con el método **onActivityResult(int requestCode, int resultCode, Intent data)** podemos comprobar que número nos devuelve a la actividad que nos pasa el control y actuar en consecuencia.

✓ Referencias:

- <http://developer.android.com/reference/android/app/Activity.html#startActivityResult%28android.content.Intent,%20int%29>
- <http://developer.android.com/reference/android/app/Activity.html#onActivityResult%28int,%20int,%20android.content.Intent%29>

```
package com.example.intents;

import androidx.appcompat.app.AppCompatActivity;

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

public class UD06_01_A1_Intents extends AppCompatActivity {

    public final static String NOMBRE = "nombre"; // Es el nombre de la variable
    public final static String APELLIDO = "apellido"; // Es el nombre de la variable
    private final int COD_PETICION = 33;

    private void gestionarEventos() {

        Button btnLanzarActivity = findViewById(R.id.btnLanzar_UD06_01_A1_Intents);
        btnLanzarActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CheckBox chkDestruir = findViewById(R.id.chkDestruir_UD06_01_A1_Intents);
                EditText etNombre = findViewById(R.id.etNombre_UD06_01_A1_Intents);

                Intent intent = new Intent(getApplicationContext(),
                UD06_01_A2_Intents.class);
                intent.putExtra(NOMBRE, etNombre.getText().toString());

                //startActivity(intent);
                startActivityForResult(intent, COD_PETICION);

                if (chkDestruir.isChecked()) {
                    finish();
                }
            }
        });
    }
}
```

```

    });

}

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == COD_PETICION) {
        if (resultCode == RESULT_OK) {
            if (data.hasExtra(UD06_01_A1_Intents.APELLIDO)) {
                EditText etNome = findViewById(R.id.etNombre_UD06_01_A1_Intents);
                Toast.makeText(this, etNome.getText() + "\ny tu apellido es: " +
                    data.getExtras().getString(UD06_01_A1_Intents.APELLIDO), To-
ast.LENGTH_SHORT).show();
            }
        } else
            Toast.makeText(this, "Saliste de la actividad secundaria sin pulsar el
botón Cerrar", Toast.LENGTH_SHORT).show();
    }
}

public void finish() {
    super.finish();
    Toast.makeText(this, "Mataste la actividad principal", Toast.LENGTH_SHORT).show();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_u_d06_01__a1__intents);

    gestionarEventos();
}
}

```

- ✓ **Línea 17:** Al igual que el caso de 'nombre', tenemos otra variable que guardará el apellido enviado desde la activity segunda a la primera.
- ✓ **Línea 18:** Creamos una constante entera y le asignamos un número cualquiera, que se le va a pasar a la actividad secundaria cuando se lance y que ella nos va a devolver cuando se cierre.
- ✓ **Línea 33:** Llamamos al intent asociado a la actividad secundaria haciendo uso del método **startActivityForResult**, pasándole un número que luego nos devolverá cuando esta se cierre y devuelva el control a la actividad principal. La diferencia de la forma anterior, llamando a este método indicamos que la activity a la que llamamos va a devolver información, la cual será recogida en el método onActivityResult.
- ✓ **Línea 43:** el método `onActivityResult (int requestCode, int resultCode, Intent data)` se activa cuando se vuelve de una actividad secundaria.
 - En **requestCode**: recogemos el código que nos envía esa actividad secundaria, y así controlando ese código podemos saber de qué actividad secundaria se regreso a la principal.
 - En **resultCode**: recibimos el código que nos pasaron desde la actividad secundaria:
 - `public static final int RESULT_CANCELED= 0 (0x00000000)`. Para cuando queremos indicar que en la actividad secundaria se cancelo algo.

- `public static final int RESULT_OK= -1 (0xffffffff)`. Para cuando queramos indicar que lo que se tiene que hacer en la activity secundaria se hizo correctamente.
- **data**: recibimos el intent que nos envía la clase secundaria, y que podemos ver si nos pasa algún valor del estilo CLAVE-VALOR, url, etc.
- ✓ **Línea 45**: Comprobamos si la vuelta a la actividad principal y por la actividad secundaria: **RecibirDatos** que llamamos antes con el número 33.
- ✓ **Línea 46**: Comprobamos si en la activity secundaria se concluye de forma exitosa. Si no es así, una de las razones por las que no se pudo terminar fue porque se pulso el botón **Back** (línea 54).
- ✓ **Línea 47**: comprobamos si el intent **data** tiene un campo **APELLIDO** de ser así ...
- ✓ **Línea 49**: ... extraeremos su valor y los mostraremos a través de un Toast.

3.2. Pasar datos de la secundaria a la principal: El XML de la activity secundaria

- ✓ Solo se añadió un EditText.

```
<?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:layout_height="match_parent"
tools:context=".UD06_01_A2_Intents">

    <TextView
        android:id="@+id/tvResultado_UD06_01_A2_Intent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:contentDescription="Nombre enviado"
        android:text="Resultado"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btnCerrar_UD06_01_A2_Intent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:text="Cerrar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.511"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/tvResultado_UD06_01_A2_Intent"
        app:layout_constraintVertical_bias="0.272" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```

<EditText
    android:id="@+id/etApellido_UD06_01_A2_Intents"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="56dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:hint="Apellido"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvResultado_UD06_01_A2_Intent" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.3. Pasar datos de la secundaria a la principal: La clase Java de la Activity secundaria

```

package com.example.intents;

import androidx.appcompat.app.AppCompatActivity;

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

public class UD06_01_A2_Intents extends AppCompatActivity {

    private String formaCerrarActivity = null;

    private void gestionarEventos() {
        findViewById(R.id.btnCerrar_UD06_01_A2_Intent).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                formaCerrarActivity = " Pulsando botón Cerrar";

                EditText et_apellido = (EditText) findViewById(R.id.etApellido_UD06_01_A2_Intents);

                Intent datos_vuelta = new Intent();
                datos_vuelta.putExtra(UD06_01_A1_Intents.APELLIDO,
et_apellido.getText().toString());
                setResult(RESULT_OK, datos_vuelta);

                finish();
            }
        });
    }

    public void finish() {
        super.finish(); // IMPORTANTE. SIEMPRE HAY QUE LLAMAR A SUPER.XXXX en cualquier
método del ciclo de vida de la activity
                        // que sobrecarguemos
    }
}

```

```

    /* if (formaCerrarActivity ==null){ // Quiere decir que hemos pulsado el botón atrás, no
    el botón cerrar, ya que si pulsamos
        // el botón Cerrar le damos un valor a la variable
        formaCerrarActivity = " Pulsando botonera Retroceso";
    }
    Toast.makeText(this, "Saliste de la actividad secundaria: " + formaCerrarActivity, To-
    ast.LENGTH_SHORT).show();*/

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_u_d06_01_a2_intents);

    // Recuperamos el nombre enviado desde la Activity 1
    TextView tvResultado = (TextView) findViewById(R.id.tvResultado_UD06_01_A2_Intent);
    Intent intent = getIntent();
    tvResultado.setText("Hola " + intent.getExtras().getString(UD06_01_A1_Intents.NOMBRE));

    gestionarEventos();
}
}

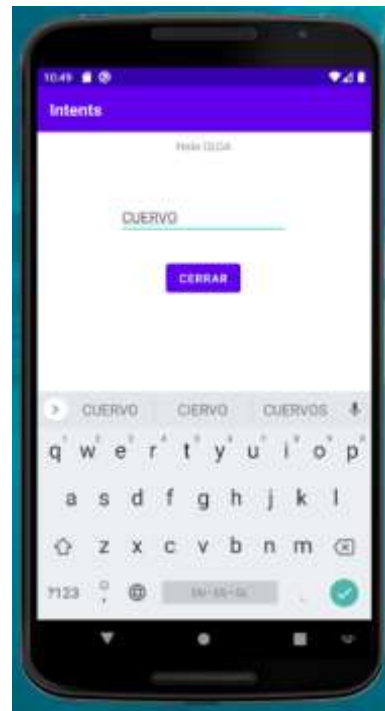
```

- ✓ **Líneas 14, 20, 36-41:** Comentar que ahora en la actividad principal ya podemos saber si el usuario pulso o no el botón Cerrar o Back por el código de vuelta de la activity.
- ✓ **Líneas 24:** Creamos un nuevo intent.
- ✓ **Líneas 25:** Añadimos datos extendidos a el intent, en este caso el par: APELLIDO-Valor.
- ✓ **Líneas 26:** setResult le envía el resultado a la activity llamadora con el código RESULT_OK. Podríamos enviar el código RESULT_CANCELED nuestro otro supuesto programado por nosotros.

4. Continuación caso práctico 1: Llamar a una activity de otra aplicación: la calculadora

- ✓ A continuación vamos a añadir más código a las clases y xmls anteriores para poder llamar a una activity de otra aplicación, en este caso del sistema.
- ✓ Vamos a hacerlo de forma explícita.

Llamar a la calculadora



Pulsar en el botón calculadora.

Se abre la Activity de otra aplicación. En este caso la calculadora.

4.1. Llamar a la calculadora: el xml de la activity principal.

✓ Simplemente añadimos un nuevo botón (recuadro rojo).

```
<?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:layout_height="match_parent"
    tools:context=".UD06_01_A1_Intents">

    <TextView
        android:id="@+id/tvNombre_UD06_01_A1_Intent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="8dp"
        android:text="Dime tu nombre:"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.137" />

    <EditText
        android:id="@+id/etNombre_UD06_01_A1_Intents"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="44dp"
        android:ems="10"
```



```

    android:hint="Nombre a enviar"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.12" />

<CheckBox
    android:id="@+id/ckDestruir_UD06_01_A1_Intents"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:text="Destruir esta activity al llamar a la segunda"
    android:textSize="14sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.292"
    tools:ignore="UnknownId" />

<Button
    android:id="@+id/btnLanzar_UD06_01_A1_Intent"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:text="Llamar 2ª Activity: Recibir Datos"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/ckDestruir_UD06_01_A1_Intents"
    app:layout_constraintVertical_bias="0.262" />

```

```

<Button
    android:id="@+id/btnCalculadora_UD06_01_A1_Intents"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="Lanzar Calculadora"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnLanzar_UD06_01_A1_Intent" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

4.2. Llamar a la calculadora: La clase java de la activity principal.

✓ Añadimos el código para procesar el evento onClick del botón.

```

package com.example.intents;

import androidx.appcompat.app.AppCompatActivity;

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

```

```

import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;

public class UD06_01_A1_Intents extends AppCompatActivity {

    public final static String NOMBRE = "nombre";           // Es el nombre de la variable
    public final static String APELLIDO = "apellido";       // Es el nombre de la variable
    private final int COD_PETICION = 33;

    private void gestionarEventos(){

        Button btnLanzarActivity = findViewById(R.id.btnLanzar_UD06_01_A1_Intent);
        btnLanzarActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CheckBox chkDestruir = findViewById(R.id.chkDestruir_UD06_01_A1_Intents);
                EditText etNombre = findViewById(R.id.etNombre_UD06_01_A1_Intents);

                Intent intent = new Intent(getApplicationContext(), UD06_01_A2_Intents.class);
                intent.putExtra(NOMBRE, etNombre.getText().toString());

                //startActivity(intent);
                startActivityForResult(intent, COD_PETICION);

                if (chkDestruir.isChecked()) {
                    finish();
                }
            }
        });

        //Botón para lanzar la calculadora del S.O.
        Button btnCalculadora = findViewById(R.id.btnCalculadora_UD06_01_A1_Intents);
        btnCalculadora.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setClassName("com.android.calculator2", "com-
m.android.calculator2.Calculator");
                startActivity(intent);
            }
        });
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == COD_PETICION) {
            if (resultCode == RESULT_OK) {
                if (data.hasExtra(UD06_01_A1_Intents.APELLIDO)) {
                    EditText etNome = findViewById(R.id.etNombre_UD06_01_A1_Intents);
                    Toast.makeText(this, etNome.getText() + "\ny tu apellido es: " +
                        data.getExtras().getString(UD06_01_A1_Intents.APELLIDO), To-
ast.LENGTH_SHORT).show();
                }

                } else
                Toast.makeText(this, "Saliste de la actividad secundaria sin pulsar el botón
Cerrar", Toast.LENGTH_SHORT).show();
            }
        }

        public void finish() {
            super.finish();
            Toast.makeText(this, "Mataste la actividad principal", Toast.LENGTH_SHORT).show();
        }
    }
}

```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_u_d06_01__a1__intents);

    gestionarEventos();
}
}
```