

# 1 Intents explícitos

## 1.1 Sumario

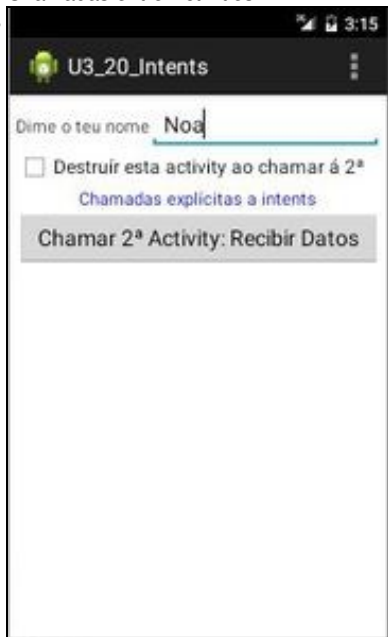
- 1 Intents explícitos
- 2 Lanzar unha segunda Activity propia da aplicación
- 3 Creación dunha segunda Activity
  - ◆ 3.1 O XML do layout da Activity principal
  - ◆ 3.2 A clase Java a activity principal
  - ◆ 3.3 O XML da Activity que recibe a chamada
  - ◆ 3.4 A clase Java da Activity que recibe a chamada. Activity Secundaria.
- 4 Pasar datos da actividade secundaria á principal
  - ◆ 4.1 Pasar datos da secundaria á principal: A clase Java da Activity principal
  - ◆ 4.2 Pasar datos da secundaria á principal: O XML da activity secundaria
  - ◆ 4.3 Pasar datos da secundaria á principal: A clase Java da Activity secundaria
- 5 Chamar a unha activity doutra aplicación: á calculadora
  - ◆ 5.1 Chamar á calculadora: o xml da activity principal.
  - ◆ 5.2 Chamar á calculadora: A clase java da activity principal.

## 1.2 Intents explícitos

- Crear o proxecto: **U3\_20\_Intents**.
- Imos comezar usando os Intents de forma **explícita**, primeiro chamando a unha segunda activity que imos crear nós e logo chamando á calculadora.
- Finalmente chamaremos a distintas activities, como contactos, navegador web, etc de forma **implícita**.
- En calquera dos casos imos pasar información entre as Activities, ben dende a que chama á chamada, como viceversa.
- Como indicamos imos comezar creando unha Aplicación con 2 Activities: unha principal e outra secundaria.
- Logo lanzaremos a calculadora do sistema.

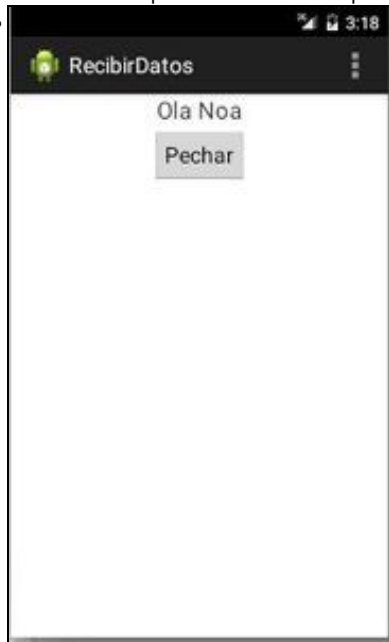
## 1.3 Lanzar unha segunda Activity propia da aplicación

- Crear o proxecto: **U3\_20\_Intents**.
- Chamadas entre Activities

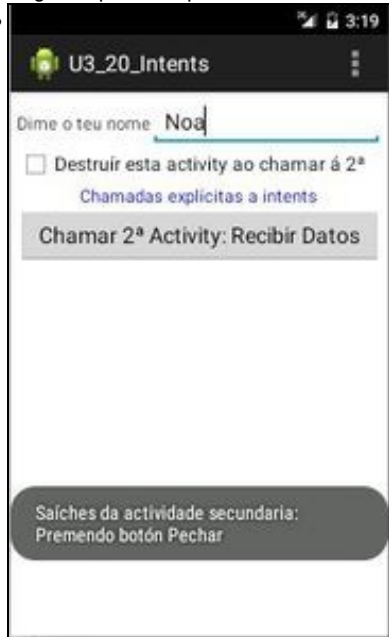


Dende a pantalla principal vaise introducir un nome. Cando se prema o botón chamarase á segunda activity que recibira o valor do cadro de texto. Ademais nesta primeira pantalla temos a opción de destruír esa activity no momento de chamar á segunda.

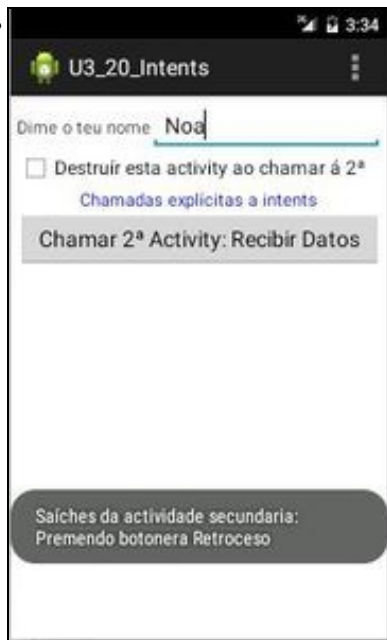
Aconséllase experimentar con esa opción e usar o botón **Back** unha vez que se está na segunda Activity.



Segunda pantalla que recibe a información da primeira e ademais ten un botón para destruír esta segunda Activity ...



... se prememos o botón **Pechar** lánzase un Toast dende o método **finish()** da segunda Activity que indica se a segunda Activity foi pechada dende o botón **Pechar** ou ....

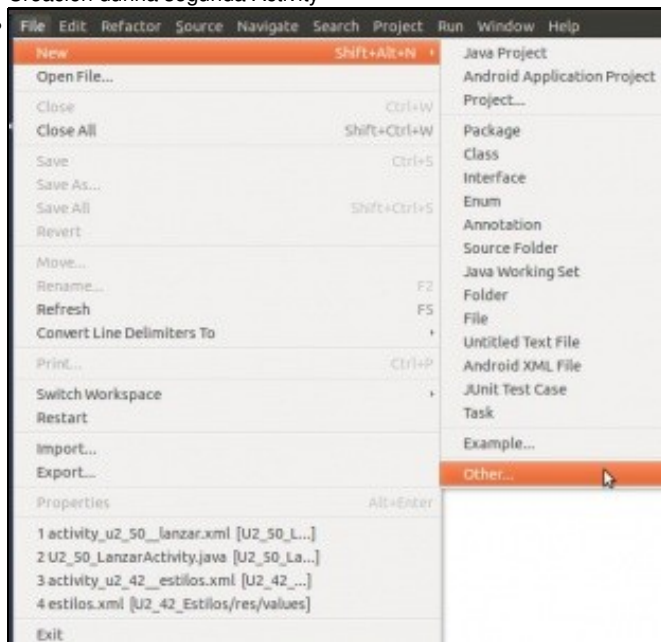


... dende o botón **Retroceso/Back** da botonera.

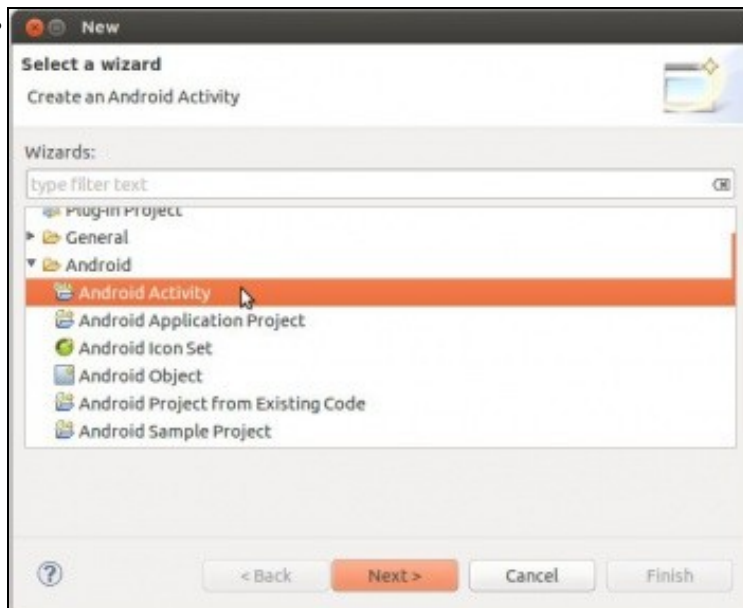
## 1.4 Creación dunha segunda Activity

- Pódese facer manualmente ou a través da utilidade que proporciona o IDE para crear unha Activity.

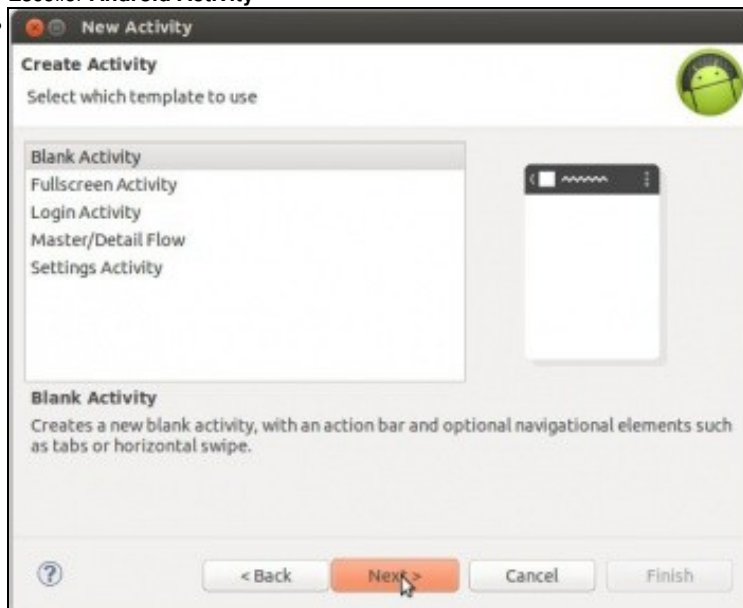
### • Creación dunha segunda Activity



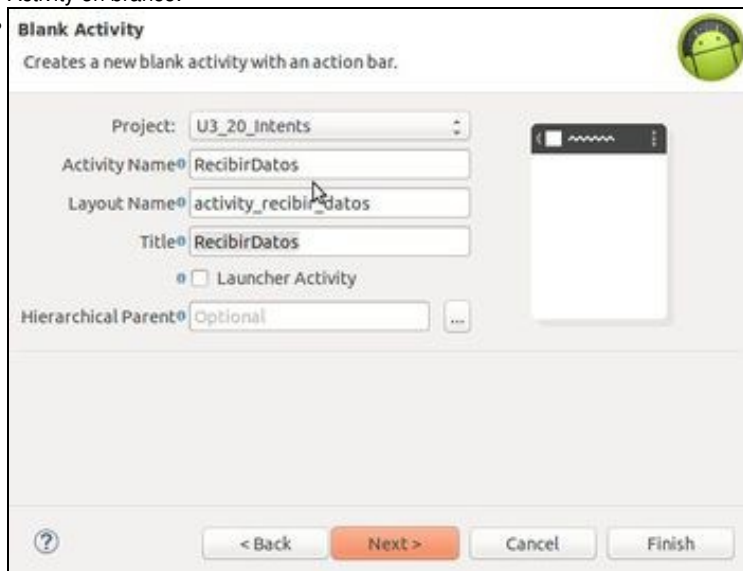
Ir ao menú **File->New->Other**



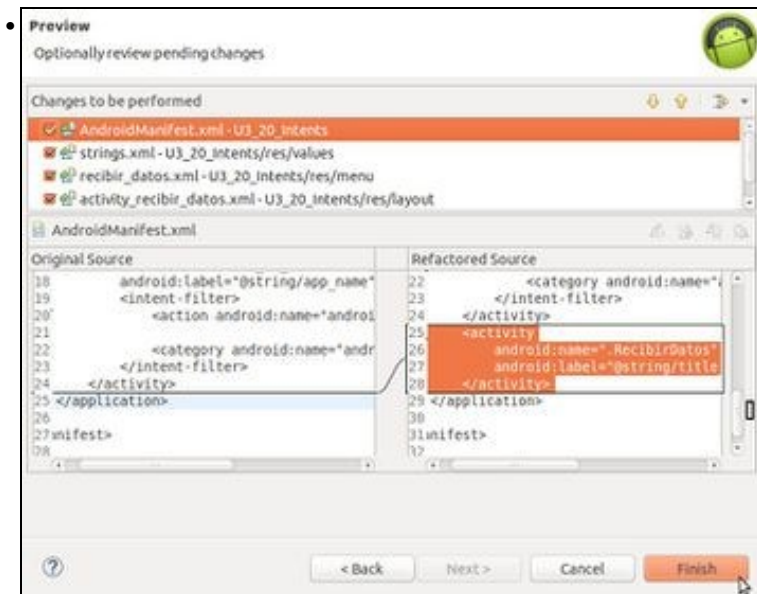
### Escoler **Android Activity**



### Activity en blanco.



Escoler en que proxecto crea e o nome, neste caso: RecibirDatos. Premer **Next**.



Observar como nos indica que vai modificar por nós o Arquivo **AndroidManifest.xml**



Observar que a ferramenta xa nos creou

- un XML e

- unha Clase Java para esa Activity.

Agora só resta codificar e deseñar o layout.

```

• <?xml version="1.0" encoding="utf-8"?>
  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.u3_20_intents"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
      android:minSdkVersion="16"
      android:targetSdkVersion="21" />

    <application
      android:allowBackup="true"
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >
      <activity
        android:name=".U3_20_Intents"
        android:label="@string/app_name" >
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />

          <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
      <activity
        android:name=".RecibirDatos"
        android:label="@string/title_activity_recibir_datos" >
      </activity>
    </application>
  </manifest>

```

O ficheiro **AndroidManifest.xml**, creando a entrada para a nova Activity. Para indicar que agora temos unha nova actividade na aplicación. Indicando o seu nome e etiqueta.

Observar como a Activity principal (U3\_20\_Intents) ten un filtro que indica que é a Activity principal da aplicación e que é a través da que se lanza a Aplicación.

### 1.4.1 O XML do layout da Activity principal

- Observar na **líña 41** que chamamos á segunda actividade co atributo xml: android:onClick

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="5dp" >

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Dime o teu nome" />

    <EditText
      android:id="@+id/et_nome"
      android:layout_width="match_parent"
      android:layout_height="wrap_content" />
  </LinearLayout>

  <CheckBox
    android:id="@+id/chk_destruir"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:checked="false"
    android:text="Destruir esta activity ao chamar á 2ª" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="Chamadas explícitas a intents"
    android:textColor="#00F" />

  <Button
    android:id="@+id/btn_enviar"
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:onClick="onEnviarClick"
        android:text="Chamar 2ª Activity: Recibir Datos" />
</LinearLayout>

```

## 1.4.2 A clase Java a activity principal

```

package com.example.u3_20_intents;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.CheckBox;
import android.widget.TextView;
import android.widget.Toast;

public class U3_20_Intents extends Activity {

    public final static String NOME = "nome";

    TextView tvNome;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u3_20__intents);
        tvNome = (TextView) findViewById(R.id.et_nome);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.u3_20__intents, menu);
        return true;
    }

    public void onEnviarClick(View view) {
        CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);

        Intent intent = new Intent(this, RecibirDatos.class);
        intent.putExtra(NOME, tvNome.getText().toString());

        startActivity(intent);

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

    public void finish() {
        super.finish();
        Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
    }

}

```

- **Liña 14:** Definimos unha constante de tipo `String`, chamada `NOME`. Observar que é pública.
- **Liña 35:** Creamos un obxecto de tipo **Intent**. O construtor recibe 2 parámetros:
  - ◆ O primeiro parámetro é unha referencia ao contexto (a clase `Activity` é unha subclase de `Context`, por iso poñemos `this`).
  - ◆ O segundo parámetro é a clase que o sistema ?intentará? cargar (no noso caso o nome da clase asociada á `Activity` que desexamos cargar).
- **Liña 36:** Asignamos ao `intent` un par `CHAVE-VALOR` a través do método **putExtra()**. Neste caso un valor `String` a través de `NOME`, definido na liña 14.
  - ◆ Tamén poderíamos non usar a constante: `intent.putExtra("nome", tvNome.getText().toString())`
- **Liña 38:** Lanzamos a `Activity`
- **Liña 40-41:** Se o `CheckBox` do `Layout` está marcado entón destruímos esta `Activity`:
  - ◆ Realizar probas marcando e desmarcando esta marca e premendo despois o botón **Back** da Segunda `Activity`..

### 1.4.3 O XML da Activity que recebe a chamada

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tv_resultado"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="resultado"
        android:textSize="20sp" />

    <Button
        android:id="@+id/btn_pechar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:onClick="onPeacharClick"
        android:text="Peachar" />

</LinearLayout>
```

### 1.4.4 A classe Java da Activity que recebe a chamada. Activity Secundaria.

```
package com.example.u3_20_intents;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

public class RecibirDatos extends Activity {
    String Forma_Peachar_Activity;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recibir_datos);
        Forma_Peachar_Activity = " Premendo botonera Retroceso";

        TextView tvResultado = (TextView) findViewById(R.id.tv_resultado);

        Intent intent = getIntent();

        tvResultado.setText("Ola " + intent.getExtras().getString(U3_20_Intents.NOME));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.recibir_datos, menu);
        return true;
    }

    public void onPeacharClick(View view) {
        Forma_Peachar_Activity = " Premendo botón Peachar";

        finish();
    }

    public void finish() {
        super.finish();
        Toast.makeText(this, "Saíches da actividade secundaria: " + Forma_Peachar_Activity, Toast.LENGTH_SHORT).show();
    }
}
```



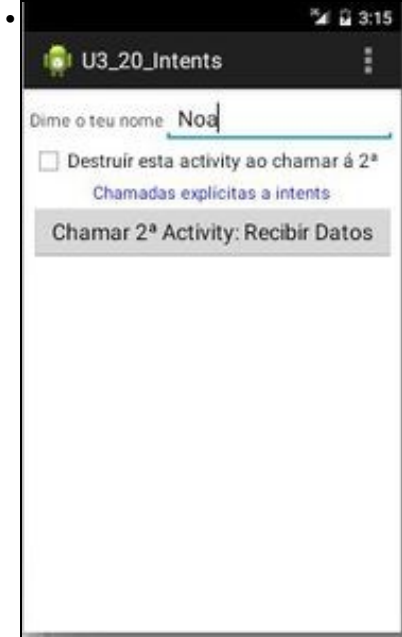
```
}  
  
}
```

- Para recuperar a información na segunda Activity hai que realizalo cando se está creando a Activity, pode ser nun método a parte chamado dende o onCreate() ou no propio onCreate();
- **Liña 11:** Observar o nome da nova Clase.
- **Liña 12:** Variable tipo String para gardar a forma na que se sae da segunda activity: Botón **Pechar** ou tecla **Back**.
- **Liña 18:** Por defecto asignamos a esa variable que se sae da segunda Activity premendo o botón **Back**.
- **Liña 12:** Creamos un intent onde recollemos cal foi o intent que iniciou esta Activity.
- **Liña 24:** Collemos a información que viña en NOME e mostrámola nunha etiqueta de texto.
- **Liña 35-38:** se se preme o botón de pechar destruírase esta Activity e cambiamos o valor da variable String.
- **Liña 42-44:** Como xa se sabe da parte anterior do Ciclo de Vida dunha Activity este método (finish()) execútase cando se destrúe a Activity.

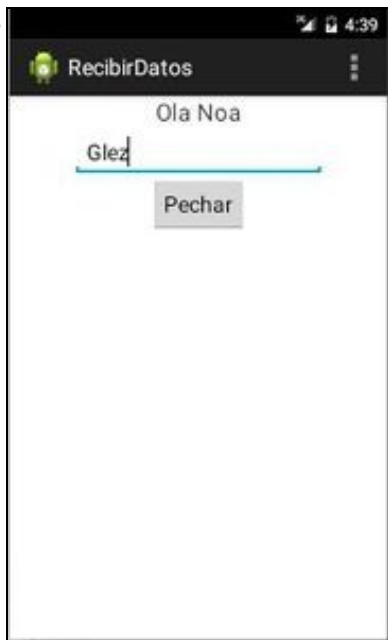
## 1.5 Pasar datos da actividade secundaria á principal

- Até agora pasamos datos da actividade principal á secundaria.
- Imos modificar a aplicación anterior, para que na secundaria se poida introducir o apelido dunha persoa e este se pase á actividade principal.
- Ademais imos controlar na Actividade Principal se se volveu da actividade secundaria premendo o botón **Pechar** ou a botonera **Back**.

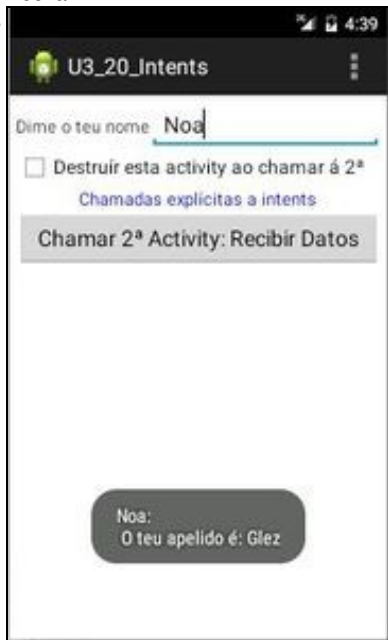
- Paso de valores da Activity Secundaria á Principal



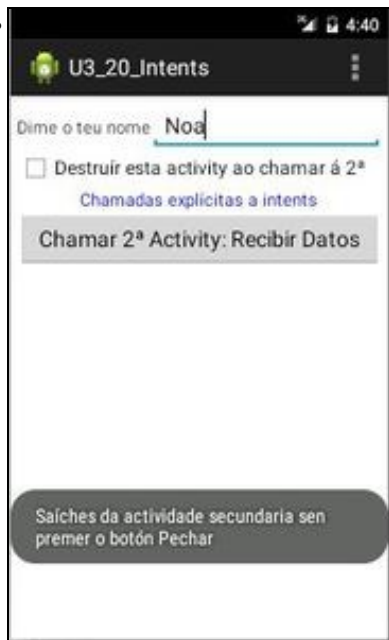
Igual que antes dende a pantalla principal vaise introducir un nome. Premer no botón que lanza a segunda actividade.



Agora aparte de recibir o Nome da Actividade Principal, podemos escribir na Actividade Secundaria os apelidos da persoa e premer en Pegar.



Observar como a actividade Principal amosa os apelidos introducidos na Actividade Secundaria.



Nesta ocasión retornouse da actividade secundaria á Principal premendo na botonera **Back**.

- A continuación vanse ver os cambios introducidos no código anterior.

### 1.5.1 Pasar datos da secundaria á principal: A clase Java da Activity principal

- A continuación vaise indicar o código que se engadiu ou modificou con respecto ao anterior.
- Agora á actividade secundaria chámase con método **startActivityForResult(intent, número)**. Este método indícalle que esperamos que nos devolva un resultado cando esta remate.
- O número que lle pasamos á actividade secundaria é o que nos vai devolver esta cando se peche e así cando se volva á actividade principal, nesta, co método **onActivityResult(int requestCode, int resultCode, Intent data)** podemos comprobar que número nos devolve a actividade que nos pasa o control e actuar en consecuencia.

#### • Referencias:

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

```
package com.example.u3_20_intents;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.CheckBox;
import android.widget.TextView;
import android.widget.Toast;

public class U3_20_Intents extends Activity {

    public final static String NOME = "com.example.NOME";
    private static final int COD_PETICION = 33;

    TextView tvNome;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u3_20__intents);
        tvNome = (TextView) findViewById(R.id.et_nome);
    }
}
```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.u3_20__intents, menu);
return true;
}

public void onEnviarClick(View view) {
CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);

Intent intent = new Intent(this, RecibirDatos.class);
intent.putExtra(NOME, tvNome.getText().toString());

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

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

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

if (requestCode == COD_PETICION) {
if (resultCode == RESULT_OK) {
if (data.hasExtra("APELIDO"))
Toast.makeText(this, tvNome.getText() + "\nO teu apelido é: " + data.getExtras().getString("APELIDO"), Toast.LENGTH_SHORT).show();
} else
Toast.makeText(this, "Saiches da actividade secundaria sen premer o botón Pechar", Toast.LENGTH_SHORT).show();
}

}

public void finish() {
super.finish();
Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
}

}

```

- **Liña 15:** Creamos unha constante enteira e asinámoslle un número calquera, que se lle vai pasar á actividade secundaria cando se chame e que ela nos vai devolver cando se peche.
- **Liña 39:** Comentamos o método co que chamabamos antes á actividade secundaria.
- **Liña 40:** Chamamos ao intent asociado á actividade secundaria, pasándolle un número que logo nos devolverá cando esta se peche e devolva o control á actividade principal.
- **Liña 46:** o método `onActivityResult (int requestCode, int resultCode, Intent data)` actívase cando se volve dunha actividade secundaria.
  - ◆ En **requestCode**: recollemos o código que nos envía esa actividade secundaria, e así controlando ese código podemos saber de que actividade secundaria se regresou á principal.
  - ◆ En **resultCode**: recibimos o código que nos pasaron dende a actividade secundaria:
    - ◇ *public static final int RESULT\_CANCELED= 0 (0x00000000)*. Para cando queremos indicar que na actividade secundaria se cancelou algo.
    - ◇ *public static final int RESULT\_OK= -1 (0xffffffff)*. Para cando queremos indicar que o que se tiña que facer na activity secundaria fíxose correctamente.
  - ◆ **data**: recibimos o intent que nos envía a clase secundaria, e que podemos ver se nos pasa algún valor do estilo CHAVE-VALOR, url, etc.
- **Liña 48:** Comprobamos se á volta á actividade principal é por mor da actividade secundaria: **RecibirDatos** que chamamos antes co número 33.
- **Liña 49:** Comprobamos se na activity secundaria se concluíu dun xeito exitoso. Se non é así, unha das razóns polas que non se puido rematar foi porque se premeu o botón **Back** (liña 54).
- **Liña 50:** comprobamos se o intent **data** ten un campo **APELIDO** de ser así ...
- **Liña 51:** ... extraemos o seu valor e amosamos a través dun Toast.

## 1.5.2 Pasar datos da secundaria á principal: O XML da activity secundaria

- Só se engadiu un EditText.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >

<TextView
    android:id="@+id/tv_resultado"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="resultado"
    android:textSize="20sp" />

<EditText
    android:id="@+id/et_apelido"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:hint="Introduce o teu apelido" />

<Button
    android:id="@+id/btn_pechar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:onClick="onPeacharClick"
    android:text="Peachar" />

</LinearLayout>

```

### 1.5.3 Pasar datos da secundaria á principal: A clase Java da Activity secundaria

```

package com.example.u3_20_intents;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class RecibirDatos extends Activity {
    // String Forma_Peachar_Activity;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recibir_datos);
        // Forma_Peachar_Activity = " Premendo botonera Retroceso";

        TextView tvResultado = (TextView) findViewById(R.id.tv_resultado);

        Intent intent = getIntent();

        tvResultado.setText("Ola " + intent.getExtras().getString(U3_20_Intents.NOME));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.recibir_datos, menu);
        return true;
    }

    public void onPeacharClick(View view) {
        // Forma_Peachar_Activity = " Premendo botón Peachar";
        EditText et_apelido = (EditText) findViewById(R.id.et_apelido);

        Intent datos_volta = new Intent();
        datos_volta.putExtra("APELIDO", et_apelido.getText().toString());
        setResult(RESULT_OK, datos_volta);
    }
}

```

```

finish();

}

public void finish() {
super.finish();
// Toast.makeText(this, "Saiches da actividade secundaria: " +
// Forma_Pechar_Activity, Toast.LENGTH_SHORT).show();
}

}

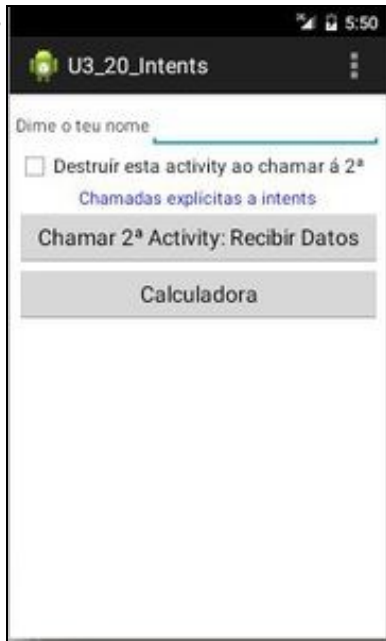
```

- **Liñas 12,18,36,48 e 49:** Comentámolas pois agora na actividade principal xa somos quen de saber se o usuario premeu ou non o botón Pechar ou Back.
- **Liña 39:** Creamos un novo intent.
- **Liña 40:** Engadimos datos extendidos ao intent, neste caso o par: APELIDO-Valor.
- **Liña 41:** setResult envíalle o resultado á activity chamada.

## 1.6 Chamar a unha activity doutra aplicación: á calculadora

- A continuación imos engadir máis código ás clases e xmls anteriores para poder chamar a unha activity doutra aplicación, neste caso do sistema.
- Imos facelo de forma explícita.

- Chamar á calculadora



Premer no botón calculadora.



Ábrese a Activity doutra aplicación. Neste caso a calculadora.

### 1.6.1 Chamar á calculadora: o xml da activity principal.

- Simplemente engadimos un novo botón (Liñas 44-49).

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="5dp" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dime o teu nome" />

        <EditText
            android:id="@+id/et_nome"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </LinearLayout>

    <CheckBox
        android:id="@+id/chk_destruir"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:checked="false"
        android:text="Destruir esta activity ao chamar á 2ª" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Chamadas explícitas a intents"
        android:textColor="#00F" />

    <Button
        android:id="@+id/btn_enviar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onEnviarClick"
        android:text="Chamar 2ª Activity: Recibir Datos" />
```

```

        <Button
            android:id="@+id/btn_calculadora"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:onClick="onCalculadoraClick"
            android:text="Calculadora" />

    </LinearLayout>

```

## 1.6.2 Chamar á calculadora: A clase java da activity principal.

- Engadimos o código para procesar o evento onClick do botón (Liñas 59-65).

```

package com.example.u3_20_intents;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.CheckBox;
import android.widget.TextView;
import android.widget.Toast;

public class U3_20_Intents extends Activity {

    public final static String NOME = "com.example.NOME";
    private static final int COD_PETICION = 33;

    TextView tvNome;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u3_20__intents);
        tvNome = (TextView) findViewById(R.id.et_nome);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.u3_20__intents, menu);
        return true;
    }

    public void onEnviarClick(View view) {
        CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);

        Intent intent = new Intent(this, RecibirDatos.class);
        intent.putExtra(NOME, tvNome.getText().toString());

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

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

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

        if (requestCode == COD_PETICION) {
            if (resultCode == RESULT_OK) {
                if (data.hasExtra("APELIDO"))
                    Toast.makeText(this, tvNome.getText() + "\nO teu apelido é: " + data.getExtras().getString("APELIDO"), Toast.LENGTH_SHORT).show();
            } else
                Toast.makeText(this, "Saíches da actividade secundaria sen premer o botón Pechar", Toast.LENGTH_SHORT).show();
        }
    }
}

```



```
public void onCalculadoraClick(View v) {

    Intent intent = new Intent();
    intent.setClassName("com.android.calculator2", "com.android.calculator2.Calculator");

    startActivity(intent);
}

public void finish() {
    super.finish();
    Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
}

}
```

- **Liña 62:** Nesta ocasión co método **setClassName(String packageName, String className)** indicámoslle no nome do paquete que contén a clase e nome da clase.

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).