

# Intents explícitos

---

## Sumario

---

### Intents explícitos

#### Lanzar unha segunda Activity propia da aplicación

##### Creación dunha segunda Activity

- O XML do layout da Activity principal
- A clase Java a activity principal
- O XML da Activity que recibe a chamada
- A clase Java da Activity que recibe a chamada. Activity Secundaria.

##### Pasar datos da actividade secundaria á principal

- Pasar datos da secundaria á principal: A clase Java da Activity principal
- Pasar datos da secundaria á principal: O XML da activity secundaria
- Pasar datos da secundaria á principal: A clase Java da Activity secundaria

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

- Chamar á calculadora: o xml da activity principal.
- Chamar á calculadora: A clase java da activity principal.

## Intents explícitos

---

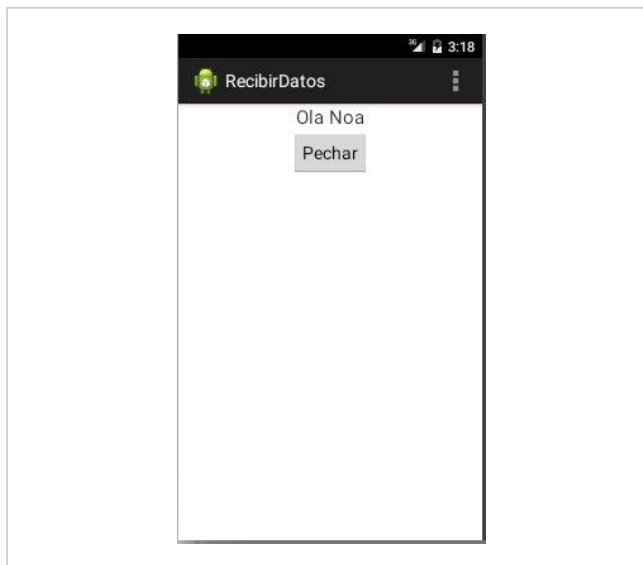
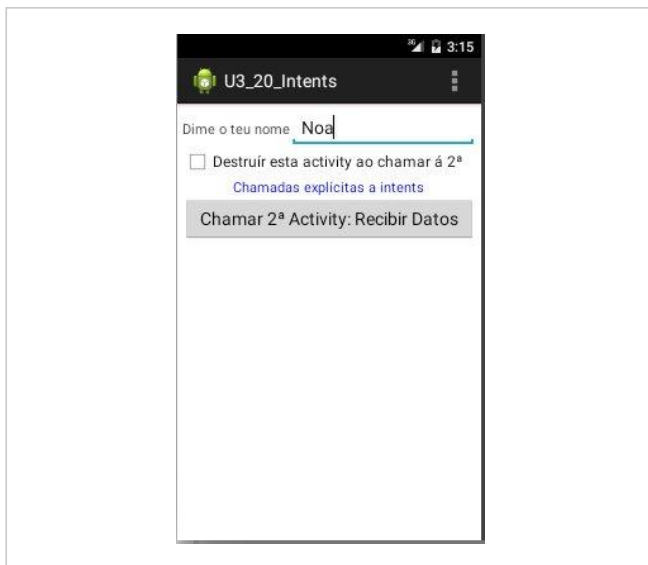
- Crear o proxecto: **U3\_20\_Intents**.
- Imos comezar usando os Intents de forma **explícita**, primeiro chamando a unha segunda activity que imos crear nós e logo chamando á calculadora.
- Finalmente chamaremos a distintas activities, como contactos, navegador web, etc de forma **implícita**.
- En 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.

## Lanzar unha segunda Activity propia da aplicación

---

- Crear o proxecto: **U3\_20\_Intents**.

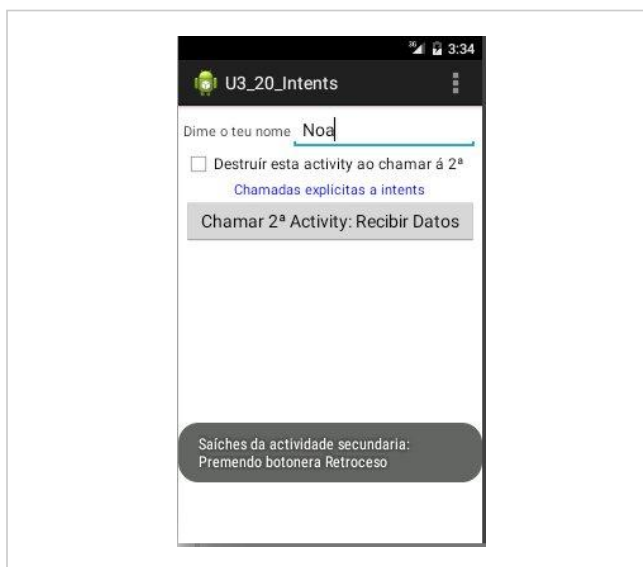
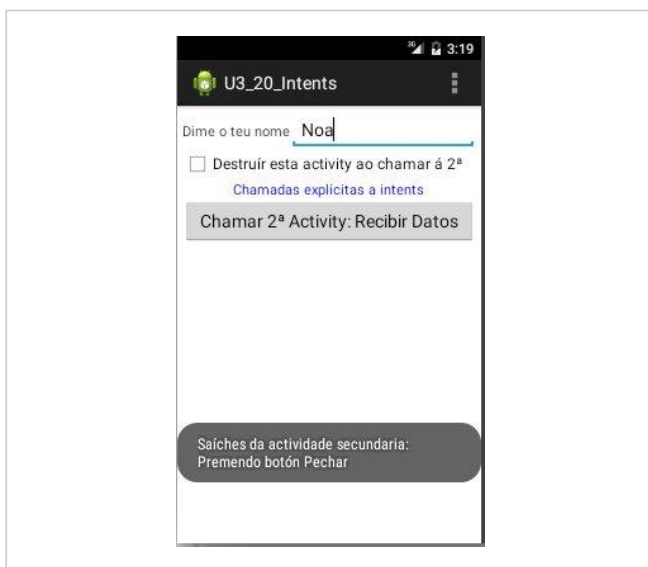
### Chamadas entre Activities



Dende a pantalla principal vaise introducir un nome. Cando se preme 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 ...



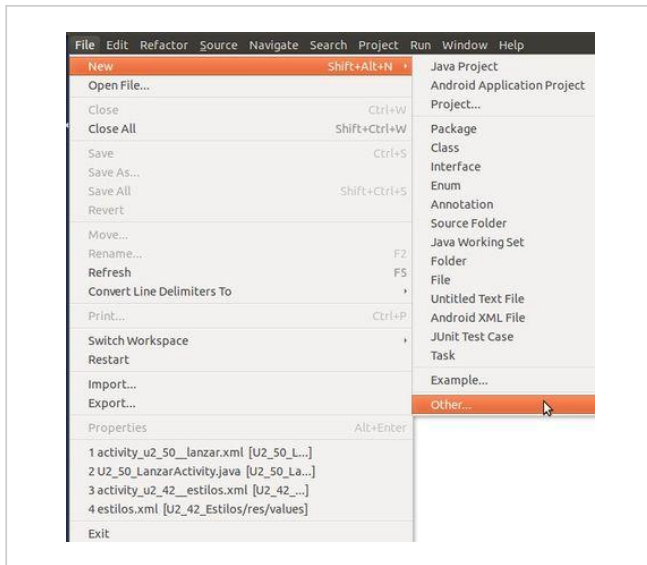
... se prememos o botón **Pegar** lánzase un Toast dende o método **finish()** da segunda Activity que indica se a segunda Activity foi pechada dende o botón **Pegar** ou ....

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

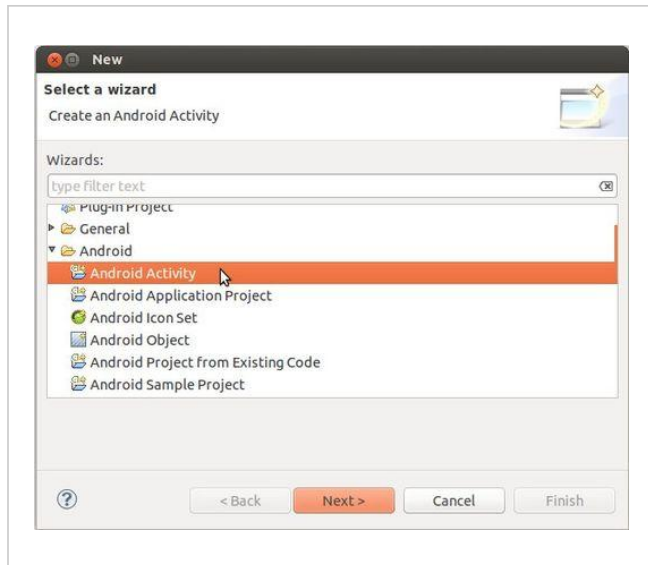
## Creación dunha segunda Activity

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

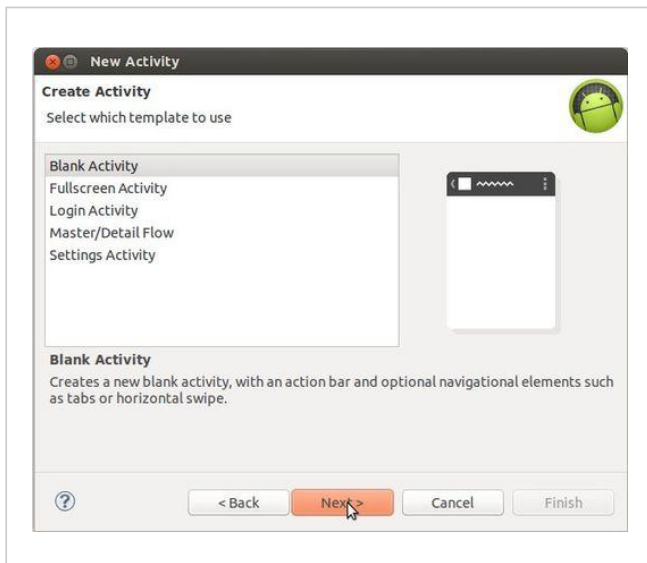
### Creación dunha segunda Activity



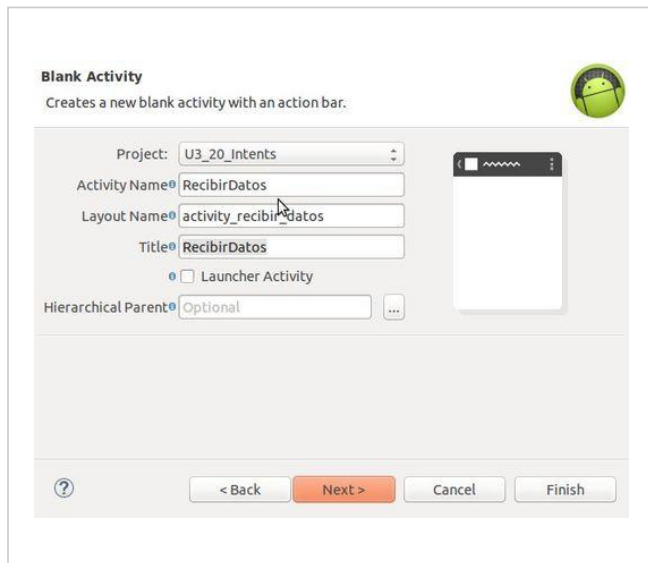
Ir ao menú **File->New->Other**



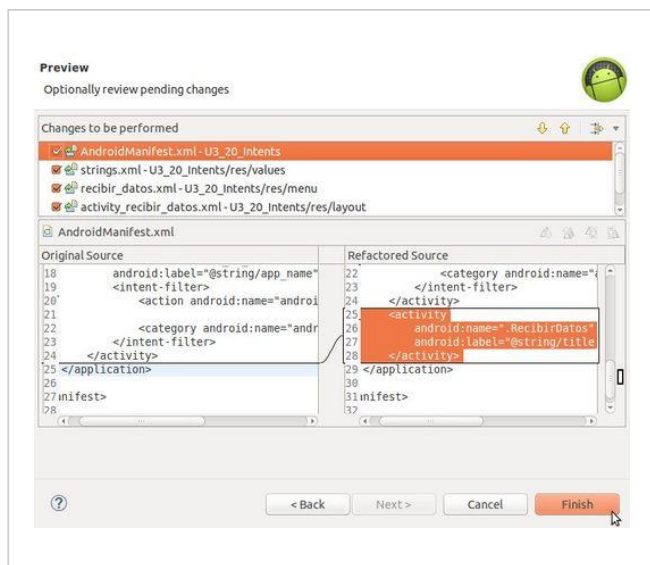
Escolher **Android Activity**



Activity en branco.



Escolher en que proxecto creala 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.



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.

## O XML do layout da Activity principal

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

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   android:layout_width="match_parent"
3   android:layout_height="match_parent"
4   android:orientation="vertical"
5   android:padding="5dp" >
6
7   <LinearLayout
8     android:layout_width="match_parent"

```

```

9      android:layout_height="wrap_content"
10      android:orientation="horizontal" >
11
12      <TextView
13          android:layout_width="wrap_content"
14          android:layout_height="wrap_content"
15          android:text="Dime o teu nome" />
16
17      <EditText
18          android:id="@+id/et_nome"
19          android:layout_width="match_parent"
20          android:layout_height="wrap_content" />
21  </LinearLayout>
22
23  <CheckBox
24      android:id="@+id/chk_destruir"
25      android:layout_width="match_parent"
26      android:layout_height="wrap_content"
27      android:checked="false"
28      android:text="Destruir esta activity ao chamar á 2ª" />
29
30  <TextView
31      android:layout_width="wrap_content"
32      android:layout_height="wrap_content"
33      android:layout_gravity="center_horizontal"
34      android:text="Chamadas explícitas a intents"
35      android:textColor="#00F" />
36
37  <Button
38      android:id="@+id/btn_enviar"
39      android:layout_width="match_parent"
40      android:layout_height="wrap_content"
41      android:onClick="onEnviarClick"
42      android:text="Chamar 2ª Activity: Recibir Datos" />
43
44  </LinearLayout>

```

## A classe Java a activity principal

```

1  package com.example.u3_20_intents;
2
3  import android.app.Activity;
4  import android.content.Intent;
5  import android.os.Bundle;
6  import android.view.Menu;
7  import android.view.View;
8  import android.widget.CheckBox;
9  import android.widget.TextView;
10 import android.widget.Toast;
11
12 public class U3_20_Intents extends Activity {
13
14     public final static String NOME = "nome";
15
16     TextView tvNome;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_u3_20_intents);
22         tvNome = (TextView) findViewById(R.id.et_nome);
23     }
24
25     @Override
26     public boolean onCreateOptionsMenu(Menu menu) {
27         // Inflate the menu; this adds items to the action bar if it is present.
28         getMenuInflater().inflate(R.menu.u3_20_intents, menu);
29         return true;
30     }
31
32     public void onEnviarClick(View view) {
33         CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);
34
35         Intent intent = new Intent(this, RecibirDatos.class);
36         intent.putExtra(NOME, tvNome.getText().toString());
37
38         startActivity(intent);
39
40         if (chkDestruir.isChecked())
41             finish();
42     }
43
44     public void finish() {
45         super.finish();

```

```

46     Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
47 }
48
49 }

```

- **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..

## O XML da Activity que recibe a chamada

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <TextView
8     android:id="@+id/tv_resultado"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:layout_gravity="center_horizontal"
12    android:text="resultado"
13    android:textSize="20sp" />
14
15   <Button
16     android:id="@+id/btn_pechar"
17     android:layout_width="wrap_content"
18     android:layout_height="wrap_content"
19     android:layout_gravity="center_horizontal"
20     android:onClick="onPecharClick"
21     android:text="Pechar" />
22
23 </LinearLayout>

```

## A clase Java da Activity que recibe a chamada. Activity Secundaria.

```

1 package com.example.u3_20_intents;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.Menu;
7 import android.view.View;
8 import android.widget.TextView;
9 import android.widget.Toast;
10
11 public class RecibirDatos extends Activity {
12     String Forma_Pechar_Activity;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_recibir_datos);
18         Forma_Pechar_Activity = " Premendo botonera Retroceso";
19
20         TextView tvResultado = (TextView) findViewById(R.id.tv_resultado);
21
22         Intent intent = getIntent();
23
24         tvResultado.setText("Ola " + intent.getExtras().getString(U3_20_Intents.NOME));
25     }
26
27     @Override
28     public boolean onCreateOptionsMenu(Menu menu) {

```

```

30    // Inflate the menu; this adds items to the action bar if it is present.
31    getMenuInflater().inflate(R.menu.recibir_datos, menu);
32    return true;
33 }
34
35 public void onPecharClick(View view) {
36     Forma_Pechar_Activity = " Premendo botón Pechar";
37
38     finish();
39 }
40
41
42 public void finish() {
43     super.finish();
44     Toast.makeText(this, "Saíches da actividade secundaria: " + Forma_Pechar_Activity, Toast.LENGTH_SHORT).show();
45 }
46
47 }

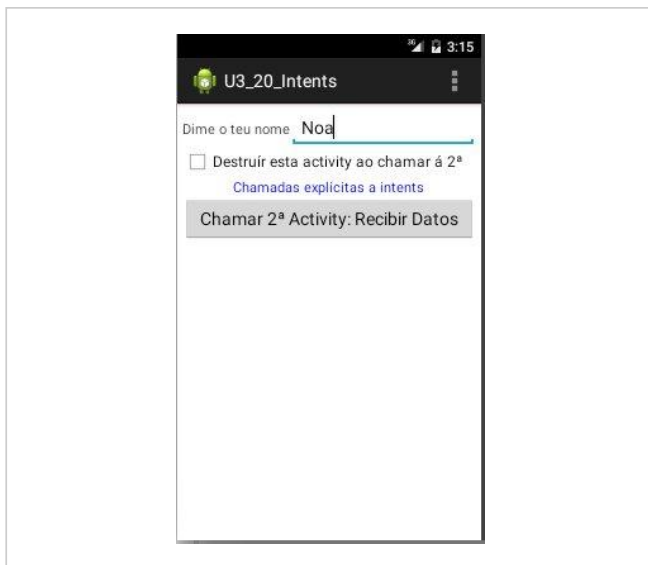
```

- 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.

## Pasar datos da actividade secundaria á principal

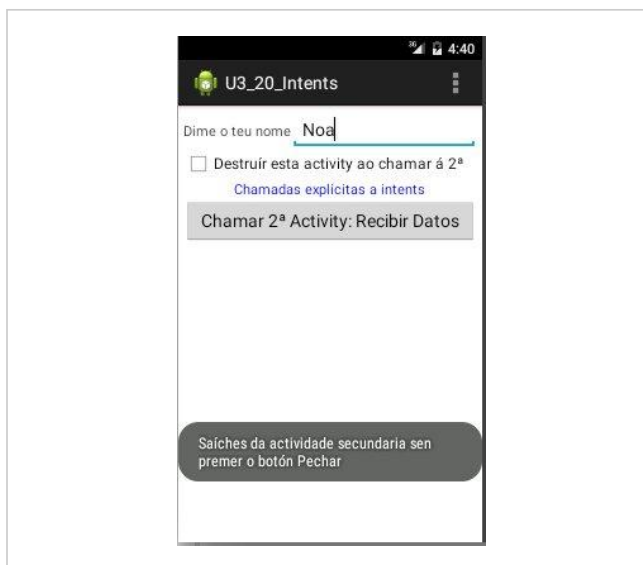
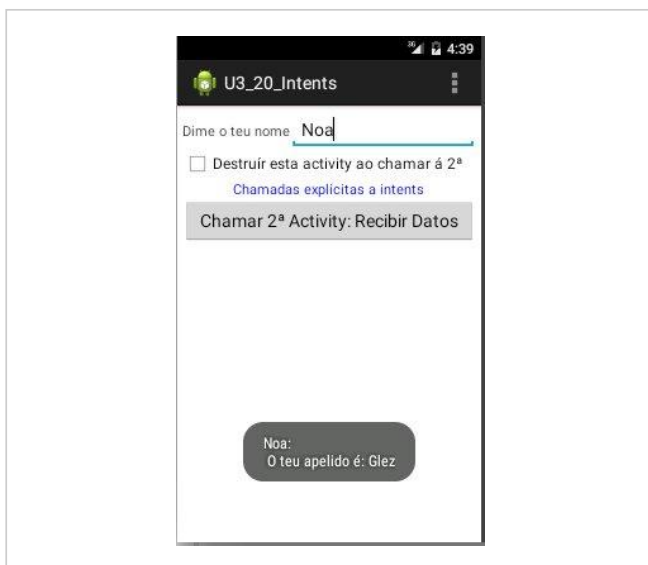
- Até agora pasamos datos da actividade principal á secundaria.
- Imos modificar a aplicación anterior, para que na secundaria se poida introducir o apelido dunha persoa e este se pase á actividade principal.
- Ademais imos controlar na Actividade Principal se se volveu da actividade secundaria premendo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

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

## 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>



```

1 package com.example.u3_20_intents;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.Menu;
7 import android.view.View;
8 import android.widget.CheckBox;
9 import android.widget.TextView;
10 import android.widget.Toast;
11
12 public class U3_20_Intents extends Activity {
13
14     public final static String NOME = "com.example.NOME";
15     private static final int COD_PETICION = 33;
16
17     TextView tvNome;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_u3_20_intents);
23         tvNome = (TextView) findViewById(R.id.et_nome);
24     }
25
26     @Override
27     public boolean onCreateOptionsMenu(Menu menu) {
28         // Inflate the menu; this adds items to the action bar if it is present.
29         getMenuInflater().inflate(R.menu.u3_20_intents, menu);
30         return true;
31     }
32
33     public void onEnviarClick(View view) {
34         CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);
35
36         Intent intent = new Intent(this, RecibirDatos.class);
37         intent.putExtra(NOME, tvNome.getText().toString());
38
39         // startActivity(intent);
40         startActivityForResult(intent, COD_PETICION);
41
42         if (chkDestruir.isChecked())
43             finish();
44     }
45
46     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
47
48         if (requestCode == COD_PETICION) {
49             if (resultCode == RESULT_OK) {
50                 if (data.hasExtra("APELLIDO"))
51                     Toast.makeText(this, tvNome.getText() + "\n0 teu apelido é: " + data.getExtras().getString("APELLIDO"),
52                         Toast.LENGTH_SHORT).show();
53             } else
54                 Toast.makeText(this, "Saíches da actividade secundaria sen premer o botón Pechar", Toast.LENGTH_SHORT).show();
55         }
56     }
57
58     public void finish() {
59         super.finish();
60         Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
61     }
62 }
63
64 }

```

- **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.

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

- Só se engadiu un EditText.

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <TextView
8       android:id="@+id/tv_resultado"
9       android:layout_width="wrap_content"
10      android:layout_height="wrap_content"
11      android:layout_gravity="center_horizontal"
12      android:text="resultado"
13      android:textSize="20sp" />
14
15   <EditText
16       android:id="@+id/et_apellido"
17       android:layout_width="wrap_content"
18       android:layout_height="wrap_content"
19       android:layout_gravity="center_horizontal"
20       android:hint="Introduce o teu apelido" />
21
22   <Button
23       android:id="@+id/btn_pechar"
24       android:layout_width="wrap_content"
25       android:layout_height="wrap_content"
26       android:layout_gravity="center_horizontal"
27       android:onClick="onPecharClick"
28       android:text="Pechar" />
29
30 </LinearLayout>

```

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

```

1 package com.example.u3_20_intents;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.Menu;
7 import android.view.View;
8 import android.widget.EditText;
9 import android.widget.TextView;
10
11 public class RecibirDatos extends Activity {
12     // String Forma_Pechar_Activity;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_recibir_datos);
18         // Forma_Pechar_Activity = " Premendo botonera Retroceso";
19
20         TextView tvResultado = (TextView) findViewById(R.id.tv_resultado);
21
22         Intent intent = getIntent();
23
24         tvResultado.setText("Ola " + intent.getExtras().getString(U3_20_Intents.NOME));
25     }
26
27     @Override
28     public boolean onCreateOptionsMenu(Menu menu) {

```

```

30 // Inflate the menu; this adds items to the action bar if it is present.
31 getMenuInflater().inflate(R.menu.recibir_datos, menu);
32 return true;
33 }
34
35 public void onPecharClick(View view) {
36 // Forma_Pechar_Activity = " Premendo botón Pechar";
37 EditText et_apellido = (EditText) findViewById(R.id.et_apellido);
38
39 Intent datos_volta = new Intent();
40 datos_volta.putExtra("APELLIDO", et_apellido.getText().toString());
41 setResult(RESULT_OK, datos_volta);
42 finish();
43
44 }
45
46 public void finish() {
47 super.finish();
48 // Toast.makeText(this, "Saiches da actividade secundaria: " +
49 // Forma_Pechar_Activity, Toast.LENGTH_SHORT).show();
50 }
51
52 }

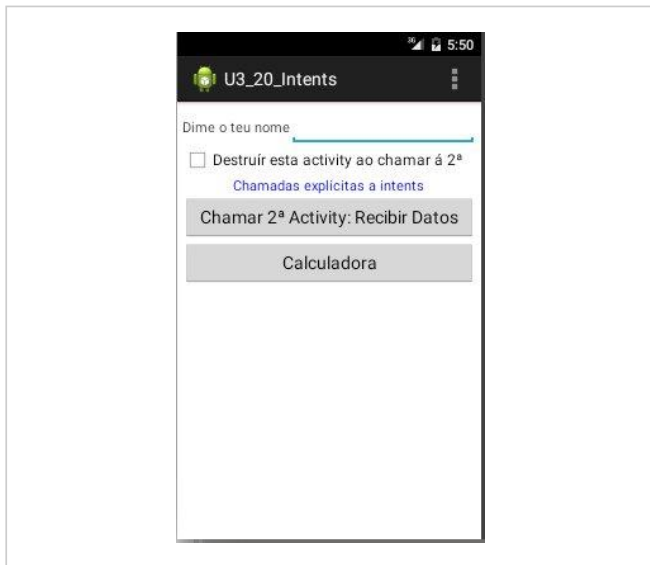
```

- **Liñas 12,18,36,48 e 49:** Comentámolos 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: APELLIDO-Valor.
- **Liña 41:** setResult envíalle o resultado á activity chamadora.

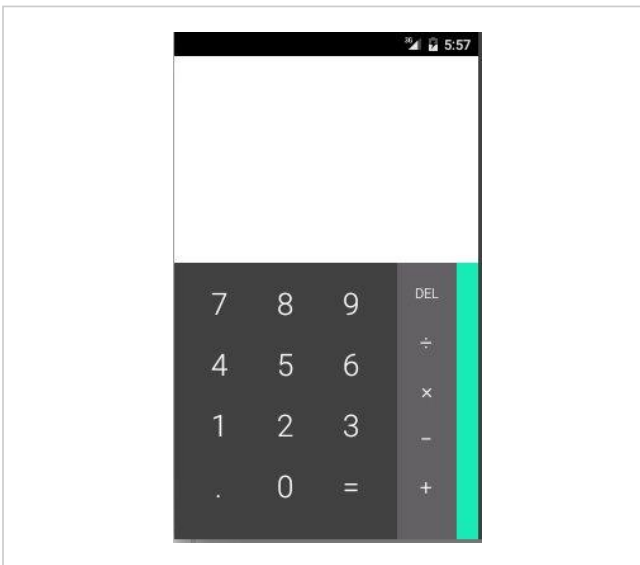
## Chamar a unha activity doutra aplicación: á calculadora

- 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.

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

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

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   android:layout_width="match_parent"
3   android:layout_height="match_parent"
4   android:orientation="vertical"
5   android:padding="5dp" >
6

```

```

7   <LinearLayout
8       android:layout_width="match_parent"
9       android:layout_height="wrap_content"
10      android:orientation="horizontal" >
11
12       <TextView
13           android:layout_width="wrap_content"
14           android:layout_height="wrap_content"
15           android:text="Dime o teu nome" />
16
17       <EditText
18           android:id="@+id/et_nome"
19           android:layout_width="match_parent"
20           android:layout_height="wrap_content" />
21   </LinearLayout>
22
23   <CheckBox
24       android:id="@+id/chk_destruir"
25       android:layout_width="match_parent"
26       android:layout_height="wrap_content"
27       android:checked="false"
28       android:text="Destruir esta activity ao chamar á 2ª" />
29
30   <TextView
31       android:layout_width="wrap_content"
32       android:layout_height="wrap_content"
33       android:layout_gravity="center_horizontal"
34       android:text="Chamadas explícitas a intents"
35       android:textColor="#00F" />
36
37   <Button
38       android:id="@+id/btn_enviar"
39       android:layout_width="match_parent"
40       android:layout_height="wrap_content"
41       android:onClick="onEnviarClick"
42       android:text="Chamar 2ª Activity: Recibir Datos" />
43
44   <Button
45       android:id="@+id/btn_calculadora"
46       android:layout_width="match_parent"
47       android:layout_height="wrap_content"
48       android:onClick="onCalculadoraClick"
49       android:text="Calculadora" />
50
51 </LinearLayout>

```

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

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

```

1 package com.example.u3_20_intents;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.Menu;
7 import android.view.View;
8 import android.widget.CheckBox;
9 import android.widget.TextView;
10 import android.widget.Toast;
11
12 public class U3_20_Intents extends Activity {
13
14     public final static String NOME = "com.example.NOME";
15     private static final int COD_PETICION = 33;
16
17     TextView tvNome;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_u3_20_intents);
23         tvNome = (TextView) findViewById(R.id.et_nome);
24     }
25
26     @Override
27     public boolean onCreateOptionsMenu(Menu menu) {
28         // Inflate the menu; this adds items to the action bar if it is present.
29         getMenuInflater().inflate(R.menu.u3_20_intents, menu);
30         return true;
31     }
32
33     public void onEnviarClick(View view) {
34         CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);

```

```

35
36     Intent intent = new Intent(this, RecibirDatos.class);
37     intent.putExtra(NOME, tvNome.getText().toString());
38
39     // startActivity(intent);
40     startActivityForResult(intent, COD_PETICION);
41
42     if (chkDestruir.isChecked())
43         finish();
44 }
45
46 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
47
48     if (requestCode == COD_PETICION) {
49         if (resultCode == RESULT_OK) {
50             if (data.hasExtra("APELLIDO"))
51                 Toast.makeText(this, tvNome.getText() + "\n0 teu apelido é: " + data.getExtras().getString("APELLIDO"),
52                     Toast.LENGTH_SHORT).show();
53         } else
54             Toast.makeText(this, "Saíches da actividade secundaria sen premer o botón Pechar", Toast.LENGTH_SHORT).show();
55     }
56 }
57
58
59 public void onCalculadoraClick(View v) {
60
61     Intent intent = new Intent();
62     intent.setClassName("com.android.calculator2", "com.android.calculator2.Calculator");
63
64     startActivity(intent);
65 }
66
67 public void finish() {
68     super.finish();
69     Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
70 }
71
72 }

```

- **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).

---

Obtenido de «[https://manuais.iessanclemente.net/index.php?title=Intents\\_explicitos&oldid=57148](https://manuais.iessanclemente.net/index.php?title=Intents_explicitos&oldid=57148)»

---

Esta página se editó por última vez el 28 jul 2015 a las 09:25.

El contenido está disponible bajo la licencia [Creative Commons: CC-BY-NC-SA](#), a menos que se indique lo contrario.