

Intents implícitos. Permisos. Filtros.

Sumario

Introdución

Permisos

Chamadas a intents de modo implícito

Filtros de intencións (Intent Filters)

Caso práctico

XML do Layout principal

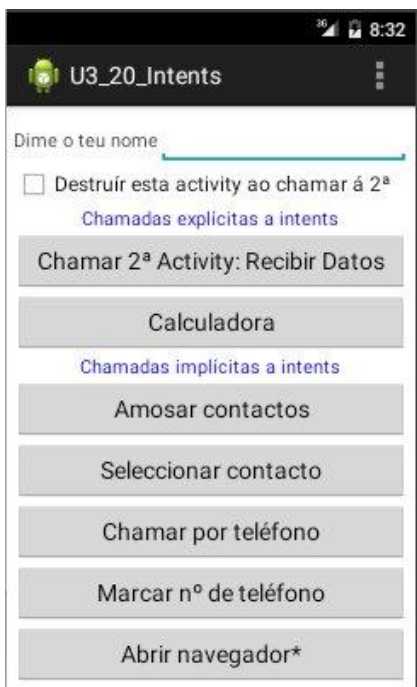
Código Java da activity principal

XML do Layout da Activity Secundaria: RecibirDatos

O código Java da Activity Secundaria: RecibirDatos

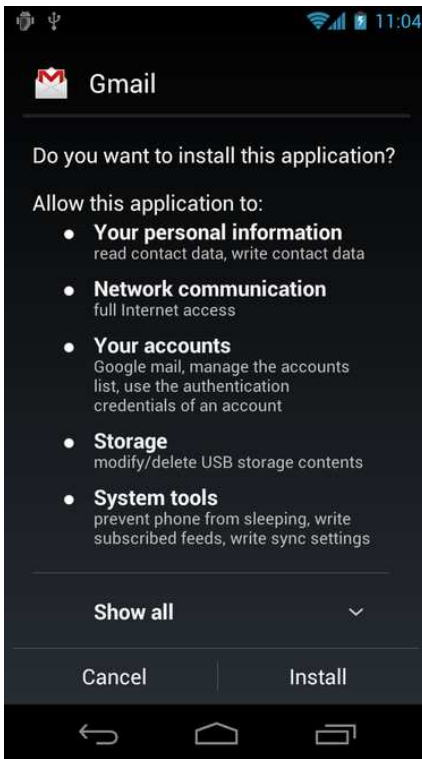
Introdución

- Imos continuar coa aplicación desenvolvida no apartado anterior: **U3_20_Intents**.
- Engadiránselle, permisos, filtros, e intents implícitos.



Permisos

- Cando instalamos unha aplicación nun dispositivo real (non nun AVD) se precisa acceder a características que esixen algún tipo de permiso, o proceso de instalación pregunta se estamos dispostos a dar eses permisos para que a aplicación poida funcionar con tódalas súas funcionalidades.
- Por exemplo, instalando Gmail:



- O proceso de instalación pregunta ao usuario se por exemplo lle deixa acceder a esta aplicación á súa lista de contactos.
- No ficheiro **AndroidManifest.xml** é onde se declaran os permisos que precisa a aplicación para poder facer uso de funcionalidades protexidas: contactos, cámara, mermoria usb, gps, etc.
- A través de unha ou varias etiquetas **<uses-permission>** vanse declarar os permisos que precisa a aplicación.

Exemplos de permisos no ficheiro AndroidManifest.xml

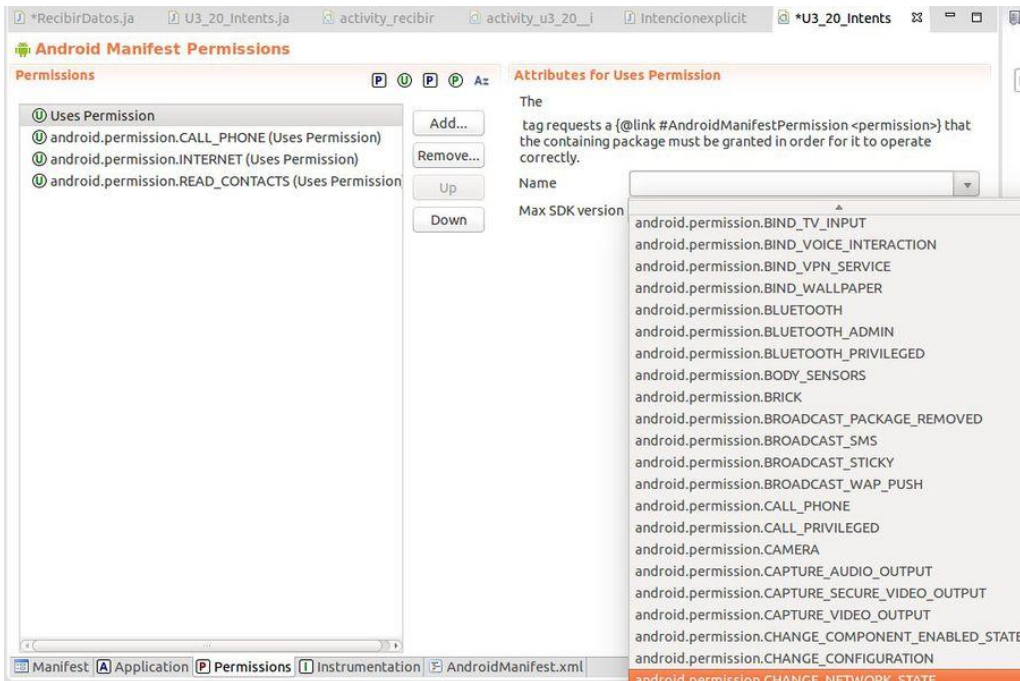
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.u3_20_intents"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="16"
9         android:targetSdkVersion="21" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:label="@string/app_name"
15         android:theme="@style/AppTheme" >
16         <activity
17             android:name=".U3_20_Intents"
18             android:label="@string/app_name" >
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25         <activity
26             android:name=".RecibirDatos"
27             android:label="@string/title_activity_recibir_datos" >
28         </activity>
29     </application>
30
31     <uses-permission android:name="android.permission.CALL_PHONE" />
32     <uses-permission android:name="android.permission.INTERNET" />
33     <uses-permission android:name="android.permission.READ_CONTACTS" />
34
35 </manifest>

```

- **Liña 31:** Permiso que "permite" realizar unha chamada de telefono sen pasar a través do interface do teléfono para que o usuario marque.
- **Liña 32:** Permítelle á aplicación abrir conexións de rede.
- **Liña 33:** Da súa lectura pódese concluir o que permite.

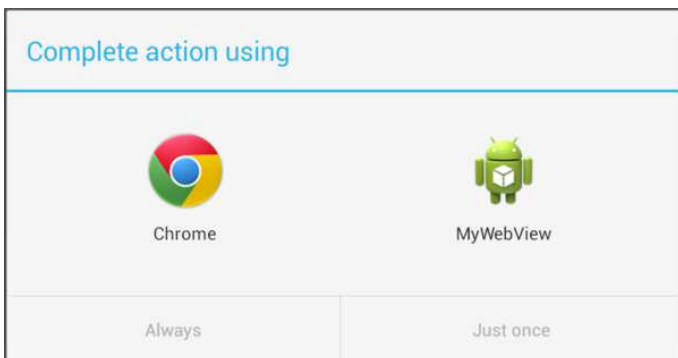
- Para coñecer os permisos dispoñibles podemos consultar o seguinte enlace e incluílos directamente no xml:
 - <http://developer.android.com/reference/android/Manifest.permission.html>
- Tamén se pode realizar de modo gráfico.
- No ficheiro AndroidManifest.xml premer en **Permissions**



- Lembrar que os permisos asígnanse no momento da instalación a aplicación. Nun dispositivo real pídeselle consentimento ao usuario, non así nun AVD.

Chamadas a intents de modo implícito

- Como xa se dixo, un compoñente pode ser lanzado de forma implícita cando se indica a **acción** que se desexa realizar e se é o caso os datos sobre os que se desexa realizar a acción.
- Non se vai especificar cal é a activity que vai atender o intent. Incluso se hai varias Activities que poidan atender ese intent o sistema ofrecerá ao usuario as distintas posibilidades para que escolla.
- Por exemplo, cando dende contactos se desexa enviar un mensaxe a un número de teléfono o sistema ofrece a posibilidade de que se envíe a través de sms, Whatsapp, Viber, etc, se se teñen instalados estes últimos.
- A seguinte imaxe amosa un exemplo no que para abrir unha URL hai dúas aplicacións que o poden facer.

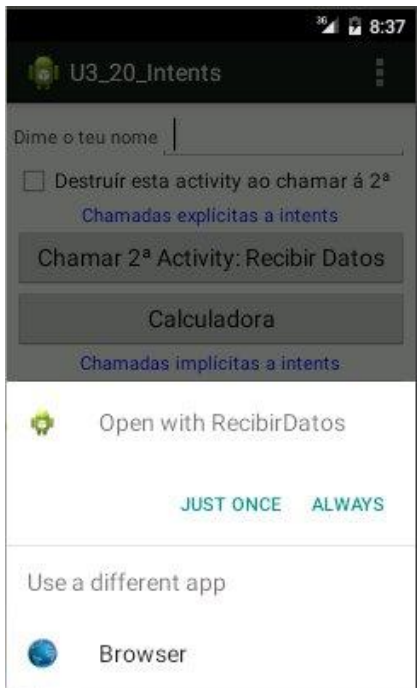


- Para lanzar un intent de modo implícito precísase indicar:
 - Acción (action):** A acción que se desexa levar a cabo. Por exemplo,
 - ACTION_VIEW, para mostrar datos ao usuario
 - ACTION_EDIT, para editar os datos que nos pasan
 - ACTION_PICK. selecciona un ítem dun conxunto de datos e devolve o seleccionado.
 - ACTION_WEB_SEARCH. busca no navegador as palabras indicadas.
 - etc.

- **Datos (data):** os datos sobre os que se vai operar, por exemplo, unha url, os datos dun contacto, a cadea de busca, etc.
- Exemplo:
 - ACTION_VIEW content://contacts/people/1 -- Amosa a información sobre a persoa con identificador 1.
- **Referencias:**
 - <http://developer.android.com/reference/android/content/Intent.html>

Filtros de intencións (Intent Filters)

- Android busca os compoñentes que poden responder a unha **Acción** nos filtros de intención que se definen no ficheiro **AndroidManifest.xml** de tódalas aplicacións que están instaladas no dispositivo.
- Cando se constrúe unha activity nunha aplicación, pódese indicar os filtros de intención de manifestan que esa activity pode dar resposta as **Accións** que lanzan outras aplicacións.
- No noso caso, imos modificar un chisco o código da actividade secundaria para que poida atender a peticións de visionar urls de tipo **http**.
- Así cando se lance, por exemplo, que se desexa ver <http://www.google.es> o sistema vai ofrecer 2 opcións: o navegador de internet e a nosa actividade secundaria do proxecto (RecibirDatos).



- Para iso é preciso definir que esa Activity pode atender Accións de visionado de datos tipo http.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.u3_20_intents"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="16"
9         android:targetSdkVersion="21" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:label="@string/app_name"
15         android:theme="@style/AppTheme" >
16         <activity
17             android:name=".U3_20_Intents"
18             android:label="@string/app_name" >
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />

```

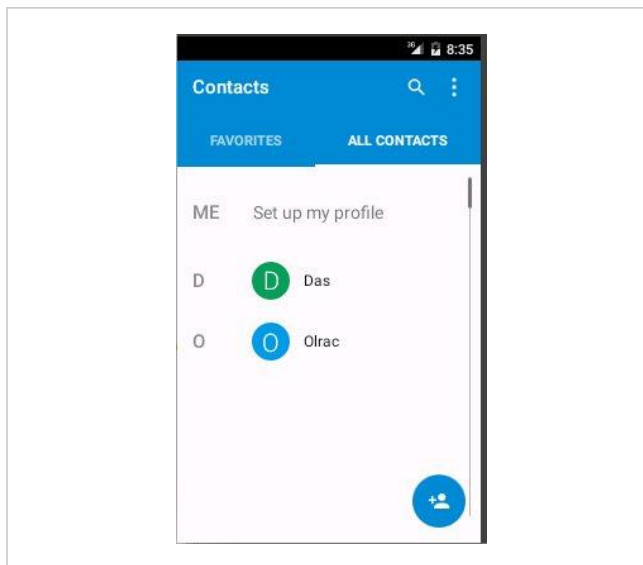
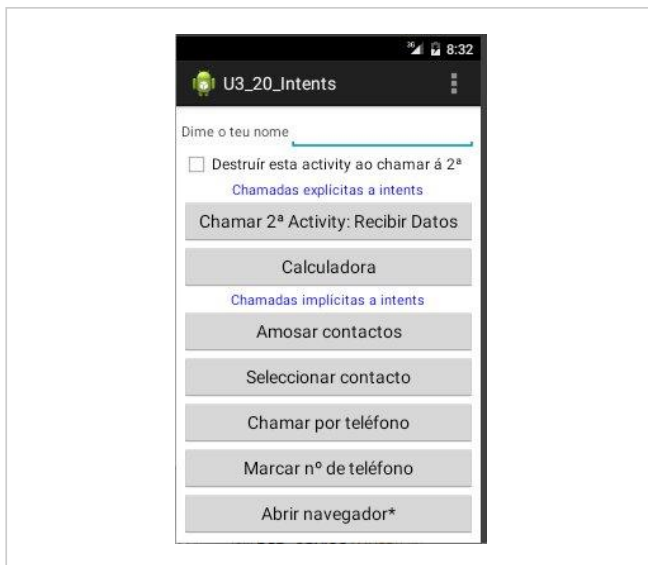
```
21
22     <category android:name="android.intent.category.LAUNCHER" />
23 </intent-filter>
24 </activity>
25 <activity
26     android:name=".RecibirDatos"
27     android:label="@string/title_activity_recibir_datos" >
28     <intent-filter>
29         <action android:name="android.intent.action.VIEW" />
30
31         <category android:name="android.intent.category.DEFAULT" />
32
33         <data android:scheme="http" />
34     </intent-filter>
35 </activity>
36 </application>
37
38 <uses-permission android:name="android.permission.CALL_PHONE" />
39 <uses-permission android:name="android.permission.INTERNET" />
40 <uses-permission android:name="android.permission.READ_CONTACTS" />
41
42
43 </manifest>
```

- **Liñas 28 e 25:** delimitan os filtros de intencións para a activity RecibirDatos.
- **Liña 29:** indica para que tipo de accións esta dispoñible esta activity.
- **Liña 30:** A categoría indica as circunstancias nas que se debe desenvolver a acción.
- **Liña 33:** o tipo de datos que pode atender esta activity cando haxa unha intención preguntando por recursos http.

Caso práctico

- A continuación imos engadir unha serie de botóns, modificar o código, etc ao proxecto.
- Continuar co proxecto **U3_20_Intents**.

Intents implícitos

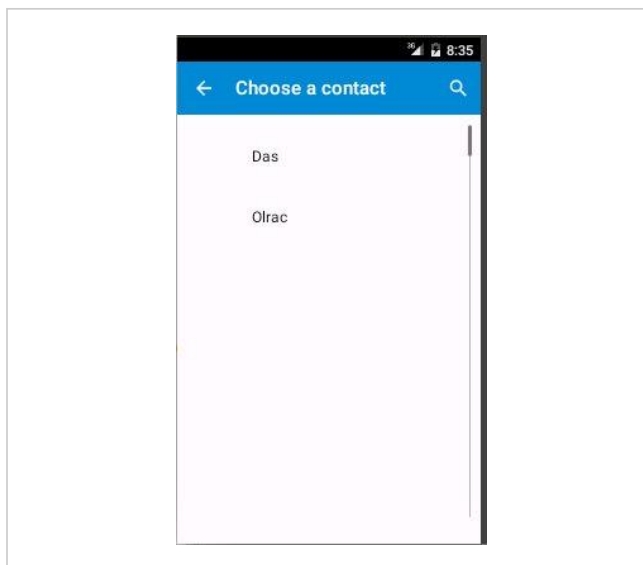
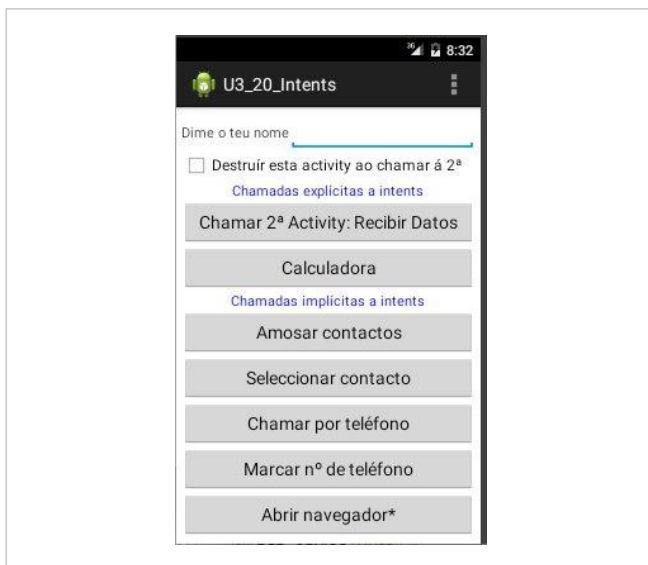


Premer en **Amosar contactos**.

A lista de contactos.

No ficheiro AndroidManifest demos permiso á aplicación para que acceda aos contactos. Para lanzar a seguinte activity da aplicación contactos o intent ten o seguinte formato:

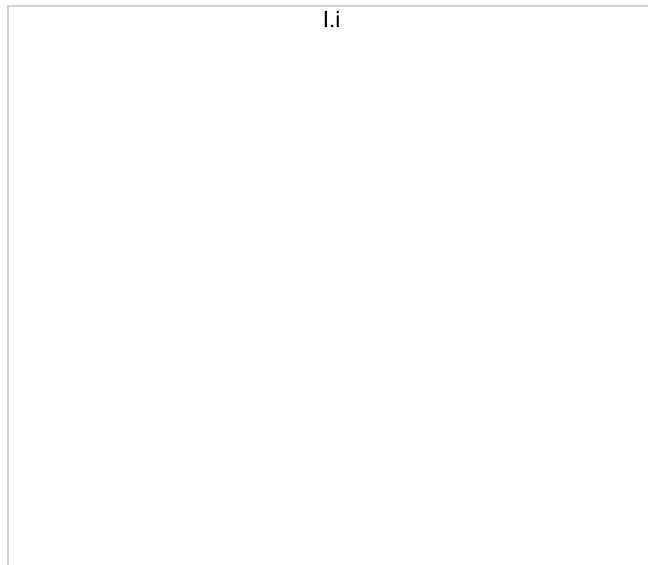
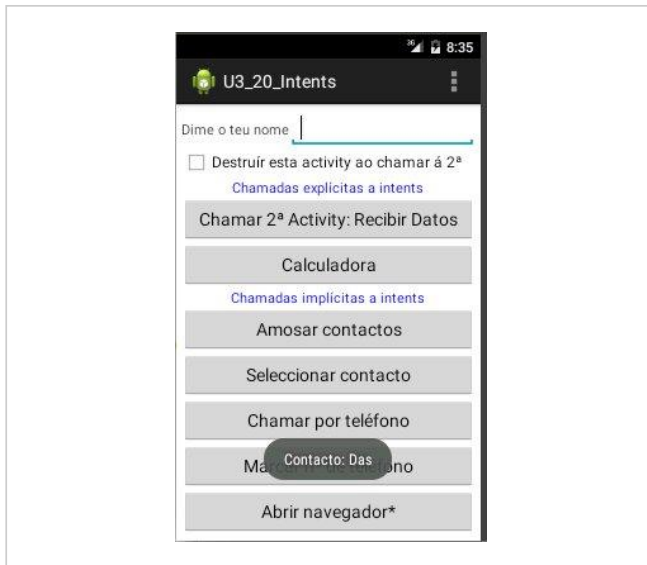
```
intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("content://contacts/people/"));
```



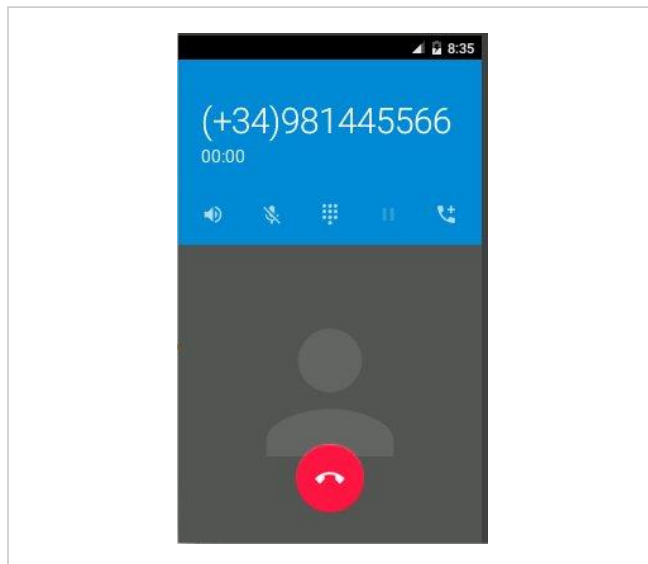
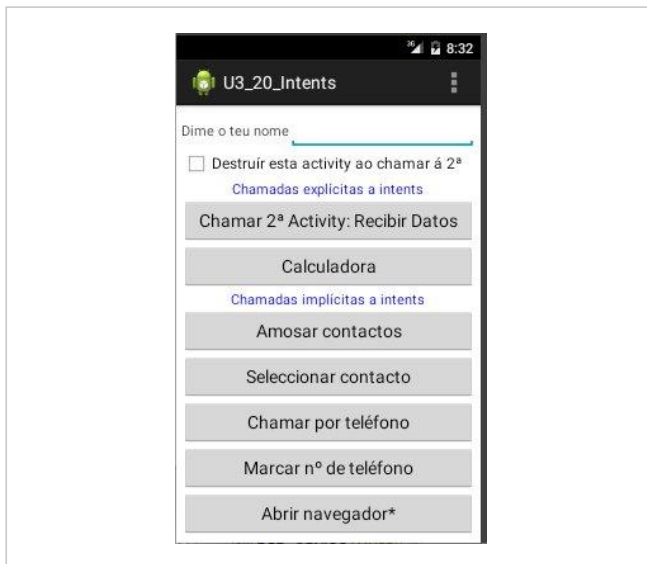
Premer en **Seleccionar contactos**. Lánzase o seguinte intent:

```
intent = new Intent(Intent.ACTION_PICK,
Uri.parse("content://contacts/people/"));
startActivityForResult(intent, COD_CONTACTOS);
```

Seleccionamos un contacto e grazas á Acción coa que se lanzou o intent podemos devolver á actividade principal os datos dese contacto.

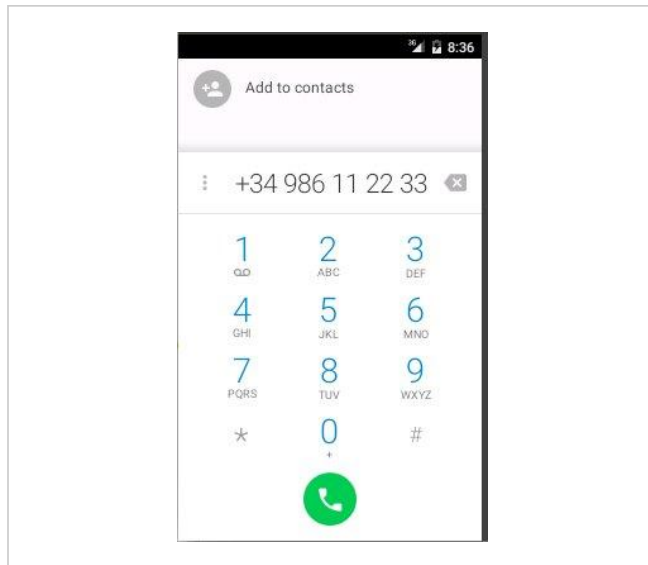
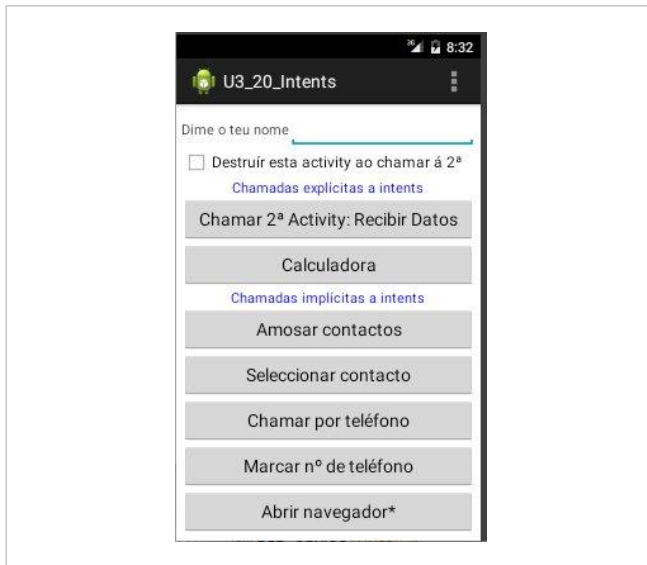


Cando volvemos á actividade principal podemos recuperar o nome do contacto seleccionado.



Premer en **Chamar por teléfono**. Vaise lanzar o seguinte intent: Xa iniciou a marcación ao teléfono indicado.

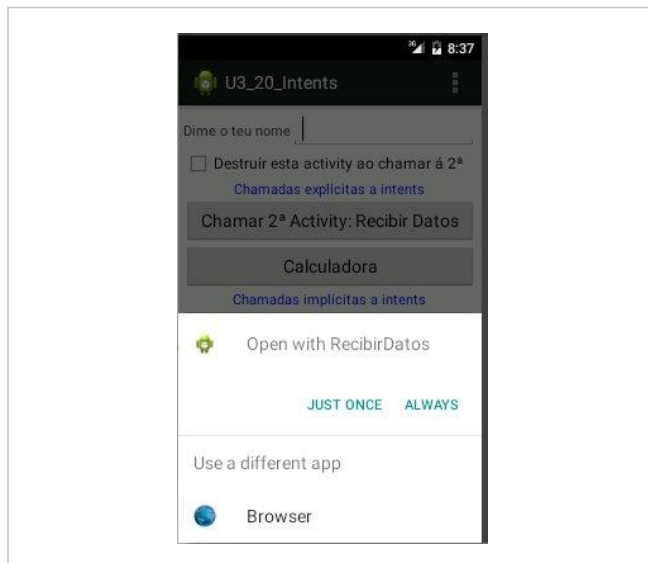
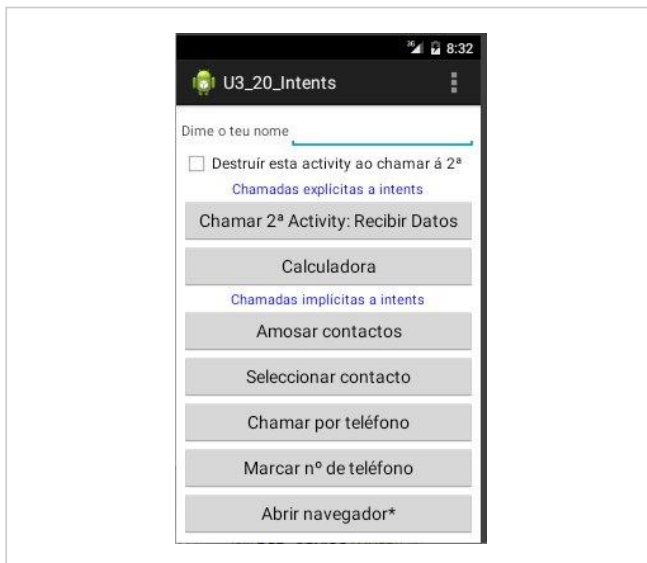
```
intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:
(+34)981445566"));
```



Premer en **Marcar nº teléfono**. Vainos lanzar a activity na que nos decidimos se marcamos o número a marcar.

Activity na que o usuario marca o número de teléfono, pero que xa lle vén cuberto o campo.

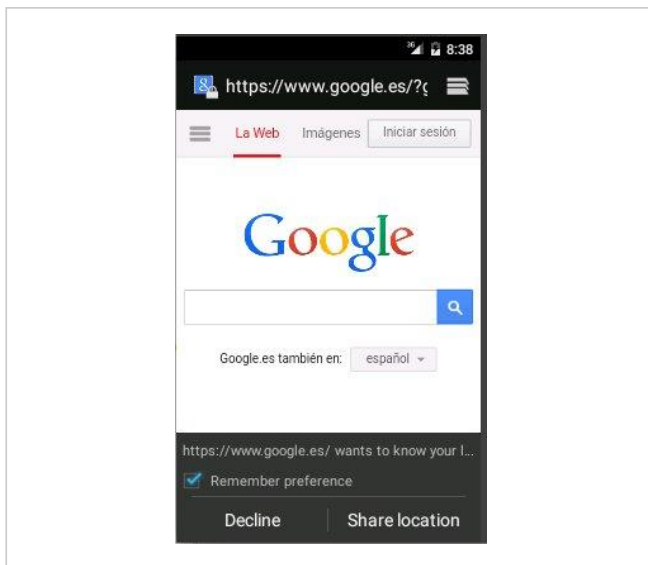
```
intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:
(+34)986112233"));
```



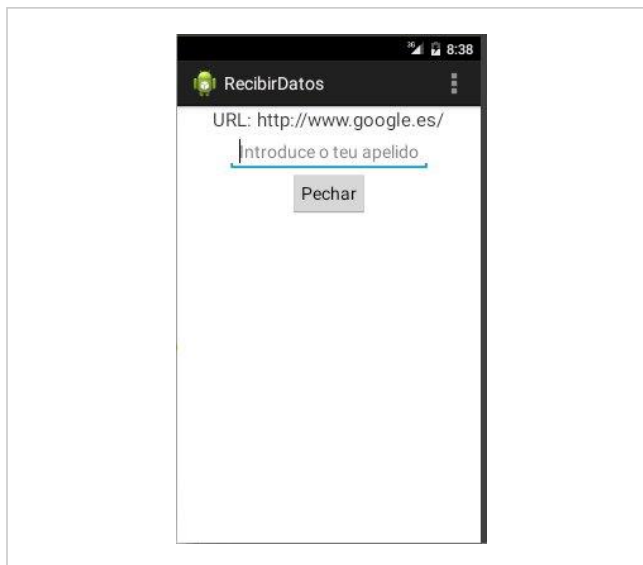
Premer agora en **Abrir navegador***. Onde se van ofrecer un par de opcións para abrir a url especificada.

Activities que teñen definidos filtros de intención nos seus ficheiros AndroidManifest para procesar datos de tipo http.

```
intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://www.google.es/"));
```

Neste caso seleccionouse o navegador web.



Neste outro caso, seleccionouse a nosa actividade secundaria (RecibirDatos) e o que fai é amosar a URL.

XML do Layout principal

- Engadiuse unha serie de botóns, todos chaman ao mesmo método. (Liñas 51-91)

```

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     <TextView
52         android:layout_width="wrap_content"
53         android:layout_height="wrap_content"
54         android:layout_gravity="center_horizontal"
55         android:text="Chamadas implícitas a intents"
56         android:textColor="#00F" />
57
58     <Button
59         android:id="@+id/btn_amosar_contactos"
60         android:layout_width="match_parent"
61         android:layout_height="wrap_content"
62         android:onClick="chamarIntent"
63         android:text="Amosar contactos" />
64
65     <Button
66         android:id="@+id/btn_seleccionar_contacto"
67         android:layout_width="match_parent"
68         android:layout_height="wrap_content"
69         android:onClick="chamarIntent"
70         android:text="Seleccionar contacto" />
71
72     <Button
73         android:id="@+id/btn_chamar_tel"
74         android:layout_width="match_parent"
75         android:layout_height="wrap_content"
76         android:onClick="chamarIntent"
77         android:text="Chamar por teléfono" />
78
79     <Button
80         android:id="@+id/btn_marcar_tel"
81         android:layout_width="match_parent"
82         android:layout_height="wrap_content"
83         android:onClick="chamarIntent"
84         android:text="Marcar nº de teléfono" />
85
86     <Button
87         android:id="@+id/btn_navegador"
88         android:layout_width="match_parent"
89         android:layout_height="wrap_content"
90         android:onClick="chamarIntent"
91         android:text="Abrir navegador*" />
92
93 </LinearLayout>

```

Código Java da activity principal

```

1 package com.example.u3_20_intents;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.database.Cursor;
6 import android.net.Uri;
7 import android.os.Bundle;
8 import android.provider.ContactsContract;
9 import android.view.Menu;
10 import android.view.View;
11 import android.widget.CheckBox;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 public class U3_20_Intents extends Activity {
16
17     public final static String NOME = "com.example.NOME";
18     private static final int COD_PETICION = 33;
19     private static final int COD_CONTACTOS = 7;
20
21     TextView tvNome;
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_u3_20__intents);
27         tvNome = (TextView) findViewById(R.id.et_nome);
28     }
29
30     @Override
31     public boolean onCreateOptionsMenu(Menu menu) {
32         // Inflate the menu; this adds items to the action bar if it is present.
33         getMenuInflater().inflate(R.menu.u3_20__intents, menu);
34         return true;
35     }

```

```

36
37 public void onEnviarClick(View view) {
38     CheckBox chkDestruir = (CheckBox) findViewById(R.id.chk_destruir);
39
40     Intent intent = new Intent(this, RecibirDatos.class);
41     intent.putExtra(NOME, tvNome.getText().toString());
42
43     // startActivity(intent);
44     startActivityForResult(intent, COD_PETICION);
45
46     if (chkDestruir.isChecked())
47         finish();
48 }
49
50 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
51
52     if (requestCode == COD_PETICION) {
53         if (resultCode == RESULT_OK) {
54             if (data.hasExtra("APELIDO"))
55                 Toast.makeText(this, tvNome.getText() + "\nO teu apelido é: " + data.getExtras().getString("APELIDO"),
56 Toast.LENGTH_SHORT).show();
57         } else
58             Toast.makeText(this, "Saiches da actividade secundaria sen premer o botón Pechar", Toast.LENGTH_SHORT).show();
59     } else if (requestCode == COD_CONTACTOS && resultCode == RESULT_OK) {
60         Uri contactoData = data.getData();
61         Cursor c = managedQuery(contactoData, null, null, null, null);
62         if (c.moveToFirst()) {
63             String nombre = c.getString(c.getColumnIndexOrThrow(ContactsContract.Contacts.DISPLAY_NAME));
64             Toast.makeText(this, "Contacto: " + nombre, Toast.LENGTH_LONG).show();
65         }
66     }
67 }
68
69
70 public void onCalculadoraClick(View v) {
71
72     Intent intent = new Intent();
73     intent.setClassName("com.android.calculator2", "com.android.calculator2.Calculator");
74
75     startActivity(intent);
76 }
77
78 public void chamarIntent(View view) {
79     Intent intent = null;
80     switch (view.getId()) {
81         case R.id.btn_seleccionar_contacto:
82             intent = new Intent(Intent.ACTION_PICK, Uri.parse("content://contacts/people/"));
83             startActivityForResult(intent, COD_CONTACTOS);
84             break;
85
86         case R.id.btn_amosar_contactos:
87             intent = new Intent(Intent.ACTION_VIEW, Uri.parse("content://contacts/people/"));
88             startActivity(intent);
89             break;
90
91         case R.id.btn_chamar_tel:
92             intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:(+34)981445566"));
93             startActivity(intent);
94             break;
95
96         case R.id.btn_marcar_tel:
97             intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:(+34)986112233"));
98             startActivity(intent);
99             break;
100
101         case R.id.btn_navegador:
102             intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.es/"));
103             startActivity(intent);
104             break;
105     }
106 }
107
108 public void finish() {
109     super.finish();
110     Toast.makeText(this, "Mataches a actividade principal", Toast.LENGTH_SHORT).show();
111 }
112
113 }

```

- **Liña 19:** Créase unha nova constante de tipo enteiro cun valor calquera para cando chamemos a activity de contactos para que nos devolva un contacto.
- **Liñas 59-66:** Revisamos cando se volva dunha actividade secundaria, se é a de Contactos. Nese caso (escápase aos obxectivos desta unidade) créase un curso para procesar os datos recibidos da activity contactos.
- **Liñas 78-106:** Procesado dos distintos botóns, creación dos intents en función do tipo de acción e datos que se desexan procesar.

- **Liñas 82 e 83:** Observar como se chama a activity de contactos para que logo nos devolva o contacto seleccionado.

XML do Layout da Activity Secundaria: RecibirDatos

- Non hai ningún cambio con respecto ao proxecto anterior.

```
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 </LinearLayout>
```

O código Java da Activity Secundaria: RecibirDatos

```
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         if ("http".equals(intent.getScheme()))
25
26             tvResultado.setText("URL: " + intent.getDataString());
27         else
28             tvResultado.setText("Ola " + intent.getExtras().getString(U3_20_Intents.NOME));
29
30     }
31
32     @Override
33     public boolean onCreateOptionsMenu(Menu menu) {
34         // Inflate the menu; this adds items to the action bar if it is present.
35         getMenuInflater().inflate(R.menu.recibir_datos, menu);
36         return true;
37     }
38 }
```

```
39 public void onPegarClick(View view) {  
40     // Forma_Pegar_Activity = " Premendo botón Pegar";  
41     EditText et_apellido = (EditText) findViewById(R.id.et_apellido);  
42  
43     Intent datos_volta = new Intent();  
44     datos_volta.putExtra("APELLIDO", et_apellido.getText().toString());  
45     setResult(RESULT_OK, datos_volta);  
46     finish();  
47  
48 }  
49  
50 public void finish() {  
51     super.finish();  
52     // Toast.makeText(this, "Saiches da actividade secundaria: " +  
53     // Forma_Pegar_Activity, Toast.LENGTH_SHORT).show();  
54 }  
55  
56 }
```

- **Liñas 24-28:** Engadiuse a posibilidade de poder procesar intents cuxo esquema (getSchema) sexa de tipo http. Nese caso simplemente amosamos a URL á que desexaba conectarse o usuario.

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

Obtenido de «[https://manuais.iessanclemente.net/index.php?title=Intents_implícitos._Permisos._Filtros.&oldid=57499](https://manuais.iessanclemente.net/index.php?title=Intents_impl%C3%ADcitos._Permisos._Filtros.&oldid=57499)»

Esta página se editó por última vez el 19 oct 2015 a las 09:13.

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