

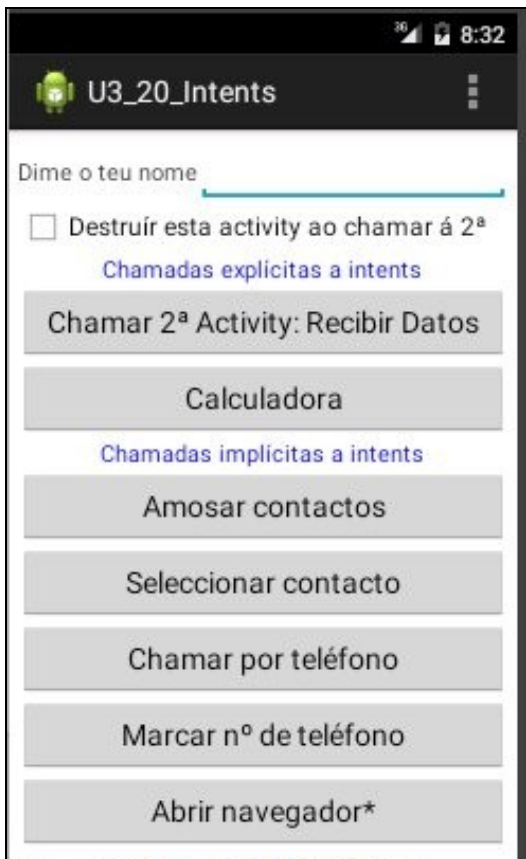
Intents implícitos. Permisos. Filtros.

Sumario

- 1 Introducción
- 2 Permisos
- 3 Chamadas a intents de modo implícito
- 4 Filtros de intencións (Intent Filters)
- 5 Caso práctico
 - ◆ 5.1 XML do Layout principal
 - ◆ 5.2 Código Java da activity principal
 - ◆ 5.3 XML do Layout da Activity Secundaria: RecibirDatos
 - ◆ 5.4 O código Java da Activity Secundaria: RecibirDatos

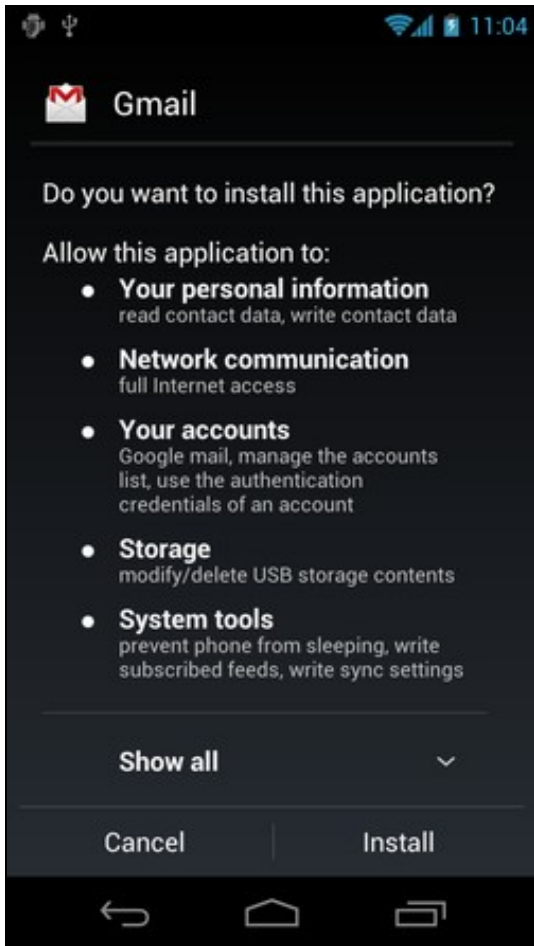
Introdución

- 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

```
<?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" />

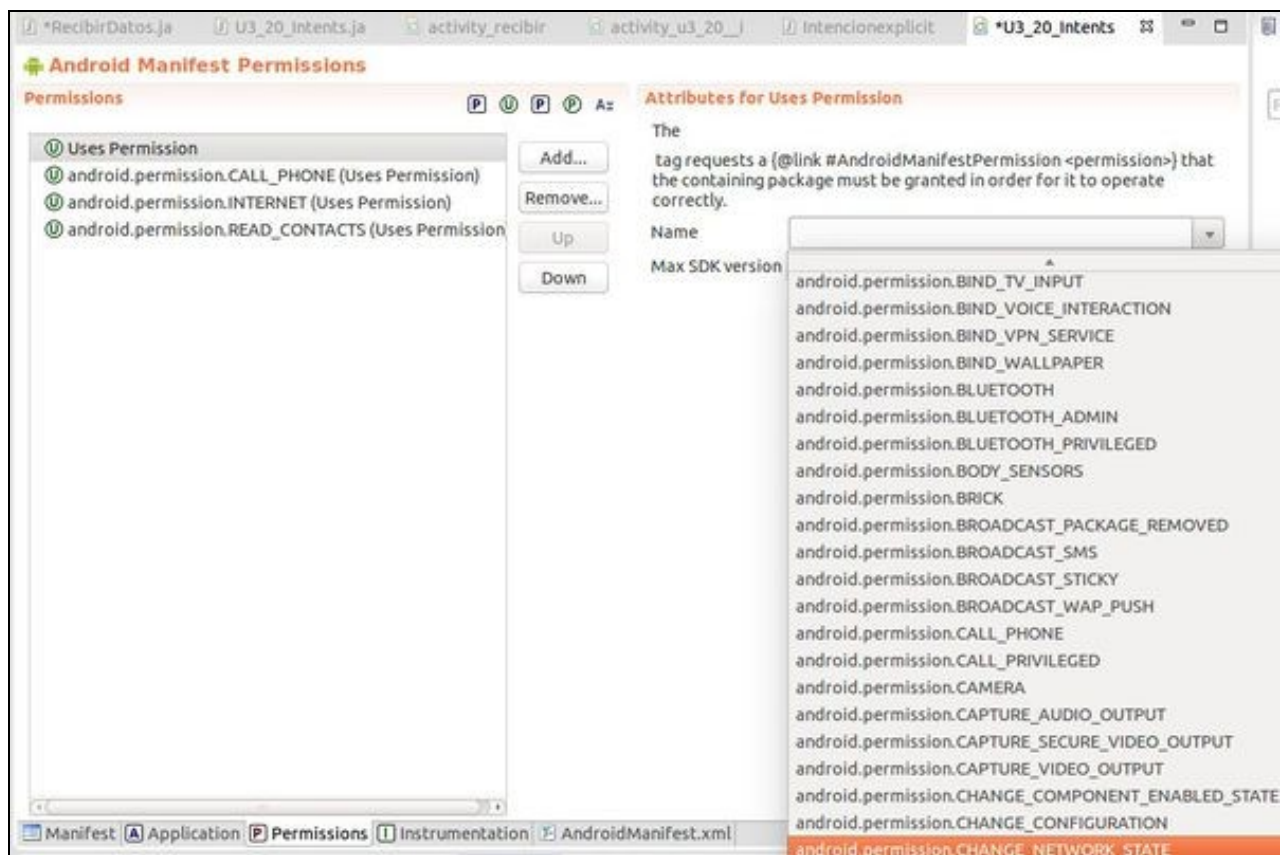
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".U3_20_Intents"
            android:label="@string/app_name" >
            <intent-filter>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

    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />

</manifest>
```

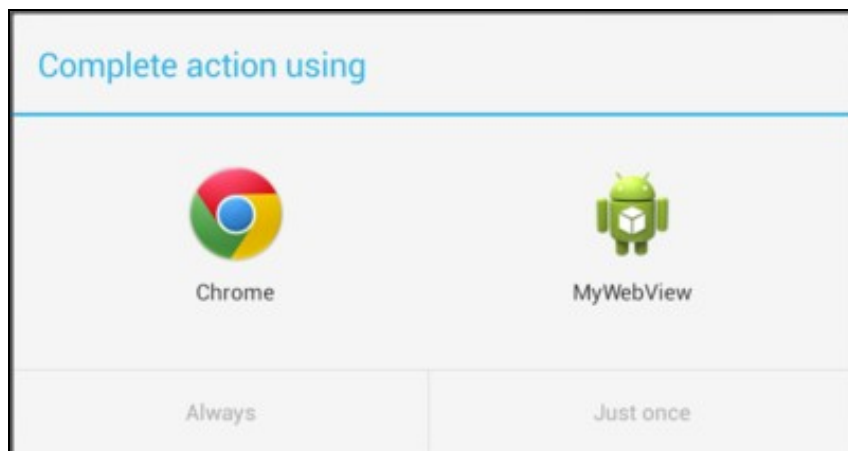
- **Liña 31:** Permisión que "permite" realizar unha chamada de teléfono sen pasar a través do interface do teléfono para que o usuario marque.
- **Liña 32:** Permisión que permite á aplicación abrir conexións de rede.
- **Liña 33:** Da súa lectura pódese concluir o que permite.
- Para coñecer os permisos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

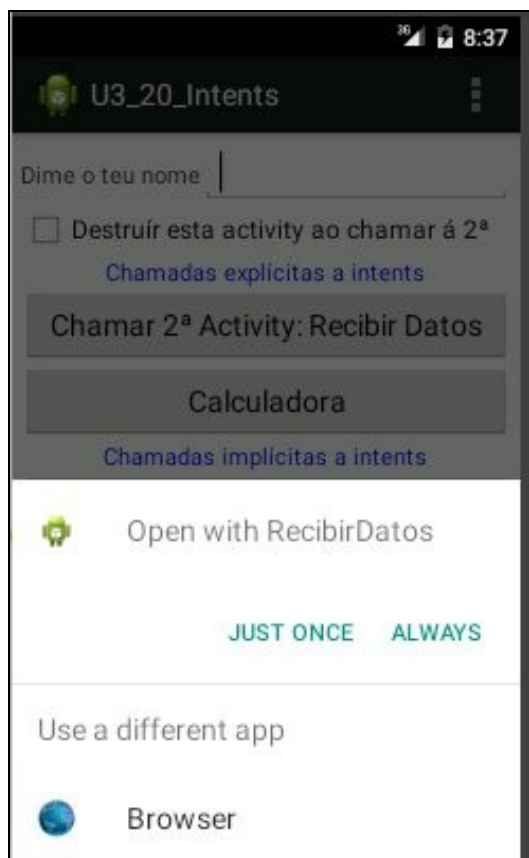
- Como xa se dixo, un compoñente pode ser lanzado de forma implícita cando se indica a **acción** que se desexa realizar e se é o caso os datos sobre os que se desexa realizar a acción.
- Non se vai especificar cal é a activity que vai atender o intent. Incluso se hai varias Activities que poidan atender 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.

```
<?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" >
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />

                <category android:name="android.intent.category.DEFAULT" />

                <data android:scheme="http" />
            </intent-filter>
        </activity>
    </application>

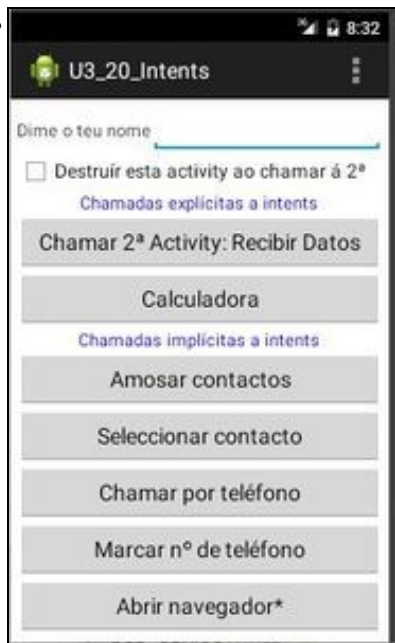
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />

</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

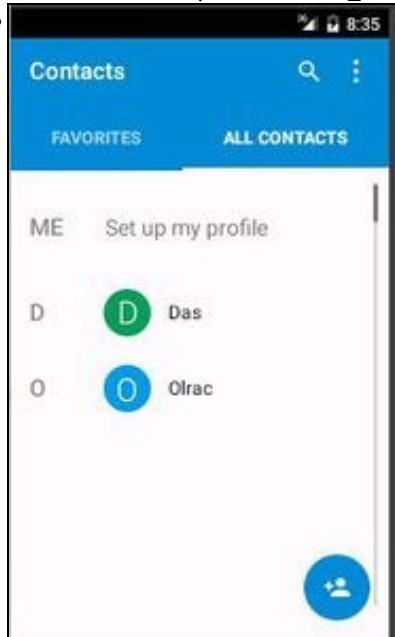
- A continuación imos engadir unha serie de botóns, modificar o código, etc ao proxecto.
- Continuar co proxecto **U3_20_Intents**.
- Intents implícitos



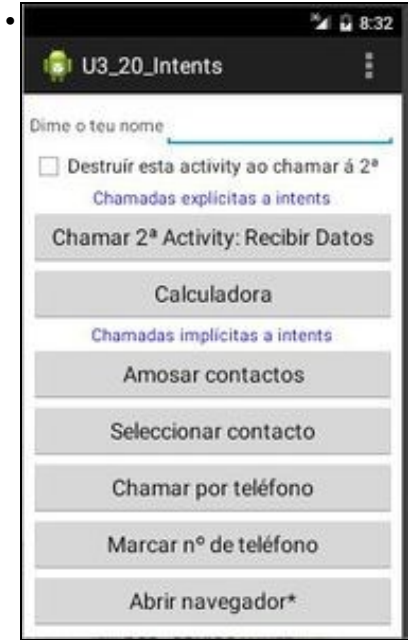
Premer en **Amosar contactos**.

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/"));
```



A lista de contactos.

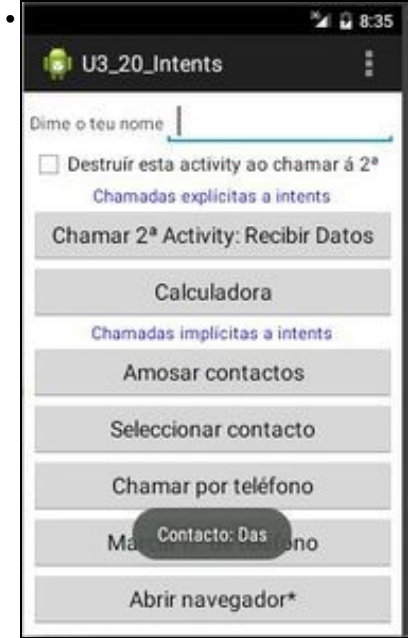


Premer en **Seleccionar contactos**. Lánzase o seguinte intent:

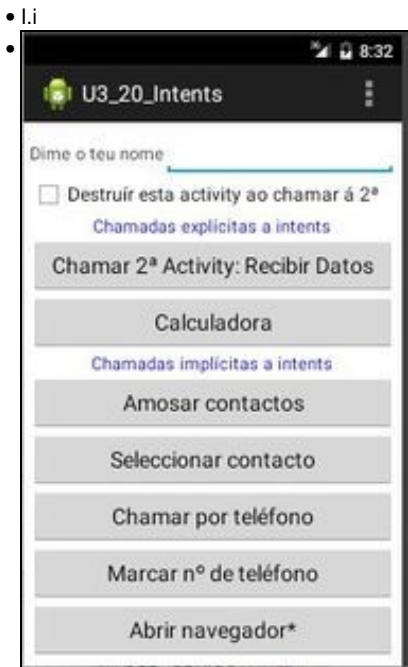
```
intent = new Intent(Intent.ACTION_PICK, Uri.parse("content://contacts/people/"));
startActivityForResult(intent, COD_CONTACTOS);
```



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.

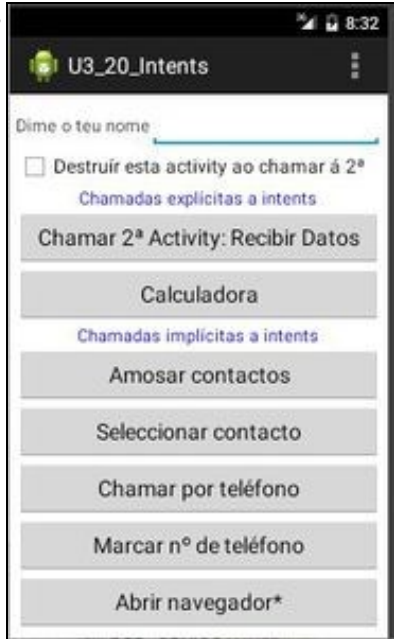


Premer en **Chamar por teléfono**. Vaise lanzar o seguinte intent:

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



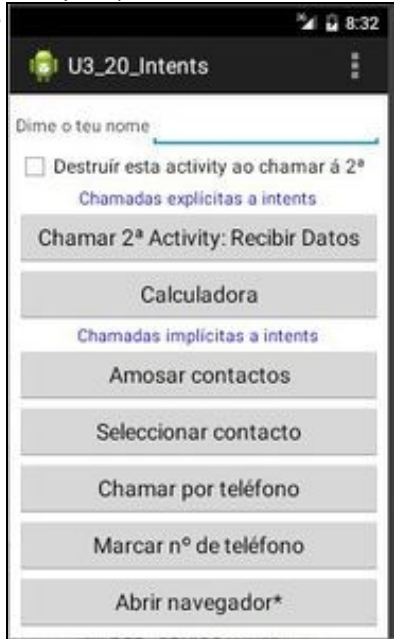
Xa iniciou a marcación ao teléfono indicado.



Premer en **Marcar nº teléfono**. Vainos lanzar a activity na que nos decidimos se marcamos o número a marcar.
`intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:(+34)986112233"));`

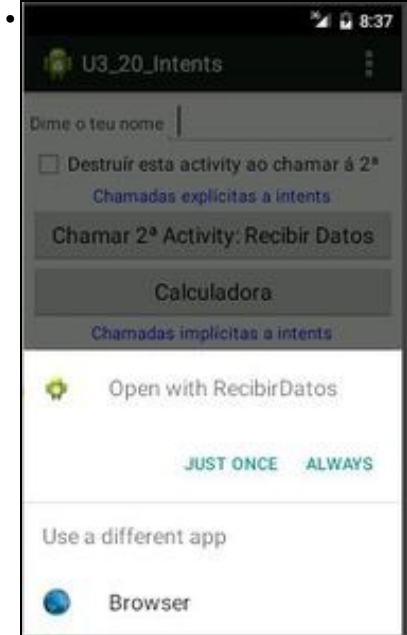


Activity na que o usuario marca o número de teléfono, pero que xa lle vén cuberto o campo.



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

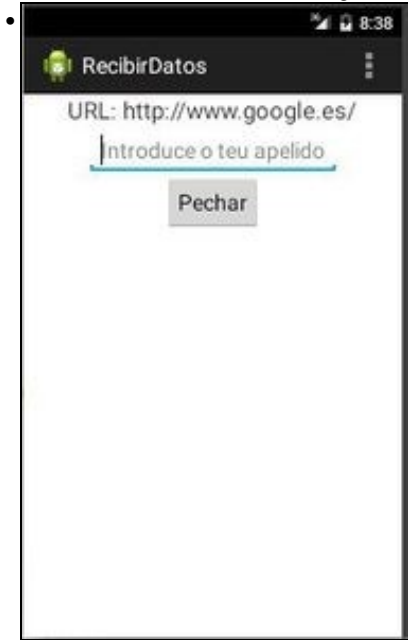
`intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.es/"));`



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



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)

```
<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" />

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

<Button
    android:id="@+id/btn_amosar_contactos"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="chamarIntent"
    android:text="Amosar contactos" />

<Button
    android:id="@+id/btn_seleccionar_contacto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="chamarIntent"
    android:text="Seleccionar contacto" />

<Button
    android:id="@+id/btn_chamar_tel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="chamarIntent"
    android:text="Chamar por teléfono" />

<Button
    android:id="@+id/btn_marcar_tel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="chamarIntent"
    android:text="Marcar nº de teléfono" />

<Button
    android:id="@+id/btn_navegador"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="chamarIntent"
    android:text="Abrir navegador*" />

</LinearLayout>

```

Código Java da activity principal

```

package com.example.u3_20_intents;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
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;
private static final int COD_CONTACTOS = 7;

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("APELLIDO"))
                Toast.makeText(this, tvNome.getText() + "\nO teu apelido é: " + data.getExtras().getString("APELLIDO"), Toast.LENGTH_SHORT).show();

        } else
            Toast.makeText(this, "Saíches da actividade secundaria sen premer o botón Pechar", Toast.LENGTH_SHORT).show();
        } else if (requestCode == COD_CONTACTOS && resultCode == RESULT_OK) {
            Uri contactoData = data.getData();
            Cursor c = managedQuery(contactoData, null, null, null, null);
            if (c.moveToFirst()) {
                String nombre = c.getString(c.getColumnIndexOrThrow(ContactsContract.Contacts.DISPLAY_NAME));
                Toast.makeText(this, "Contacto: " + nombre, Toast.LENGTH_LONG).show();
            }
        }

    }

    public void onCalculadoraClick(View v) {

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

        startActivity(intent);
    }

    public void chamarIntent(View view) {
        Intent intent = null;
        switch (view.getId()) {
            case R.id.btn_seleccionar_contacto:
                intent = new Intent(Intent.ACTION_PICK, Uri.parse("content://contacts/people/"));
                startActivityForResult(intent, COD_CONTACTOS);
                break;

            case R.id.btn_amosar_contactos:
                intent = new Intent(Intent.ACTION_VIEW, Uri.parse("content://contacts/people/"));
                startActivity(intent);
        }
    }

```

```

break;

case R.id.btn_chamar_tel:
intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:(+34) 981445566"));
startActivity(intent);
break;

case R.id.btn_marcar_tel:
intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:(+34) 986112233"));
startActivity(intent);
break;

case R.id.btn_navegador:
intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.es/"));
startActivity(intent);
break;
}
}

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

}

```

- **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.

```

<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="onPecharClick"
        android:text="Pechar" />

</LinearLayout>

```


O código Java da Activity Secundaria: RecibirDatos

```
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_Pechar_Activity;

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

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

        Intent intent = getIntent();

        if ("http".equals(intent.getScheme()))

            tvResultado.setText("URL: " + intent.getDataString());
        else
            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 onPecharClick(View view) {
        // Forma_Pechar_Activity = " Premendo botón Pechar";
        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, "Saíches da actividade secundaria: " +
        // Forma_Pechar_Activity, Toast.LENGTH_SHORT).show();
    }

}
```

- **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.