

Tecnologías y Desarrollo en Dispositivos Móviles

Apartado 10: Intents y navigation drawer

Autores:

Víctor M. Rivas Santos / Miguel Á. García Cumberras
(Antonio Rueda Ruiz)

Activity principal

- En todas las aplicaciones existe una Activity principal, identificada en el archivo AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.vrivas.intent_paises">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name="com.example.vrivas.intent_paises.MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="com.example.vrivas.intent_paises.espana" />
        <activity android:name="com.example.vrivas.intent_paises.infoPais" />
    </application>
</manifest>
```

Intent-filter

- Los Intents son los mensajes que se envían al sistema Android para arrancar una Activity. **Sirve para invocar componentes**
- *<intent-filter>* especifica qué *intents* puede procesar cada Activity
 - La acción **MAIN** indica que la Activity procesa el *intent* de arranque de la aplicación
 - Otras acciones comunes son: EDIT, VIEW, SEARCH...
 - La categoría **LAUNCHER** indica que debe ser añadido a la carpeta de aplicaciones de Android
 - Otras categorías comunes son: BROWSABLE, APP_MAPS, ...

<https://developer.android.com/reference/android/content/Intent.html>

Ejemplos Intent-filter

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

```
<activity
  android:name="com.pycitup.pyc.WebViewActivity"
  android:label="@string/title_activity_web_view" >
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="http" />
  </intent-filter>
</activity>
```

Uso de Intents

- Los *intents* se usan para invocar a una actividad desde otra:

```
public void muestraEspana( View view) {  
    try {  
  
        Intent intent = new Intent(this, espania.class);  
        startActivity(intent);  
  
    } catch (ActivityNotFoundException e) {  
  
        Toast.makeText(this  
            , "ActivityNotFound "+e.getMessage()  
            , Toast.LENGTH_SHORT).show();  
  
    }  
}
```

Aportación de información extra

- Antes de iniciar el *intent*, se le puede pasar información para que sea utilizada por la nueva actividad

```
public void muestraFrancia( View view ) {  
    Intent intent=new Intent( this, infoPais.class );  
  
    intent.putExtra(EXTRA_PAIS, "Francia");  
    intent.putExtra(EXTRA_POBLACION, 6673600);  
    intent.putExtra(EXTRA_EXTENSION, 643801.78 );  
    intent.putExtra("LENGUA_MONEDA_CAPITAL"  
        , new ArrayList<String>() {{  
            add( "Francés" );  
            add( "Euro" );  
            add( "París" );  
        }}  
    );  
    startActivity(intent);  
}
```

Recuperación de la información extra

- La actividad iniciada por el Intent puede recuperar la información que se le ha pasado

```
String nombrePais=intent.getStringExtra( MainActivity.EXTRA_PAIS );
```

```
int poblacion=intent.getIntExtra( MainActivity.EXTRA_POBLACION, 0 );
```

```
double extension=intent.getDoubleExtra( MainActivity.EXTRA_EXTENSION, 0 );
```

```
ArrayList<String> lmc=intent.getStringArrayListExtra( "LENGUA_MONEDA_CAPITAL" );
```

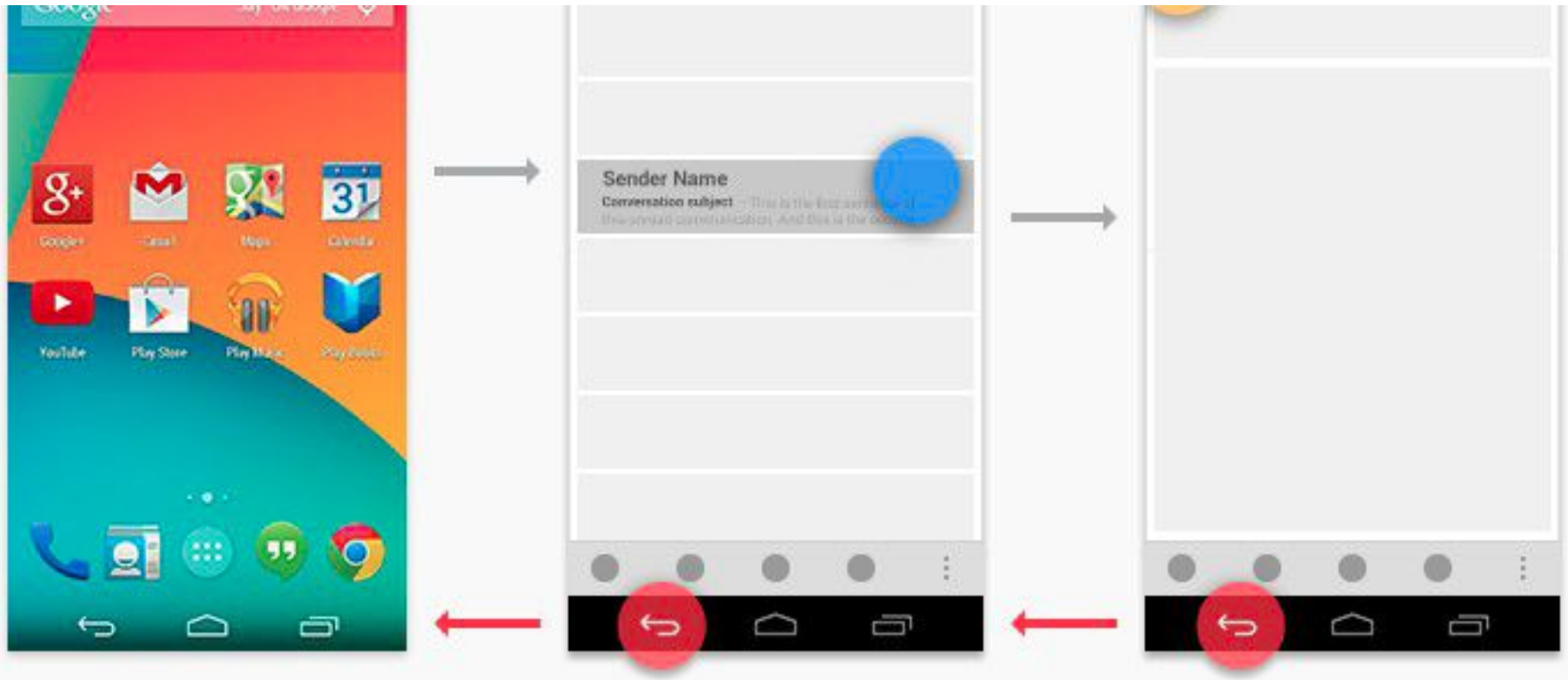
Acceder al listado de posibles *get* para *intent* (con especial atención a *Serializable*)

<https://developer.android.com/reference/android/content/Intent.html#pubmethods>

Vuelta atrás con *Back* (a)

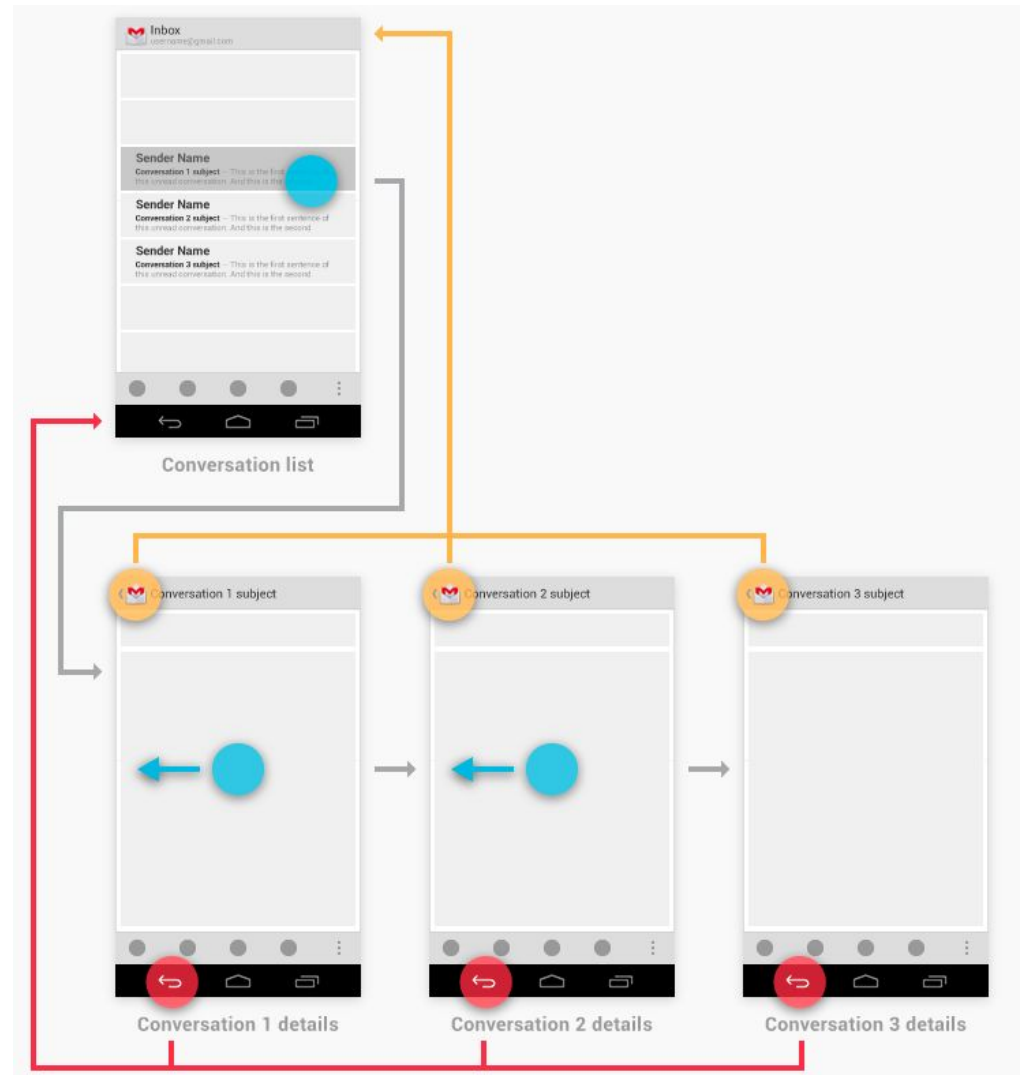
- Conforme se lanzan actividades, Android va construyendo una pila de ellas denominada *Task*
 - La primera actividad de cada Task es la propia pantalla *Launcher* de aplicaciones
 - Podemos volver a la actividad anterior de la Task pulsando el botón *Back* del dispositivo
- Salvo la última lanzada, las actividades en la pila de la Task se encuentran en estado Stop (en memoria pero inactivas)
 - Al ser recuperadas se mantienen con el contenido que tenían previamente

Vuelta atrás con *Back* (b)



Vuelta atrás con *Up* (a)

- Las últimas versiones de Android permiten la vuelta atrás en la jerarquía de actividades mediante el botón virtual: *Up*
- La jerarquía de la aplicación y el orden cronológico no tienen por qué coincidir



Vuelta atrás con *Up* (b)

- El botón de volver en la jerarquía (Up) aparece en la ActionBar de la aplicación

Layout/activity_main.xml

```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

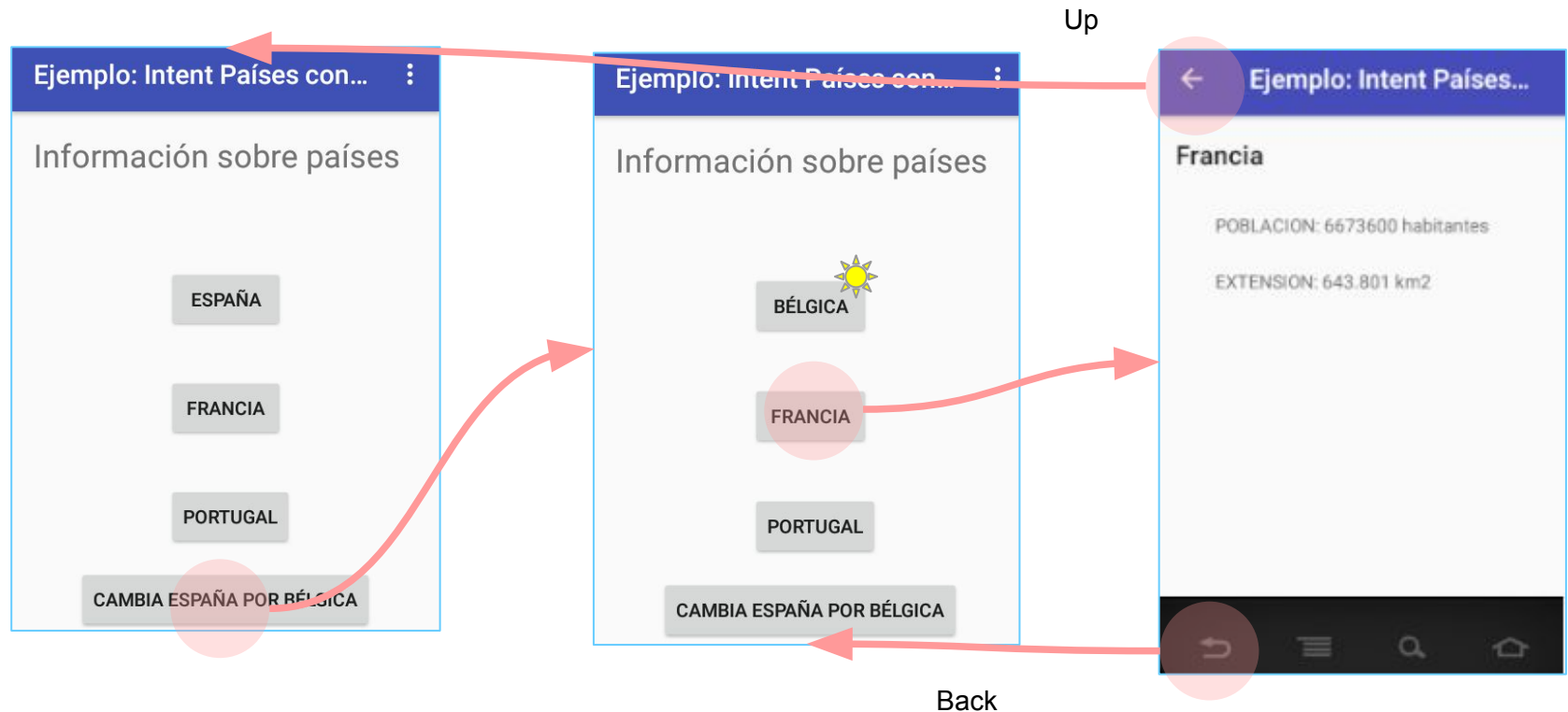
</android.support.design.widget.AppBarLayout>
```

AndroidManifest.xml

- Para activarlo la actividad debe tener declarada una *parent activity*

```
<activity android:name="com.example.vrivas.intent_paises_vuelta_atras.espana"
    android:parentActivityName=".MainActivity"/>
<activity android:name="com.example.vrivas.intent_paises_vuelta_atras.infoPais"
    android:parentActivityName=".MainActivity"/>
```

Vuelta atrás con *Up* (c)



Solucionando la vuelta atrás

- Resolvemos el problema indicando el modo *singleTop* en la actividad inicial
 - Este modo evita que existan dos instancias de una misma actividad en un Task
 - Se restaura la instancia que existía

AndroidManifest.xml

```
<activity
  android:name="com.example.vrivas.intent_paises_vuelta_atras.MainActivity"
  android:label="@string/app_name"
  android:theme="@style/AppTheme.NoActionBar"
  android:launchMode="singleTop">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Intents explícitos frente a intents implícitos

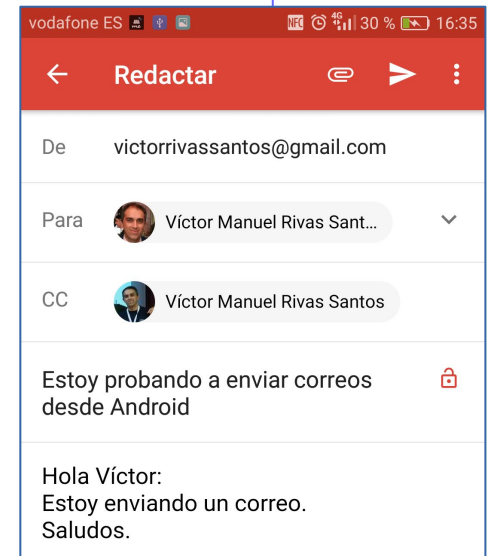
- En los *intents* **explícitos** indicamos el componente o la clase (=la actividad) a ejecutar
- En los *intents* **implícitos** se proporciona información que permite al S.O determinar qué componente debe ejecutar:
 - Una actividad
 - Un *BroadcastReceiver*
 - P. ej.: una app o componente que se lanza al reiniciarse el terminal
 - Un *Service*
 - P.ej.: una descarga de un fichero, actualización de datos ...

<https://developer.android.com/guide/components/intents-filters.html#Types>

Ejemplo de lanzamiento de intents implícitos (a)

```
public void btNavegadorClick( View v) {  
    Intent navegador = new Intent(Intent.ACTION_VIEW, Uri.parse("http://duckduckgo.com/?q=vrivas"));  
    if (navegador.resolveActivity(getPackageManager()) != null) {  
        startActivity(navegador);  
    }  
    else Toast.makeText(this, "No se puede lanzar el navegador", Toast.LENGTH_SHORT).show();  
}
```

```
public void btEmailClick(View v){  
    String datosMail = "mailto:vrivas@ujaen.es"  
        + "?cc=victorrivassantos@gmail.com"  
        + "&subject=" + Uri.encode("Estoy probando a enviar correos desde Android")  
        + "&body=" + Uri.encode("Hola Víctor:\n Estoy enviando un correo. \n Saludos.");  
  
    Intent email = new Intent(Intent.ACTION_SENDTO  
        , Uri.parse(datosMail));  
  
    if (email.resolveActivity(getPackageManager()) != null) {  
        startActivity(email);  
    }  
    else Toast.makeText(this  
        , "No se puede enviar el email"  
        , Toast.LENGTH_SHORT).show();  
}
```



Ejemplo de lanzamiento de intents implícitos (b)

```
public void btDialClick( View v ) {  
    Intent dial = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:953212344"));  
    if (dial.resolveActivity(getPackageManager()) != null) startActivity(dial);  
    else Toast.makeText(this, "No se puede realizar dial", Toast.LENGTH_SHORT).show();  
}
```

```
public void btCallClick( View v ) {  
    // Requiere gestionar permisos en función de la versión de la API  
    Intent llamada = new Intent(Intent.ACTION_CALL, Uri.parse("tel:953212344"));  
    if (llamada.resolveActivity(getPackageManager()) != null)  
        try {  
            startActivity(llamada);  
        } catch( ActivityNotFoundException e ) {  
            Toast.makeText(this  
                , "ActivityNotFoundException: " + e.getMessage(), Toast.LENGTH_SHORT).show();  
        } catch( Exception e ) {  
            Toast.makeText(this  
                , "Exception: " + e.getMessage(), Toast.LENGTH_SHORT).show();  
        }  
    else Toast.makeText(this, "No se puede realizar call", Toast.LENGTH_SHORT).show();  
}
```


Ejemplo de lanzamiento de intents implícitos (c)

```
public void btAlarmaClick(View v) {
    Intent alarma=null;
    if (Build.VERSION.SDK_INT >= 24) alarma=btAlarma24(v); else alarma=btAlarma23(v);
    if (alarma.resolveActivity(getPackageManager()) != null) startActivity(alarma);
    else Toast.makeText(this, "No se puede realizar alarma", Toast.LENGTH_SHORT).show();
}

public Intent btAlarma23(View v) {
    return new Intent(AlarmClock.ACTION_SET_ALARM)
        .putExtra(AlarmClock.EXTRA_MESSAGE, "¡P'arriba, gandul!")
        .putExtra(AlarmClock.EXTRA_HOUR, (new Date()).getHours())
        .putExtra(AlarmClock.EXTRA_MINUTES, (new Date()).getMinutes() + 1);
}

@TargetApi(24)
public Intent btAlarma24(View v) {
    return new Intent(AlarmClock.ACTION_SET_ALARM)
        .putExtra(AlarmClock.EXTRA_MESSAGE, "¡P'arriba, dormilón!")
        .putExtra(AlarmClock.EXTRA_HOUR, (Calendar.getInstance()).get(Calendar.HOUR))
        .putExtra(AlarmClock.EXTRA_MINUTES, (Calendar.getInstance()).get(Calendar.MINUTE)+1);
}
```

Delegador automático

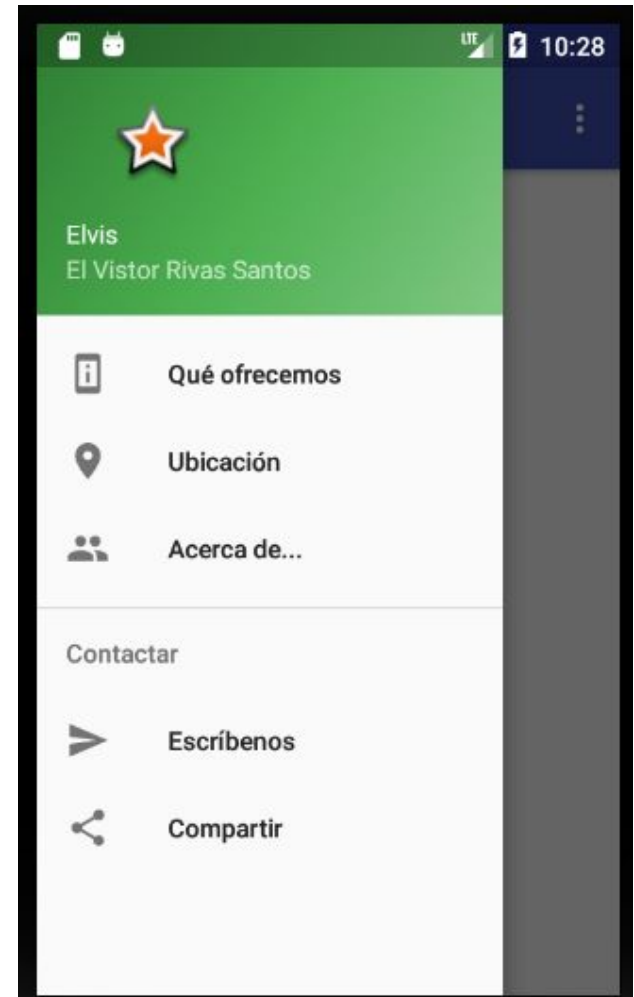
MTV: Utilizar *intents* implícitos

- Crea una app que use *intents* implícitos para:
 - Mostrar una web remota
 - Mostrar los datos de contacto de la primera persona que haya en la agenda
 - Posiblemente debas añadir a una persona en la lista de contactos del emulador ;)
 - Mostrar la galería de imágenes

Panel Lateral

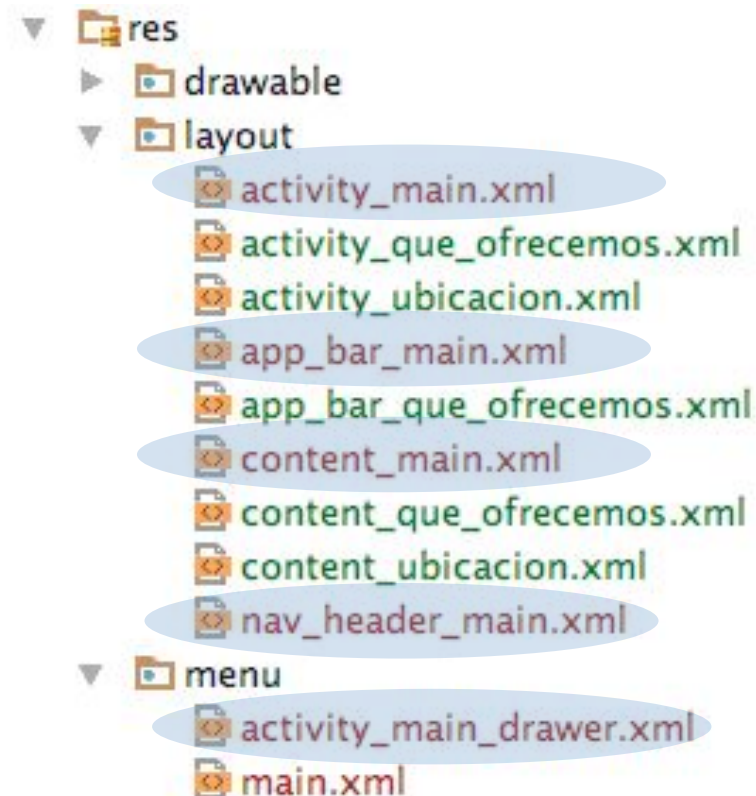
(*Navigation Drawer*)

- Es uno de los elementos más habituales en la mayoría de aplicaciones
- Permite acceder a un listado de opciones en el lateral izquierdo de nuestra app
- Está normalmente relacionado con la creación y lanzamiento de *intents*



Estructura de Layouts

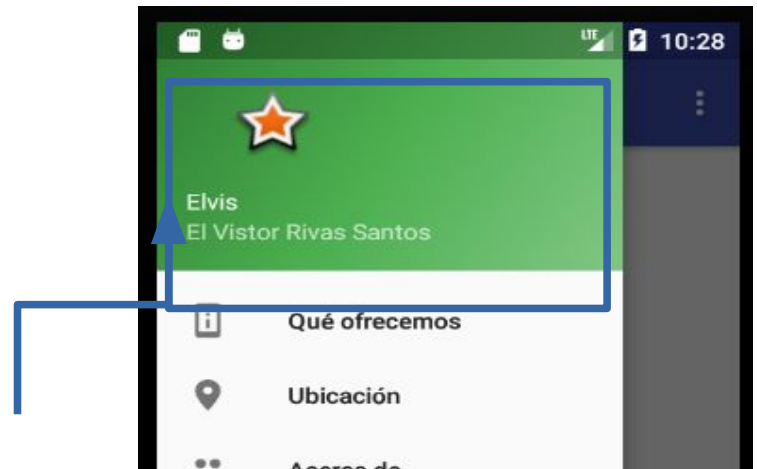
- Android Studio crea una estructura algo compleja para almacenar el panel lateral



res/layout/activity_main.xml

```
<include
    layout="@layout/app_bar_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_main"
    app:menu="@menu/activity_main_drawer" />
```



res/menu/activity_main_drawer.xml



Gestión de los eventos en el panel lateral: *MainActivity.java*

```
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
```

```
    int id = item.getItemId();
```

```
    if (id == R.id.nav_que_ofrecemos) {
```

```
    } else if (id == R.id.nav_ubicacion) {
```

```
        try {
```

```
            Intent intent = new Intent(activity, Ubicacion.class);
            activity.startActivity(intent);
```

```
        } catch (ActivityNotFoundException e) {
```

```
            Toast.makeText(activity,
                "ActivityNotFound "+e.getMessage(),
                Toast.LENGTH_SHORT).show();
        }
```

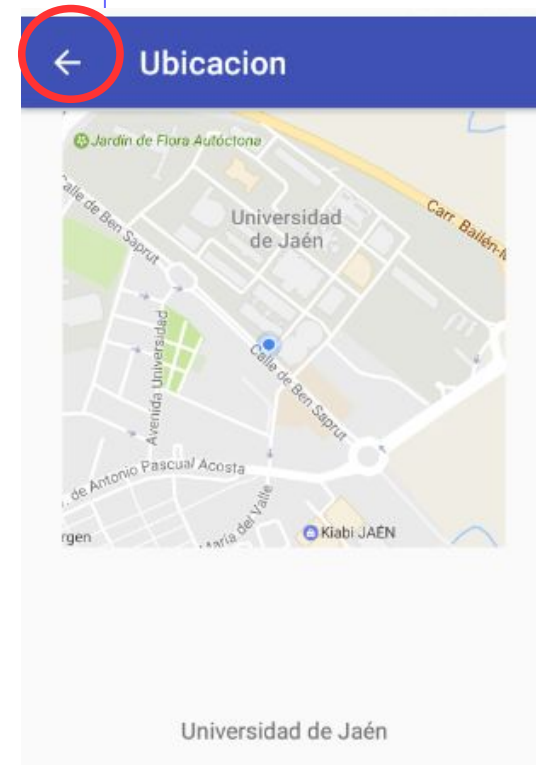
```
    } else if (id == R.id.nav_acerca_de) {
```

```
    } else if (id == R.id.nav_escribenos) {
```

```
    } else if (id == R.id.nav_compartir) {
```

```
    }
```

```
    DrawerLayout drawer = (DrawerLayout) activity.findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```



Vuelta hacia atrás (*flashback*)

AndroidManifest.xml

```
<activity
  android:name=".Ubicacion"
  android:label="@string/title_activity_ubicacion"
  android:parentActivityName=".MainActivity"
  android:theme="@style/AppTheme.NoActionBar">
  <meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value="com.example.vrivas.navigationdrawer.MainActivity" />
</activity>
```


Compartir panel lateral (I)

MainActivity.java

```
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(Menuitem item) {
    return (new DrawerManager()).onNavigationItemSelected(this, item);
}
```



Compartir panel lateral (II)

DrawerManager.java

```
package com.example.vrivas.navigationdrawer;
import ...;
public class DrawerManager {
    public boolean onNavigationItemSelected(Activity activity, MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.nav_que_ofrecemos) {
            try {
                Intent intent = new Intent(activity, QueOfrecemos.class);
                activity.startActivity(intent);
            } catch (ActivityNotFoundException e) {
                Toast.makeText(activity,
                    "ActivityNotFound "+e.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        } else if (id == R.id.nav_ubicacion) {
            try {
                Intent intent = new Intent(activity, Ubicacion.class);
                activity.startActivity(intent);
            } catch (ActivityNotFoundException e) {
                Toast.makeText(activity,
                    "ActivityNotFound "+e.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        } else if (id == R.id.nav_acerca_de) {
        } else if (id == R.id.nav_escribenos) {
        } else if (id == R.id.nav_compartir) {
        }

        DrawerLayout drawer = (DrawerLayout) activity.findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }
}
```



Opciones básicas de un panel lateral

MTV: Practicar los intents desde el panel lateral

- Realiza una app con un panel lateral que tenga, al menos, las siguientes opciones:
 - *Contactar*: llevará al usuario a una nueva actividad donde se le permitirá introducir los datos “remitente” y “tema de la pregunta/queja/sugerencia”; al pulsar el botón “Enviar” se enviará un correo a nuestra dirección de email con dichos datos.
 - *Acerca de...*: llevará al usuario a una nueva actividad en la que se mostrará nuestra foto e información acerca de la versión de la app, fecha de lanzamiento, etc.
 - *Ayuda*: llevará al usuario a una nueva actividad en la que se cargará una página web local con ayuda sobre la aplicación