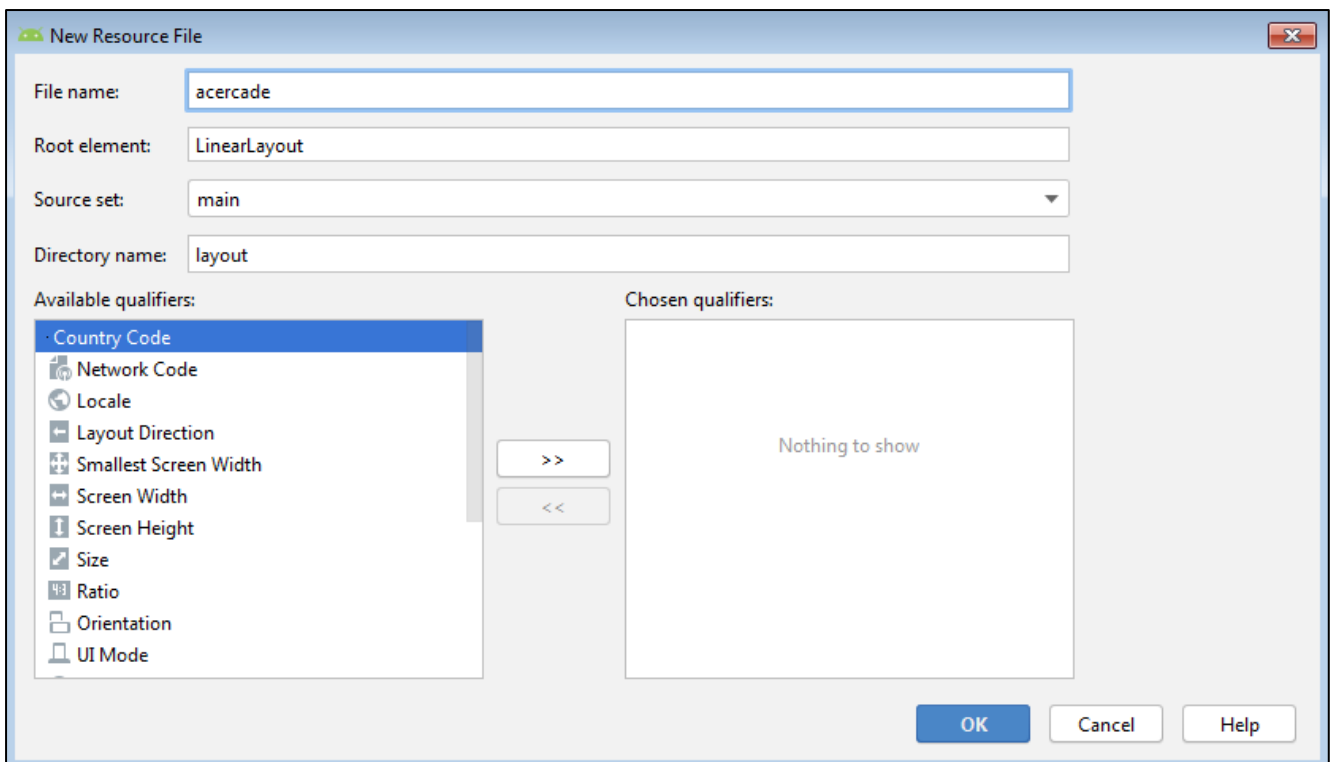
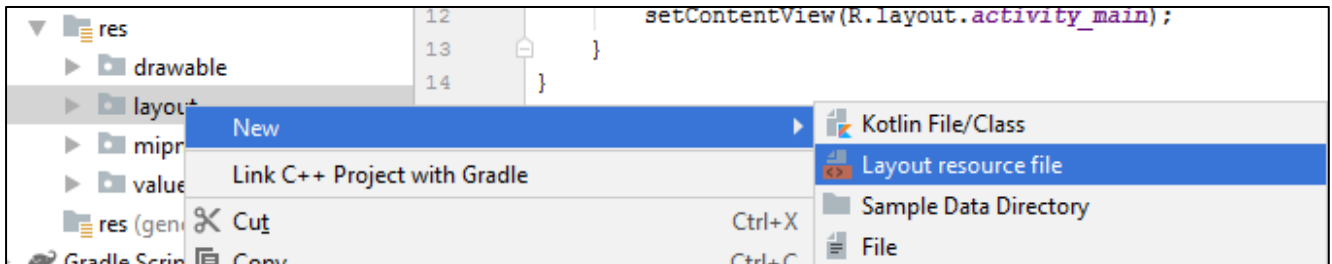


## PRACTICA 11: ASTEROIDES

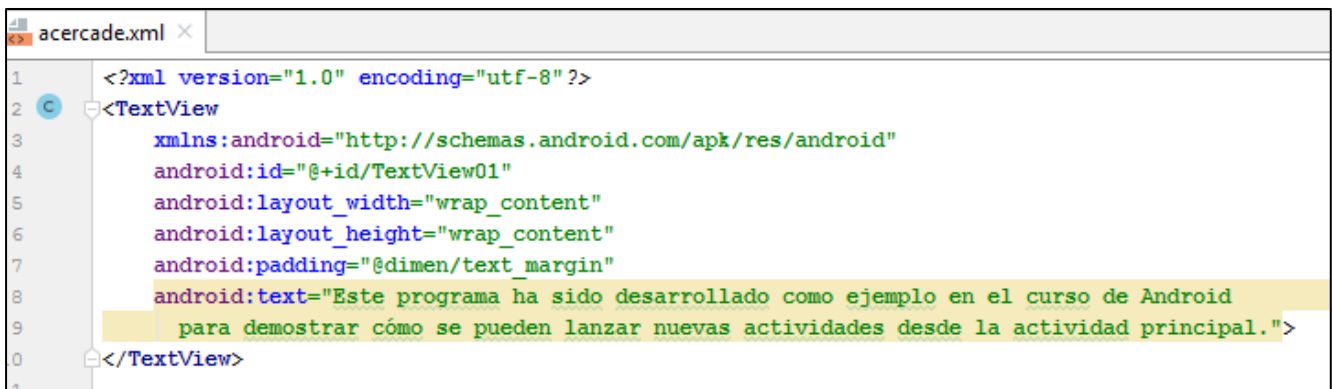
### Parte I: Implementación de una caja Acerca de

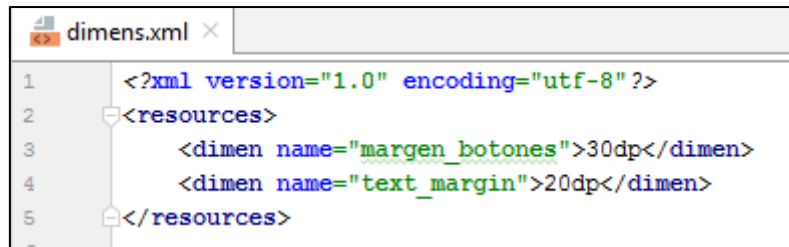
Dentro del proyecto Asteroides, vamos a crear una caja *Acerca de...* y visualizarla cuando se pulse el botón adecuado.

**Paso 1.** En primer lugar crea el fichero `res/layout/acercade.xml`. Para ello pulsa con el botón derecho sobre el explorador del proyecto en la carpeta `res/layout` y selecciona `New > Layout resource file`. Indica en File name: `acercade`.



**Paso 2.** Selecciona la lengüeta `Text` y copia el siguiente contenido:



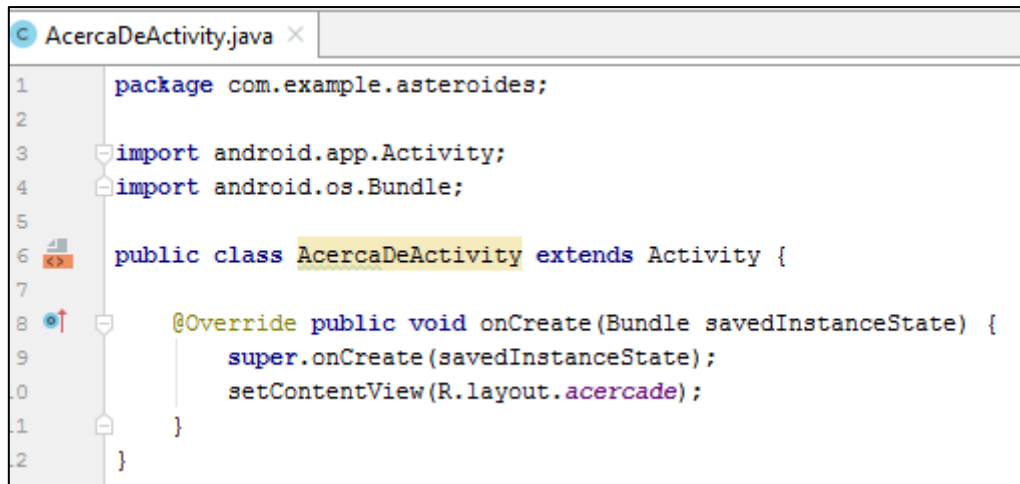


```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <resources>
3      <dimen name="margen_botones">30dp</dimen>
4      <dimen name="text_margin">20dp</dimen>
5  </resources>

```

**Paso 3.** Creamos ahora una nueva actividad, que será la responsable de visualizar esta vista. Para ello crea el fichero `AcercaDeActivity.java`, pulsando con el botón derecho sobre el nombre del paquete de la aplicación y seleccionando `New > Java Class`. En el campo `Name` introduce `AcercaDeActivity` y pulsa `Finish`. Reemplaza el código por:

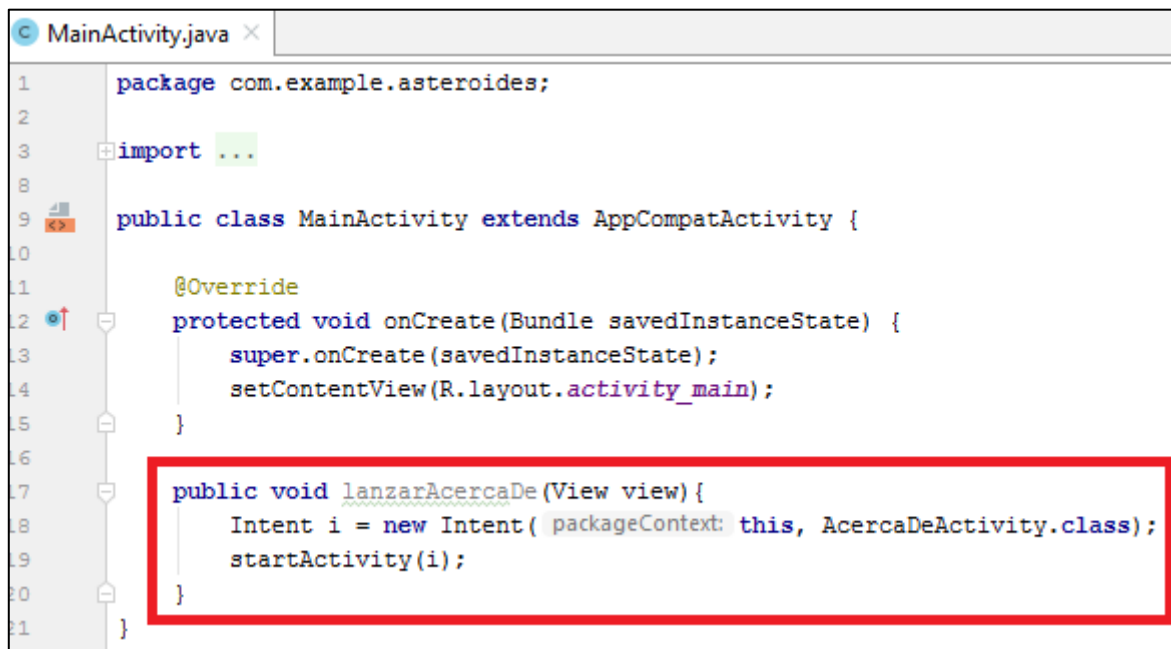


```

1  package com.example.asteroides;
2
3  import android.app.Activity;
4  import android.os.Bundle;
5
6  public class AcercaDeActivity extends Activity {
7
8      @Override public void onCreate(Bundle savedInstanceState) {
9          super.onCreate(savedInstanceState);
10         setContentView(R.layout.acercade);
11     }
12 }

```

**Paso 4.** Pasemos ahora a crear un método en la clase `Asteroides.java` que será ejecutado cuando sea pulsado el botón `Acerca de`.



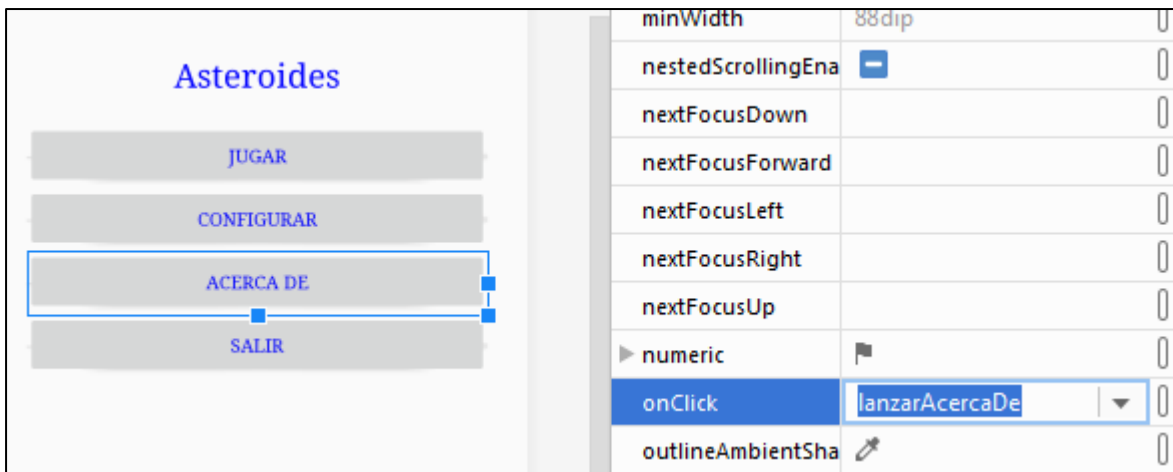
```

1  package com.example.asteroides;
2
3  import ...
4
5  public class MainActivity extends AppCompatActivity {
6
7      @Override
8      protected void onCreate(Bundle savedInstanceState) {
9          super.onCreate(savedInstanceState);
10         setContentView(R.layout.activity_main);
11     }
12
13     public void lanzarAcercaDe(View view) {
14         Intent i = new Intent( packageContext: this, AcercaDeActivity.class);
15         startActivity(i);
16     }
17 }

```

**Nota sobre Java:** Pulsa **Alt-Intro**, para que automáticamente se añadan los paquetes que faltan.

**Paso 5.** Para asociar este método al botón edita el `Layout activity_main.xml`. Selecciona la lengüeta `Design` y pulsa sobre el botón `Acerca de...` y en la vista `Properties` busca el atributo `onClick` e introduce el valor `lanzarAcercaDe`.



Se ha de hacer esto para las el **activity\_main.xml** y para el **activity\_main.xml (land)**.

**Paso 6.** Pasa a la edición xml pulsando en la lengüeta *main.xml* y observa como en la etiqueta `<Button>` correspondiente, se ha añadido el atributo:

`android:onClick="lanzarAcercaDe"`

```
<Button
    android:id="@+id/Button03"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="lanzarAcercaDe"
    android:text="Acerca De" />
```

**NOTA:** En caso de que exista algún recurso alternativo para el layout repite el mismo proceso.

**Paso 7.** Ejecuta ahora la aplicación y pulsa en el botón *Acerca de*. Observarás que el resultado no es satisfactorio ¿Qué ha ocurrido?

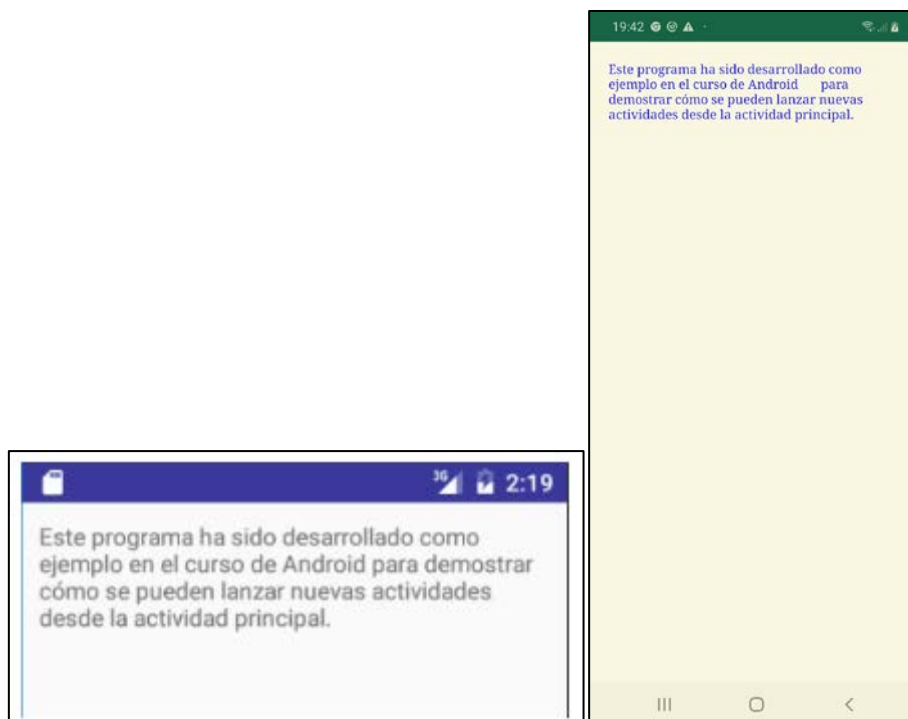
```
2019-10-29 19:19:22.549 15285-15285/? E/AndroidRuntime: FATAL EXCEPTION: main
Process: com.example.asteroides, PID: 15285
java.lang.IllegalStateException: Could not execute method for android:onClick
    at androidx.appcompat.app.AppCompatActivity$DeclaredOnClickListener.onClick(AppCompatActivity.java:402)
    at android.view.View.performClick(View.java:7339)
    at android.widget.TextView.performClick(TextView.java:14222)
    at android.view.View.performClickInternal(View.java:7305)
    at android.view.View.access$3200(View.java:245)
    at android.view.View$PerformClick.run(View.java:27787)
    at android.os.Handler.handleCallback(Handler.java:873)
    at android.os.Handler.dispatchMessage(Handler.java:99)
    at android.os.Looper.loop(Looper.java:214)
    at android.app.ActivityThread.main(ActivityThread.java:7078) <1 internal call>
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:493)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:975)
Caused by: java.lang.reflect.InvocationTargetException <1 internal call>
    at androidx.appcompat.app.AppCompatActivity$DeclaredOnClickListener.onClick(AppCompatActivity.java:397) <9 more...> <1 internal call> <2 more...>
Caused by: android.content.ActivityNotFoundException: Unable to find explicit activity class {com.example.asteroides/com.example.asteroides.AcercaDeActivity}; have you declared this activity in your AndroidManifest.xml?
    at android.app.Instrumentation.checkStartActivityResult(Instrumentation.java:12020)
    at android.app.Activity.startActivityForResult(Activity.java:4689)
    at androidx.fragment.app.FragmentActivity.startActivityForResult(FragmentActivity.java:676)
    at android.app.Activity.startActivityForResult(Activity.java:4647)
    at androidx.fragment.app.FragmentActivity.startActivityForResult(FragmentActivity.java:663)
    at android.app.Activity.startActivity(Activity.java:5008)
    at android.app.Activity.startActivity(Activity.java:4976)
    at com.example.asteroides.MainActivity.lanzarAcercaDe(MainActivity.java:19) <1 internal call> <10 more...> <1 internal call> <2 more...>
```

El problema es que toda actividad que ha de ser lanzada por una aplicación ha de ser registrada en el fichero **AndroidManifest.xml**. Para registrar la actividad, abre **AndroidManifest.xml**. Añade el siguiente texto dentro de la etiqueta `<application ...>` `</application>`:

`<activity android:name=".AcercaDeActivity" android:label="Acerca de ..."/>`



**Paso 8.** Ejecuta de nuevo el programa. El resultado ha de ser similar al mostrado a continuación:



La vista mostrada en el ejemplo anterior no parece muy atractiva. Tratemos de mejorarla aplicando un tema. Como vimos en el capítulo anterior un tema es una colección de estilos que define el aspecto de una actividad o aplicación. Puedes utilizar alguno de los temas disponibles en Android o crear el tuyo propio.

**Paso 9.** En este caso utilizaremos uno de los de Android. Para ello abre *AndroidManifest.xml* e introduce la línea subrayada:

```
<activity android:label="Acerca de ..." android:name=".AcercaDeActivity"
        android:theme="@style/Theme.AppCompat.Light.Dialog" />
```

**Paso 10.** Ejecuta de nuevo el programa y observa cómo mejora la apariencia:



## Parte II: Un escuchador de evento por código

Como acabamos de ver en un layout podemos definir el atributo XML `android:onClick` que nos permite indicar un método que será ejecutado al hacer click en una vista. A este método se le conoce como escuchador de evento. Resulta muy sencillo y además está disponible en cualquier descendiente de la clase `View`. Sin embargo esta técnica presenta dos inconvenientes. Solo está disponible para el evento `onClick()`. La clase `View` tiene otros eventos (`onLongClick()`, `onFocusChange()`, `onKey()`,...) para los que no se han definido un atributo xml. Entonces, ¿qué hacemos si queremos definir un evento distinto de `onClick()`. La respuesta la encontrarás en este ejercicio:

**Paso 1.** Abre la clase `Asteroides.java` y añade las líneas que aparecen subrayadas:

```
public class MainActivity extends AppCompatActivity {

    private Button bAcercaDe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bAcercaDe = findViewById(R.id.button03);
        bAcercaDe.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                lanzarAcercaDe( view: null);
            }
        });

        public void lanzarAcercaDe(View view){
            Intent i = new Intent( packageContext: this, AcercaDeActivity.class);
            startActivity(i);
        }
    }
}
```

**Nota sobre Java:** Pulsa **Alt-Intro** para que automáticamente se añadan los paquetes que faltan. Para la clase OnClickListener selecciona android.view.View.OnClickListener.

**Paso 2.** Elimina el atributo añadido al botón:

android:onClick="lanzarAcercaDe"

```
<Button
    android:id="@+id/Button03"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Acerca De"/>

```

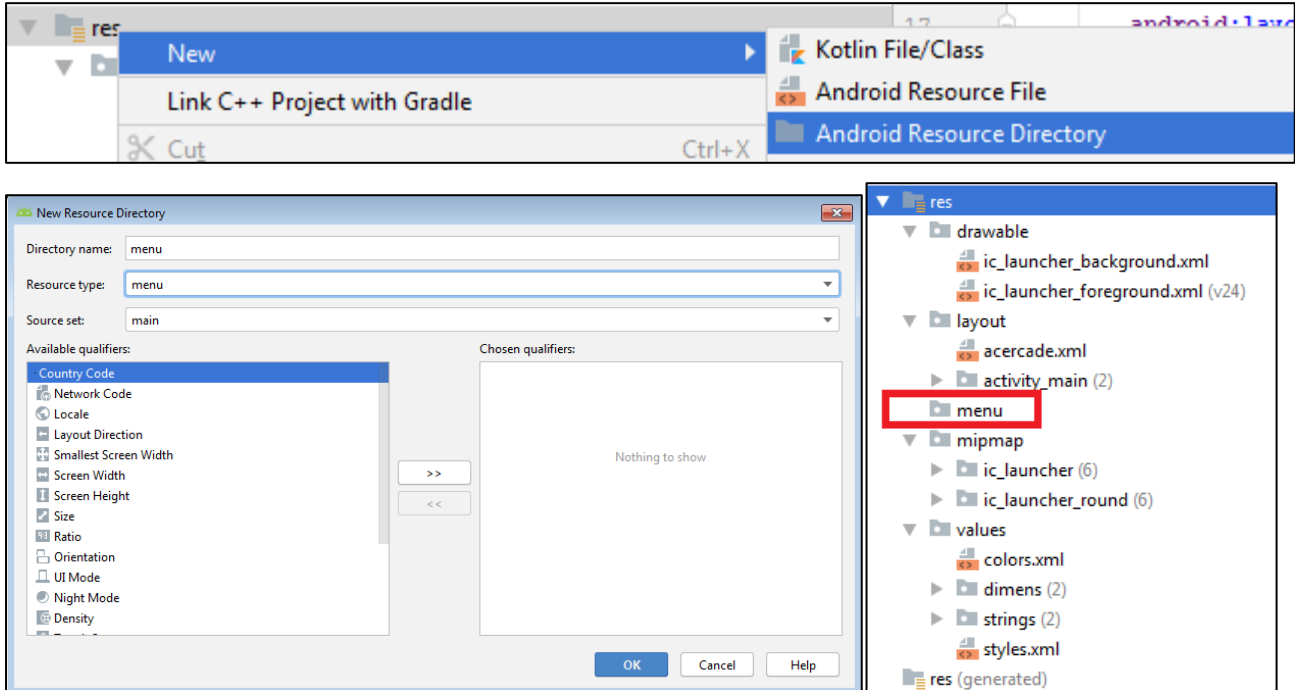
**Paso 3.** Verifica que el resultado es el mismo en ambos casos.

**NOTA:** No es conveniente que en tus actividades incluyas un botón para cerrarlas. Un dispositivo Android siempre dispone de la tecla «retorno», que tiene la misma función.

#### **Parte IV: Añadiendo un menú a una actividad**

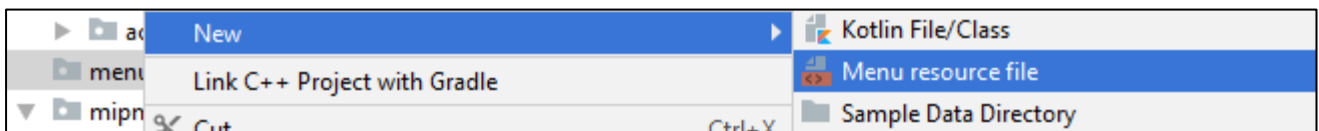
Podemos asignar un menú a nuestra actividad de forma muy sencilla.

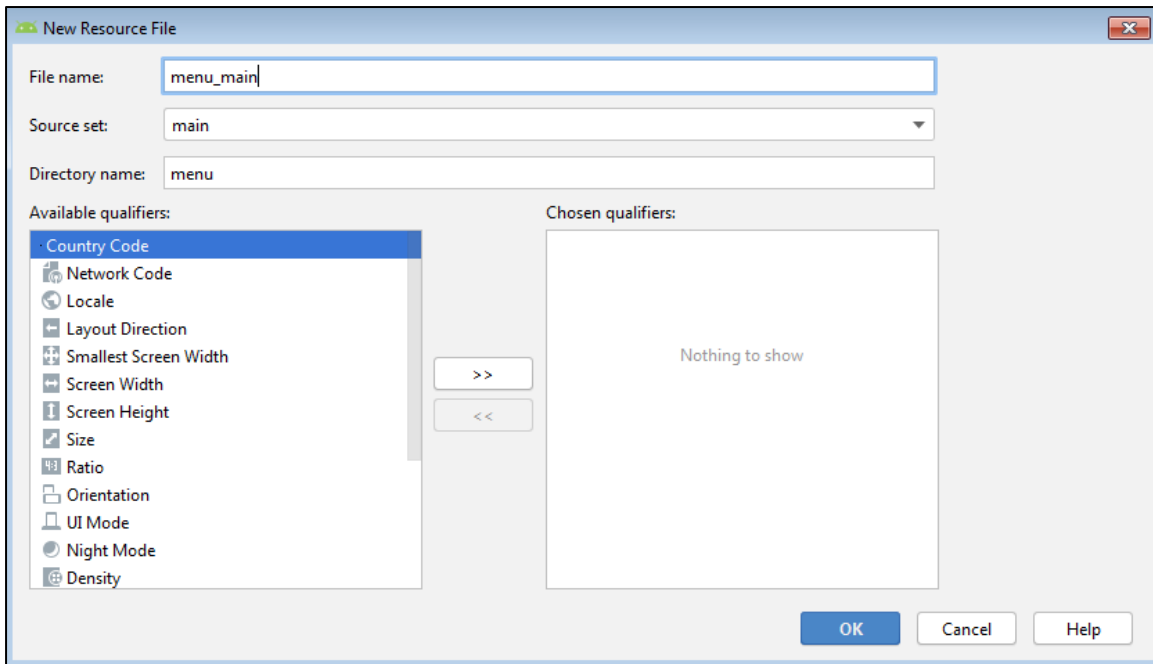
**Paso 1.** Abre el proyecto Asteroides. Crea una carpeta de recursos Android llamada menú



**Paso 2.** Crea un nuevo menú, usa File > New > Android resource file. En el campo File name: selecciona menu\_main y en el campo Resource type: selecciona Menu.

Si desarrollas Mis Lugares abre el fichero res / menu / menu\_main.xml.





**NOTA:** El fichero de menú se crea automáticamente si seleccionas una actividad de tipo: Basic Activity o Scrolling Activity.

**Paso 3.** Reemplaza el contenido que se muestran a continuación:

```

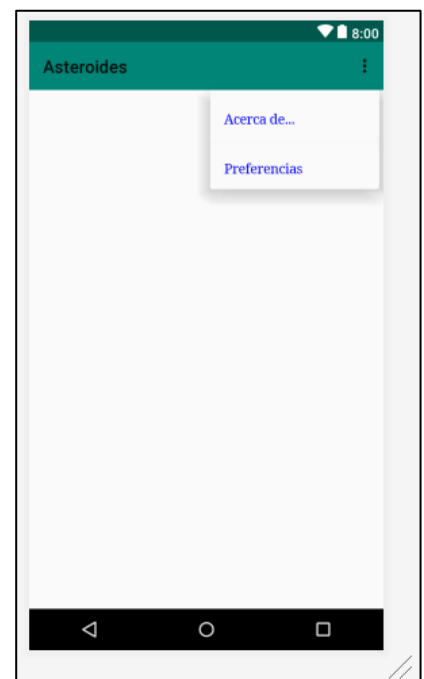
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto">
4      <item android:id="@+id/action_settings"
5          android:title="Preferencias"
6          android:icon="@android:drawable/ic_menu_preferences"
7          android:orderInCategory="100"
8          app:showAsAction="never"/>
9      <item android:title="Acerca de..."
10         android:id="@+id/acercaDe"
11         android:icon="@android:drawable/ic_menu_info_details"/>
12  </menu>

```

Como puedes ver cada ítem de menú tiene tres atributos principales: id que permite identificarlo desde el código, title, para asociarle un texto e icon, para asociarle un icono. Los otros dos atributos se aplican cuando el menú actúa como una barra de acciones y serán explicados en el próximo punto.

**Paso 4.** A continuación se muestra la apariencia de este menú. Esta apariencia puede cambiar dependiendo de la versión de Android.

**Paso 5.** Para activar el menú, has de introducir el siguiente código en la actividad que muestra el menu. Posiblemente solo tengas que incluir el texto subrayado.





```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true; /** true -> el menú ya está visible */
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    if (id == R.id.acercaDe) {
        lanzarAcercaDe( view: null);
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

**Paso 6.** Ejecuta el programa y pulsa en el icono del menú en la esquina superior derecha. Han de aparecer los dos ítems de menú. Selecciona Acerca de... para pasar a la actividad correspondiente.

