

PRACTICA 12: ASTEROIDES

Parte V: Añadiendo una barra de acciones a nuestra aplicación.

Paso 1. Reemplaza el contenido del fichero `res/menu/menu_main.xml` por:



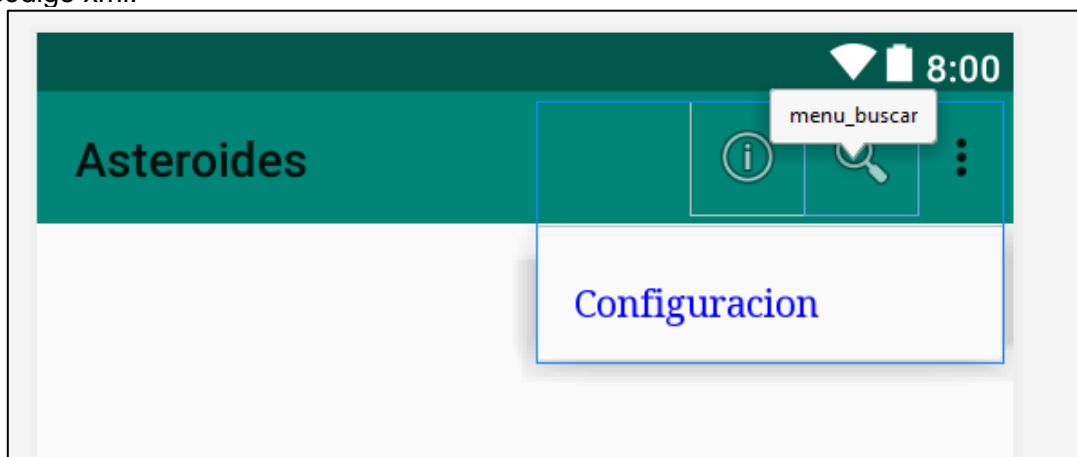
En este fichero se define los iconos y las acciones a mostrar en la barra. Las acciones que indiquen en el atributo `showAsAction` la palabra `always` se mostrarán siempre, sin importar si caben o no. El uso de estas acciones debería limitarse, lo ideal es que haya una o dos, ya que al forzar que se visualicen todas podrían verse incorrectamente. Las acciones que indiquen `ifRoom` se mostrarán en la barra de acciones si hay espacio disponible, y se moverán al menú de *Overflow* si no lo hay. En esta categoría se deberían encontrar la mayoría de las acciones. Si se indica `never`, la acción nunca se mostrará en la barra de acciones, sin importar el espacio disponible. En este grupo se deberían situar acciones como modificar las preferencias, que deben estar disponibles para el usuario, pero no visibles en todo momento.

Las acciones se ordenan de izquierda a derecha según lo indicado en `orderInCategory`, con las acciones con un número más pequeño más a la izquierda. Si no caben todas las acciones en la barra, las que tienen un número mayor se mueven al menú de *Overflow*.

Paso 2. Ejecuta la aplicación. Podrás ver como aparece la barra de acciones en la parte de arriba, con los botones que hemos definido.



Android Studio incorpora un editor visual de menús que nos permite crear menús sin necesidad de escribir código xml.



Parte VI: Creación de iconos personalizados para Asteroides

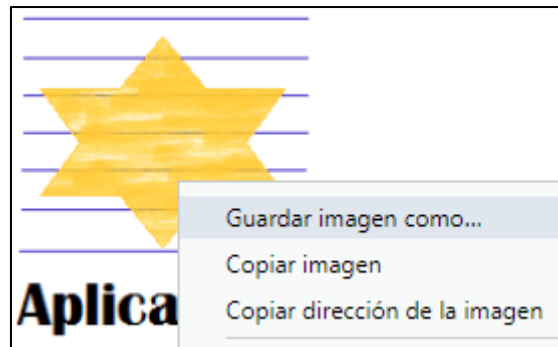
Veamos un ejemplo práctico de cómo crear un icono: El diseñador gráfico de nuestra empresa nos acaba de pasar el icono asociado para iniciar la aplicación que estamos diseñando (launcher icon).



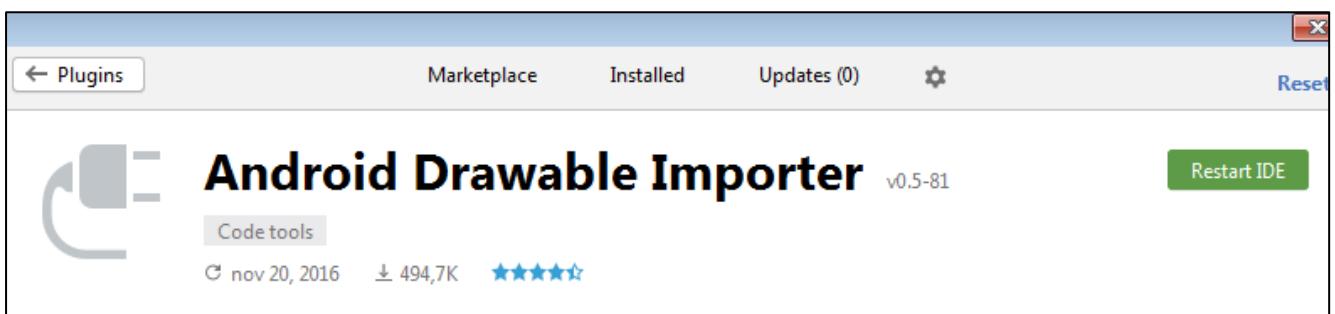
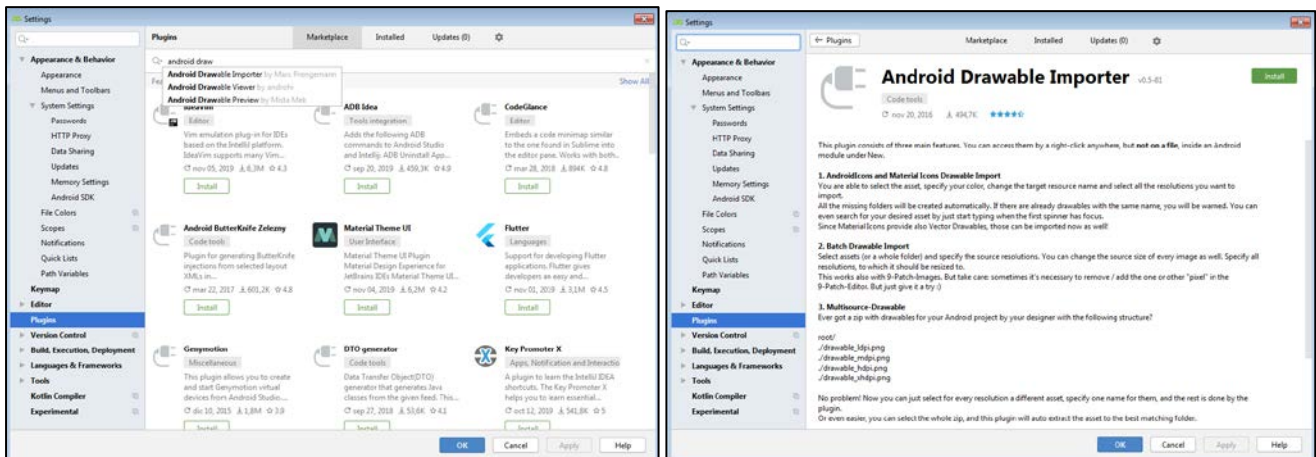
(Encontrarás este icono en el moodle)

Para asignarlo a una aplicación realiza los siguientes pasos:

Paso 1. Descarga el gráfico anterior del Moodle debajo de este enunciado y llámale *icon.png*.

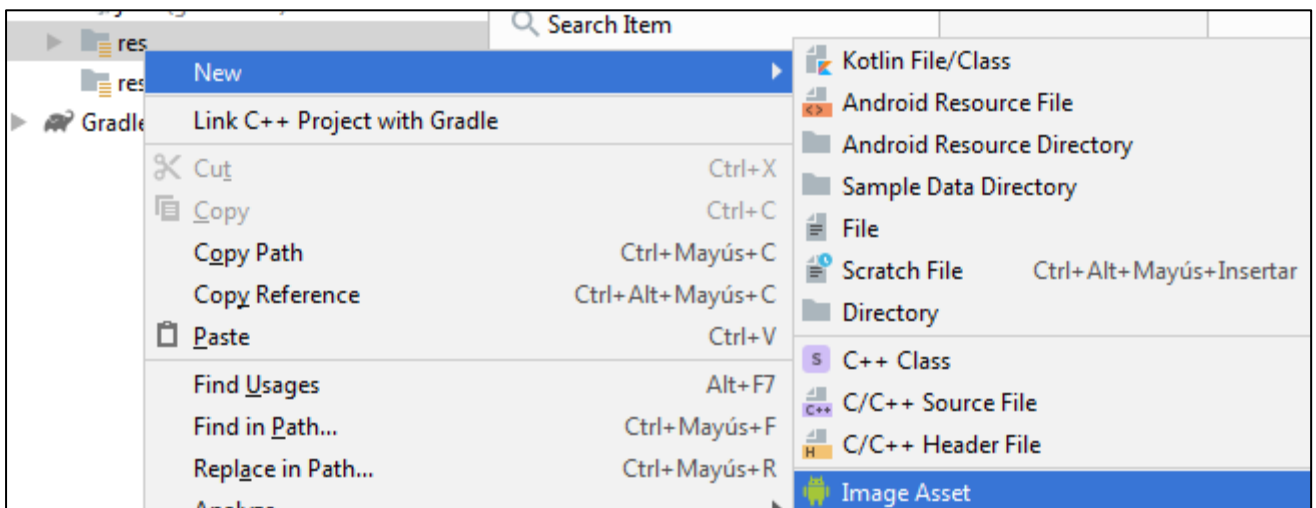


Paso 2. Instala el plugin Android Drawable Importer. Ve a File/Settings/Plugins y busca el plugin. Instálalo y reinicia Android Studio.

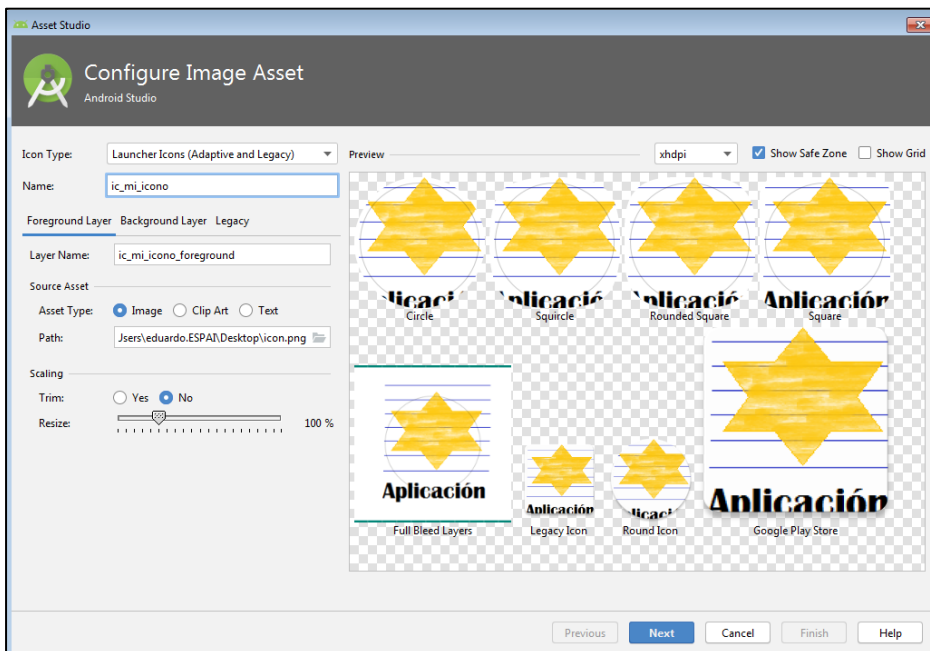
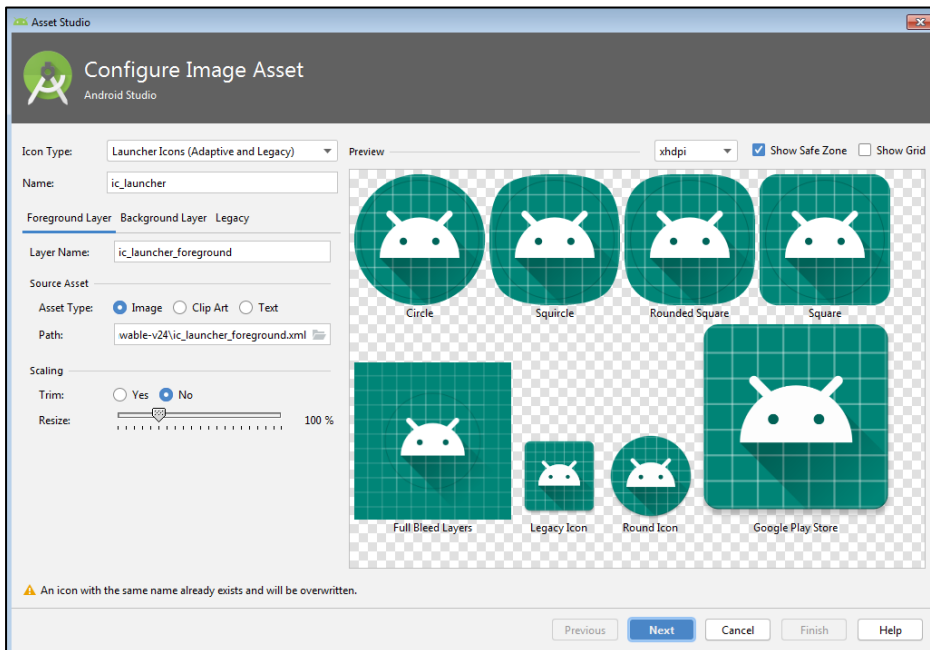


Si no te aparece en pluguins es probable que tu versión de Android ya incluya el plugin.

Paso 3. En el explorador proyecto pulsa con el botón derecho sobre la carpeta res y selecciona File / New / Image Assets.

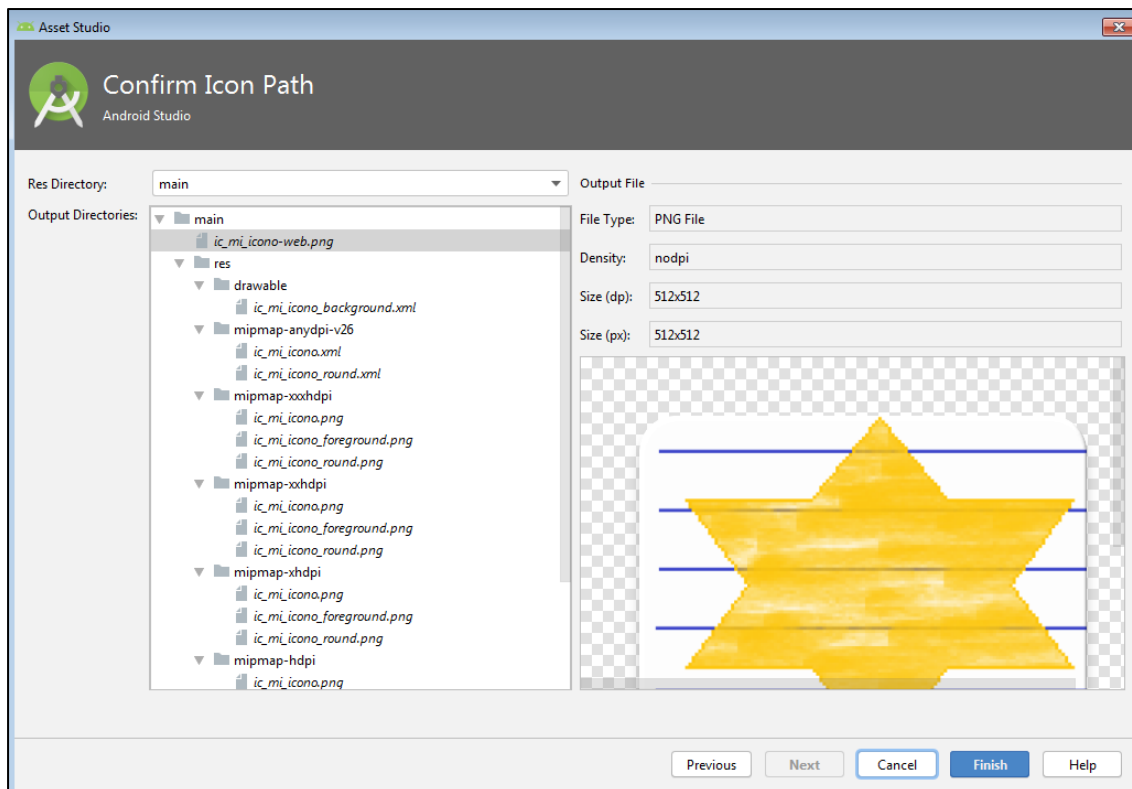


Paso 4. En el campo Name: introduce ic_mi_icono; en Asset Type: Image y en Path: indica el fichero descargado en el paso 1. El resultado ha de ser similar a:

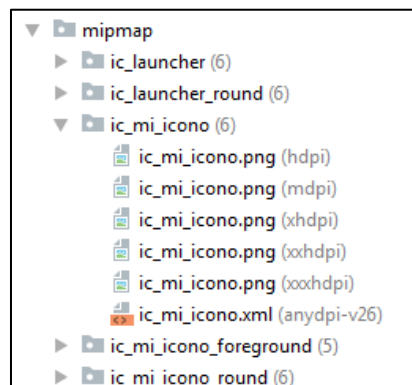


En la parte inferior de la página podrás previsualizar cómo quedarán las imágenes para diferentes densidades gráficas.

Paso 5. Pulsa el botón Next y en la siguiente pantalla pulsa Finish.



Paso 6. Verifica como dentro de la carpeta res/mipmap se han creado 5 recursos alternativos para diferentes densidades gráficas:



Paso 7. El resultado es aceptable para algunas opciones. Pero en algunos casos el texto no se lee o unas líneas horizontales quedan más gruesas que otras. Puede ser interesante retocar estos ficheros png usando un editor de gráficos. Por ejemplo, retoca los iconos hdpi y mdpi para que se pueda leer más claramente el texto “Aplicación” y redibuja las líneas horizontales para que tenga el mismo grosor y separación.

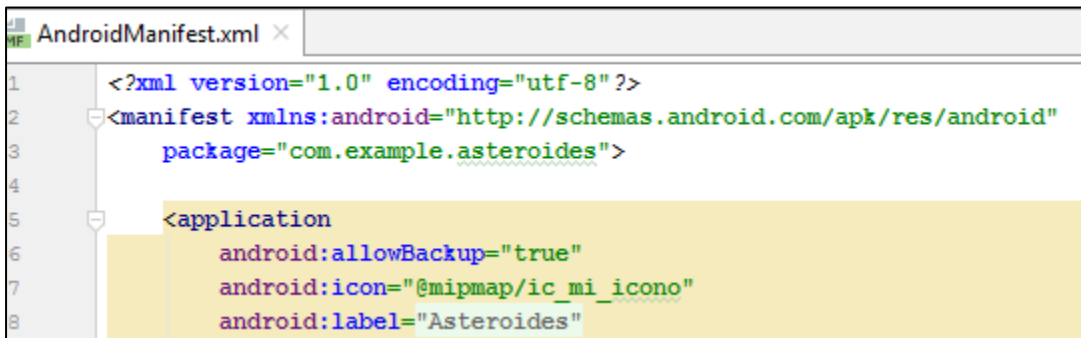


Paso 8. Para que el nuevo icono sea utilizado como lanzador de nuestra aplicación, abre AndroidManifest.xml y reemplaza el valor del atributo icon como se muestra a continuación:

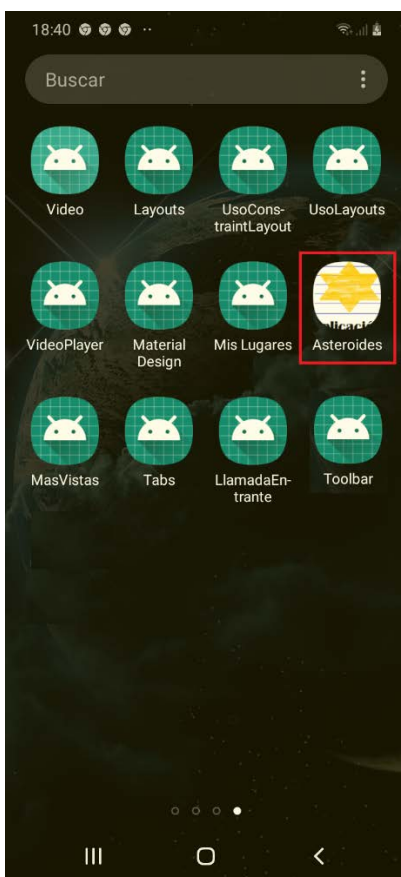
<application

android:allowBackup="true"

android:icon="@mipmap/ic_launchermi_icono"



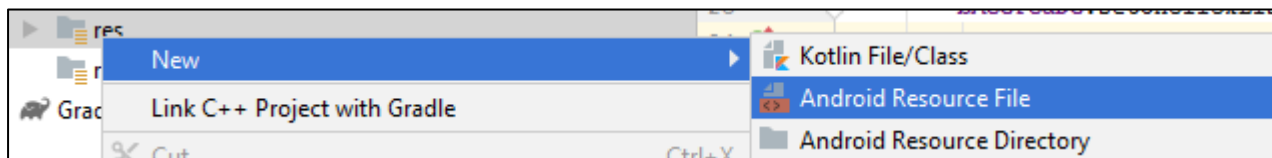
Paso 9. Visualiza el resultado instalando la aplicación en diferentes dispositivos con diferentes densidades gráficas.



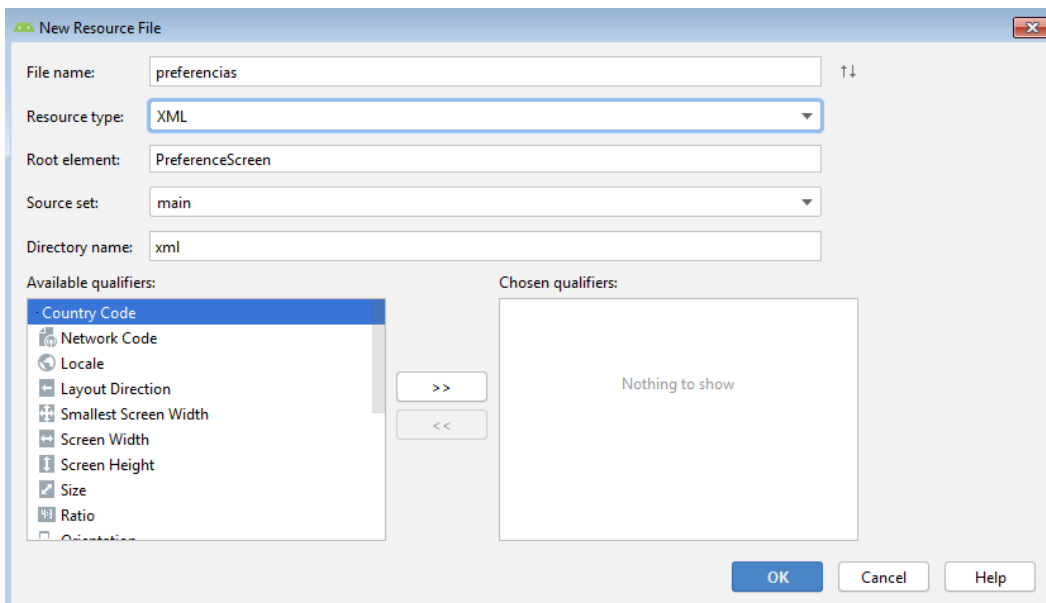
Parte VII: Añadiendo preferencias Asteroides

Paso 1. Abre el proyecto Asteroides.

Paso 2. Pulsa con el botón derecho sobre la carpeta *res* y selecciona la opción *New > Android resource File*.



Paso 3. Completa los campos File name: preferencias y Resource type: XML. Se creará el fichero *res/xml/preferencias.xml*.

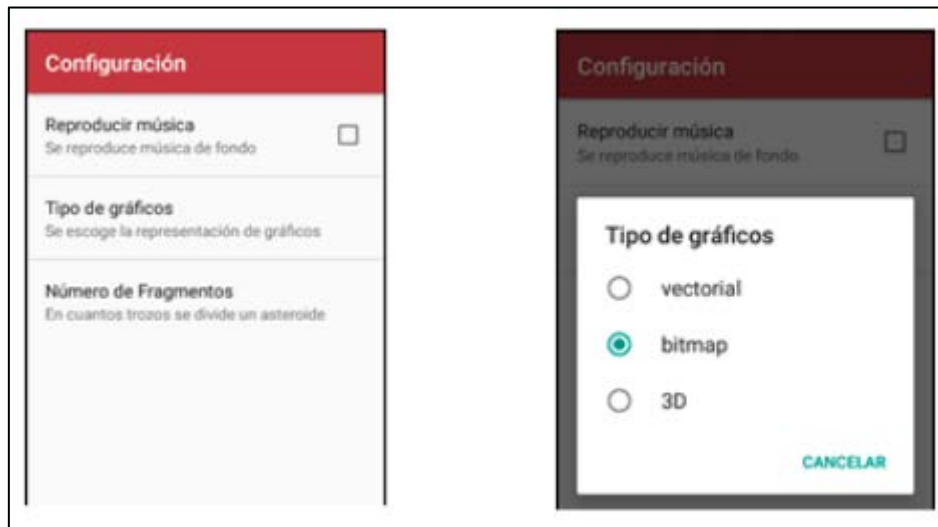


Paso 4. Edita el fichero e introduce el siguiente código:

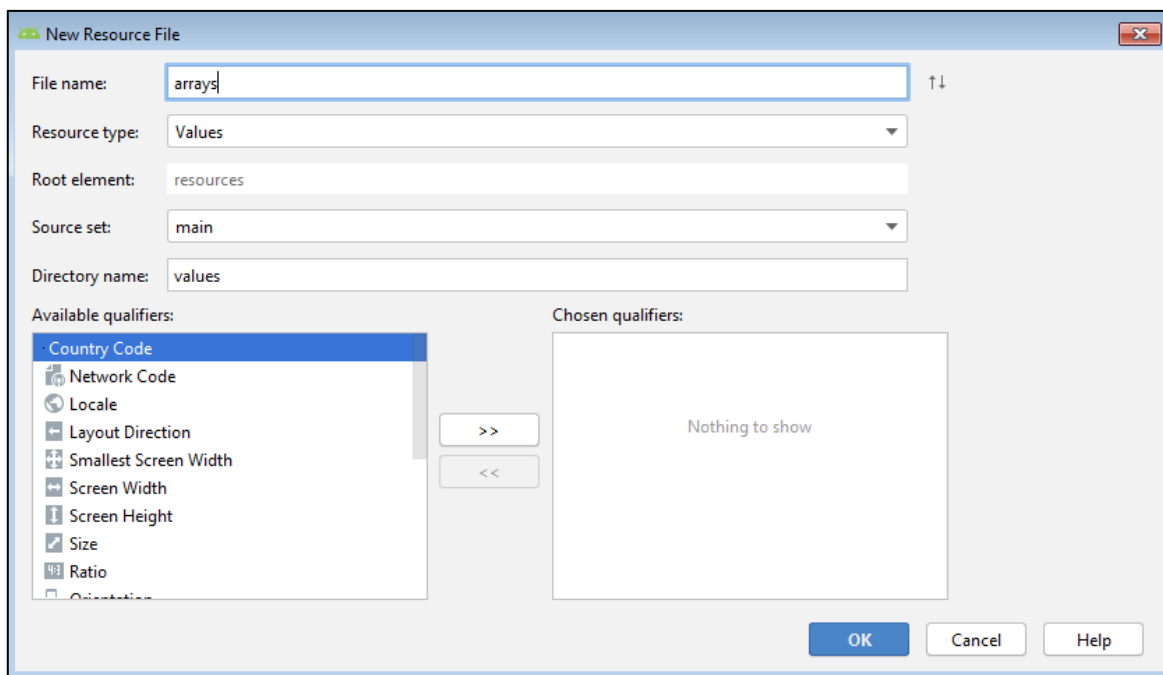


El significado de cada etiqueta y atributo se descubre fácilmente si observas el resultado obtenido que se muestra a continuación. El atributo *inputType* permite configurar el tipo de teclado que se mostrará

para introducir el valor. Coinciden con el atributo de EditText. Para ver los posibles valores consultar developer.android.com/reference/android/widget/TextView.html#attr_android:inputType



Paso 5. Para almacenar los valores del desplegable, has de crear el fichero `/res/values/arrays.xml` con el siguiente contenido. Para ello pulsa con el botón derecho sobre la carpeta `res` y selecciona la opción `New > Android resource file`. Completa los campos `File name: arrays` y `Resource type: values`.



```
arrays.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <string-array name="tiposGraficos">
4          <item>vectorial</item>
5          <item>bitmap</item>
6          <item>3D</item>
7      </string-array>
8      <string-array name="tiposGraficosValores">
9          <item>0</item>
10         <item>1</item>
11         <item>2</item>
12     </string-array>
13 </resources>
```

Paso 6. Crea ahora una nueva clase `Preferencias.java` con el siguiente código:


```
Preferencias.java x
package com.example.asteroides;

import android.app.Activity;
import android.os.Bundle;
import android.preference.PreferenceFragment;

public class Preferencias extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getFragmentManager().beginTransaction().
            replace(android.R.id.content, new MyPreferenceFragment()).commit();
    }

    public static class MyPreferenceFragment extends PreferenceFragment {
        @Override
        public void onCreate(final Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            addPreferencesFromResource(R.xml.preferencias);
        }
    }
}
```

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

NOTA: Desde el nivel de API 11, el método `addPreferencesFromResource()` se ha marcado como obsoleto. En lugar de usar la clase `PreferenceActivity`, se recomienda utilizar `PreferenceFragment`. Sin embargo, si estamos trabajando con un nivel mínimo de API inferior al 11 (en nuestro caso el 9), no disponemos de esta clase. Cuando un método o clase está marcado como obsoleto, se nos indica que hay otra forma más moderna de hacerlo, incluso que en un futuro es posible que deje de estar incluido en el API. Pero esto es algo que nunca ha pasado. En Android es frecuente seguir utilizando métodos marcados como obsoletos. En el siguiente apartado se explica el uso de `PreferenceFragment`.

NOTA: Recientemente, Google ha creado las librerías compatibilidad `preference-v7` y `preference-v14` donde se define `PreferenceFragmentCompact` una versión de `PreferenceFragment` compatible con versiones anteriores. El uso de `PreferenceFragmentCompact` se muestra en el Gran Libro de Android Avanzado.

Paso 7. No hay que olvidar registrar toda nueva actividad en `AndroidManifest.xml`.



Paso 8. Añade a MainActivity.java el método lanzarPreferencias(). Este método ha de tener el mismo código que lanzarAcercaDe() pero lanzando la actividad Preferencias. En el Layout activity_main.xml añade al botón con texto “Configurar” en el atributo onClick el valor lanzarPreferencias.

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

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



Paso 9. Para activar la configuración desde la opción de menú añade el siguiente código en el fichero MainActivity.java en el método onOptionsItemSelected() dentro del switch:

```

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

```

Paso 10. Arranca la aplicación y verifica que puedes lanzar las preferencias mediante las dos alternativas.

