

SEMINARIO DE LENGUAJES

OPCIÓN ANDROID

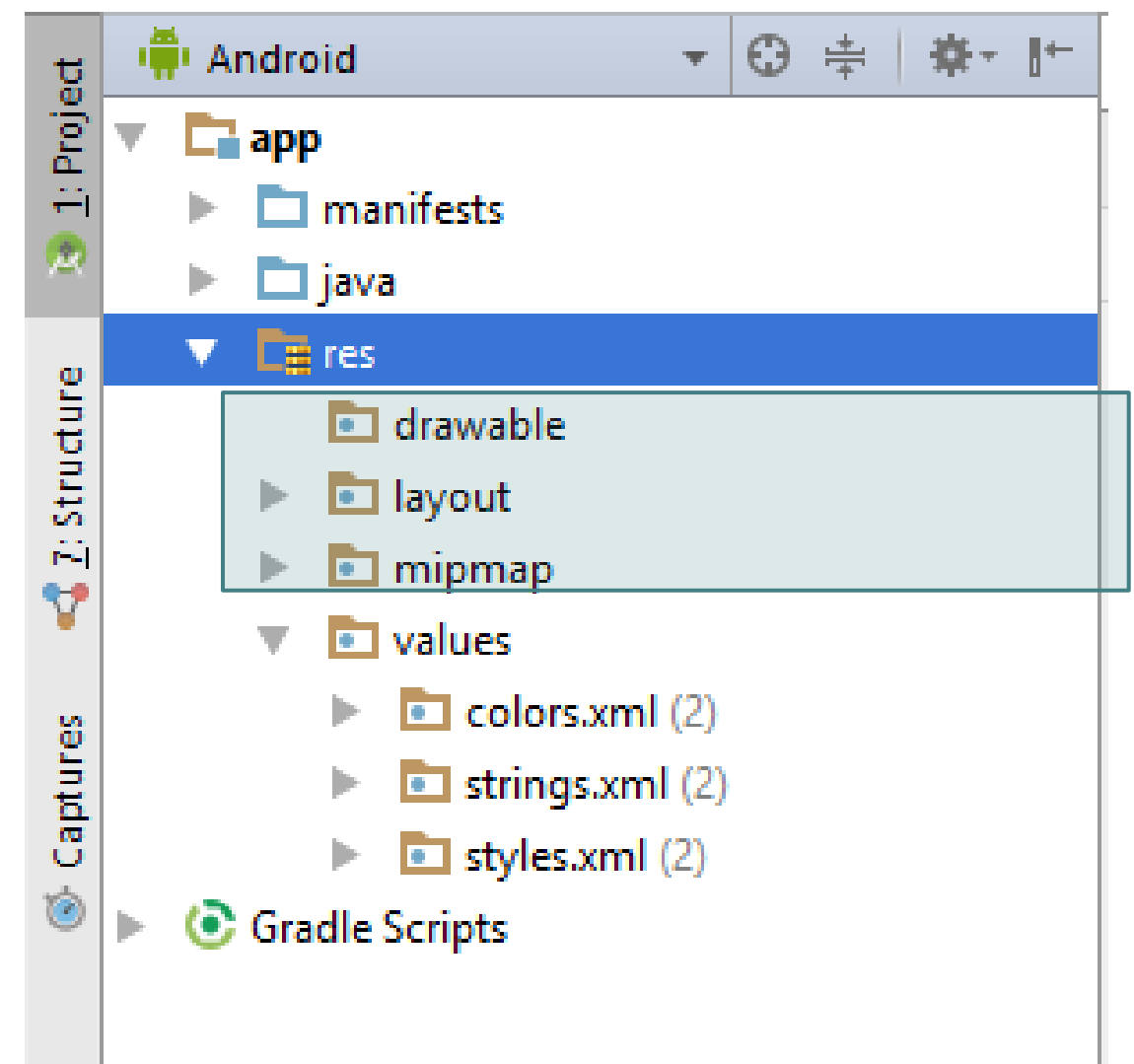


Recursos de archivo. Menús.

Mg. Corbalán Leonardo, Esp. Delía Lisandro

Recursos de archivos

- En **Android Studio**, las carpetas contenidas en **res/** (a excepción de **res/value/**) se utilizan para definir recursos de archivos.
- Para estos recursos se crea un **identificador** automático que coincide con el **nombre del archivo sin la extensión**.
- Según la carpeta donde se cree, se conocerá su tipo de recurso.



Recursos de archivos

- Algunos de los tipos de recursos de archivos más utilizados son:
 - **drawable/** Archivos que definen recursos de imágenes: bitmaps (.png, .jpg o gif), XML con descriptores de gráficos (Ej. shape)
 - **layout/** Archivos XML que definen el diseño de una interfaz de usuario
 - **mipmap/** Archivos de elemento de diseño para diferentes densidades de los íconos lanzadores.
 - **menu/** Archivos XML que definen menús de aplicaciones, como un menú de opciones, un menú contextual o un submenú.
 - **raw/** Archivos arbitrarios para guardar sin procesar (Ej. audio o video)

Actividad guiada

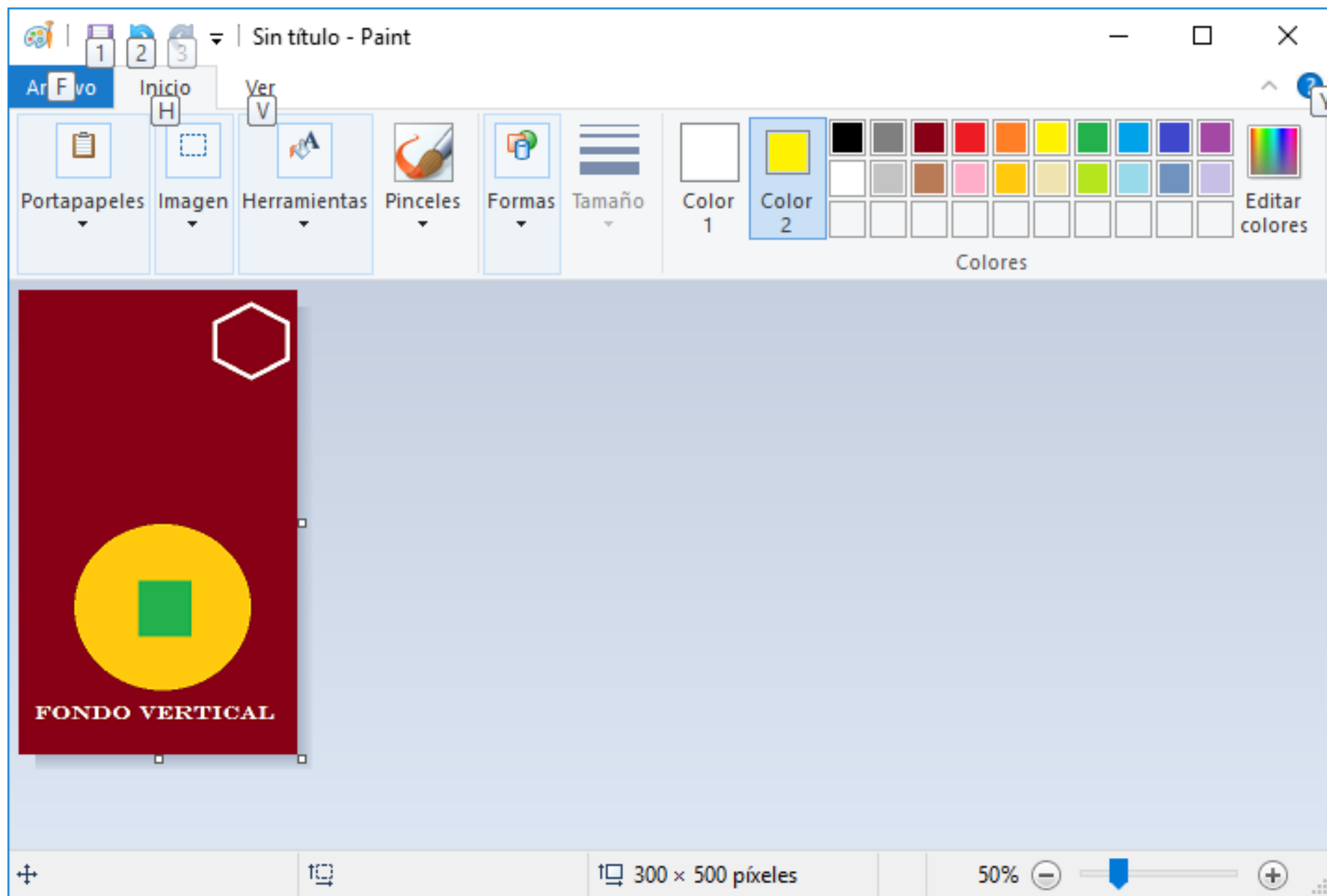
- Crear un nuevo proyecto **Android Studio** llamado **"RecursosDeArchivo"** basado en la siguiente **Empty Activity**

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>

</LinearLayout>
```

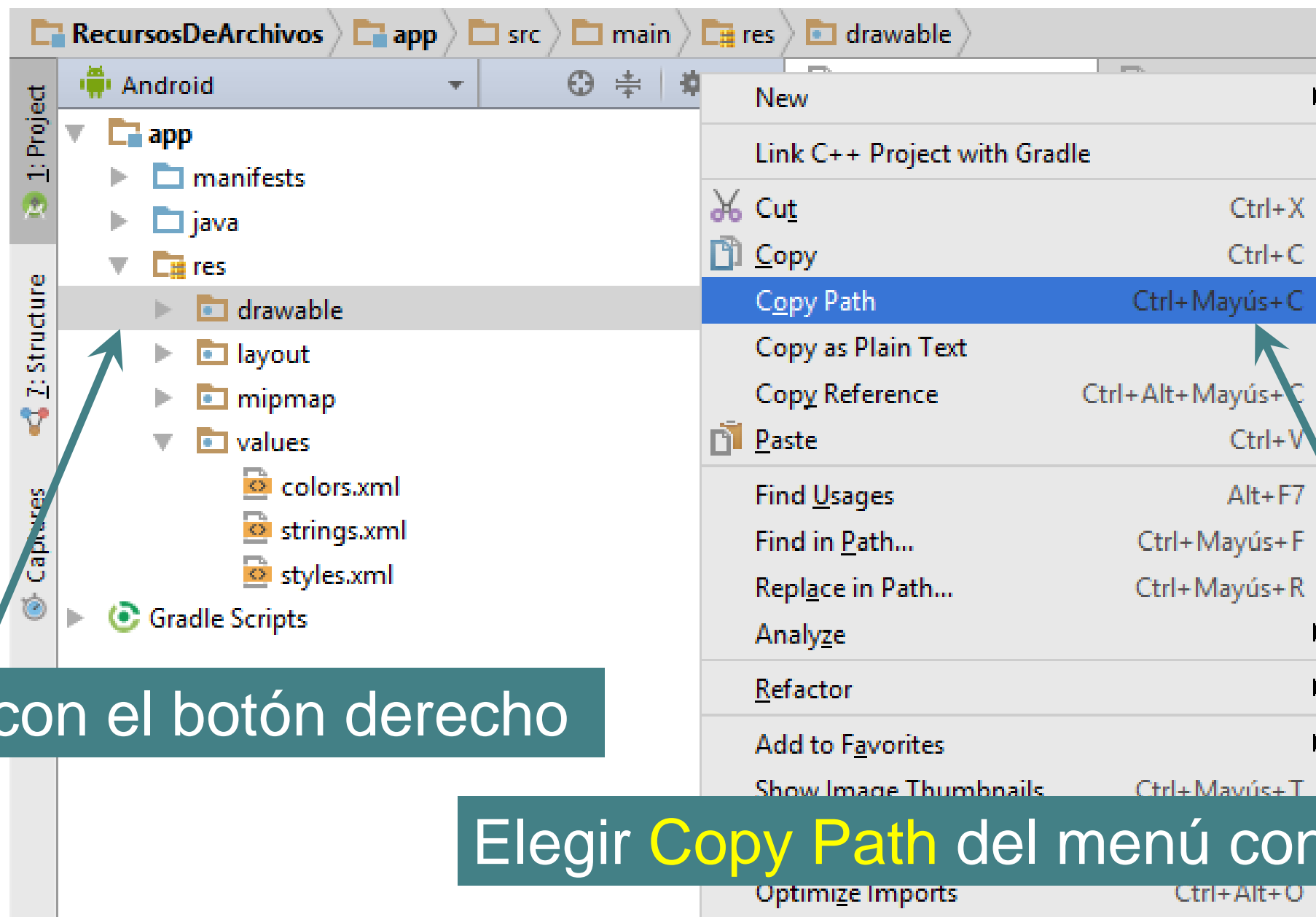
Actividad guiada

Crear en el **Paint** una imagen con relación 3:5 (por ejemplo 300 x 500)



Actividad guiada

Para guardar la imagen va a ser necesario ubicar el **path** completo de la carpeta **res/drawable** del proyecto



Actividad guiada

Guardar imagen creada en Paint con el nombre **fondo.png** en la carpeta **res/drawable** del proyecto



Actividad guiada

Establecer la imagen como fondo del **layout** de la **activity principal**

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

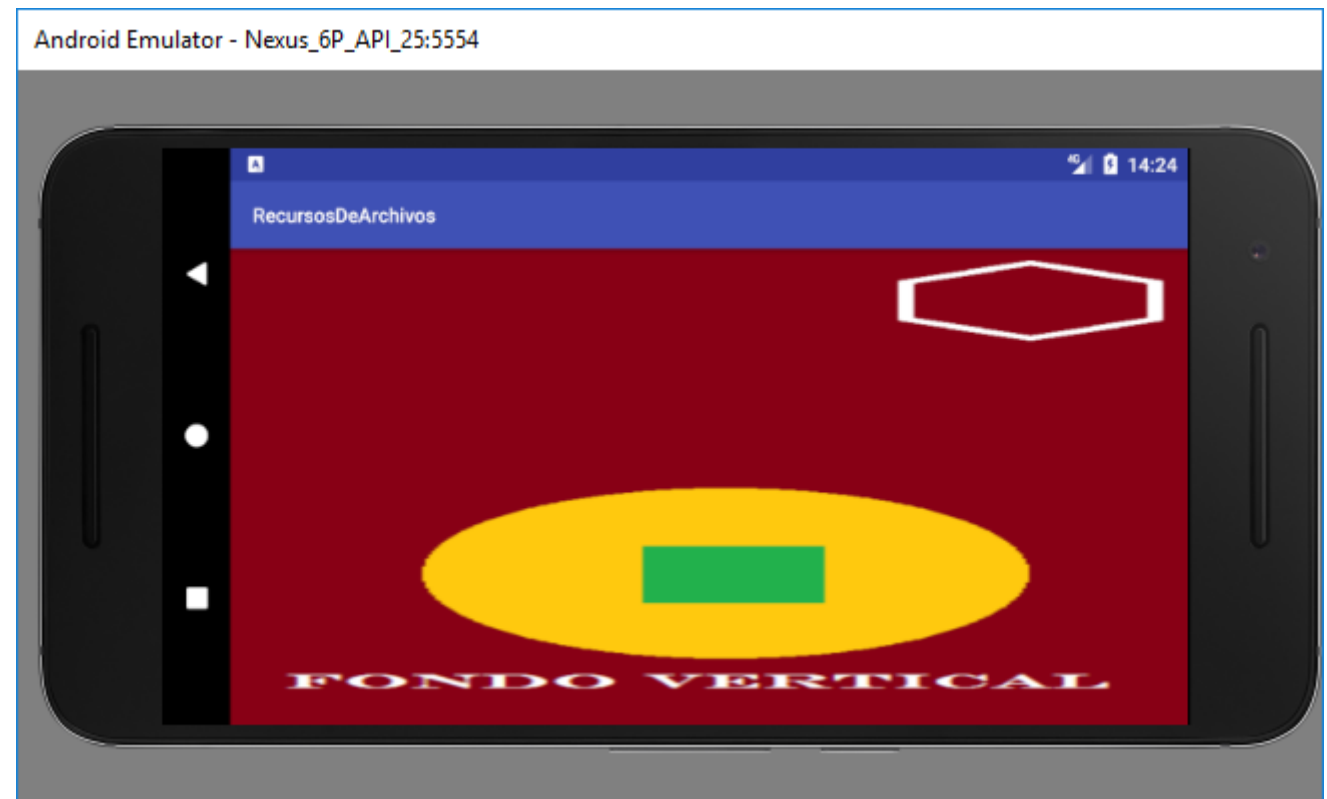
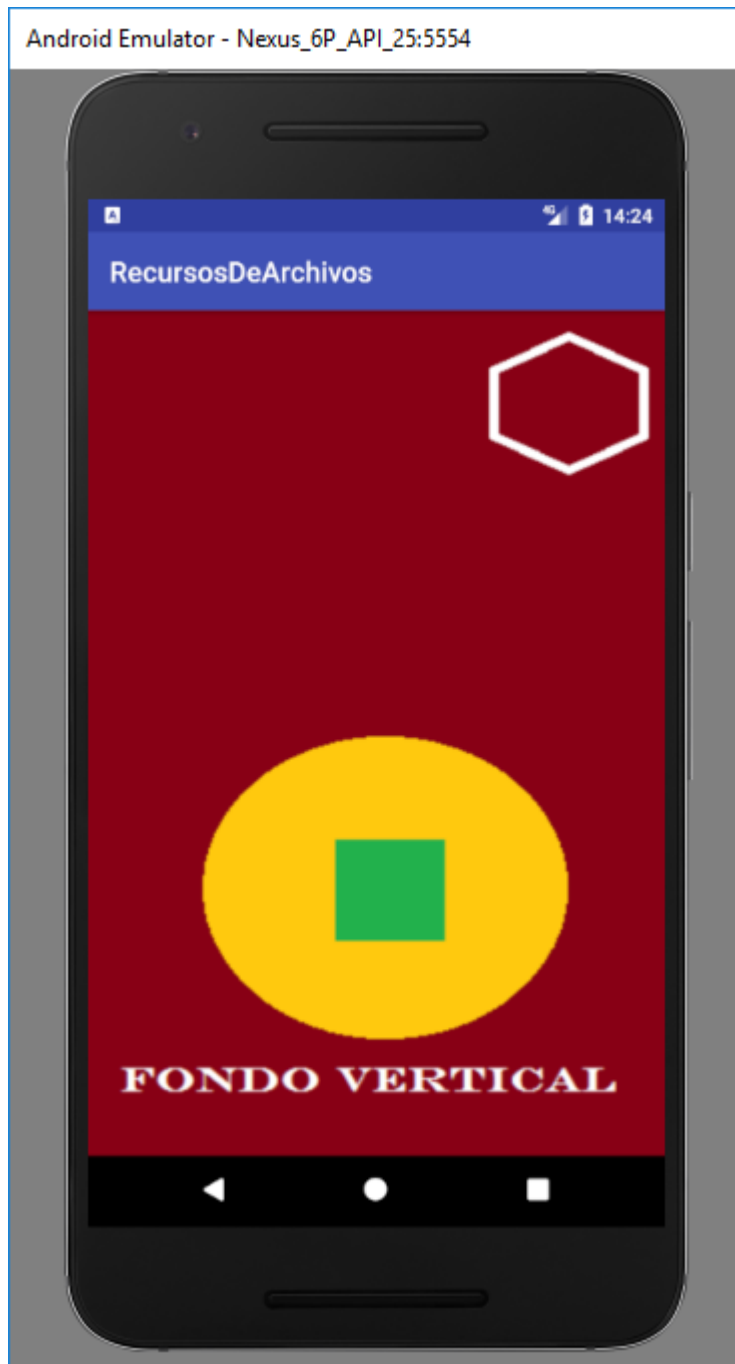
```
    android:background="@drawable/fondo"
```

```
>
```

```
</LinearLayout>
```

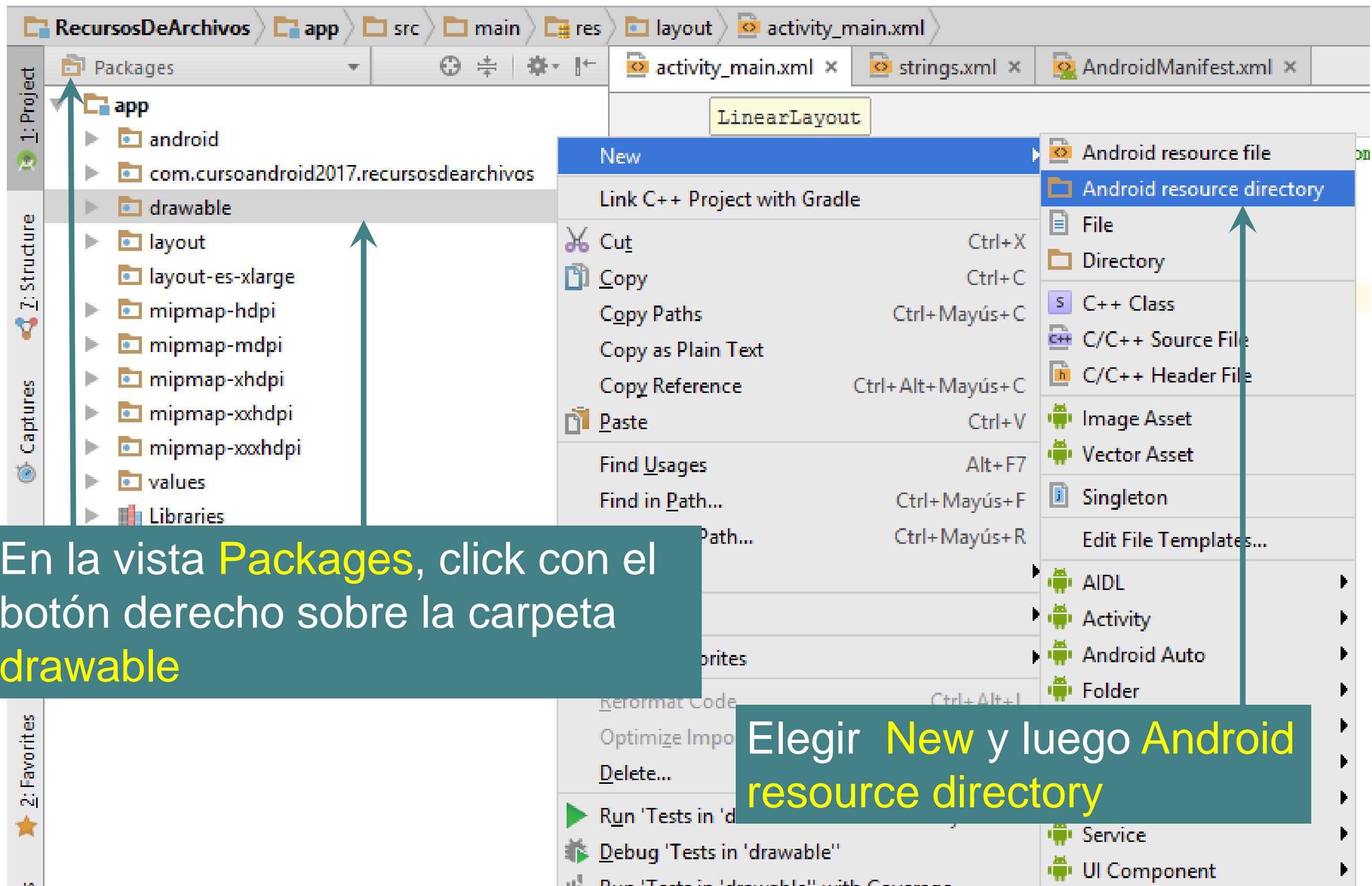

Actividad guiada

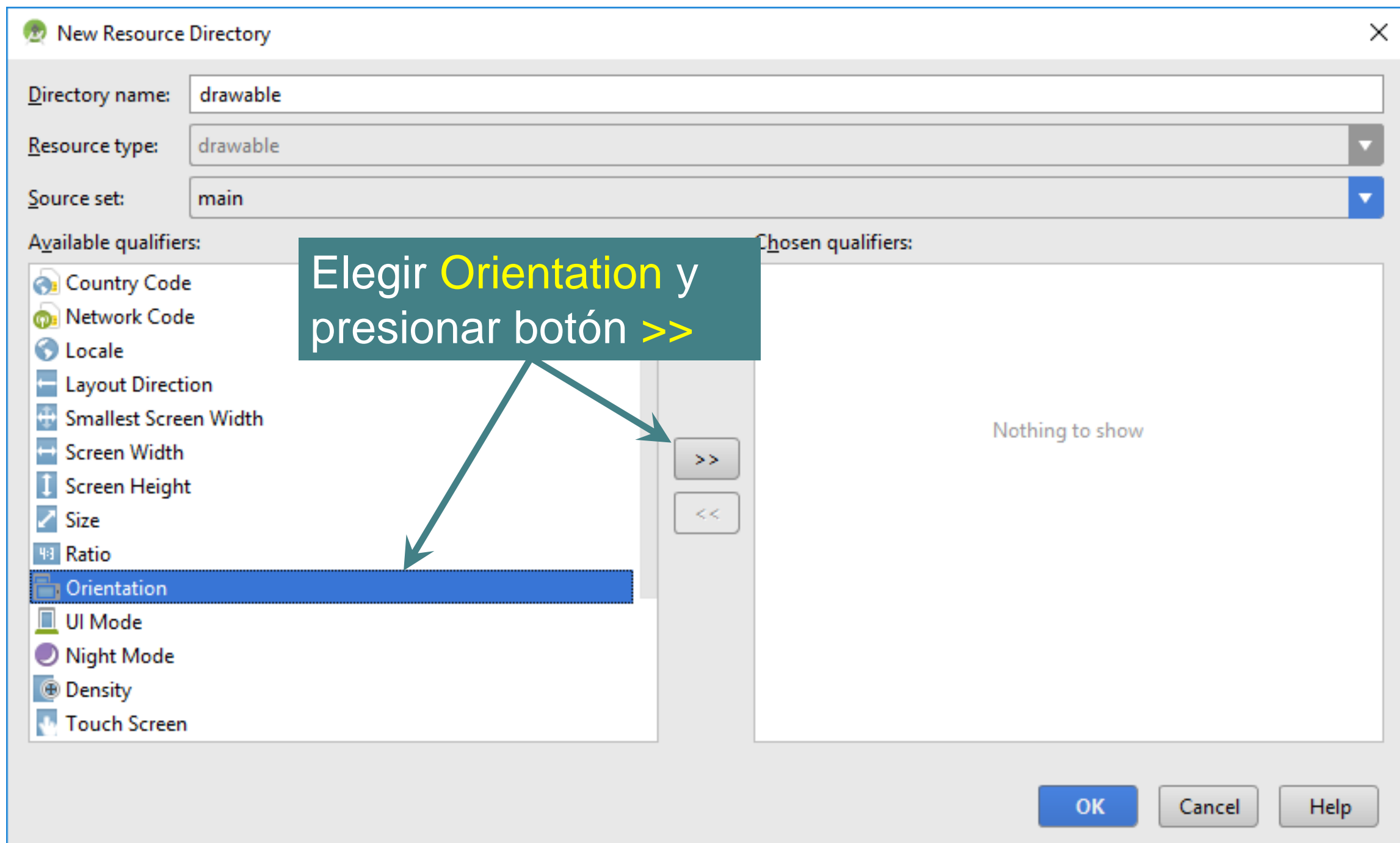
Observar que la imagen de fondo no se ve bien cuando el dispositivo se encuentra en posición horizontal

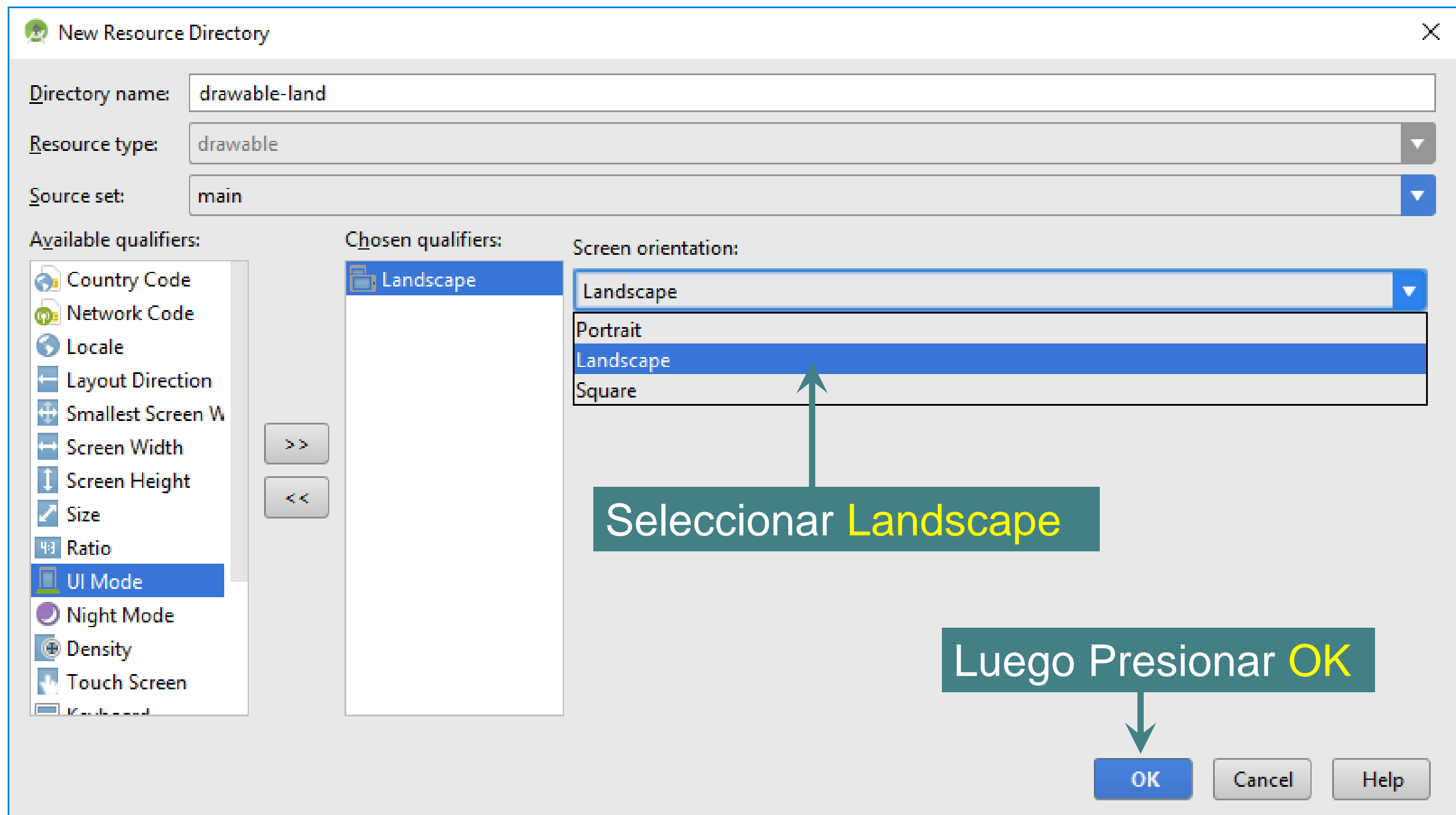


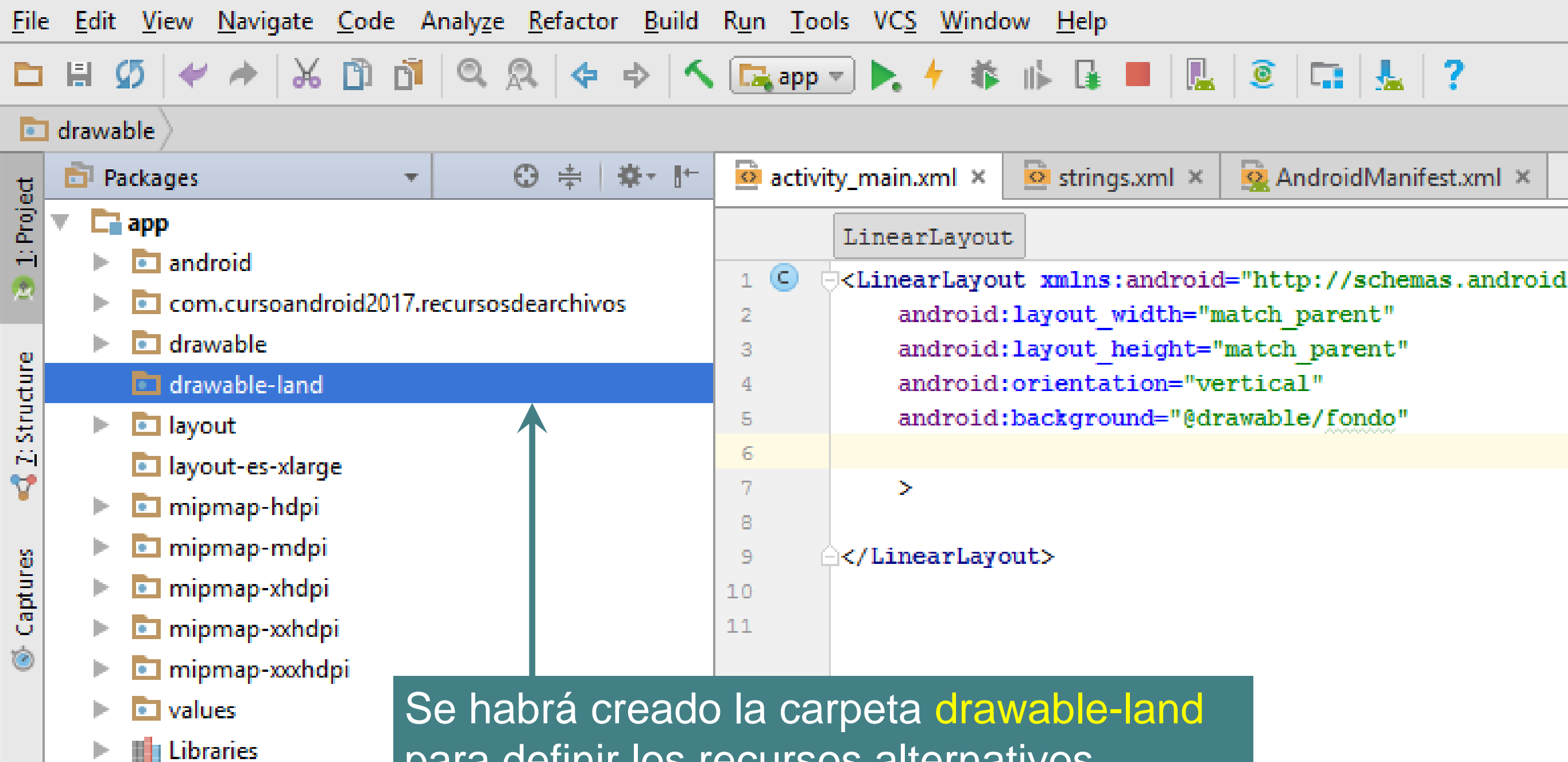
Actividad guiada

- Vamos a crear un recurso alternativo **drawable** para establecer otra imagen de fondo en la disposición horizontal (**landscape**)
- Para ello es necesario crear el directorio de recursos **res/drawable-land** para guardar en él una imagen que llamaremos también **fondo.png**
- El sufijo **-land** es un calificador que hace referencia a los recursos alternativos para la disposición **landscape**, al igual que el sufijo **-en** lo hace para el **idioma inglés** (visto en la clase anterior)





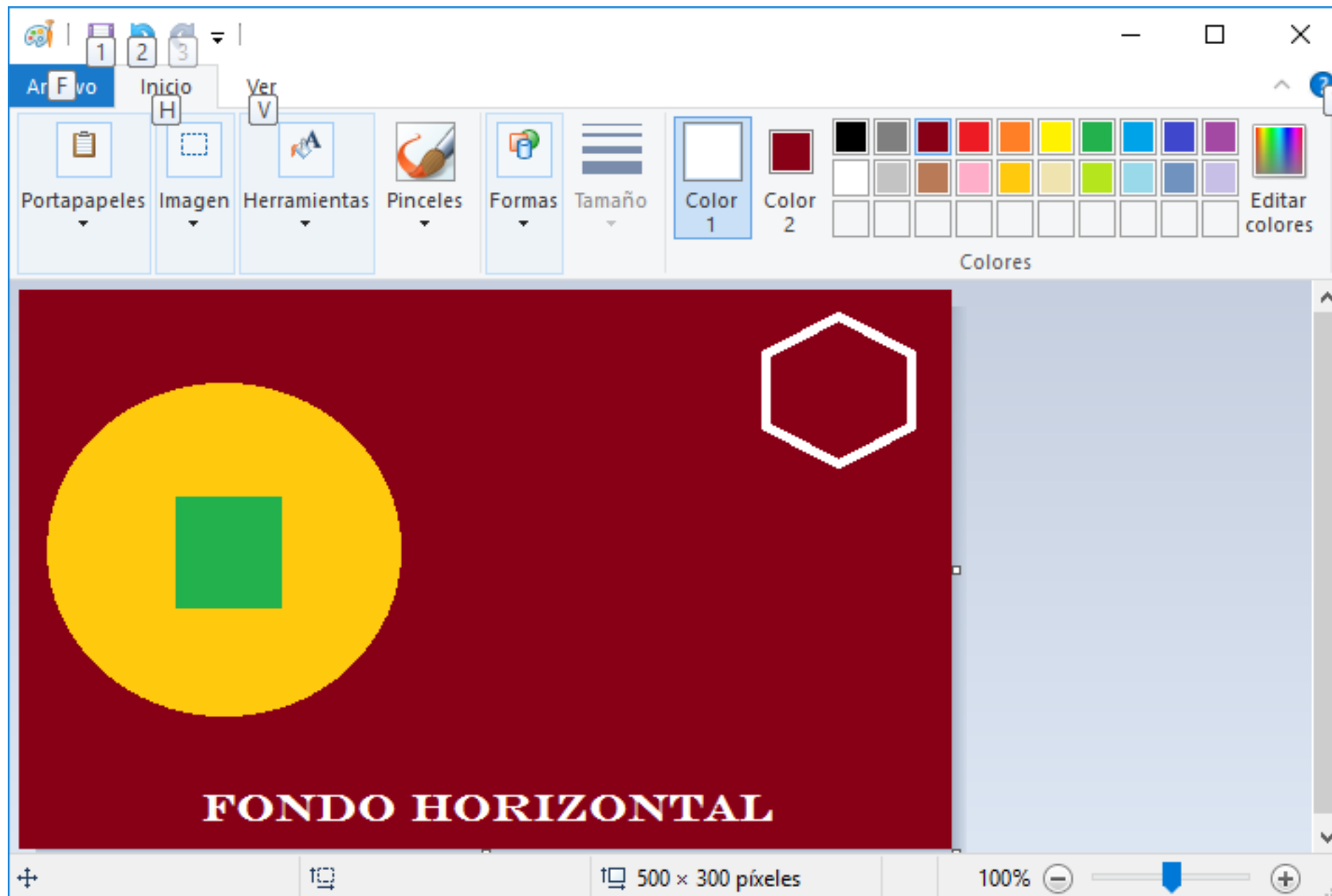




Se habrá creado la carpeta **drawable-land** para definir los recursos alternativos correspondientes a la **orientación horizontal** del dispositivo

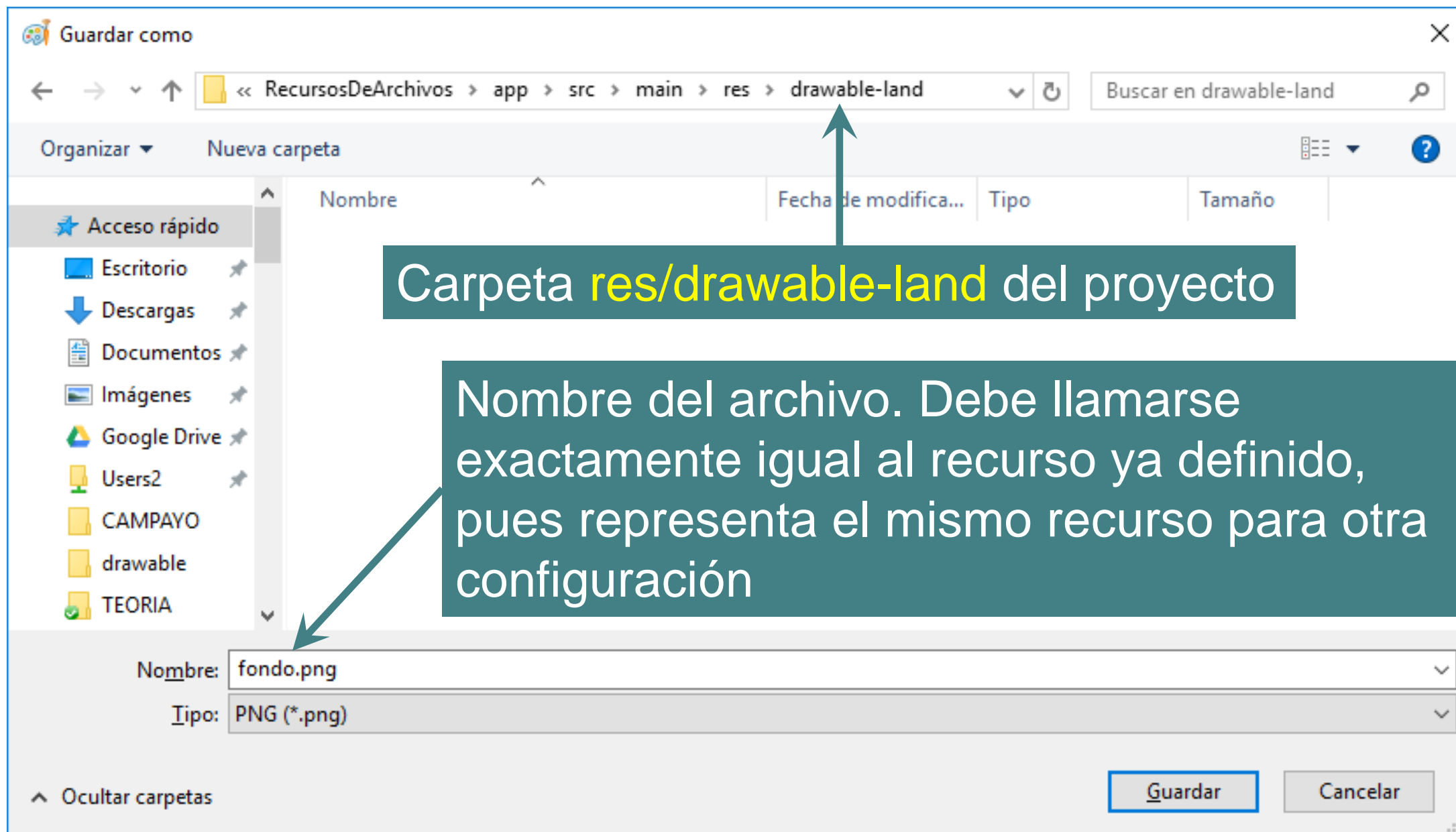
Actividad guiada

Crear en el **Paint** una imagen con relación 5:3 (por ejemplo 500 x 300)



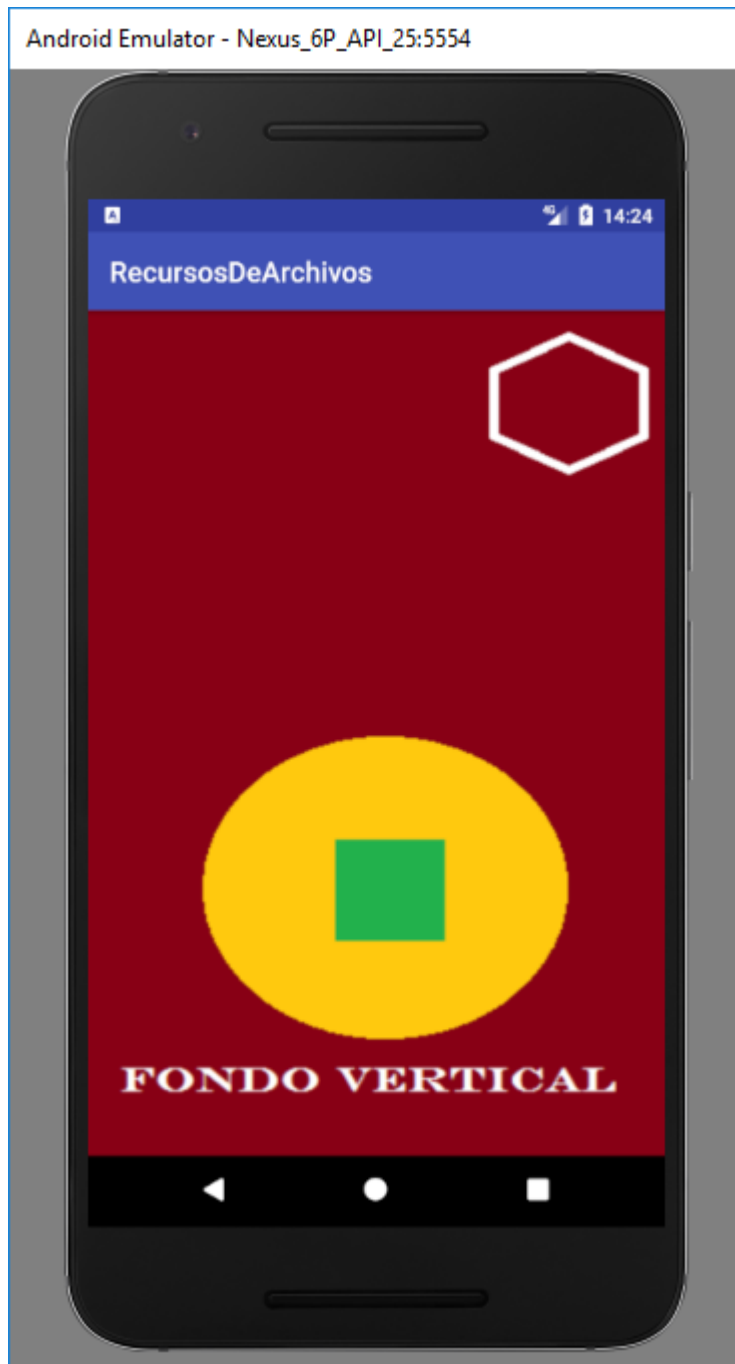
Actividad guiada

Guardar imagen creada en Paint con el nombre **fondo.png** en la carpeta **res/drawable-land** del proyecto



Actividad guiada

Verificar ahora que al colocar el dispositivo en disposición horizontal cambia la imagen de fondo.



Recursos mipmap

- En la carpeta **mipmap/** se colocan los archivos de imágenes que constituyen los recursos predeterminados de íconos lanzadores de la aplicación
- En la carpeta **mipmap-{calificadores}/** se colocan los archivos de imágenes para los recursos alternativos de íconos de la aplicación
- Al crear un proyecto, **Android Studio** establece recursos **mipmap** para cinco densidades distintas: **mdpi** (~160dpi), **hdpi**(~240dpi), **xhdpi** (~320dpi), **xxhdpi** (~480dpi) y **xxxhdpi**(~640dpi)

RecursosDeArchivos > app > src > main > AndroidManifest.xml

Packages

1: Project

2: Structure

Captures

Build Variants

app

- android
- com.cursoandroid2017.recursosdearchivos
- drawable
- drawable-land
- layout
- mipmap-hdpi
 - ic_launcher.png
 - ic_launcher_round.png
- mipmap-mdpi
 - ic_launcher.png
 - ic_launcher_round.png
- mipmap-xhdpi
 - ic_launcher.png
 - ic_launcher_round.png
- mipmap-xxhdpi
 - ic_launcher.png
 - ic_launcher_round.png
- mipmap-xxxhdpi
 - ic_launcher.png
 - ic_launcher_round.png
- values
- Libraries

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.cursoandroid2017.recursosdearchivos">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="RecursosDeArchivos"
9         android:roundIcon="@mipmap/ic_launcher"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12         <activity android:name=".MainActivity">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15                 <category android:name="android.intent.category.LAUNCHER" />
16             </intent-filter>
17         </activity>
18     </application>
19
20 </manifest>
```

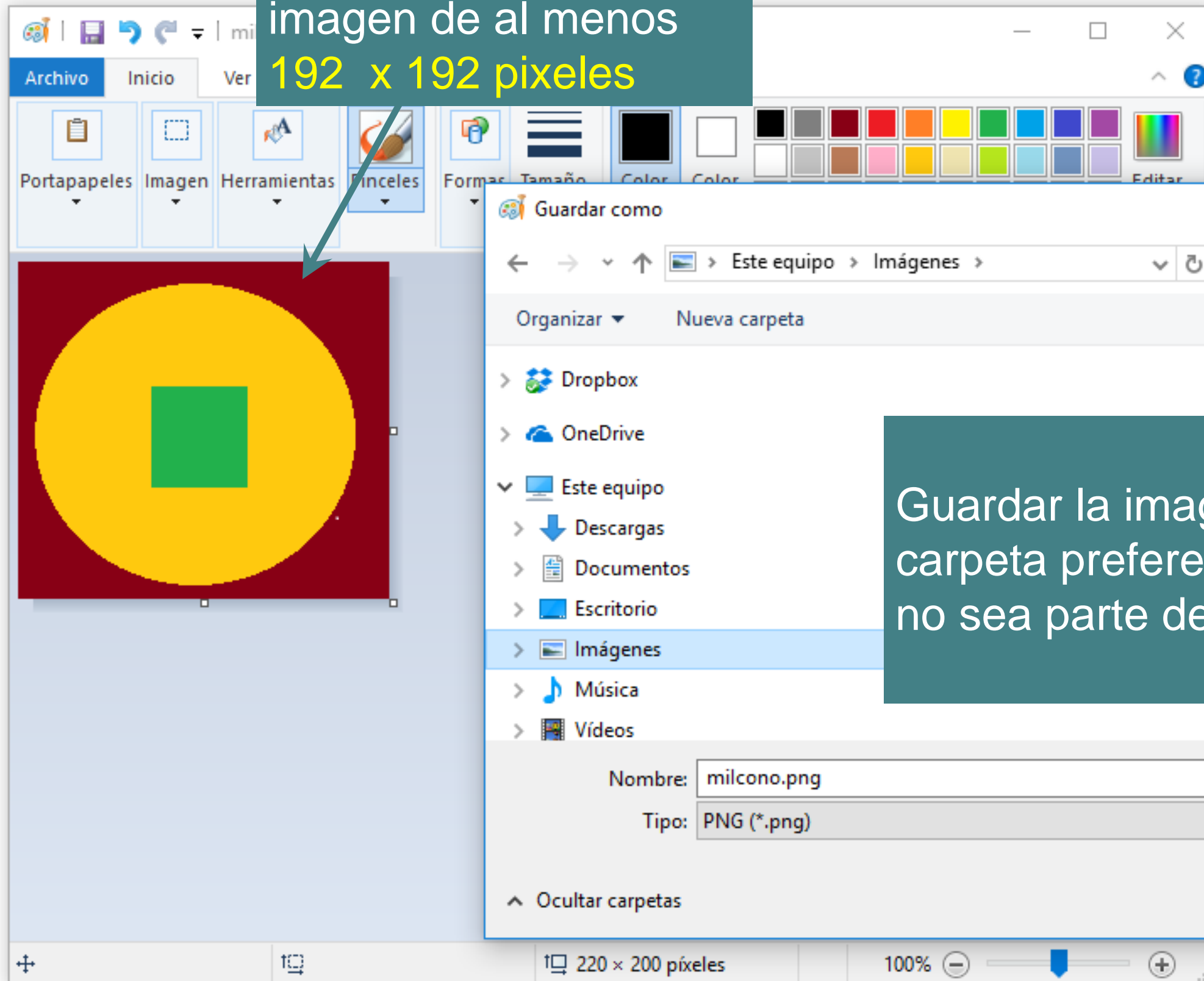
Los íconos lanzadores se establecen en el manifiesto de la aplicación

Recursos alternativos para cinco densidades distintas

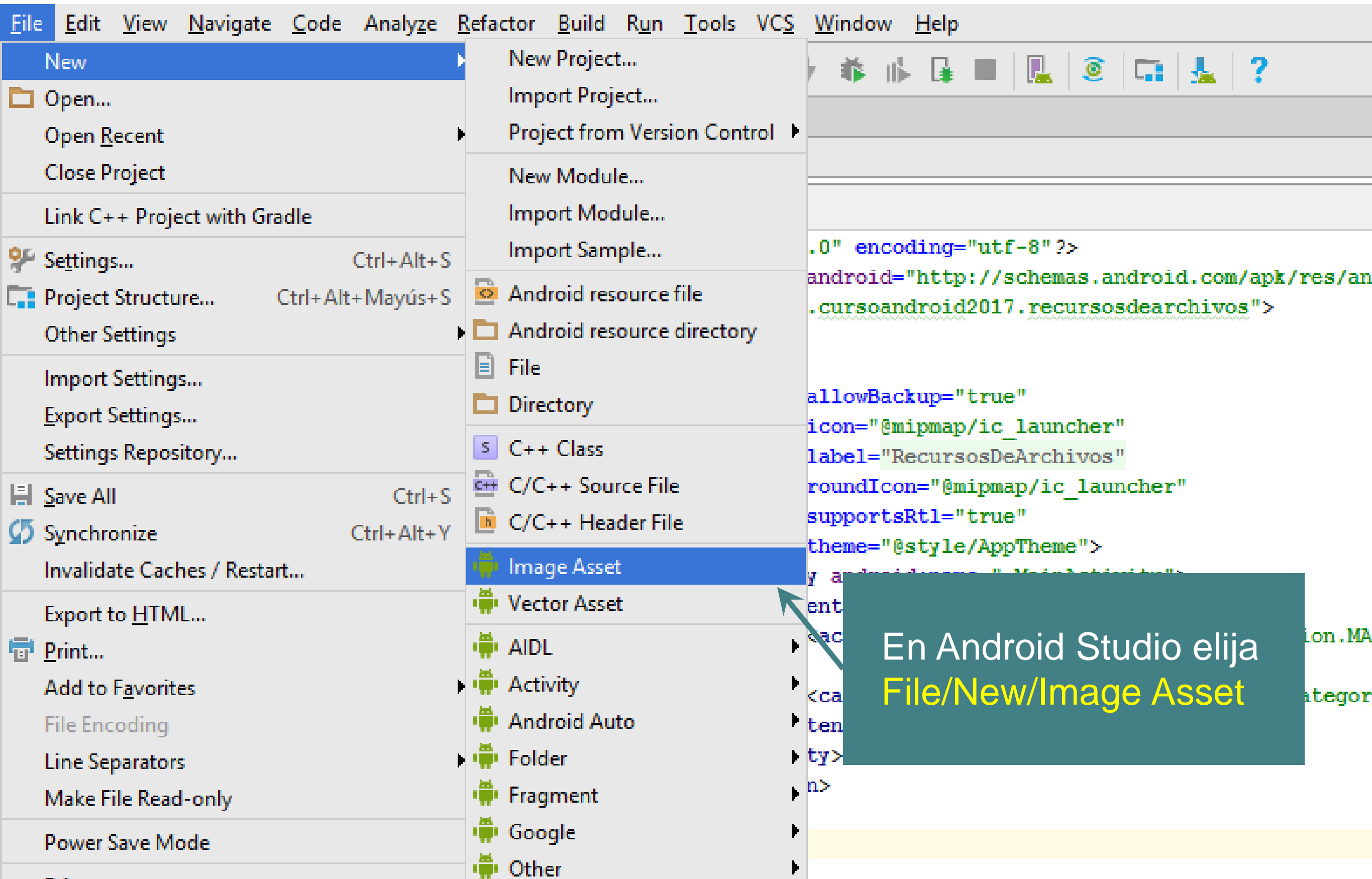
Recursos mipmap

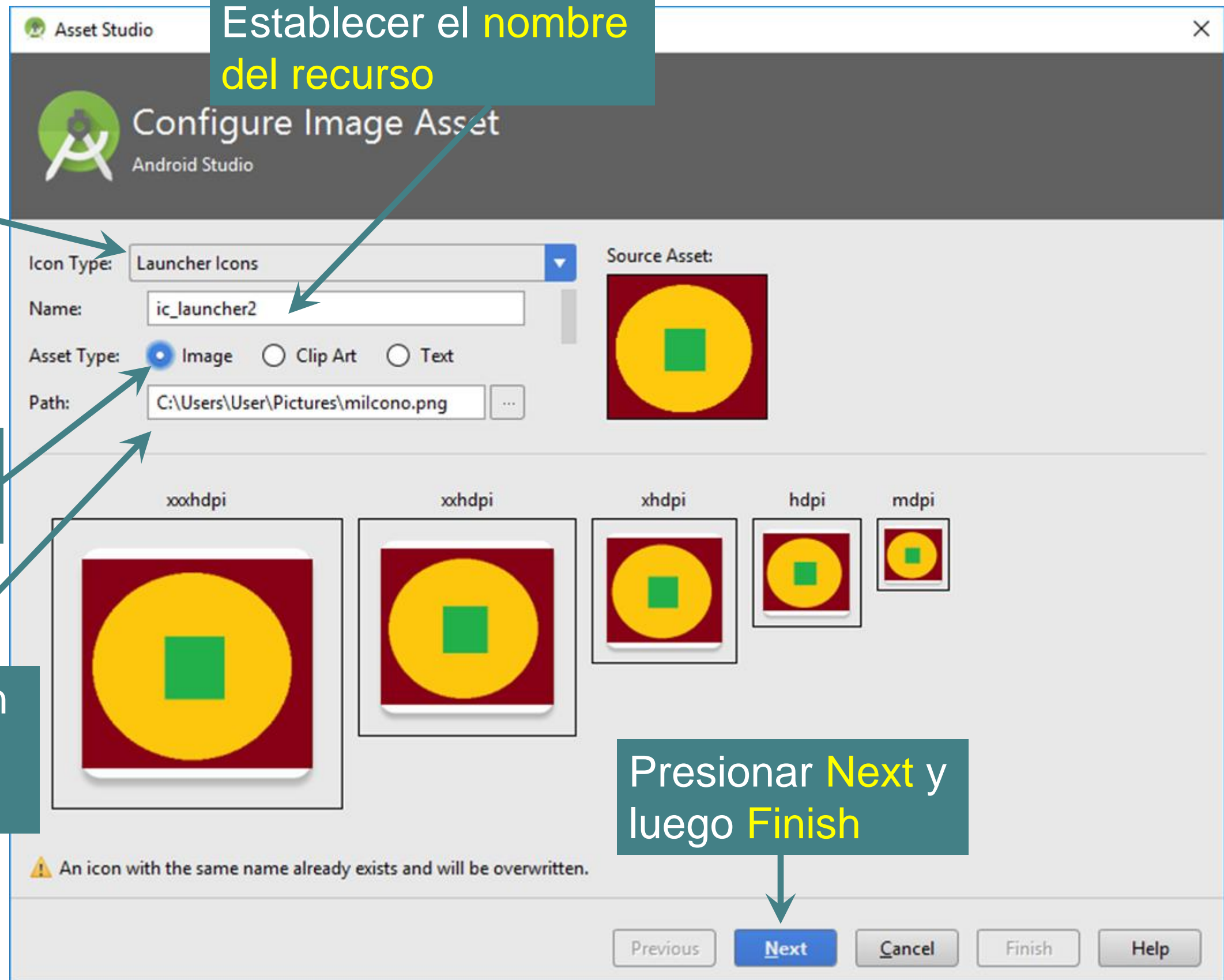
- Vamos a crear un nuevo **ícono lanzador** para nuestra aplicación
- Luego utilizaremos una herramienta provista por **Android Studio** para generar las distintas versiones alternativas para cada densidad

Crear en el **Paint** una imagen de al menos **192 x 192 píxeles**



Guardar la imagen en alguna carpeta preferentemente que no sea parte del proyecto





The screenshot displays the Android Studio interface. On the left, the 'Project' tab shows the file structure of an Android application. The 'app' directory is expanded, revealing several resource folders: 'android', 'com.cursoandroid2017.recursosdearchivos', 'drawable', 'drawable-land', 'layout', and 'mipmap-hdpi'. The 'mipmap-hdpi' folder is selected, showing three launcher icons: 'ic_launcher.png', 'ic_launcher2.png' (highlighted), and 'ic_launcher_round.png'. Below this, other mipmap folders (mdpi, xhdpi, xxhdpi, xxxhdpi) are also visible, each containing the same three launcher icons. On the right, the 'AndroidManifest.xml' file is open in the 'manifest' editor. The XML code defines the application's manifest, including the package name 'com.cursoandroid2017.recursosdearchivos', the application name 'RecursosDeArchivos', and the main activity 'MainActivity'. The application is configured with 'android:allowBackup="true"', 'android:icon="@mipmap/ic_launcher"', 'android:roundIcon="@mipmap/ic_launcher"', and 'android:supportsRtl="true"'. The main activity is defined with the name '.MainActivity' and an intent filter for the action 'android.intent.action.MAIN' and category 'android.intent.category.LAUNCHER'. A lightbulb icon is visible next to the closing tag of the application element, indicating a suggestion or warning.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/a
3     package="com.cursoandroid2017.recursosdearchivos">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="RecursosDeArchivos"
9         android:roundIcon="@mipmap/ic_launcher"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.M
15
16                <category android:name="android.intent.catego
17            </intent-filter>
18        </activity>
19    </application>
20
```

Se crean los recursos alternativos para cada densidad

En **AndroidManifest.xml** establecer los íconos de lanzamiento con el nuevo recurso definido

The screenshot displays the Android Studio environment. On the left, the 'Project' pane shows the package structure: 'app' contains 'android', 'com.cursoandroid2017.recursosdearchivos', and 'drawable'. Below this, the 'Build Variants' pane shows the 'mipmap-xhdpi' and 'mipmap-xxhdpi' folders, each containing 'ic_launcher.png', 'ic_launcher2.png', and 'ic_launcher_round.png'. The main editor shows the 'AndroidManifest.xml' file with the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.cursoandroid2017.recursosdearchivos">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher2"
        android:label="RecursosDeArchivos"
        android:roundIcon="@mipmap/ic_launcher2"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN">
                <category android:name="android.intent.category.LAUNCHER">
            </intent-filter>
        </activity>
    </application>
```

Two arrows point from the text box on the left to the 'android:icon' and 'android:roundIcon' attributes in the XML code. A red box at the bottom right contains the text: 'Ejecutar en el emulador y verificar que ha cambiado el ícono de lanzamiento de la aplicación'.

Actividad guiada - continuación

Con la misma estrategia utilizada al establecer el fondo de la aplicación, implemente dos **layout** distintos para la **activity** principal



```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/fondo"
    android:gravity="center"
>
```

```
<Button
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Nuevo Juego"
/>
```

```
<Button
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Continuar jugando"
/>
```

```
<Button
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Configuración"
/>
```

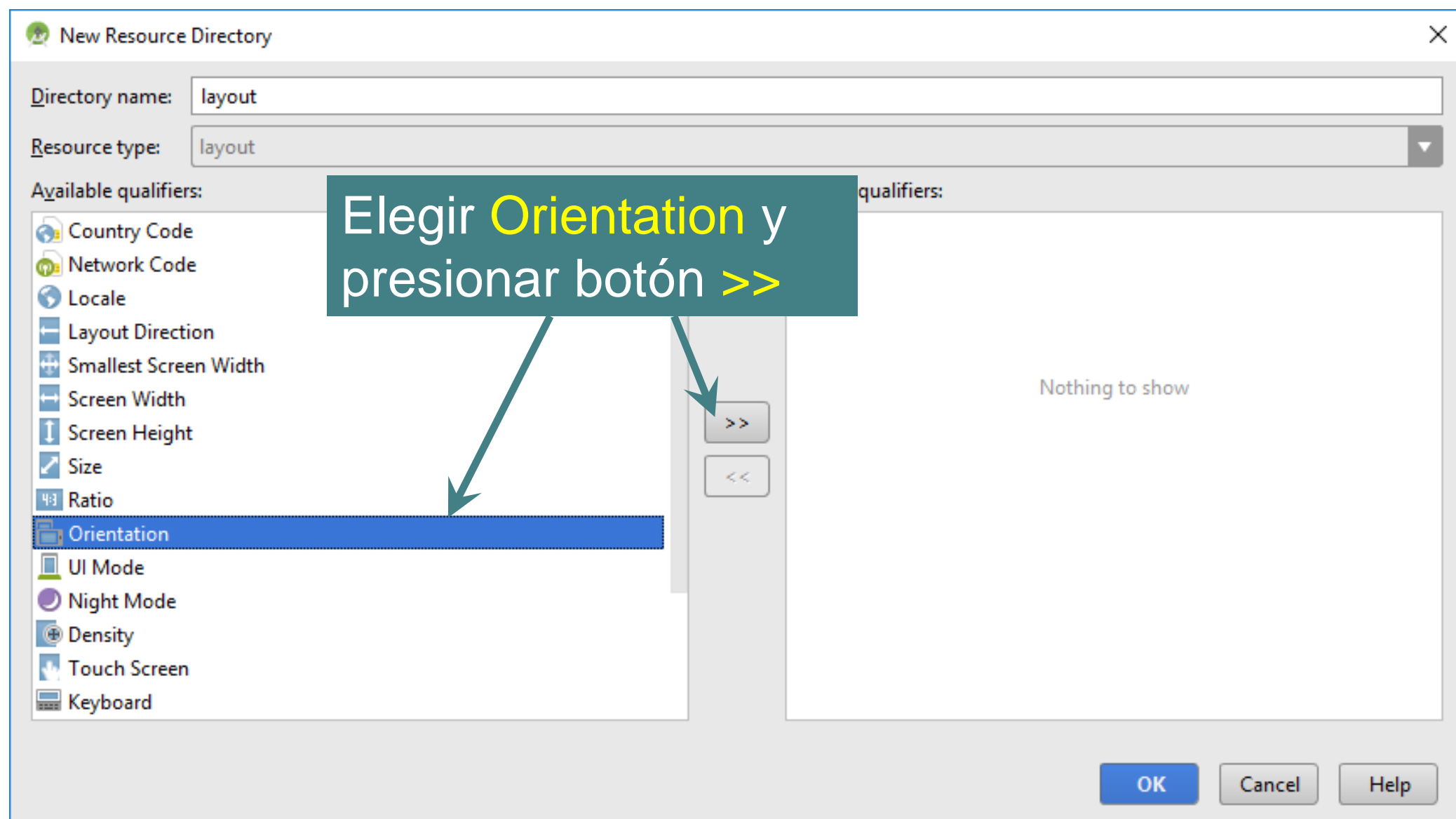
```
</LinearLayout>
```

activity_main.xml

Layout predeterminado
de la activity principal

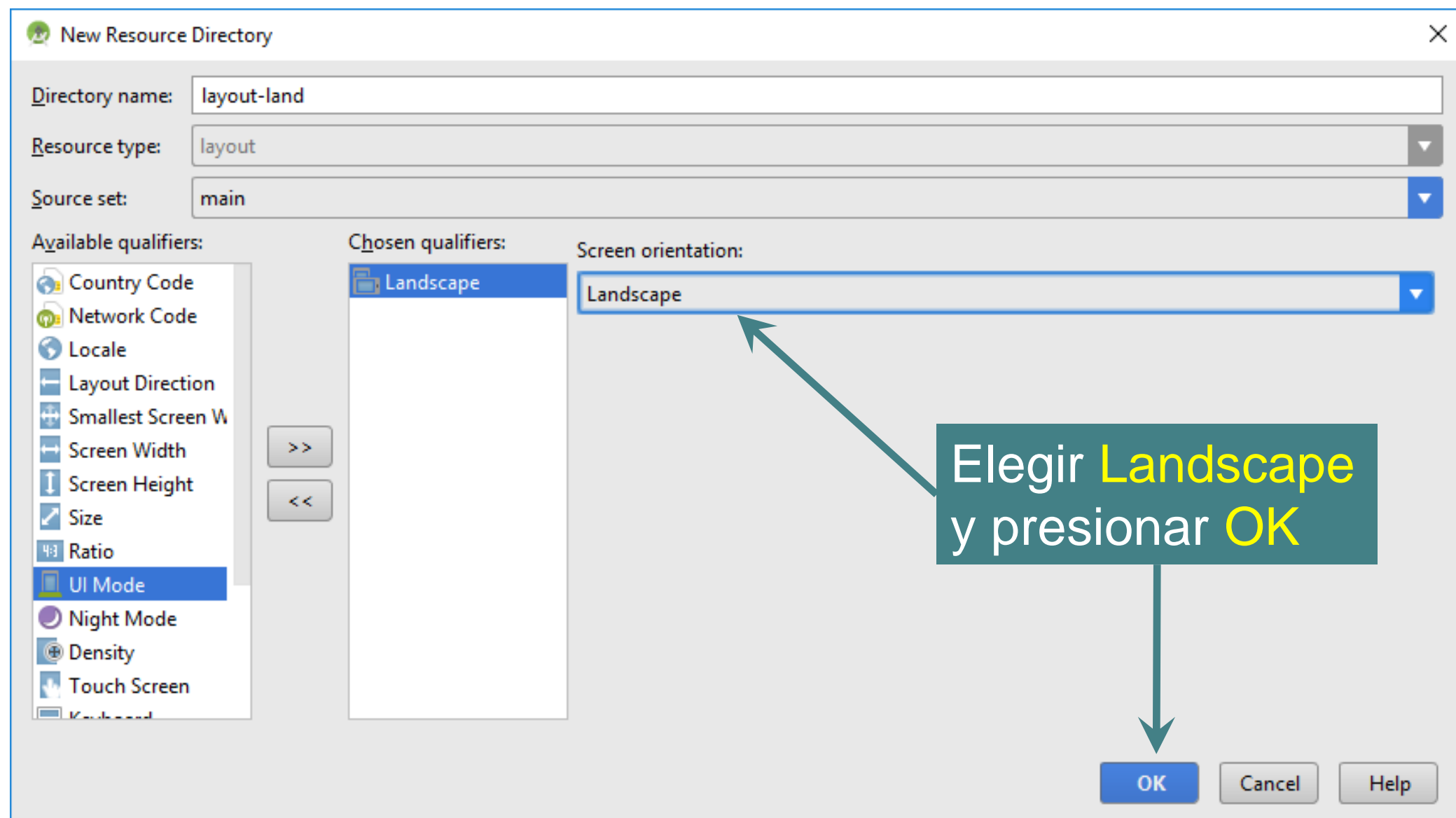
Actividad guiada - continuación

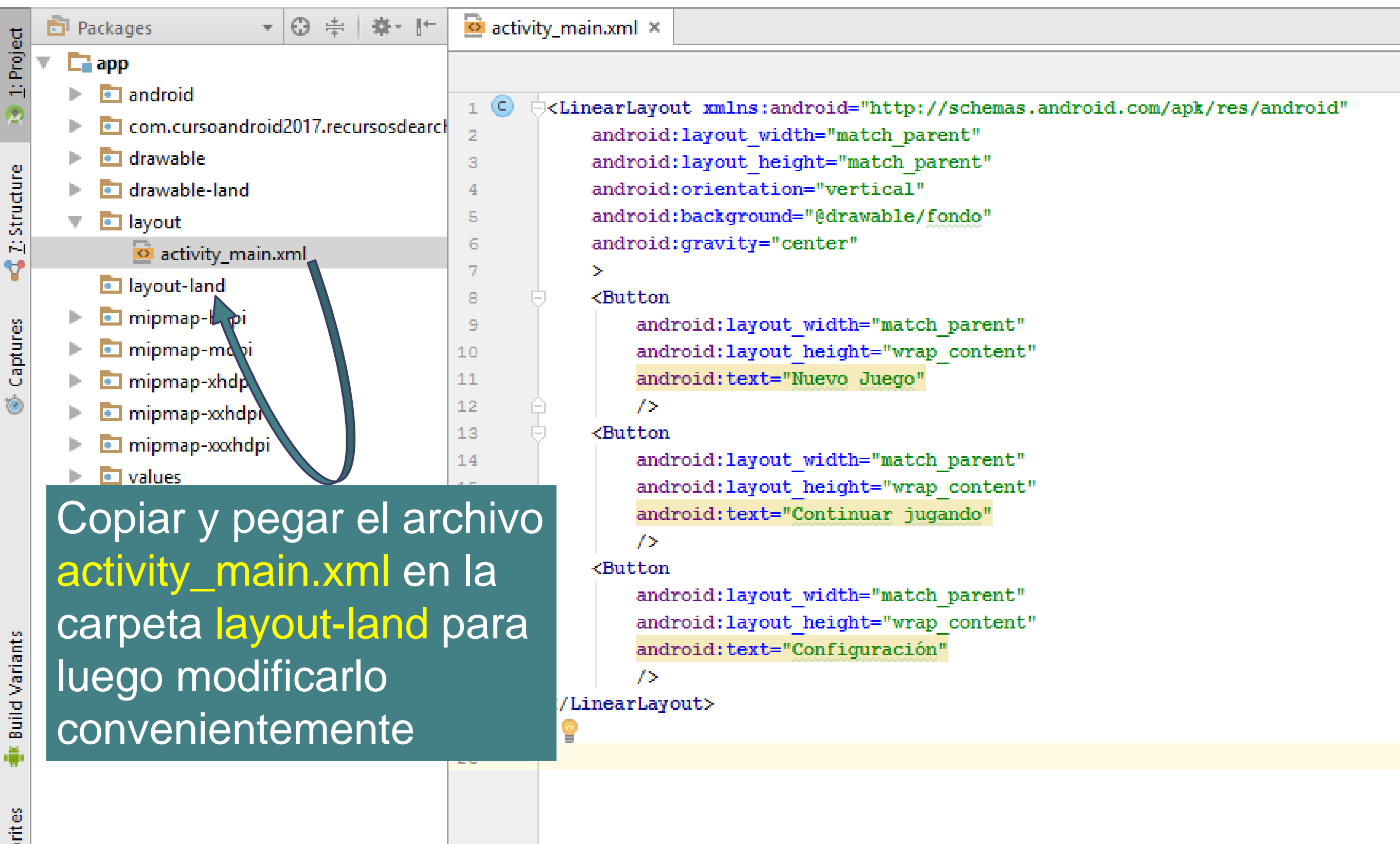
Crear un nuevo directorio de recursos **layout-land** para definir el **layout** de la **activity principal** en el caso de la **disposición horizontal** del dispositivo



Actividad guiada - continuación

Crear un nuevo directorio de recursos **layout-land** para definir el **layout** de la **activity principal** en el caso de la **disposición horizontal** del dispositivo





Project Structure:

- app
 - android
 - com.cursoandroid2017.recursosearch
 - drawable
 - drawable-land
 - layout
 - activity_main.xml
 - layout-land
 - mipmap-hdpi
 - mipmap-mdpi
 - mipmap-xhdpi
 - mipmap-xxhdpi
 - mipmap-xxxhdpi
 - values

activity_main.xml content:

```
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:background="@drawable/fondo"
6     android:gravity="center"
7 >
8     <Button
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:text="Nuevo Juego"
12    />
13    <Button
14        android:layout_width="match_parent"
15        android:layout_height="wrap_content"
16        android:text="Continuar jugando"
17    />
18    <Button
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:text="Configuración"
22    />
23 /LinearLayout>
```

Copiar y pegar el archivo **activity_main.xml** en la carpeta **layout-land** para luego modificarlo convenientemente

```

<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:background="@drawable/fondo"
    android:stretchColumns="*">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nuevo Juego" />
    <TableRow>
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Continuar jugando" />
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Configuración" />
    </TableRow>
</TableLayout>

```

activity_main.xml en la carpeta layout-land
Recurso de layout alternativo para la orientación landscape

Ejecutar en el emulador

Para poner en práctica - Ejercicio:

Crear un fondo alternativo para la activity principal para el caso en que el dispositivo esté dispuesto horizontalmente con la configuración del idioma en francés.

TIP: Se debe crear el directorio de recursos `drawable-fr-land/`