INDICE

1.1. Introducción

- ✓ Debemos tener en cuenta que los dispositivos que 'ejecutan' el SO Android son muy variados, en tamaño de pantalla y en puntos por pulgada, por lo que las resoluciones a utilizar pueden variar considerablemente.
- ✓ Por tanto, a la hora de diseñar una aplicación, debemos ser capaces de adaptarla a diferentes tamaños, para que se visualice correctamente y aproveche el espacio disponible.
- ✓ Para hacer esto, Android implementa un mecanismo de "sufijo" en las carpetas donde se almacenan los layouts.

  Cada uno de los sufijos indica diferentes características físicas da pantalla (tamaño, orientación, dp,...)

  Todos los layouts definidos en estas carpetas se cargarán cuando los sufijos coincidan con las características físicas del dispositivo.

- ✓ Los sufijos 'no son excluyentes' y se pueden combinar, por ejemplo, podemos tener un layout definido cuando dispongamos de una tablet de 720dp y está posicionado en horizontal.

- ✓ También podemos limitar en *AndroidManifiest.xml* los dispositivos que pueden instalar nuestra aplicación en función de la pantalla del mismo, con la etiqueta '*supports-screens*'.

  Más información en este enlace.´

1.2. En función de la orientación

1.2.1. Introducción

- ✓ Algunos dispositivos permiten que la pantalla gire, colocándola en posición horizontal-vertical (apaisado-retrato)
- ✓ Si queremos tener un diseño diferente para cada orientación, necesitamos crear una carpeta dentro de *'res'* llamada ***/res / layout-land***.
- ✓ Aquí dentro ponemos los diseños que queremos tener con una orientación horizontal del dispositivo, mientras dejamos en ***/res/layout*** los diseños para una orientación vertical.

Nota: en el emulador, para cambiar la orientación debemos presionar los botones **CTRL + F11** o **7** del teclado numérico, sin que el teclado numérico esté activado.

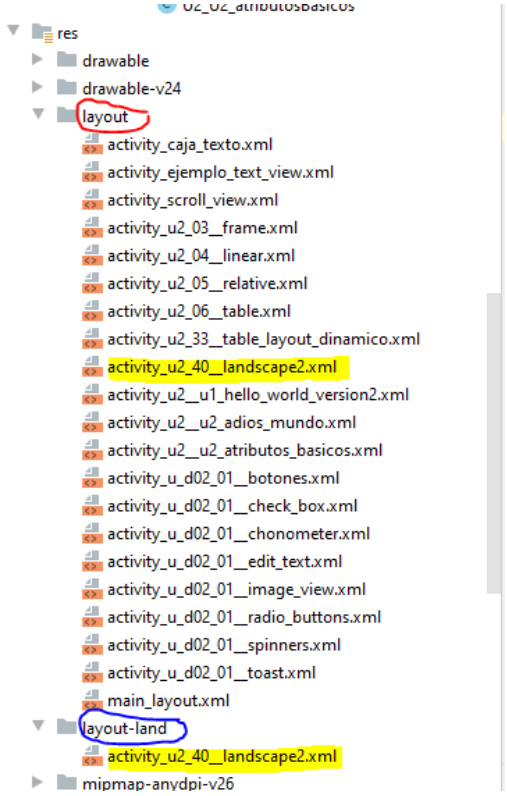
Veremos más adelante, en el tema Internacionalización , cómo podemos generar estos "sufijos" automáticamente cuando creamos el archivo de recursos.

- ✓ **Referencias:**
  - **Diferentes pantallas**: https://developer.android.com/guide/practices/screens_support

1.2.2. Caso práctico

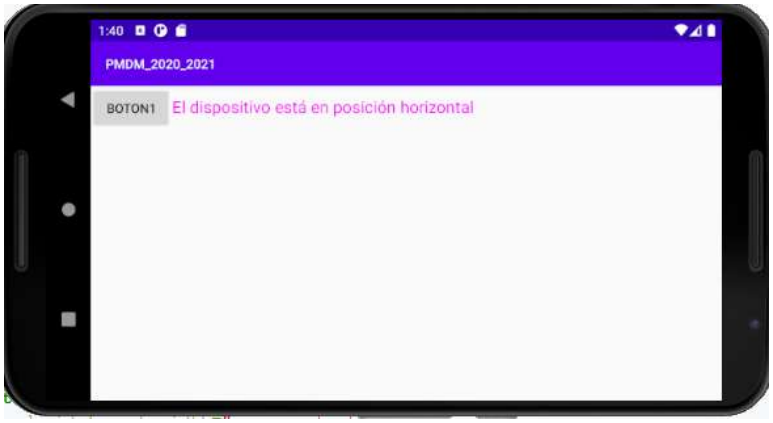
En este caso práctico vamos a ver los diferentes tipos de orientación que puede tener nuestra app que puede ser vertical (Portrait) y horizontal (Landscape), es decir, iremos adaptando los diseños a la orientación que tome el dispositivo (en este caso, el teléfono).

Cambio de orientación



Los ficheros de los layouts para cada orientación.

El dispositivo en vertical dispone de los elementos de forma distinta.



… a si esta en apaisado.

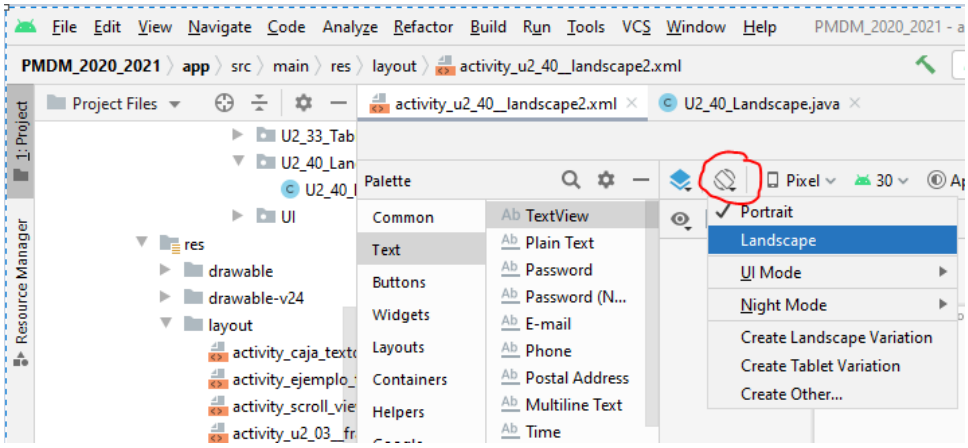
Comenzamos:

- ✓ Crear el proyecto: **U2_40_Landscape**
- ✓ Creamos una activity '*Empty Activity*' con el mismo nombre que el proyecto.
- ✓ Creamos el diseño (xml) vertical.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".U2_40_Landscape.U2_40_Landscape">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#00F"
        android:textSize="18sp"
        android:text="El dispositivo está en posición vertical" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Boton1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Esta etiqueta solo se muestra cuando el dispositivo está en vertical" />
</LinearLayout>
```

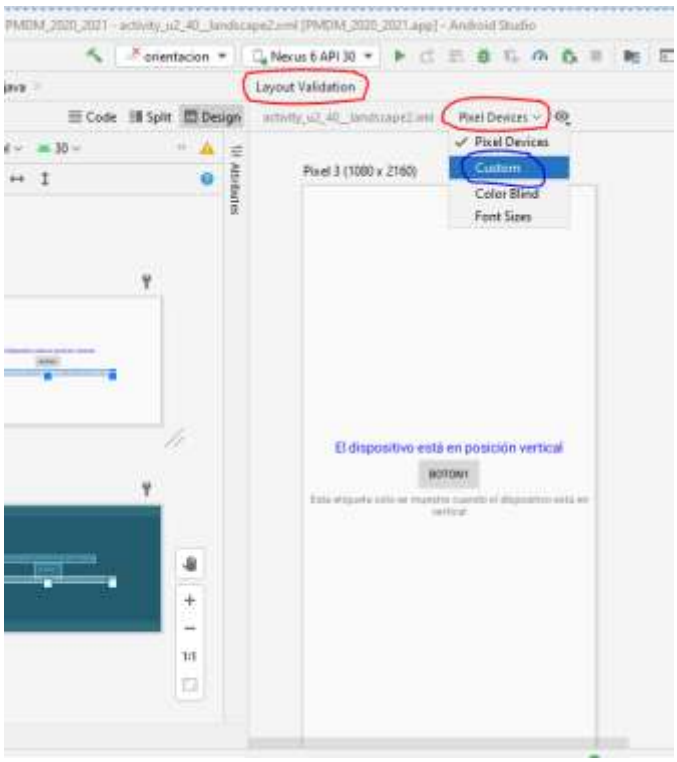
Visualización en tiempo real

Para poder ver el diseño en el modo horizontal seleccionamos la opción '*Landscape'* (Orientation for Preview (O)).



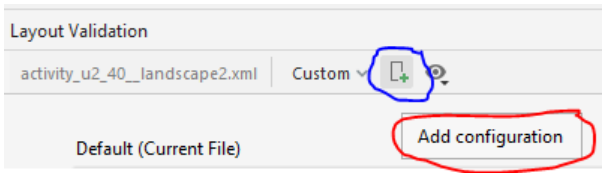
Para cualquier cambio que realicemos en los diseños, estar intercambiando entre Portrait – Landscape puede resultar muy engorroso. Nosotros lo que queremos es ver los cambios en ambas orientaciones en tiempo real, veamos cómo hacerlo:

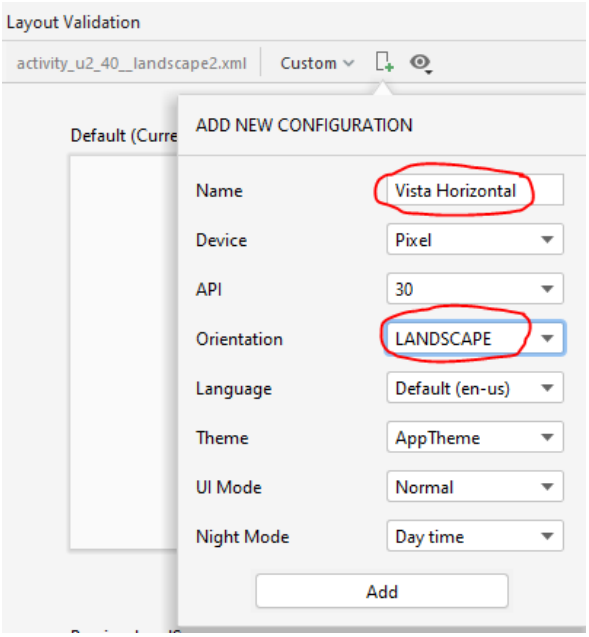
✓ Haremos una copia del diseño en modo Portrait.



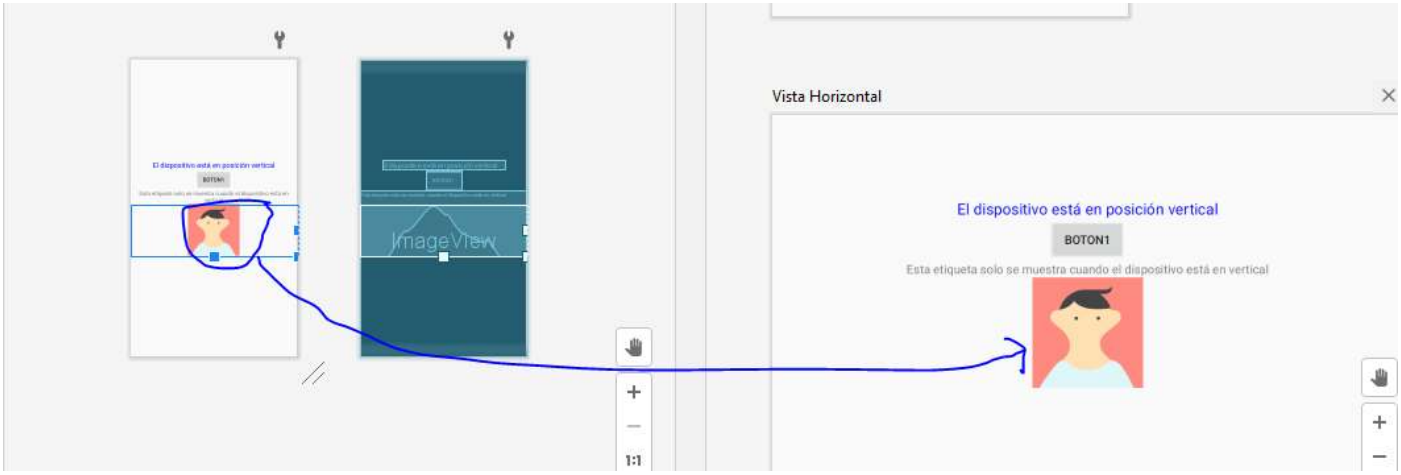
Pixel Device es simplemente para ir adaptando diseños de acuerdo a diferentes resoluciones de dispositivos.

- Seleccionamos '*Custom'* y añadimos una nueva configuración.

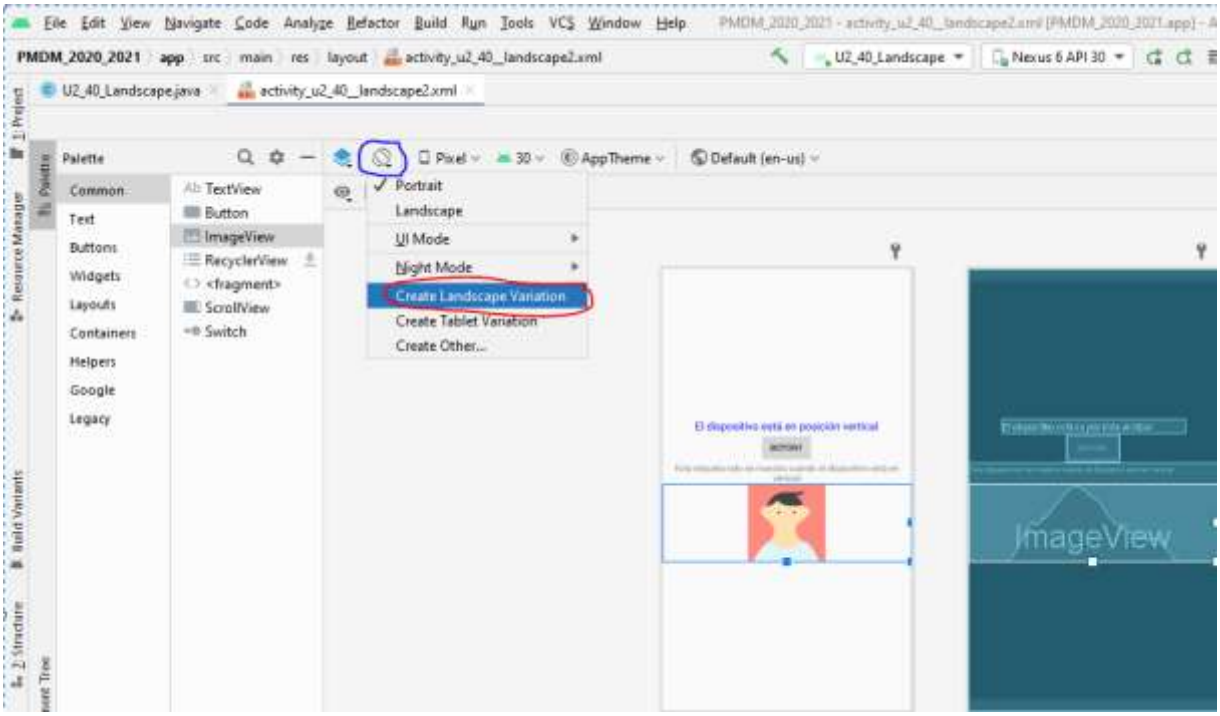




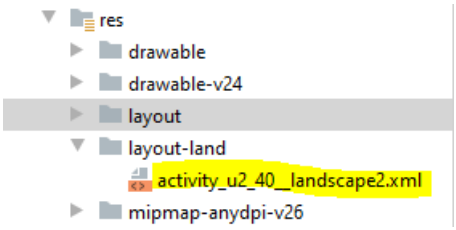
Ya tenemos nuestra aplicación en modo horizontal y la podemos ver en tiempo real. Por supuesto que cualquier modificación que realicemos en el diseño se reflejará en la vista horizontal. Por ejemplo, añadimos un ImageView.



✓ Creamos el diseño (xml) horizontal.



Esta acción genera un nuevo archivo xml cuyo nombre es el mismo que para el diseño vertical la diferencia es la ubicación que será **/res/layout-land**



A partir de aquí los diseños son independientes lo que significa que los cambios en uno de ellos no se reflejan en el otro.

✓ **XML layout horizontal**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".U2_40_Landscape.U2_40_Landscape">


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Boton1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#F0F"
```

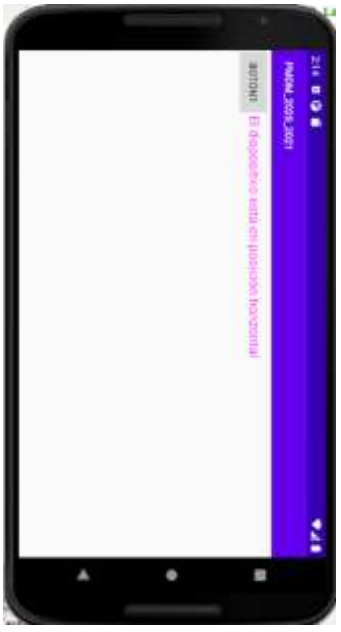
```
            android:textSize="18sp"
            android:text="El dispositivo está en posición horizontal" />

    </LinearLayout>
```

Forzar un diseño concreto

Es interesante saber que podemos forzar a que la app siempre muestre un diseño concreto aunque se cambie la orientación del dispositivo. Por ejemplo, vamos a hacer que siempre se muestre el diseño apaisado o landscape. Para ello modificaremos el archivo 'AndroidManifest.xml' añadiendo una nueva línea de código con el parámetro '*android:screenOrientation*'.

```
<activity android:name=".U2_40_Landscape.U2_40_Landscape"
    android:screenOrientation="landscape">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Independientemente de que cambiemos la orientación del dispositivo siempre se mostrará la vista landscape.



1.3. En función del tamaño

✓ A partir de la versión de Android 3.2 se cambia la forma de indicarle al S.O. que layout tiene que coger en función del tamaño de la pantalla. Más información en este enlace.

Anteriormente, en el directorio '**res/**' se ponía un sufijo (como **res/layout-xlarge/**) para indicar que se layout fuese cargado para un tamaño de pantalla grande.

Ahora, el sufijo que se va a añadir va a indicar el espacio necesario en pp (puntos por pulgada) que se necesita para poder visualizar el layout.

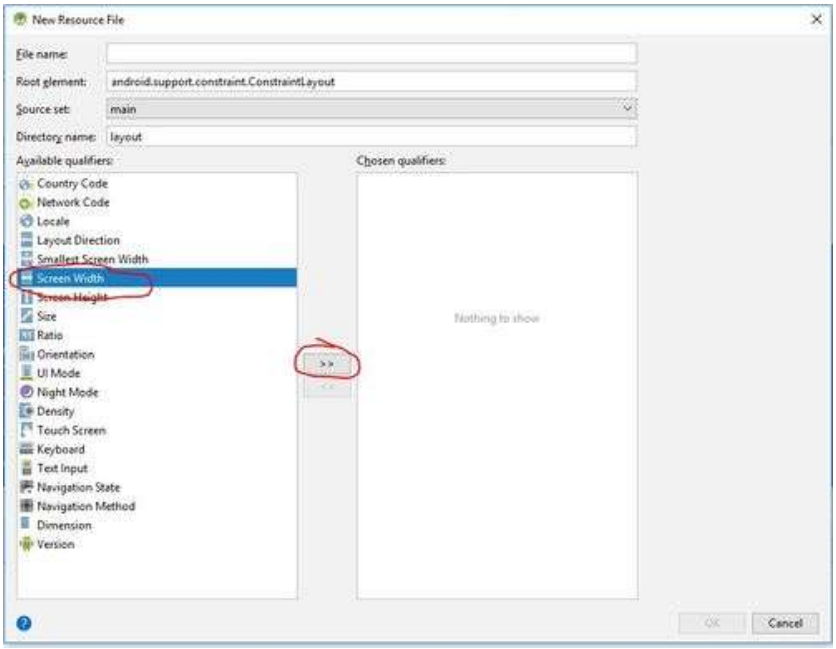
Esto se implementó para dar una respuesta de diseño a las tablet's de 7' ya que dichas tablet's entraban dentro de la misma categoría que las de 5' pero el espacio es bastante mayor.

El nombre que debe de tener la carpeta para que cargue el layout que se encuentre definido dentro de ella, debe de ser: **nombre_carpeta-sw<N>dp** siendo <N> el número de dp's necesario para que se pueda visualizar. Por ejemplo, 'res/clientes-sw600dp/.

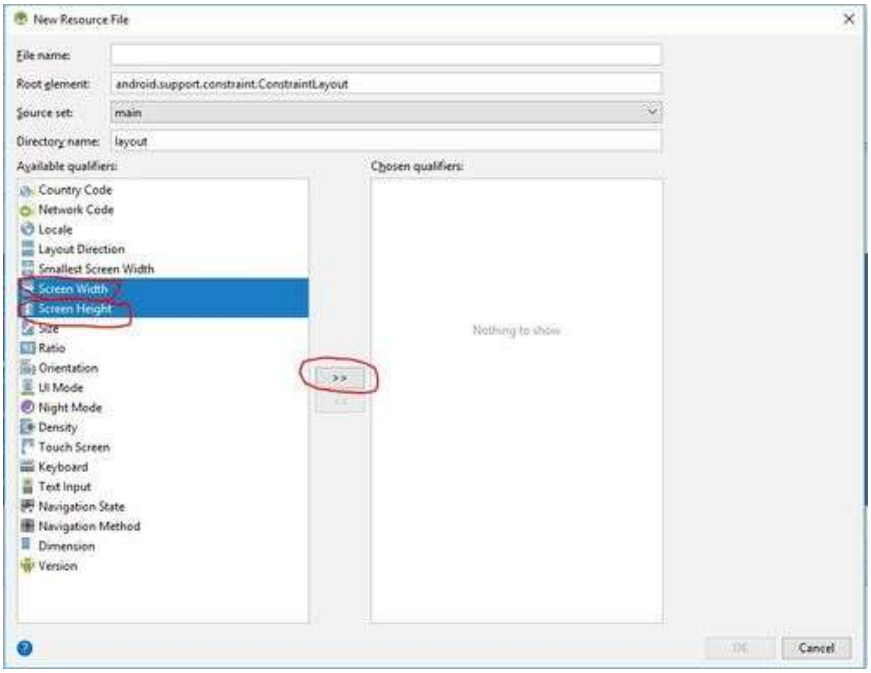
Este formato es independiente de que la pantalla esté en horizontal o vertical.

Por lo tanto, refleja el tamaño mínimo posible en dp para que la interface de un determinado layout pueda ser visualizada correctamente.

Esta es la opción que recomienda Android para el diseño de las Activities.



- ✓ Existen otros sufijos referidos solamente al ancho-alto de la pantalla:
  - **nombre_carpeta-w<N>dp**: Ancho mínimo en dp. Por ejemplo: res/productos-w150dp/
  - **nombre_carpeta-h<N>dp:** Alto mínimo en dp. Por ejemplo: res/productos-h300dp/



- ✓ Estos son algunos valores que puedes utilizar para tamaños de pantallas comunes:
  - **320dp**, para dispositivos con configuraciones de pantalla como las siguientes:
    - 240 x 320 ldpi (teléfono celular QVGA)
    - 320 x 480 mdpi (teléfono celular)
    - 480 x 800 hdpi (teléfono celular de alta densidad)
  - **480dp**, para pantallas de 480 x 800 mdpi (tablet/teléfono celular).
  - **600dp**, para pantallas de 600 x 1024 mdpi (tablet de 7 pulgadas).
  - **720dp**, para pantallas de 720 x 1280 mdpi (tablet de 10 pulgadas).