

Cómo admitir varias densidades de pantalla

En esta sección se explica cómo proporcionar diferentes recursos y utilizar unidades de medida de resolución independiente para admitir diferentes densidades de pantalla.

Cómo utilizar píxeles independientes de la densidad

Un error común que debes evitar al crear tus diseños es utilizar píxeles absolutos para definir distancias o tamaños. Definir dimensiones de diseño mediante píxeles es problemático porque cada pantalla tiene densidades de píxeles diferentes, por lo que es posible que el mismo número de píxeles corresponda a varios tamaños físicos en distintos dispositivos. Por tanto, al especificar dimensiones, debes utilizar siempre unidades `dp` o `sp`. `dp` es un píxel independiente de la densidad que corresponde al tamaño físico de un píxel a 160 dpi. `sp` es la misma unidad de base, pero aumentada en función del tamaño de letra preferido por el usuario (se trata de un píxel independiente de la escala). Por tanto, debes utilizar esta unidad de medida para definir el tamaño de letra (pero no para tamaños de diseños).

Por ejemplo, al especificar un espacio entre dos vistas, debes utilizar `dp` en lugar de `px`:

```
<Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/clickme"
        android:layout_marginTop="20dp" />
```

Al especificar el tamaño de letra, debes usar siempre `sp`:

```
<TextView android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:textSize="20sp" />
```

Cómo proporcionar mapas de bits alternativos

Dado que Android se ejecuta en dispositivos con diferentes densidades de pantalla, siempre debes proporcionar tus recursos de mapas de bits adaptados a los conjuntos de densidades generalizados: baja, media, alta y muy alta. De este modo, podrás conseguir un rendimiento y una calidad gráfica óptimos en todas las densidades de pantalla.

Para generar estas imágenes, debes empezar con tu recurso genérico en formato vectorial y generar las imágenes para cada densidad con la siguiente escala de tamaños:

- `xhdpi`: 2,0
- `hdpi`: 1,5
- `mdpi`: 1.0 (base)
- `ldpi`: 0,75

Esto significa que si generas una imagen de 200 x 200 para dispositivos `xhdpi`, debes generar el mismo recurso en 150 x 150 para `hdpi`, en 100 x 100 para `mdpi` y, por último, una imagen de 75 x 75 para dispositivos `ldpi`.

A continuación, sitúa los archivos de imagen generados en el subdirectorio adecuado en `res/` y el sistema

EN ESTA SECCIÓN PUEDES APRENDER:

1. [Cómo utilizar píxeles independientes de la densidad](#)
2. [Cómo proporcionar mapas de bits alternativos](#)

TAMBIÉN PUEDES CONSULTAR:

- [Cómo admitir varias pantallas](#)
- [Directrices para diseñar iconos](#)

¡PRUÉBALO!

Descargar la aplicación de ejemplo

NewsReader.zip

seleccionará el archivo correspondiente automáticamente en función de la densidad de la pantalla del dispositivo en el que se esté ejecutando la aplicación:

```
MyProject/  
  res/  
    drawable-xhdpi/  
      awesomeimage.png  
    drawable-hdpi/  
      awesomeimage.png  
    drawable-mdpi/  
      awesomeimage.png  
    drawable-ldpi/  
      awesomeimage.png
```

Por tanto, cada vez que hagas referencia a `@drawable/awesomeimage`, el sistema seleccionará el mapa de bits adecuado en función de los puntos por pulgada de la pantalla.

Para consultar más sugerencias y directrices sobre cómo crear recursos de iconos para tu aplicación, consulta la sección [Directrices para diseñar iconos \(/guide/practices/ui_guidelines/icon_design.html\)](https://developer.android.com/guide/practices/ui_guidelines/icon_design.html).