

## Pantallas en Android

El sistema operativo de Android se ejecuta en múltiples y variados dispositivos, estos vienen en una diversidad de tamaños, resoluciones y densidades de pantalla. Android busca ser lo más eficiente en todos los dispositivos y se encarga de ajustar las interfaces en cada pantalla, realizando el escalado y cambio de tamaño para que la aplicación funcione en estos dispositivos.

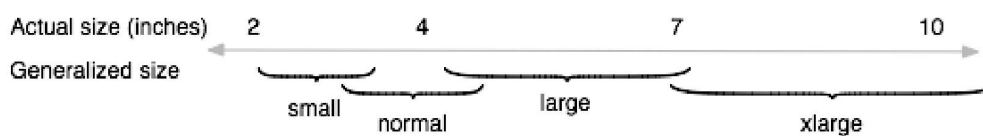
Es importante comprender algunos términos y conceptos al trabajar con interfaces gráficas en Android.

### Tamaño de Pantalla:

Este es el tamaño físico de la pantalla, se mide con la diagonal de la pantalla (de esquina a esquina, normalmente en unidades de pulgadas). Android clasifica los tamaños de pantalla en cuatro grupos muy simples: small, normal, large y extra large.

Una pantalla small se encuentra entre 2" y 3.6", normal entre 3.5" y 4.7", large entre 4.2" y 7" y por último xlarge o extra large que va desde 7" hasta 10".

Es muy usual que el fabricante del dispositivo clasifique el suyo en un grupo de los antes mencionados, hoy en día también es muy extraño encontrar dispositivos con tamaños de pantalla small.



### Densidad de Pantalla:

La densidad de pantalla hace referencia al número de píxeles dentro de un área física específica, por lo general se mide en puntos por pulgadas (dpi) o píxeles por pulgada.

Android clasifica las densidades en ocho grupos: low (baja), medium (media), high (alta), extra high (muy alta), extra extra high (muy, muy alta), extra extra extra high (extremadamente alta), nodpi y tvdpi.

Densidad	Descripción
ldpi	Recursos para pantallas de densidad baja (ldpi) (120 dpi)
mdpi	Recursos para pantallas de densidad media (mdpi) (~160 dpi; esta es la densidad de referencia)
hdpi	Recursos para pantallas de densidad alta (hdpi) (~240 dpi)
xhdpi	Recursos para pantallas de densidad muy alta (xhdpi) (~320 dpi)
xxhdpi	Recursos para pantallas de densidad muy, muy alta (xxhdpi) (~480 dpi)
xxxhdpi	Recursos para usos de densidad extremadamente alta (xxxhdpi) (~640 dpi)
nodpi	Recursos para todas las densidades. Estos son recursos independientes de la densidad. El sistema no escala recursos etiquetados con este calificador, independientemente de la densidad de la pantalla actual.
tvdpi	Recursos para pantallas entre mdpi y hdpi; de aproximadamente 213 dpi. Este no se considera un grupo de densidad "principal". Se usa más que nada para televisiones, y la mayoría de las apps no lo

	necesitarán ya que con recursos mdpi y hdpi será suficiente; el sistema los escalará según corresponda. Si crees que es necesario proporcionar recursos tvdpi, debes ajustar su tamaño con un factor de 1.33*mdpi. Por ejemplo, una imagen de 100 x 100 píxeles para pantallas mdpi debe ser de 133 x 133 píxeles para tvdpi.
--	---

Para crear elementos de diseño alternativos de mapa de bits que puedan ser utilizados en diferentes densidades, se debe seguir la proporción de escalamiento **3:4:6:8:12:16** entre las seis densidades principales.

Para el cálculo de estas densidades se puede tomar como ejemplo un mapa de bits de 48 x 48 píxeles para pantallas de densidad media, los tamaños resultantes de acuerdo a cada densidad deberían ser:

- 36 x 36 (0.75x) para densidad baja (ldpi)
- 48 x 48 (referencia de 1.0x) para densidad media (mdpi)
- 72 x 72 (1.5x) para densidad alta (hdpi)
- 96 x 96 (2.0x) para densidad muy alta (xhdpi)
- 144 x 144 (3.0x) para densidad muy, muy alta (xxhdpi)
- 192 x 192 (4.0x) para densidad extremadamente alta (xxxhdpi)

Este cálculo resulta de dividir 48 entre 4 (valor para densidad media mdpi, dentro de la proporción de escalamiento) y multiplicarlo por cada valor dentro de la proporción de escalamiento. Para una pantalla xhdmi será 12 x 8 igual a 96.

### **Orientación de Pantalla:**

La orientación de la pantalla puede ser de dos tipos horizontal (landscape) o vertical (portrait) desde el punto de vista de los usuarios. Esta posición cambia dependiendo de la posición del dispositivo, esto es posible dado que dispositivos como móviles y tabletas poseen un módulo o sensor llamado acelerómetro, el cual mide las rotaciones del dispositivo sobre tres ejes, con estos valores el sistema operativo se acomoda a una de las dos orientaciones.

La orientación es importante dado que el ancho y alto de la interfaz gráfica cambia de acuerdo a cada orientación.

El control de orientación puede ser bloqueado por el usuario y obligar al sistema operativo a omitir la disposición solicitada por el acelerómetro, el ejemplo más típico se da dentro de las aplicaciones de videojuegos que bloquean la orientación para mejorar la interacción del usuario con la interfaz. El usuario no solo puede bloquear la orientación, también le es posible crear diferentes interfaces de acuerdo a la orientación.

Algo sumamente importante que debe tener en cuenta el desarrollador es que Android reinicia la actividad en ejecución (destruye y reinicia) cuando se producen cambios de orientación en pantalla.

### **Densidad Independiente de Pixel:**

La unidad de densidad independiente de pixel o unidad de pixel virtual (dp), es una unidad flexible que escala de forma uniforme en todas las dimensiones de cualquier pantalla.

La densidad independiente de un pixel es equivalente a un pixel físico en una pantalla de 160 dpi, que es la densidad de línea de base asumida por el sistema para una pantalla de densidad "medium". Para calcularlo se usa la fórmula:

$$dp = (px \times 160) / dpi$$

$$px = dp \times (dpi / 160)$$

Por ejemplo, un margen de 2 dp para un objeto en una densidad de 240 dpi será igual a 3 pixeles mientras que ese mismo margen en una densidad de 480 dpi será equivalente a 6 pixeles.

## PARTE 1. CREACION DEL PROYECTO

Ejecute el programa *Android Studio*. En la ventana que se despliega haga clic sobre el menú *File* y la Opción *New Project*.

En la nueva ventana seleccione la pestaña *Phone and Table*, elija la opción *Empty Activity* y haga clic en el botón *Next*.

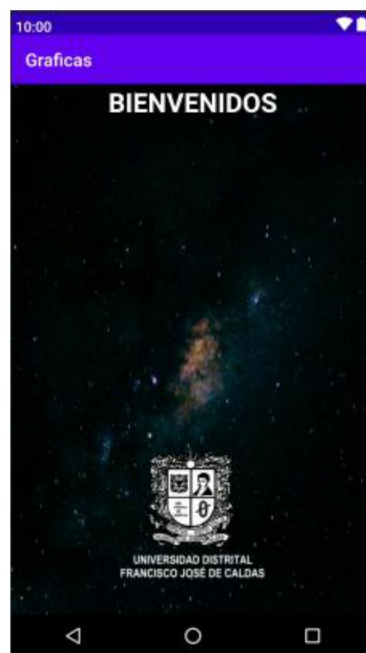
En la siguiente ventana escriba como nombre del proyecto **Graficas** y como dominio escriba **ud.example.graficas**, seleccione la mínima versión de Android de ejecución para su nueva aplicación (se recomienda la API 23, Android 6.0 Marshmallow), seleccione como lenguaje Java y haga clic sobre el botón *Finish*.

## PARTE 2. ADICION DE CODIGO

Por medio de un explorador de sistema operativo (Windows o Linux) busque el proyecto que se está realizando, y ubique dentro de la subcarpeta *drawable*, localizada dentro de la subcarpeta *res*, la imagen fondo.png que acompaña la práctica, cuidando en cambiarle el nombre a fondo1.png.

### Modificación del archivo *activity\_main.xml*

Nuestra actividad al final debería lucir de la siguiente manera:



Haga doble clic sobre el archivo activity\_main.xml del paquete layout, y cambie el modo de visualización a texto, por medio de la pestaña Text. Modifique el código de la siguiente manera:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:srcCompat="@drawable/fondo1"
        android:scaleType="centerCrop" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="159dp">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="BIENVENIDOS"
            android:textColor="#FFFFFF"
            android:textSize="30dp"
            android:textStyle="bold"
            android:textAlignment="center"/>
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

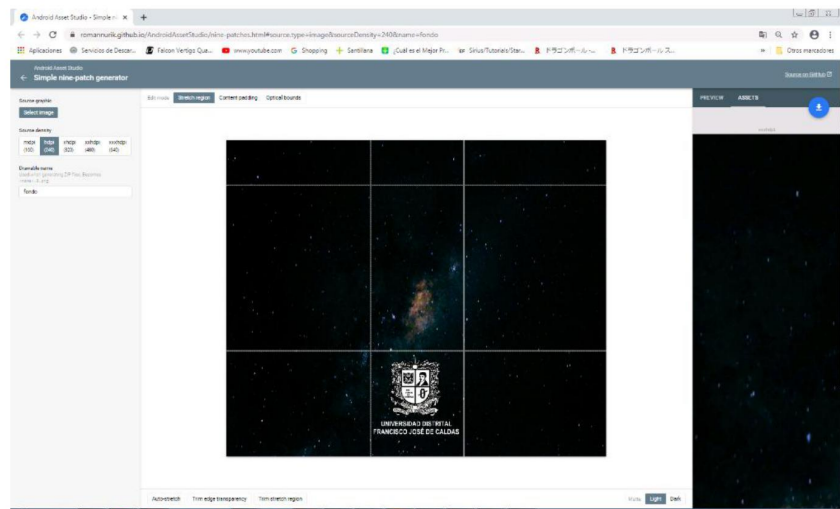
En este punto ejecute y evalúe la aplicación (rote el celular o cambie la orientación de la pantalla, para revisar el comportamiento de la aplicación, prestando especial atención a la imagen adicionada).

### **Mejorando la interfaz para adaptarse a múltiples dispositivos**

Una herramienta muy útil a la hora ayudarnos a mejorar las interfaces o imágenes es el 9-Patch, este es un sistema que utiliza Android para realizar un escalado selectivo de las imágenes mediante la adición de unos marcadores en la propia imagen, la imagen de bits está dividida en nueve secciones. Es muy común que Android dado el tamaño de la pantalla o la densidad transforme las imágenes distorsionándolas, esto lo puede evitar el 9-Patch. Un ejemplo del uso del 9-Patch son las imágenes usadas en los botones estándar de Android, estos botones deben estirarse a diferentes tamaños de acuerdo al dispositivo.

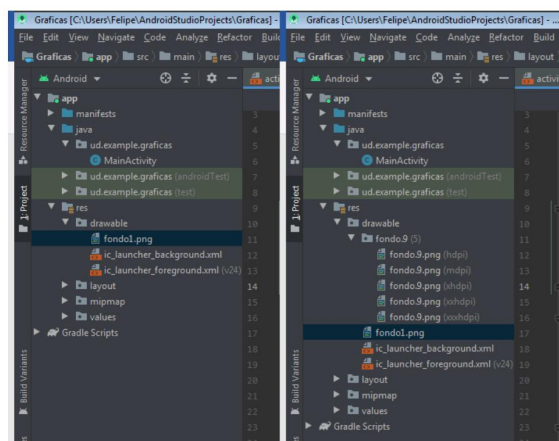
El SDK trae una herramienta para convertir una imagen a 9-Patch de una forma fácil, llamada Draw 9-Patch. Esta herramienta se encuentra en el directorio /tools/ donde se encuentra instalado el SDK de Android, solo se requiere abrir el programa (draw9patch) y hacer Drag&Drop de la imagen a la cual deseamos convertir en un 9-Patch. También es posible realizar este proceso desde paginas online, un ejemplo de estas páginas es *Simple nine-patch generator* que se puede ejecutar desde la URL: <https://romannurik.github.io/AndroidAssetStudio/nine-patches.html>, esta trabaja de igual manera basta seleccionar la imagen y soltarla en la página.

Con la ayuda de un browser vaya hasta la página mencionada y arrastre en ella la imagen fondo.png que acompaña a la practica 1. La idea es crear nuestras divisiones tal como se muestran en la siguiente imagen:



Descargue el archivo comprimido de la página y descomprima en una ubicación adecuada en el equipo. El archivo comprimido cuenta con una carpeta res y cinco carpetas de recursos drawables (las densidades que podemos manejar en los dispositivos móviles y tablets). Bastara con copiar estas carpetas dentro del proyecto en la subcarpeta res.

Por medio de un explorador de sistema operativo (Windows o Linux) busque el proyecto que se está realizando, y ubique la subcarpeta res, copie las cinco carpetas y los archivos incluidos en esta ubicación, vaya al proyecto abierto en Android Studio y observe los cambios.



Ahora basta remplazar el recurso utilizado (la imagen utilizada en la actividad) de fondo1 a fondo.

En este punto ejecute y evalúe la aplicación (rote el celular o cambie la orientación de la pantalla, para revisar el comportamiento de la aplicación, prestando especial atención a la imagen adicionada).

### **TRABAJO ADICIONAL**

1. Adicionar interfaces para la orientación landscape y para tablets.