escuelaartegranada

TEMA 2: Fundamentos técnicos del desarrollo de interfaces

Desarrollo de Interfaces
2º Desarrollo de Aplicaciones Multiplataforma
Curso 2020/2021 –Profesor: Juan Miguel Gonzálvez Craviotto

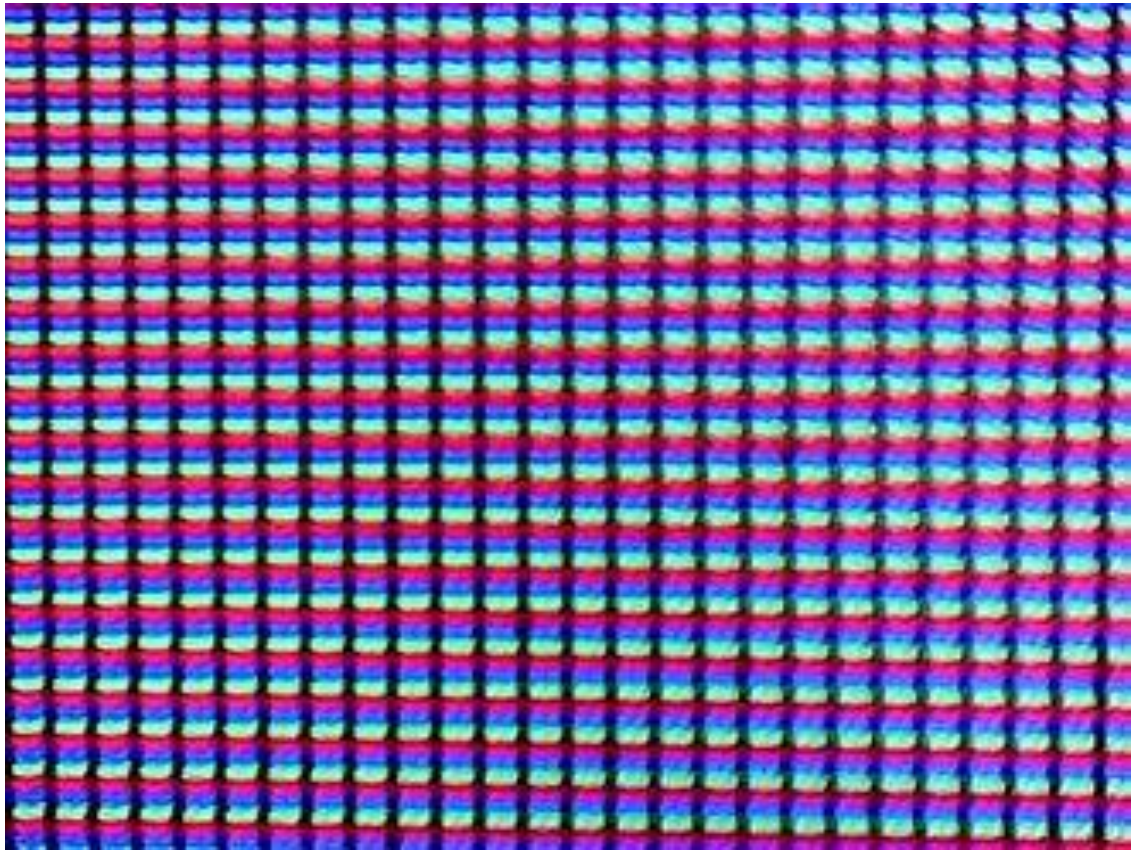
Índice

- Píxeles y resolución
- Colores
- Imágenes
- Multimedia

Píxeles y resolución

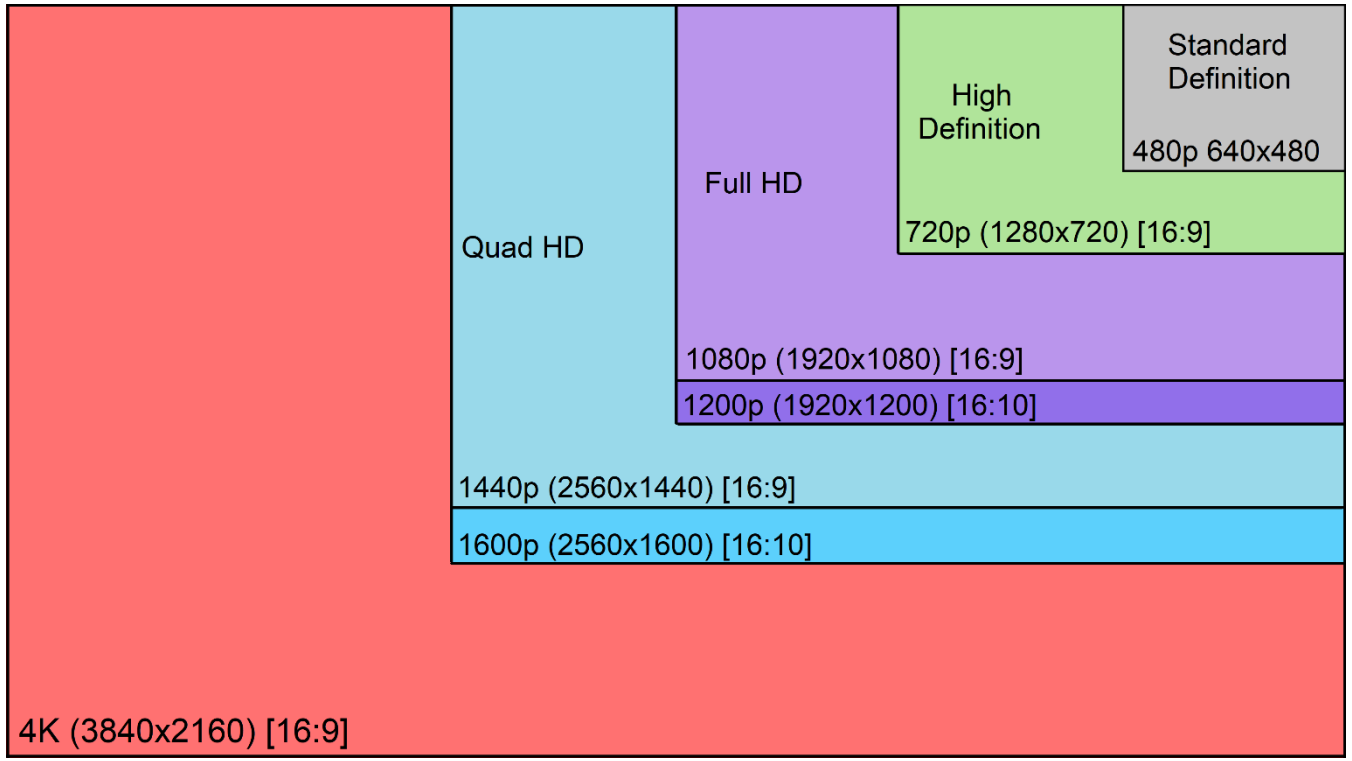
Píxel

- Unidad mínima homogénea en color que compone una pantalla
- Es decir, un pixel no puede tener dos colores, tendrá siempre un único color
- El color de cada pixel se define mediante el modelo de color RGB
- Las pantallas están formadas por una matriz de píxeles



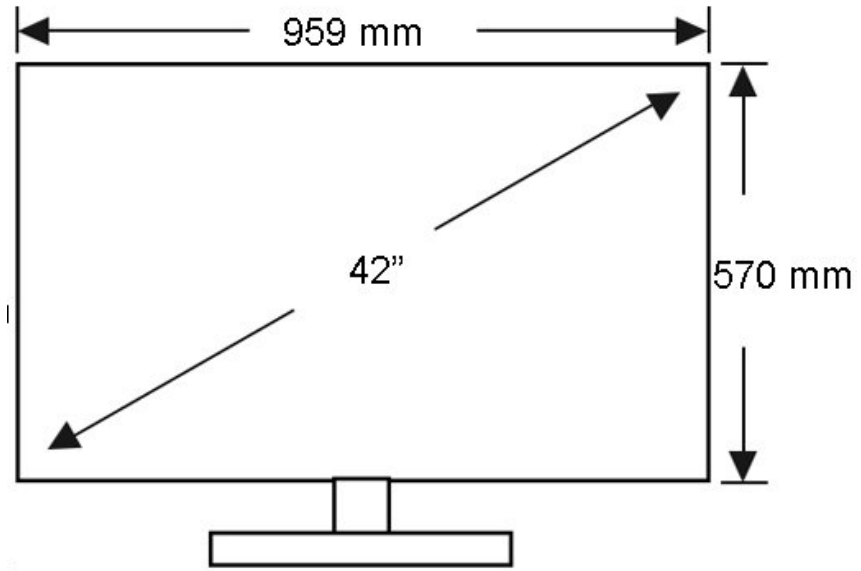
Resolución

- Número de píxeles que pueden ser mostrados por una pantalla
- Se define mediante dos número (x, y) que definen el número de píxeles a lo ancho y a lo alto de la pantalla (primero siempre el ancho)



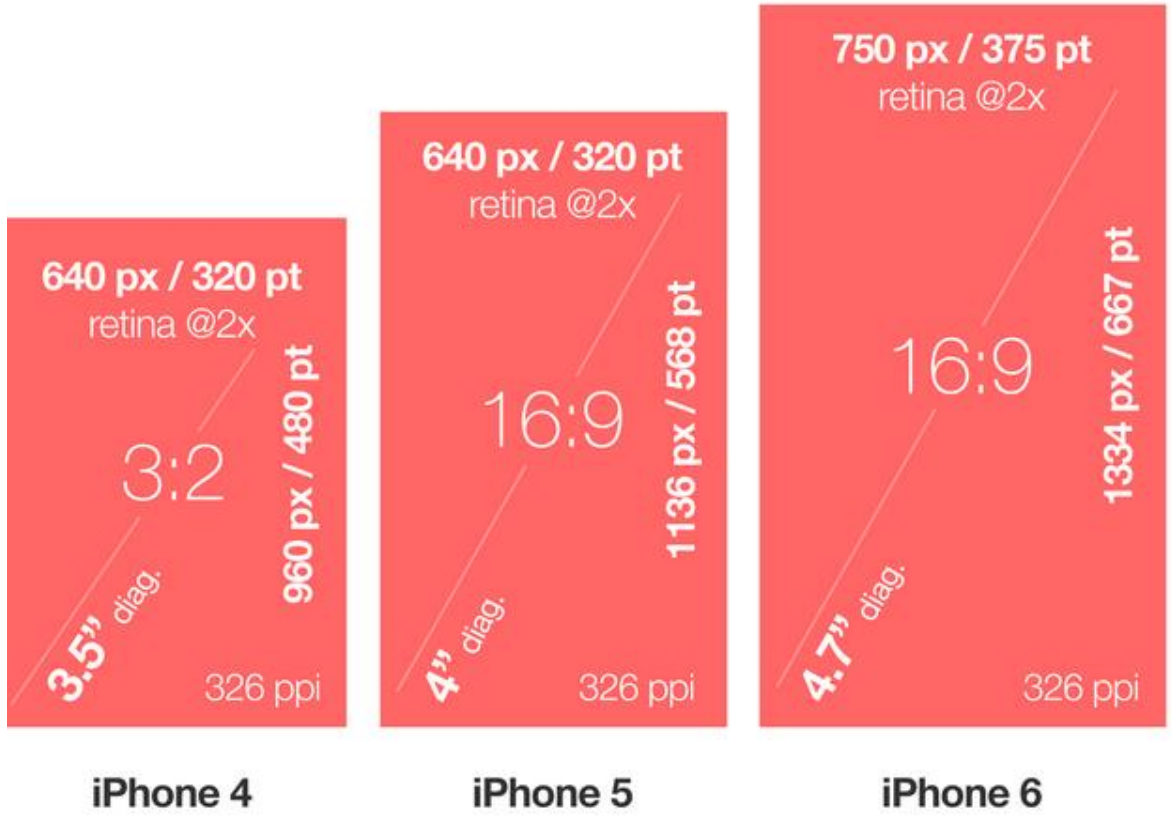
Tamaño

- Dos pantallas pueden tener la misma resolución pero distinto tamaño (en este caso una pantalla se verá más ‘pixelada’ que la otra)
- El tamaño de las pantallas se determina mediante la longitud de la diagonal
- Históricamente se ha medido en pulgadas (1 inc = 2.54 cm). Se utilizan pulgadas para medir pantallas incluso en países con el Sistema Internacional de Unidades



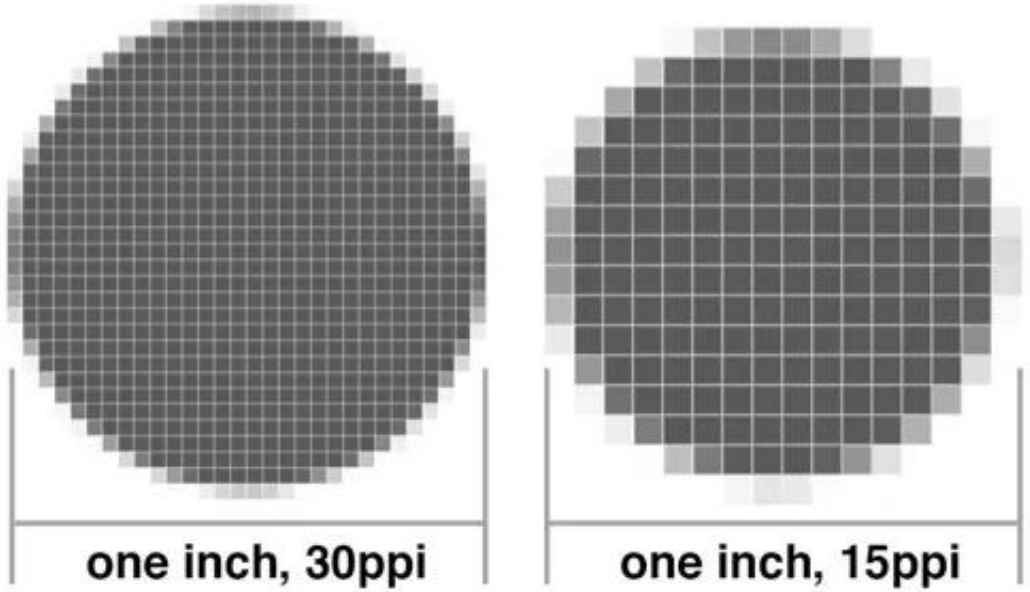
Proporción

- ¿Las pulgadas de una pantalla definen inequívocamente el tamaño de una pantalla?
- Para saber las medidas exactas no solo hay que tener en cuenta el tamaño de la diagonal, sino la relación entre el ancho y el alto
- ¡Cuidado! En los móviles algunas veces la proporción se expresa en orientación horizontal



Densidad de píxeles

- La densidad de píxeles de una pantalla se mide en píxeles por pulgada (ppi). Indican cuantos píxeles hay en una línea (horizontal o vertical) de tamaño una pulgada
- A la hora de imprimir se utiliza la medida de puntos por pulgada (dpi)



PPI

SPECS Compare

DISPLAY Benchmarks

Size:	6.2 inches
Resolution:	2400 x 1080 pixels, 20:9 ratio, 424 PPI
Technology:	Dynamic AMOLED
Refresh rate:	120Hz (adaptive)
Screen-to-body:	86.06 %
Peak brightness:	1300 cd/m2 (nit)
Features:	HDR support, Scratch-resistant glass (Corning Gorilla Glass Victus), Ambient light sensor, Proximity sensor

Unidades de medida en Android Studio

Unidades de medida en Android Studio – Unidades tradicionales

- **px**: correspondencia directa con los píxeles de la pantalla
- **in**: pulgadas
- **mm**: milímetros
- **pt**: puntos. Para medir el tamaños de la letra. Equivalente a 0.352778 mm

Points	Pixels		
6pt	8px	11pt	15px
7pt	9px	12pt	16px
7.5pt	10px	13pt	17px
8pt	11px	13.5pt	18px
9pt	12px	14pt	19px
10pt	13px	14.5pt	20px
10.5pt	14px	15pt	21px

Unidades de medida en Android Studio – Unidades tradicionales - Problemas

- ¿Qué sucede si definimos el tamaño de un elemento de la UI en píxeles y ejecutamos nuestra app en dos pantallas de distinta resolución?
- ¿Y en dos pantallas de la misma resolución y distinto tamaño?

- ¿Dos dispositivos con la misma resolución pero distinto tamaño de pantalla tienen igual ppi?
- ¿Dos dispositivos con distinta resolución pero igual tamaño de pantalla tienen igual ppi?

Unidades de medida en Android Studio – Unidades tradicionales - Problemas

- Los dispositivos Android no solo vienen con pantallas de diferentes tamaños (teléfonos celulares, tablets, TVs, etc.), sino que sus pantallas también tienen píxeles de distintos tamaños. Es decir, hay dispositivos que tienen 160 píxeles por pulgada cuadrada y otros son 480 píxeles en el mismo espacio. Si no tienes en cuenta estas variaciones de densidad de píxeles, es posible que el sistema escale tus imágenes (y se vean borrosas) o que estas se muestren en el tamaño incorrecto.

Unidades de medida en Android Studio – Unidades tradicionales - Problemas

- Definir dimensiones con píxeles es un problema, ya que las diferentes pantallas tienen distintas densidades de píxeles.

  Por lo tanto, la misma cantidad de píxeles puede corresponder a diferentes tamaños físicos en distintos dispositivos.

Píxeles independientes de la densidad - dp

- Para que el tamaño de tu IU se mantenga visible en pantallas con distintas densidades, debes diseñar tu IU con píxeles independientes de la densidad (dp) como unidad de medida.
- Un dp es una unidad de píxel virtual que prácticamente equivale a un píxel en una pantalla de densidad media (160 dpi; la densidad "de referencia"). Android traduce este valor a la cantidad apropiada de píxeles reales para cada densidad.

Density Bucket	Screen Density
ldpi	120 dpi
mdpi	160 dpi
hdpi	240 dpi
xhdpi	320 dpi
xxhdpi	480 dpi
xxxhdpi	640 dpi

Sp

- Al igual que dp es una medida independiente de la densidad.
- La diferencia es que se escala con respecto al tamaño de fuente elegido por el usuario.
- Por lo tanto se recomienda su uso a la hora de definir el tamaño del texto.

Conversión dp a píxeles

- Imagina una app en la que un gesto de desplazamiento se reconozca después de que el dedo del usuario haya recorrido al menos 16 píxeles. En una pantalla de referencia, un desplazamiento obligatorio del usuario de 16 pixels / 160 ppi, que equivale a un décimo de pulgada (o 2.5 mm) antes de que se reconozca el gesto. En un dispositivo con una pantalla de alta densidad (240 ppi), el dedo del usuario debe desplazarse 16 pixels / 240 dpi, que equivale a un quinceavo de pulgada (o 1.7 mm). La distancia es mucho más corta y, por lo tanto, la app resulta más sensible para el usuario.

Conversión dp a píxeles

- Para solucionar este problema, el umbral del gesto debe expresarse en dp en el código, y luego convertirse a píxeles reales.

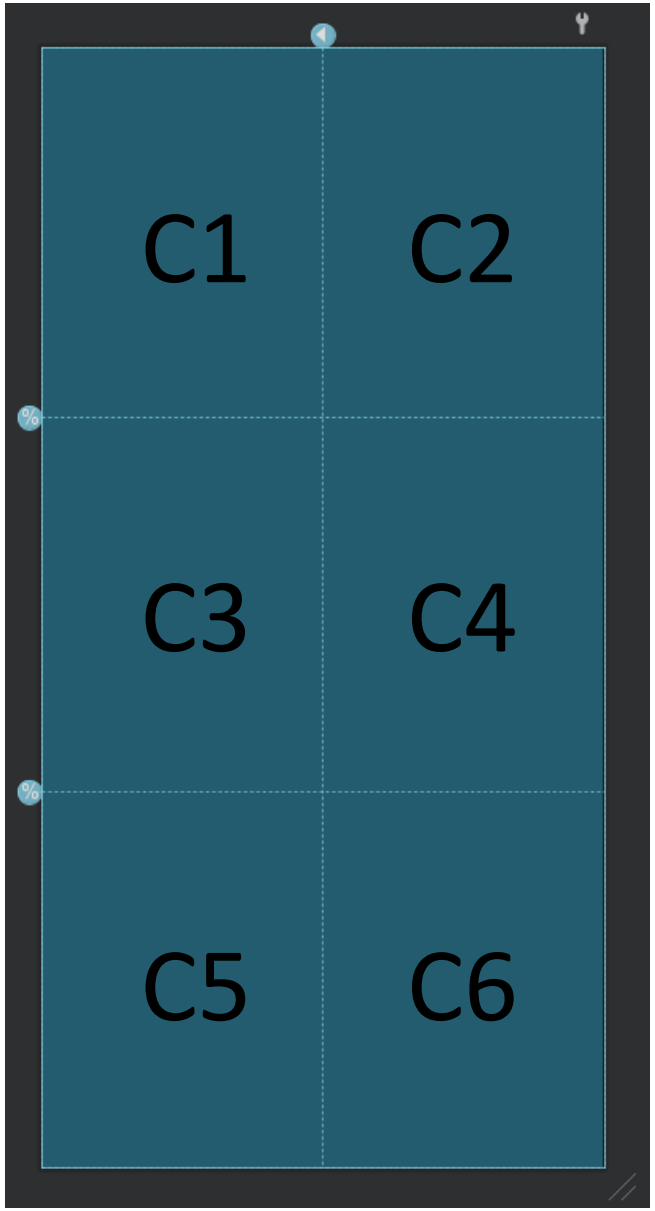
- px = dp * (ppi/ 160)

Coordenadas en la pantalla

- En Informática, al igual que en Matemáticas, definimos la posición de un punto en pantalla especificando sus dos coordenadas
- En Informática, a diferencia de en Matemáticas, el punto (0,0) representa la esquina superior izquierda

Ejercicios

- Especifica en que cuadrantes están los siguientes puntos (el móvil es el Pixel 3 (1080x2160):
  - A) (0,0)
  - B) (1080, 2160)
  - C) (0, 1000)
  - D) (720, 720)
  - E) (1005, 1400)
  - F) (0, 540)
  - G) (1150, 2100)
  - H) (100, 2000)
  - I) (600, 800)



Colores

Sistemas de colores

- En Informática hay tres formas principales de definir un color:
  - Mediante su nombre
  - Mediante su código RGB
  - Mediante su código RGBA o ARGB

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
 <color name="White">#FFFFFF</color>
 <color name="Ivory">#FFFFF0</color>
 <color name="LightYellow">#FFFFE0</color>
 <color name="Yellow">#FFFF00</color>
 <color name="Snow">#FFFAFA</color>
 <color name="FloralWhite">#FFFAF0</color>
 <color name="LemonChiffon">#FFFACD</color>
 <color name="Cornsilk">#FFF8DC</color>
 <color name="Seashell">#FFF5EE</color>
 <color name="LavenderBlush">#FFF0F5</color>
 <color name="PapayaWhip">#FFEFD5</color>
 <color name="BlanchedAlmond">#FFEBCD</color>
 <color name="MistyRose">#FFE4E1</color>
 <color name="Bisque">#FFE4C4</color>
 <color name="Moccasin">#FFE4B5</color>
 <color name="NavajoWhite">#FFDEAD</color>
```

RGB

- El color es definido por tres componentes (nivel de rojo, nivel de verde, nivel de azul)
- Cada componente toma un valor entre 0 y 255

- Ejemplos:
- Rojo → 255,0,0
- Azul → 0,0,255
- Amarillo → 255, 255, 0
- Negro → 0, 0, 0
- Blanco → 255, 255, 255

¿Por qué entre 0 y 255?

RGB

- Normalmente los colores RGB se suelen especificar en hexadecimal
- Por lo tanto… recordemos como se obtenían

Hexadecimal

- Sistema numérico en base 16.
- Los número van del 0 a la F → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B(11), C(12), D(13), E(14), F(15)
- Se suele anteceder el prefijo 0x para indicar que un número es hexadecimal

Preguntas

- ¿Cuánto es 16 en hexadecimal?
- ¿Cuánto es 0xF+0x1 en hexadecimal?
- ¿Cuánto es 0x11 en decimal?
- ¿Cuánto es 0xFF en decimal?

- ¿Problemas? https://www.calculator.net/hex-calculator.html

RGB en hexadecimal

- Ahora cada color va de 00 a FF.
- 00 significa que no ‘echamos’ nada de ese color.
- FF significa que ‘echamos’ lo máximo de dicho color
- FF0000 → Rojo, 0000FF → Azul, 000000 → Negro ...
- Valores intermedios indican la cantidad exacta del color que echamos

Preguntas

- ¿Qué color es FFFF00?
- En el color 220033, ¿de qué color básico (rojo, verde o azul) se está ‘echando’ más?
- En el color 22AA33, ¿de qué color básico (rojo, verde o azul) se está ‘echando’ más?
- En el color FFAA33, ¿de qué color básico (rojo, verde o azul) se está ‘echando’ más?
- En el color F3AAFD, ¿de qué color básico (rojo, verde o azul) se está ‘echando’ más?

¿Problemas?

- Prueba a utilizar la siguiente herramienta https://www.w3schools.com/colors/colors_picker.asp hasta que te sientas cómodo definiendo colores RGB en hexadecimal

RGB, grises

- Cuando echamos igual cantidad de todos los colores el resultado es: negro, gris o blanco. Dependerá de la cantidad
- Si echamos 00 de todos el resultado será el negro
- Si echamos FF de todos el resultado será el
- Valores intermedios son grises más o menos oscuros

Preguntas

- ¿Qué color es CCCCCC? ¿Y 313131?
- ¿Qué gris es más oscuro 4A4A4A o A4A4A4?

ARGB

- Se añade un nuevo componente (Alpha, Red, Green, Blue). El componente Alpha determina la trasparencia de un color. Su valor, al igual que los otros tres, va de 0 a FF
- Alpha = FF, indica un color opaco
- Alpha = 00, indica un colo transparente por completo (no se vería)
- Valores intermedios cambian el nivel de trasparencia del color.

Preguntas

- ¿El color RGB FF0000 es el mismo que el color ARGB FFFF0000?
- ¿Se vería alguno de los siguientes colores RGB 0031AADD, 00000000, 002312AC?

¡Cuidado! Tenemos ARGB y RGBA

- Son la misma forma de definir colores, solo que en RGBA el canal alpha se especifica al final. En la web se utiliza RGBA y en Android Studio utilizamos ARGB

Imágenes

Tipos de imágenes

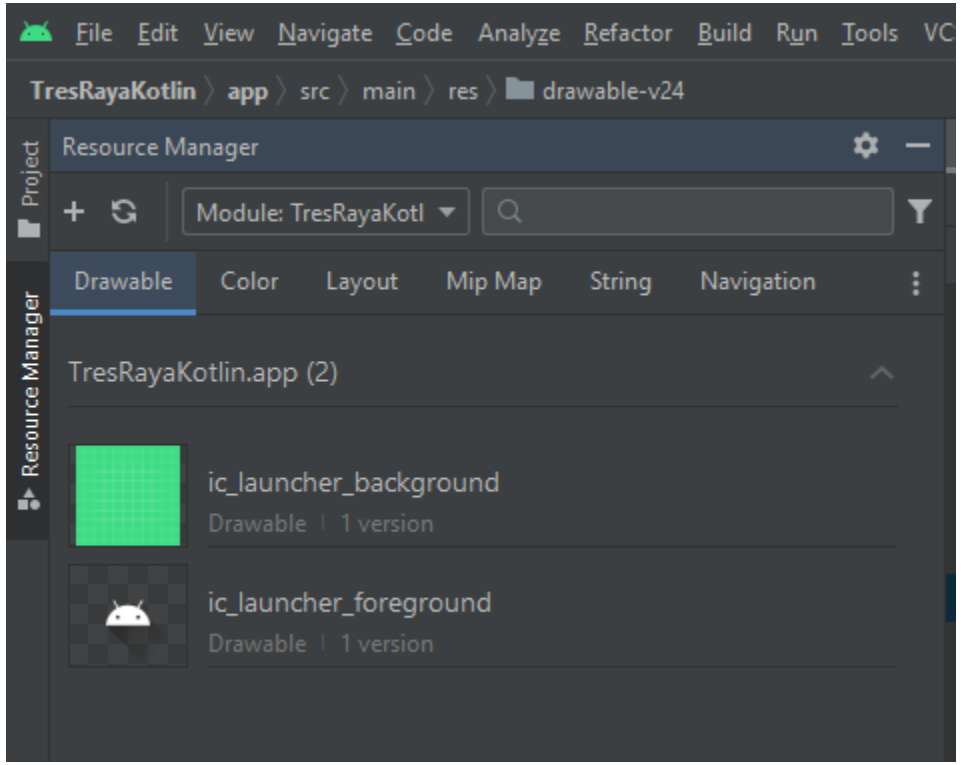
- **Imagen mapa de bits**: la imagen se guarda como una matriz de ternas, donde cada terna especifica un color
- **Imagen vectorial**: la imagen se especifica como un conjunto de puntos en el plano cartesiano

Imágenes mapa de bits

- Las imágenes representadas mediante un mapa de bits se suelen comprimir, en caso contrarío ocuparían mucho
- De hecho el formato de una imagen lo que especifica es el tipo de compresión que ha recibido, siendo las principales
  - JPEG: compresión con pérdidas. No admite canal alfa. Mejor opción para paisajes e imágenes similares con muchos colores y formas irregulares.
  - PNG: compresión sin pérdidas. Admite canal alfa.
  - GIF: comprensión sin pérdidas. Permite imágenes animadas.
  - WebP: formato de Google creado para sustituir a las imágenes JPEG, PNG y GIF, ya que podemos especificar si queremos una comprensión sin perdidas o con perdidas.

Imágenes mapa de bits

- Las imágenes mapa de bits soportadas por Android Studio son: BMP, JPEG, PNG, GIF, WebP y HEIF
- Puedes agregar imágenes desde la pestaña Resource Manager > Drawable:



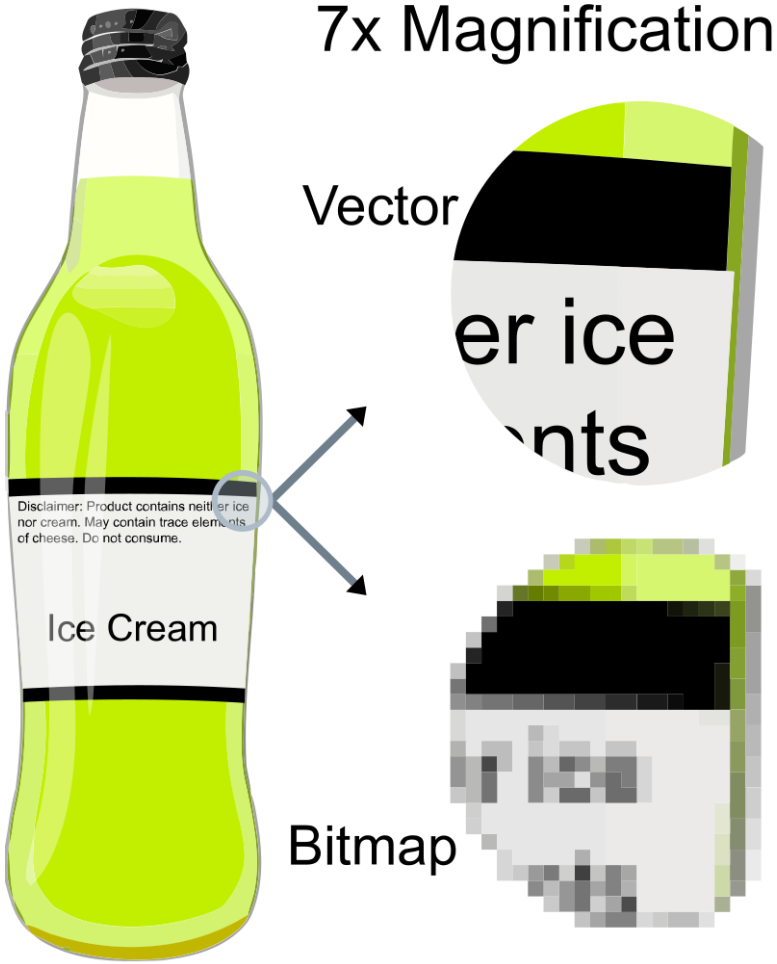
Imágenes vectoriales

- Al definirse como puntos en el plano cartesiano se les pueden aplicar trasformaciones matemáticas, permitiendo que una imagen sea escalada todo lo que se quiera sin necesidad de que se pixele
- Ocupan mucho menos espacio
- Sin embargo no toda imagen puede convertirse a formato vectorial, por ejemplo una foto quedará rara si la trasformamos a vectorial
- Los logos e iconos siempre deben de ser vectoriales
- Android Studio soporta imágenes vectoriales .SVG y .PSD

Usar SVG en Android Studio

- Puedes convertir un archivo SVG a un elemento de diseño vectorial de Android Studio fácilmente con Vector Asset Studio de la siguiente manera:
  - En la ventana Proyecto, haz clic con el botón derecho en el directorio res y selecciona Nuevo > Elemento vectorial.
  - Selecciona Archivo local (SVG, PSD).
  - Ubica el archivo que quieras importar y realiza las modificaciones necesarias.

Mapa bits vs vectoriales

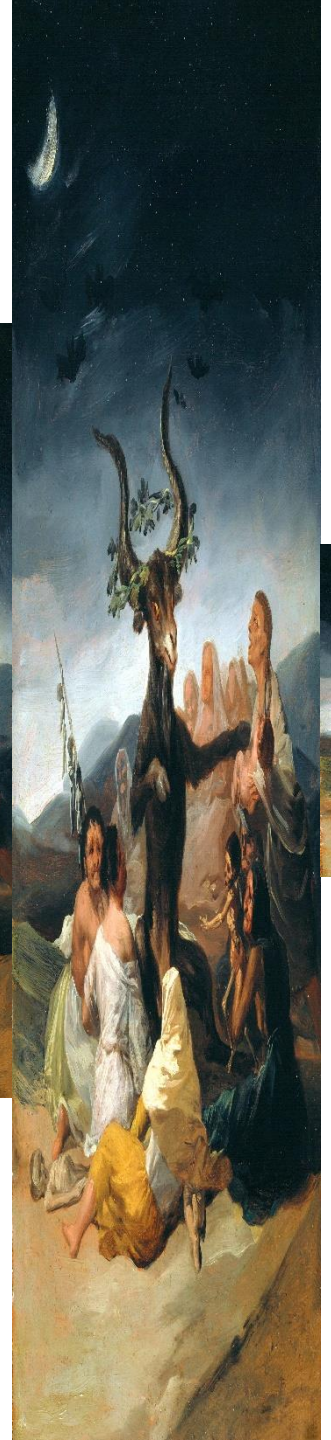


Mapa bits vs vectoriales

Redimensionar imágenes

- En mucha ocasiones querremos utilizar imágenes en un tamaño distinto al original
- Es importante que al redimensionar imágenes mantengamos su relación de aspecto original (anchura/altura)
- Será obligatorio a lo largo de todo el curso que las imágenes guarden su relación de aspecto original
- Una imagen puede hacerse más pequeña sin problema mientras guarde su relación de aspecto original
- Si una imagen se hace más grande, aunque guarde su relación de aspecto original, se pixelará

Bien ↓

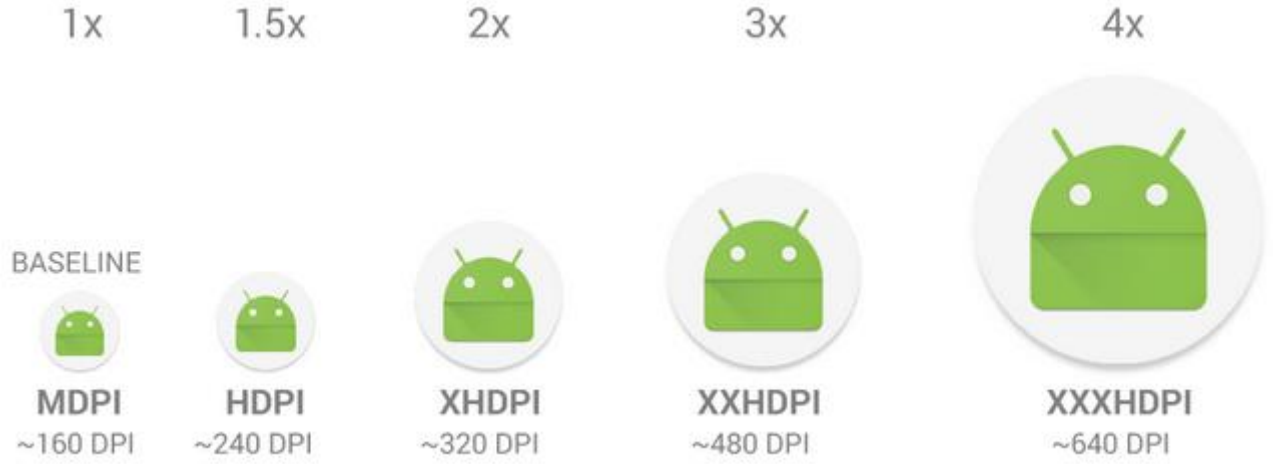


← Mal

Mal ↓

Imágenes en Android Studio

- Para proporcionar gráficos de buena calidad en dispositivos con diferentes densidades de píxeles, debes brindar varias versiones de cada mapa de bits en tu app (una para cada intervalo de densidad) en la resolución correspondiente. De lo contrario, Android deberá escalar tu mapa de bits para que ocupe el mismo espacio visible en cada pantalla, lo que generará artefactos de escalamiento como imágenes borrosas.



Calificador de 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.

Debes seguir la **proporción de escalamiento 3:4:6:8:12:16** (0.75, 1, 1.5, 2, 3, 4) entre las seis densidades principales

Multimedia

Videos

- Todo lo dicho para las imágenes en cuanto a la relación de aspecto se aplica también a los videos
- Tan solo ten en cuenta que la mayoría de los videos tienen una relación 16:9, aunque no siempre tiene que por que ser así, los videos antiguos tenían una relación de aspecto 4:3
- Los formatos soportados por Android Studio son: .mp4, .webM, .mkv y .3gp

Audio

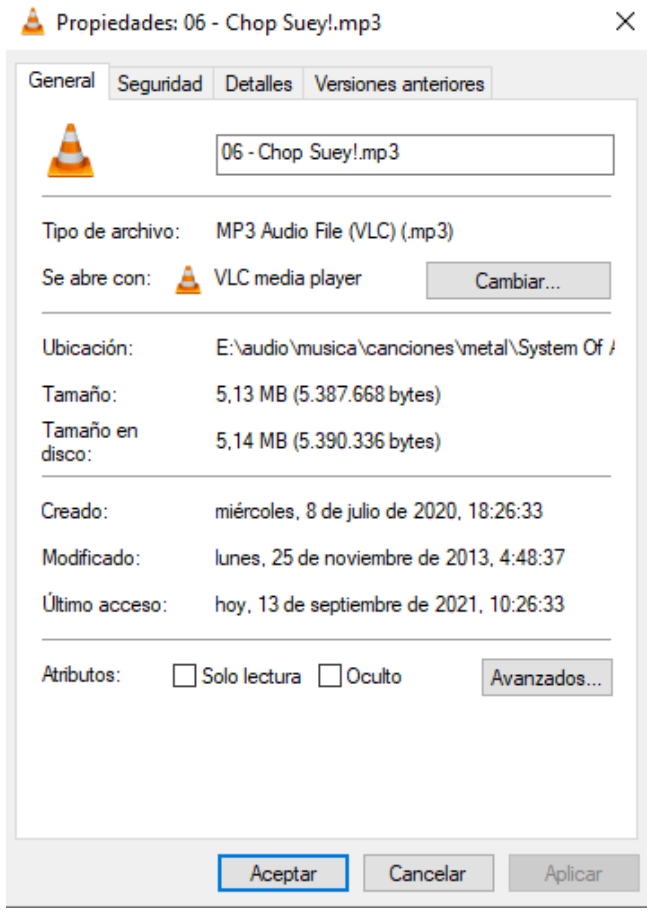
- Los ficheros de audio puede grabarse con perdidas o sin perdidas
- El principal formato sin perdidas es .wav
- Los principales formatos con pérdidas son .mp3 y .ogg
- Aunque en la práctica .mp3 y .wav se escuchen igual si vas a editar el audio es importante grabarlo en .wav o sino la edición hecha si perderá calidad
- Es bueno pasar a .mp3 o .ogg una vez el audio ha sido editado
- Android Studio soporta .wav, .mp3 y .ogg

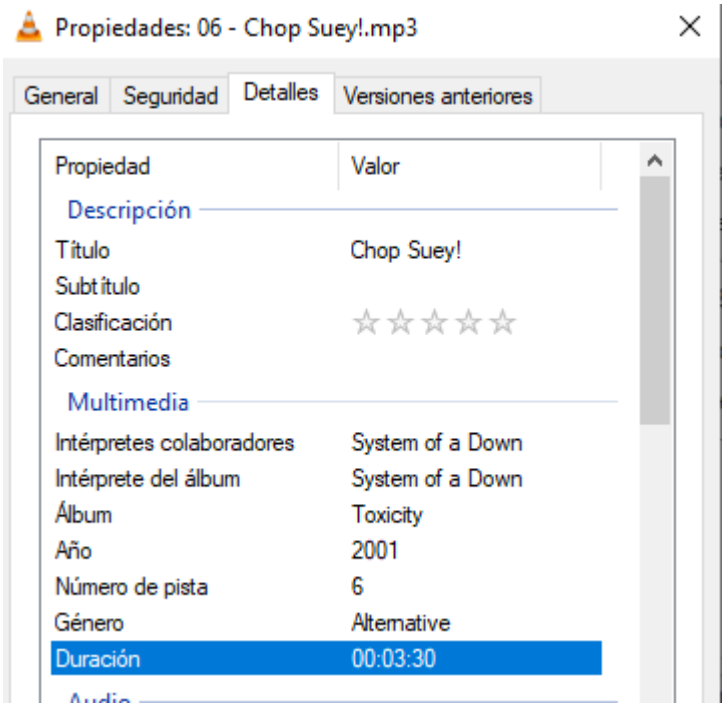
Audio bitrate

- Una métrica para medir la calidad del audio es el bitrate
- Nos indica la cantidad de bits por segundo que codifica el fichero
- Por lo general a mayor bitrate mayor calidad, pero hasta cierto punto donde lo único que conseguimos es que el fichero sea más pesado sin ganar calidad
- Por lo general el bitrate de los ficheros se encuentra en el rango (128, 320) kbps
- Con un bitratede 192 kbps es más que suficiente
- Muchos ficheros .mp3 tiene bitrate de 128kbps que puede resultar insuficiente para personas que utilicen unos buenos altavoces

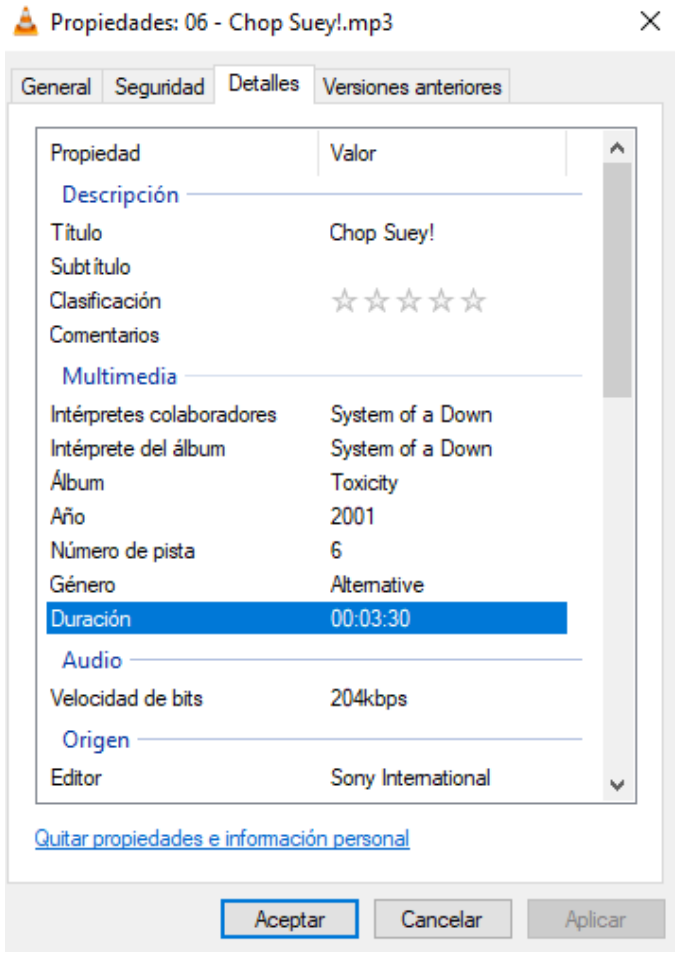
Ejercicios

- ¿Cuál es el bitrate?





Ejercicios


Propiedades: 06 - Chop Suey!.mp3
General
Seguridad
Detalles
Versiones anteriores
Propiedad
Valor
Descripción
Título
Chop Suey!
Subtítulo
Clasificación
Comentarios
Multimedia
Intérpretes colaboradores
System of a Down
Intérprete del álbum
System of a Down
Álbum
Toxicity
Año
2001
Número de pista
6
Género
Alternative
Duración
00:03:30
Audio
Velocidad de bits
204kbps
Origen
Editor
Sony International
Quitar propiedades e información personal
Aceptar
Cancelar
Aplicar