Diseño del Interfaz - II

Interfaces independientes de la pantalla

Cuando se 'inventó' Android, sólo existía un terminal capaz de soportarlo. Desde entonces, Android se ha instalado en todo tipo de dispositivos, desde móviles a tablets pasando por relojes inteligentes etc. Esto implica que **no podemos** hacer una app pensando en que se va a ver en un dispositivo en concreto. Si lo hacemos, estamos restringiendo nuestra a que nuestra app se instale únicamente en un % muy pequeño de dispositivos.

Recuerda que todo lo que se hace para Android está pensado para que funcione en todas partes. Para lograrlo, Android tiene claramente separados cada uno de sus componentes; y lo mismo ocurre con sus apps. El código, por un lado, el interfaz por otro, los recursos por otro, etc. No nos tiene que extrañar por tanto que existan formas de trabajar con cosas tan concretas como la resolución de pantalla de un dispositivo.

La resolución de pantalla de un dispositivo en Android se clasifica así:

Nombre	Resolución
QVGA	240 x 320 pixeles
HVGA	320 x 480 pixeles
WQVGA	240 x 432 pixeles
WVGA	480 x 800 pixeles
WXGA	1280 x 800 pixeles

Múltiples ficheros de recursos

Android permite crear una estructura paralela de directorios de recursos para cada **resolución de pantalla**, **idioma**, **densidad**, etc. Es decir, que podemos disponer (por ejemplo) de la misma imagen repetida para diferentes densidades de pantalla. El propio Android se encargará de mostrar la imagen adecuada automáticamente.

Esto se consigue mediante una serie de **sufijos** que se pueden añadir a las carpetas de recursos. Al hacerlo, esos recursos quedan 'marcados' para ser utilizados por una u otra pantalla.

Tamaño de pantalla: Tamaño relativo a un terminal “estándar”	
Sufijo	
-small	Tamaño menor a 3 pulgadas
-medium	Un móvil normal, desde las 3 hasta las 4.5 pulgadas
-large	Para móviles grandes o tablets pequeñas, desde 4.5 hasta 7 pulgadas
-xlarge	Para tablets, notebooks, o televisiones. Originalmente de 7 a 10 pulgadas

Densidad: Densidad de pixeles por pantalla. Se mide en puntos por pulgada (dpi)	
Sufijo	
-ldpi	Para pantallas desde los 100 hasta los 150 dpi
-mdpi	Para pantallas desde los 150 hasta los 200 dpi
-hdpi	Para pantallas desde los 200 hasta los 250 dpi
-xhdpi	Para pantallas desde los 250 hasta los 300 dpi
-xxhdpi	Para pantallas de alrededor de los 480 dpi
-xxxhdpi	Para pantallas de alrededor de los 640 dpi
-nodpi	Para recursos que no deberían ser escalados sin importar la pantalla

Relación de Aspecto: Relación de la altura con respecto a la anchura de la pantalla	
Sufijo	
-long	Para pantallas más altas que los dispositivos estándar
-notlong	Para pantallas de relación de aspecto estándar

Cada uno de estos sufijos se puede usar de forma independiente o en combinación. Por ejemplo, podemos hacer cosas como:

```
res/layout-small-long/
res/layout-large/
res/drawable-hdpi/
```

Normalmente utilizaremos las variantes de densidad para los *drawables*, y el tamaño y relación de aspecto para los *layouts*. Sin embargo, vale cualquiera de ellos.

Es posible encontrar por internet tablas de referencia como la que ves aquí en la que se puede consultar la relación que hay entre los modelos de pantalla, densidades y tamaños. Esta tabla probablemente esté obsoleta, la incluyo por referencia.

Tipos de pantallas

	Low density (120), *ldpi*	Medium density (160), *mdpi*	High density (240), *hdpi*	Extra high density (320), *xhdpi*
Small screen	**QVGA (240x320)**		480x640	
Normal screen	**WQVGA400 (240x 400)** **WQVGA (240x432)**	**HVGA (320x480)**	**WVGA (480x800)** **WVGA854 (480x854)** 600x1024	640x960
Large screen	**WVGA800 (480x800)** **WVGA854 (400x854)**	**WVGA800 (480x800)** **WVGA854 (480x854)** 600x1024		
Extra large screen	1024x600	**WXGA (1280x800)** 1024x768 1280x768	1536x1152 1920x1152 1920x1200	2048x1536 2560x1536 2560x1600

Práctica 13

[Java] Repite la práctica **05 – Llamar Walter Anónima [Java]** pero añadiendo una imagen para cada uno de las densidades indicadas en la tabla. Deberás de emular la app en diferentes dispositivos para apreciar los cambios.

Práctica 14

[Kotlin] Repite la práctica **05 – Llamar Walter Anónima [Kotlin]** pero añadiendo una imagen para cada uno de las densidades indicadas en la tabla. Deberás de emular la app en diferentes dispositivos para apreciar los cambios.

Nota. Existen páginas web que, dada una imagen, te generan automáticamente la misma imagen en diferentes densidades.

Configuraciones de pantalla soportadas

Puede ser que nuestra app no pueda mostrarse en ciertas pantallas. Podemos indicarlo mediante el **AndroidManifest.xml** de forma sencilla.

```
<supports-screens android:smallSreens="false"
                  android:normalSreens="true"
                  android:largeSreens="true"
                  android:anyDensity="true" />
```

Poner false fuerza al Android a mostrar las interfaces mediante un modo de compatibilidad que en realidad es un escalado. **No siempre funciona**, pero por lo menos se intenta.

Consejos para realizar interfaces independientes de la resolución

En principio, la regla básica es crear interfaces que se visualicen bien en diferentes clases de pantalla (pequeña, normal, grande) y resoluciones (baja, media, alta). A partir de ahí:

- **Nunca uses** tamaños absolutos basados en el número de pixeles.

- **Usa los sufijos.** Están para algo. Android Studio de automatiza mucho estas cosas.

- **Prueba la app** en el mayor número de dispositivos posible. Para algo tienes emuladores, dos procesadores en paralelo, 64GB de RAM y 10 TB de disco… ¿no?

Internacionalización

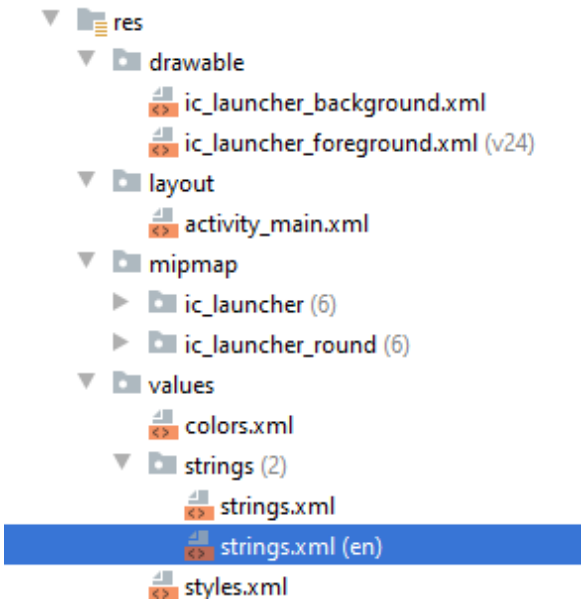
La **Internacionalización** ocurre cuando queremos que una app esté disponible en varios idiomas. Es decir que, si se instala en un móvil en idioma español, salgan los textos en español; y si están en inglés, salgan los textos en inglés. Para lograrlo basta con duplicar el fichero **strings.xml** y poner los textos en un idioma diferente. Android se encarga de escoger el fichero correcto al cargar la app. Veamos cómo se hace:

1) Vamos a la carpeta de Recursos (**res)**, le damos botón derecho y escogemos la opción **New Android Resource Directory**. Crearemos una carpeta con el nombre de values-eng, por ejemplo, para el inglés. Puedes crear tantas como quieras.

2) Le damos botón derecho sobre la carpeta de Recursos (**res)** y escogemos la opción **New Android Resource File.** Tendremos que rellenar las siguientes opciones:

- **Filename**: strings.xml
- **Directory name**: values-b+en (en si es en Inglés)

El resultado será este. Ahora ya sólo nos queda copiar el contenido del fichero original y realizar la traducción.



Es posible **cambiar el Idioma** de la App de forma manual. Para ello basta con crear un método ligado a un evento de un botón (por ejemplo). En el ejemplo que te mostramos, bastaría con pasarle el idioma por parámetro.

```
private void setLang(String lang){
    //la variable lang es al idioma que vamos a cambiar, ejemplo : eu = euskera
    Locale locale = new Locale(lang);
    Locale.setDefault(locale);
    Configuration config = new Configuration();
    config.locale = locale;
    Context context = Settings.this;
    context.getResources().updateConfiguration(config,context.getResources().getDisplayMetrics());
}
```

Cortesía de: **Luis Carlos Salvatierra**

Práctica 15

[Java] Añade Multi-idioma a una app que escojas, así como de la posibilidad de cambiar de un idioma a otro de forma manual.

Práctica 16

[Kotlin] Añade Multi-idioma a una app que escojas, así como de la posibilidad de cambiar de un idioma a otro de forma manual.