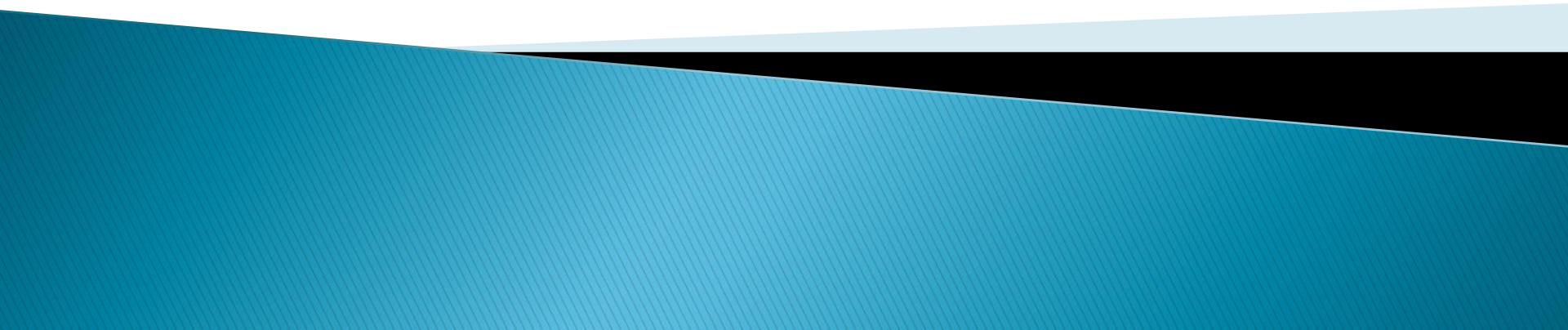


# Android: Layouts

Diseño para distintos dispositivos



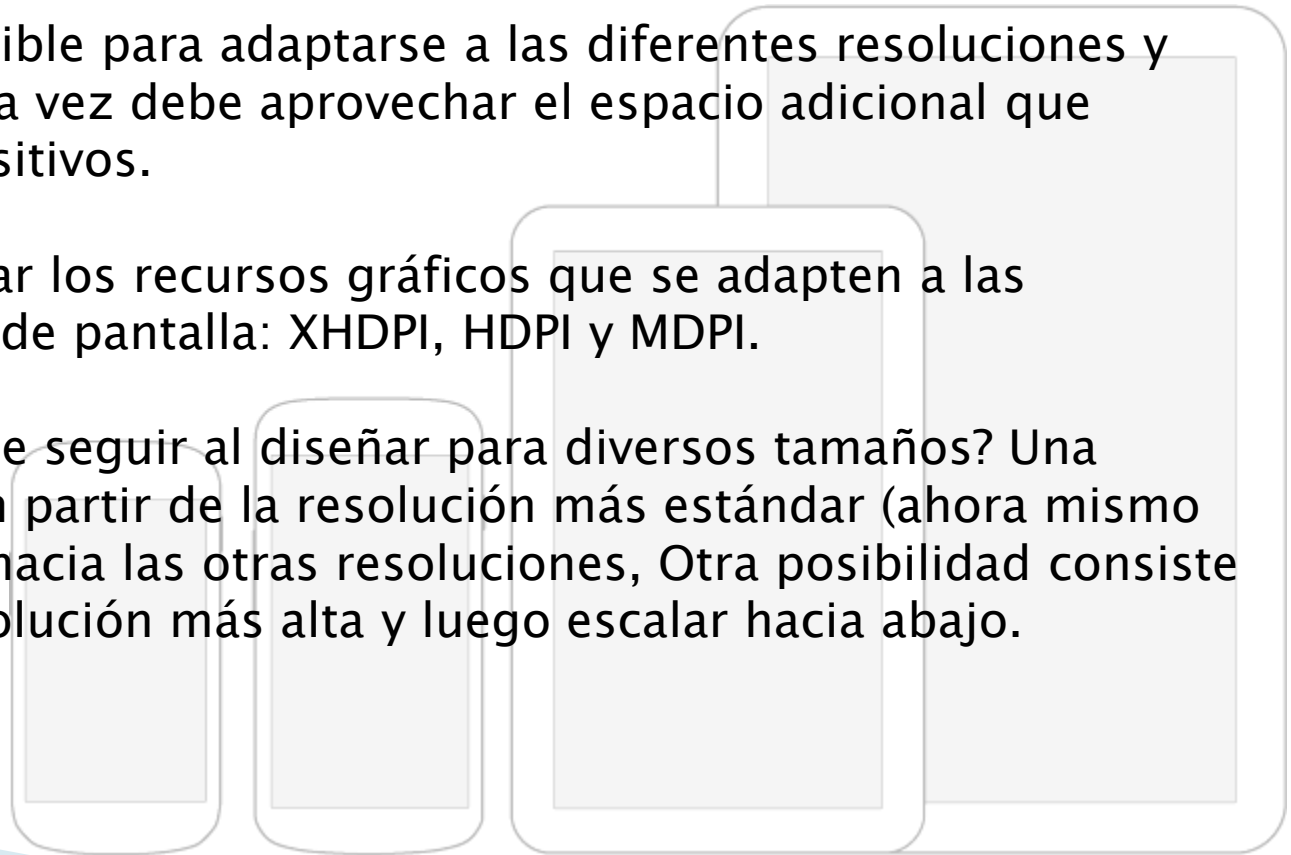
# Cómo diseñar para múltiples dispositivos

Android ofrece una serie de recursos con los que se pretende facilitar el diseño de aplicaciones que se adapten a los diferentes tamaños de pantalla.

El diseño debe ser flexible para adaptarse a las diferentes resoluciones y orientaciones, pero a la vez debe aprovechar el espacio adicional que ofrecen algunos dispositivos.

Se deberá proporcionar los recursos gráficos que se adapten a las diferentes densidades de pantalla: XHDPI, HDPI y MDPI.

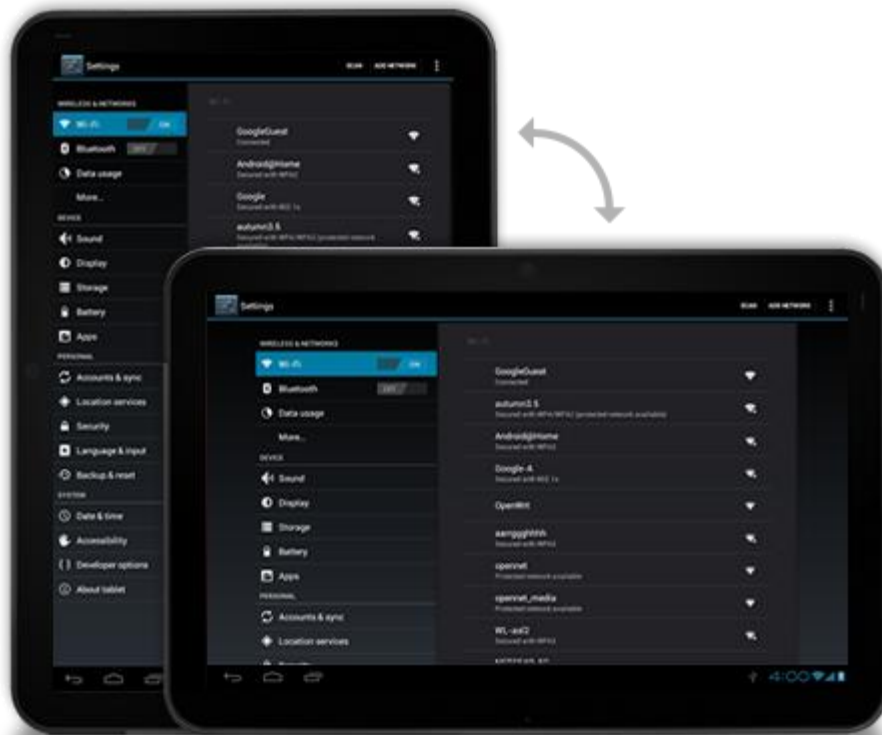
¿Qué estrategia se debe seguir al diseñar para diversos tamaños? Una posibilidad consiste en partir de la resolución más estándar (ahora mismo MDPI) y luego escalar hacia las otras resoluciones, Otra posibilidad consiste en empezar por la resolución más alta y luego escalar hacia abajo.



# Diseños multi panel

- ▶ Los paneles ofrecen la posibilidad de combinar diferentes vistas en un solo layout o en varios dependiendo del espacio.
- ▶ Ante los cambios de orientación también se pueden adoptar diferentes soluciones:
  - Estirar o comprimir el contenido de los paneles.
  - Reorganizar la distribución de los paneles.
  - Expandir o comprimir el contenido de los paneles.
  - Ocultar o mostrar algunos paneles.

# Estirar y comprimir (Stretch/compress)

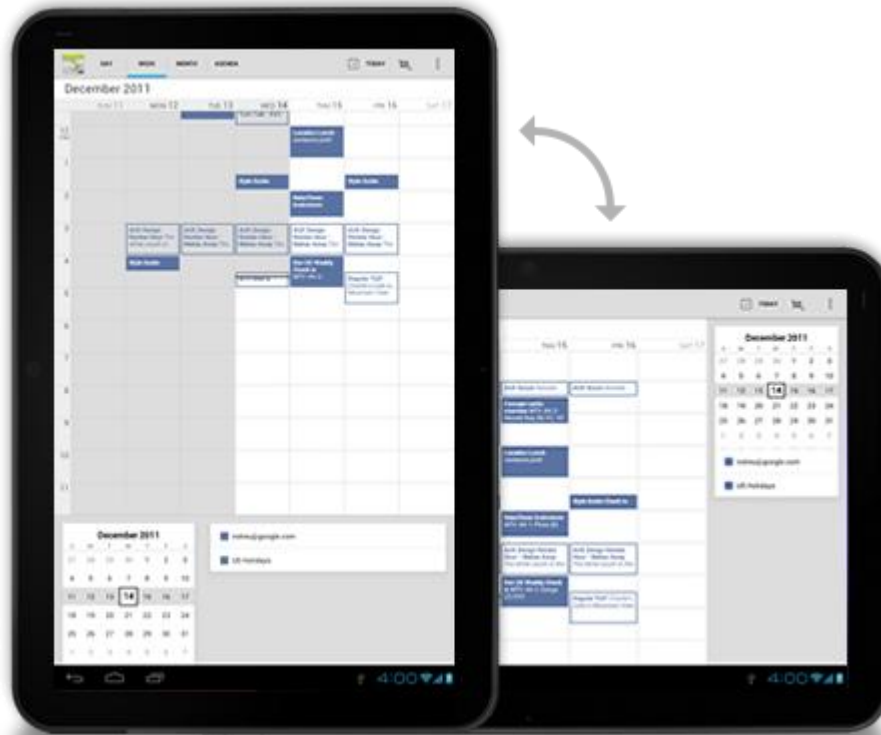


Es preferible indicar las medidas usando valores como **wrap\_content** y **match\_parent** en lugar de medidas concretas.

Los layouts más indicados son **RelativeLayout** y **LinearLayout**.

Es conveniente diseñar para los tamaños: **small**, **normal**, **large** y **xlarge**.

# Reorganizar (stack)



En Android 3.2 se ha añadido el **smallest-width qualifier**, es decir, un indicador de medida mínima:

`layout-sw600dp/main.xml`

También se pueden definir alias para facilitar la forma de nombrar los diferentes diseños.

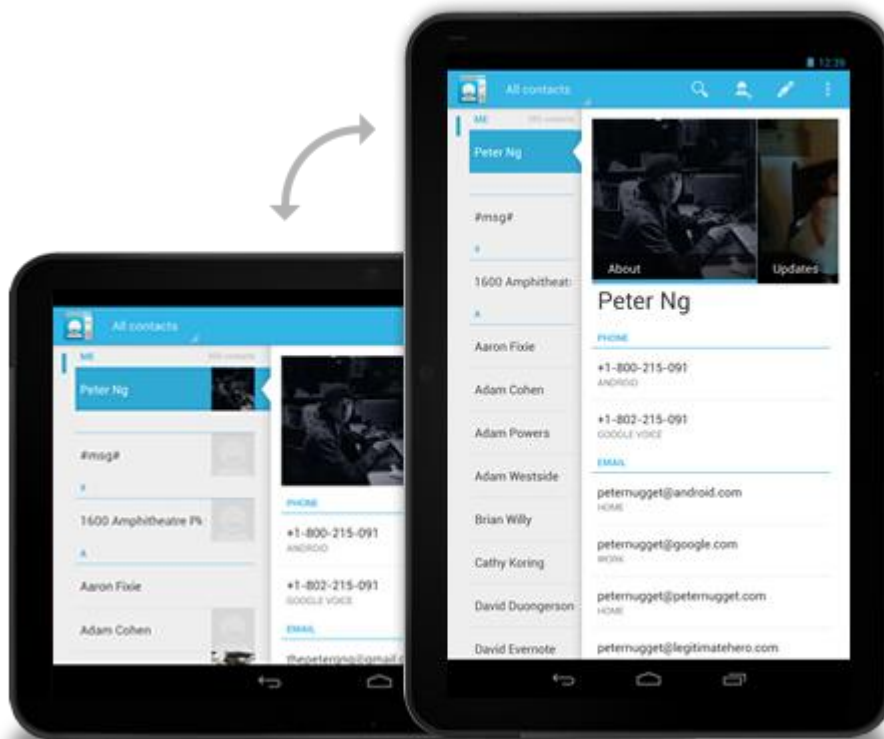
Es conveniente diseñar para ambas orientaciones: **landscape** y **portrait**.

# Expandir y comprimir (expand/collapse)

Es aconsejable usar imágenes **nine-patch**.

Este tipo de imágenes se dividen en 9 secciones: los cuatro bordes que se escalan sobre un eje, las cuatro esquinas que no se escalan y la parte central que se escala sobre ambos ejes.

Las dimensiones se deben especificar en dp (vistas) y sp (fuentes).



# Ocultar y mostrar (show/hide)

Las imágenes se deben escalar para los diferentes tamaños:

xhdpi: 2.0

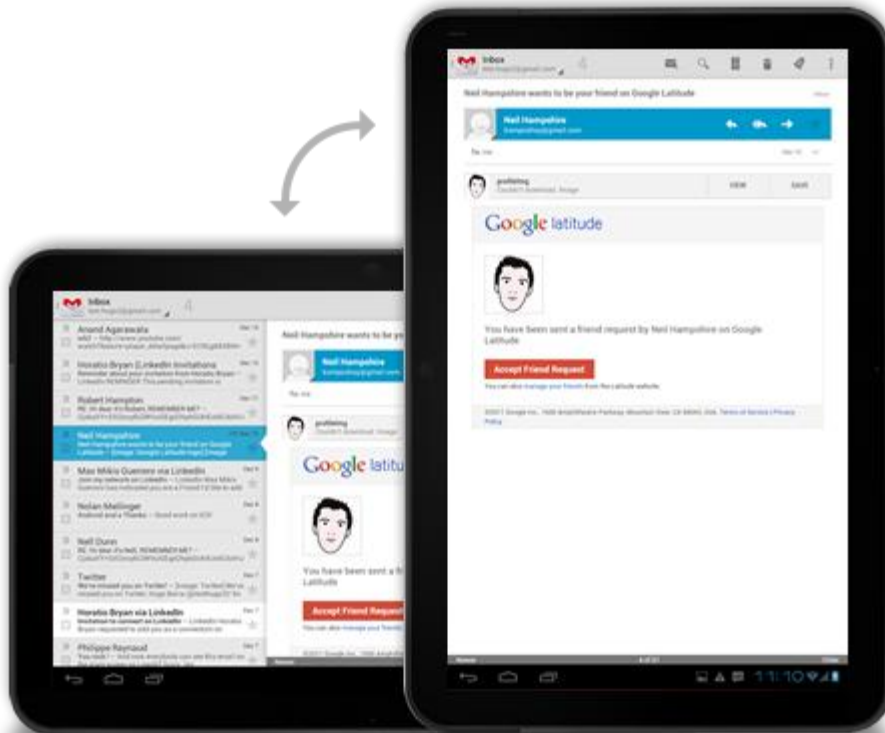
hdpi: 1.5

mdpi: 1.0 (baseline)

ldpi: 0.75

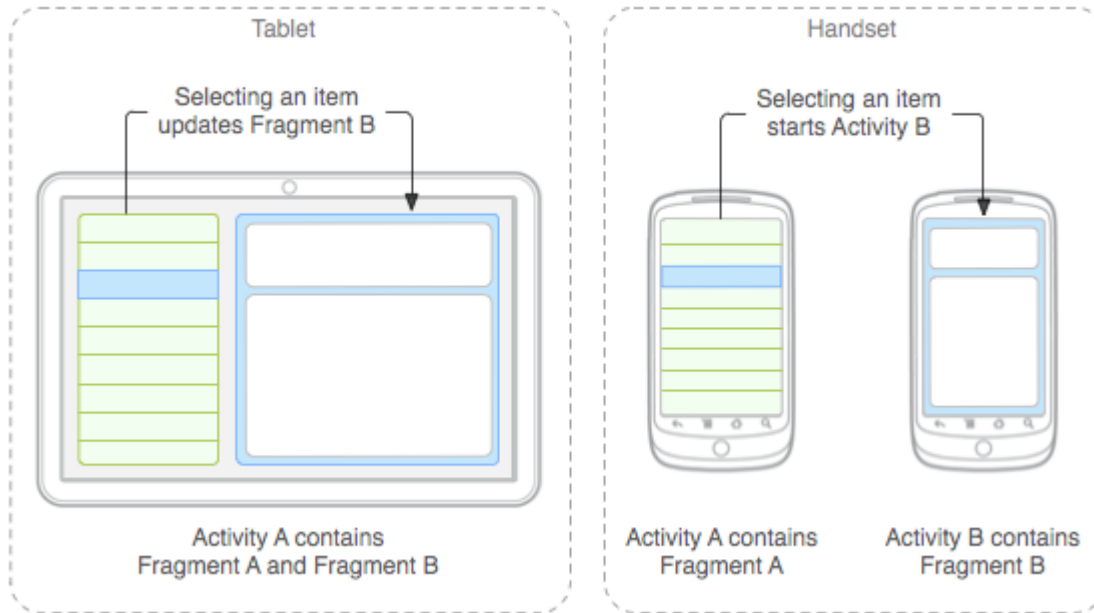
Desde el código se deberá poder detectar cuál es el diseño actual, detectar cambios de configuración y cargar los elementos variables.

Uno de los elementos más utilizados son los fragmentos, que se pueden reutilizar.





# Diseños mono y multipanel



Este tipo de diseños se basan en recursos como:

- fragmentos
- barra de acción



# Enlaces

- ▶ <http://developer.android.com/design/style/devices-displays.html>
  - ▶ <http://developer.android.com/design/patterns/multi-pane-layouts.html>
  - ▶ <http://developer.android.com/training/basics/fragments/index.html>
  - ▶ <http://developer.android.com/training/multiscreen/index.html>
  - ▶ <http://developer.android.com/guide/practices/tablets-and-handsets.html>
- 