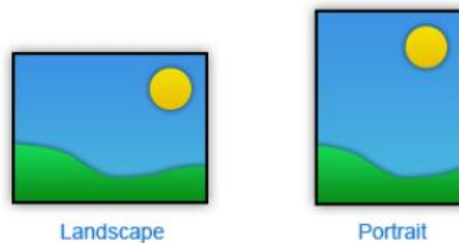


Tamaño y Orientación en Android.

Ya en otras oportunidades hemos tocado el tema de densidades en android, allí se mencionó que existen diferentes dispositivos con este sistema operativo y por esa razón es importante garantizar que nuestras aplicaciones se visualicen de la mejor manera en cada uno de ellos.

Adicional a esto por lo regular los dispositivos cuentan con 2 orientaciones de pantalla, estas son:

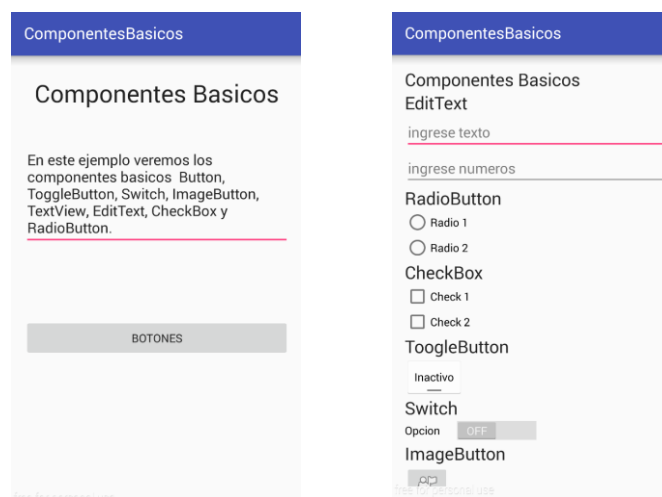
- **Portrait:** Representa el modo retrato cuando nuestros dispositivos se encuentran de forma vertical.
- **Landscape:** Representa el modo paisaje, cuando nuestros dispositivos se encuentran de forma horizontal.



En este ejemplo veremos cómo trabajar con las dos orientaciones ya que es de suma importancia que independientemente de en qué orientación nos encontremos la aplicación debe funcionar igual y brindar una muy buena experiencia de usuario, adicional a esto veremos también como estructurar nuestro proyecto para que también pueda ser visualizado dependiendo de la densidad del dispositivo.

Sin embargo antes de trabajar con las 2 orientaciones también debemos saber que se puede bloquear el uso de una u otra, esto nos permitirá definir por defecto una única orientación ajustándose a nuestras necesidades.

Para el desarrollo de esta actividad, trabajaremos con la aplicación realizada sobre los componentes Básicos.



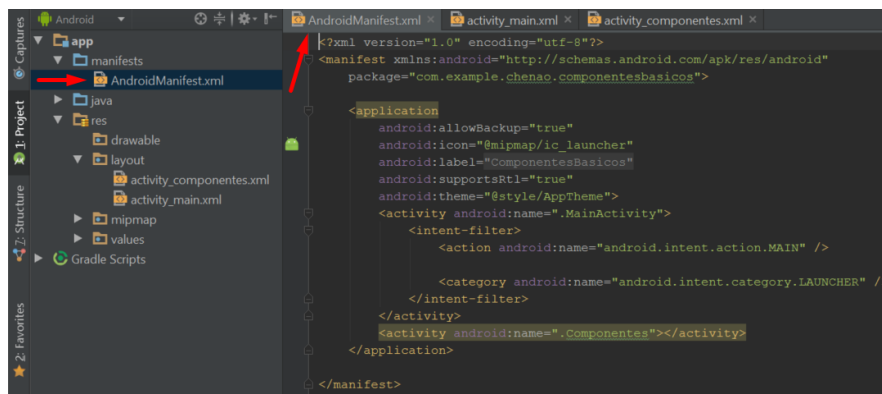
Bloquear una Orientación.

Básicamente lo que queremos aquí es evitar que al girar nuestro dispositivo la pantalla también se gire, esto depende mucho del enfoque de nuestra aplicación.

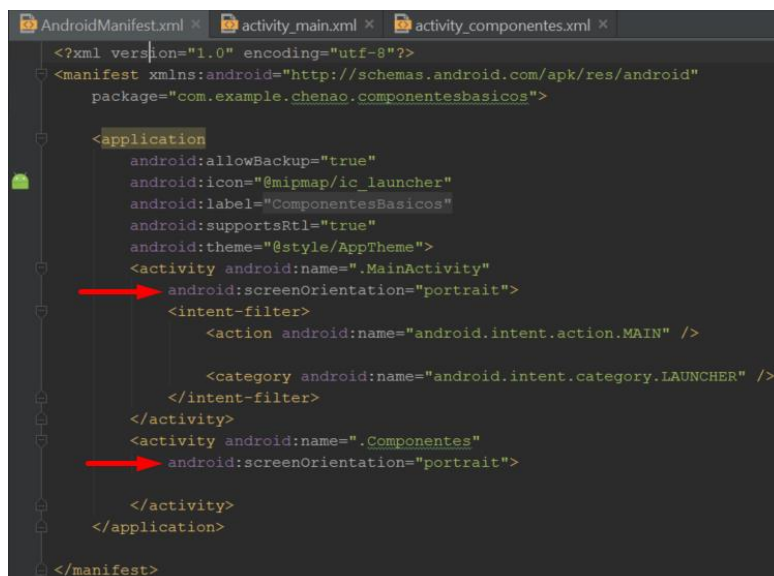
De entrada no deberíamos restringir al usuario a solo una orientación, pues tal vez a los usuarios les guste más una que otra y si lo hacemos puede que estemos obligando al usuario a trabajar con la orientación que no le gusta, estando muy ligado a el tema de UX (Experiencia de usuario), sin embargo hay ocasiones donde es importante, si nuestra aplicación es de juego por ejemplo donde es importante el uso de la pantalla modo landscape.

Archivo Manifest

El **manifest** es un archivo xml donde definimos las configuraciones básicas de nuestra aplicación, aquí se definen las actividades creadas, los permisos de nuestra aplicación entre otras configuraciones. Para definir la orientación por defecto, abrimos el archivo que se encuentra en la raíz del proyecto.



Aquí tan solo debemos agregar la propiedad **screenOrientation="portrait"** en cada una de las activity que queremos bloquear, en este caso solo se podrá visualizar la aplicación en modo portrait, si ejecutamos la aplicación, podemos ver su funcionamiento.

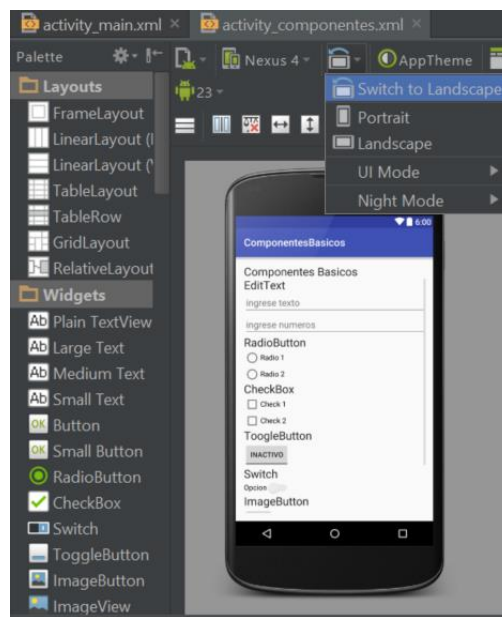


Este proceso se debe hacer solo cuando queremos definir una orientación por defecto, sino entonces no modificamos el manifest y podemos trabajar con las dos orientaciones como lo veremos a continuación.

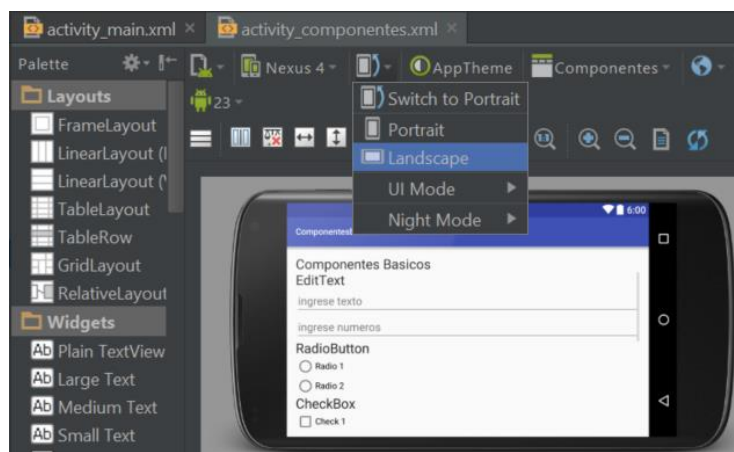
Portrait y Landscape.

Por defecto las aplicaciones se encuentran en modo portrait, si trabajamos en android studio podemos probar como se vería nuestra aplicación en alguna de las dos orientaciones, para esto desde la vista de diseño usamos la opción que nos permite realizar el cambio.

Por defecto nos encontramos en portrait.



Pero al cambiar la opción podemos ver como se muestra la aplicación en Landscape.



Como vemos la interfaz de nuestra aplicación cambia, y dependiendo de las propiedades internas los componentes se adaptan o no al tamaño de la pantalla, como es el caso de los EditText ya que ambos fueron creados con la propiedad

```
android:layout_width="match_parent"
```

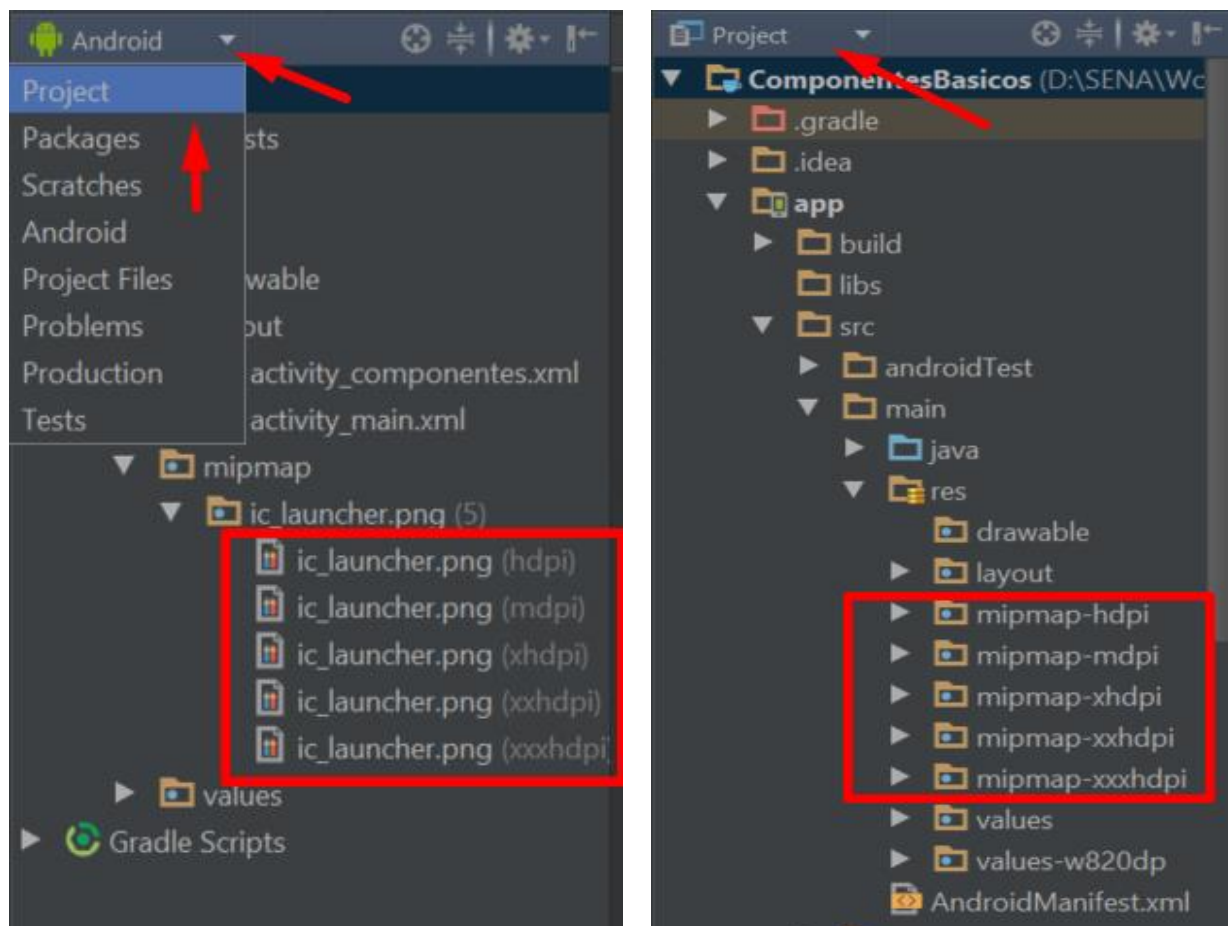
Lo que permite que el componente se adapte a lo ancho a la pantalla, sin embargo esto no es suficiente y se hace necesario poder organizar nuestra aplicación para esta orientación.

Definiendo el modo Landscape.

Como inicialmente creamos la aplicación para el modo portrait ahora es necesario optimizarla para el modo paisaje, el proceso es muy simple, tan solo debemos crear un nuevo directorio y copiar allí los layout de nuestra aplicación.

Si nuestro proyecto está en la vista **Project** podemos verificar que para el directorio **mipmap** se encuentran otros directorios con el mismo nombre pero diferente sufijo, esto es porque como se mencionó en la guía sobre imágenes y densidades, android identifica dependiendo de la densidad del dispositivo que icono tomar de que carpeta.

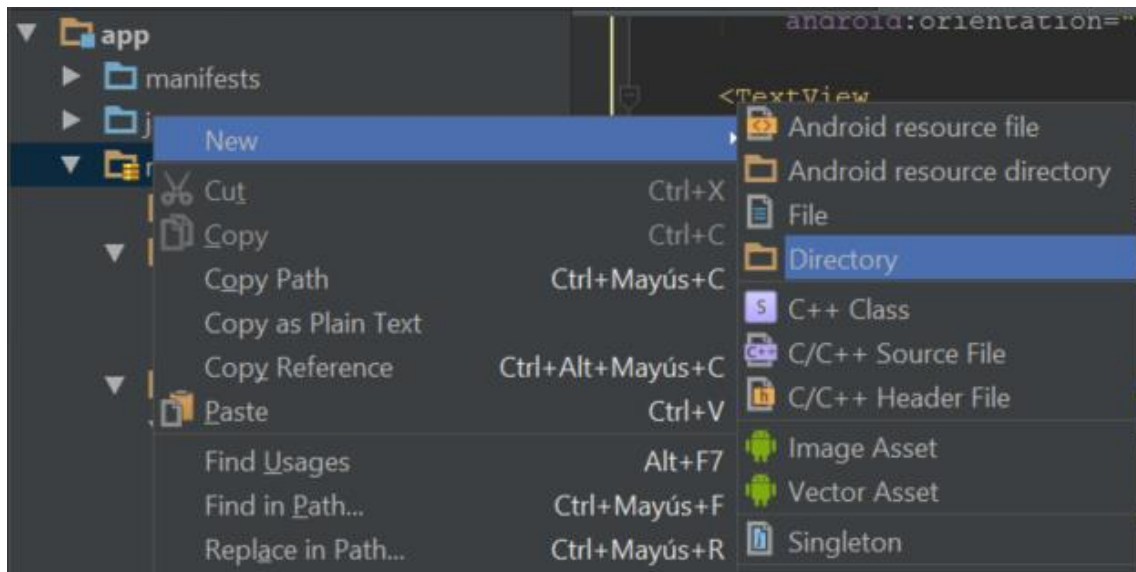
En las siguientes imágenes vemos como se muestra inicialmente en la vista Android y luego al cambiar en la vista Project.



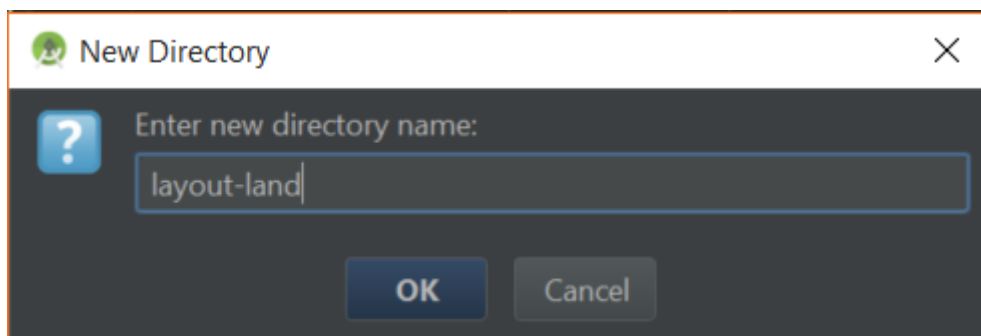
Este proceso es similar a lo que se habló con el uso de imágenes y el directorio **drawable**, lo normal es crear un directorio drawable para cada tipo de densidad (hdpi,mdpi,xdpi,xxdpi,xxxdpi).

Para el caso de las orientaciones se realiza del mismo modo, crearemos otro directorio layout pero con el sufijo land.

Hacemos clic derecho sobre el directorio res y creamos la nueva carpeta



Asignamos el nombre al directorio



Es importante que nos encontremos en la vista **Project** para visualizar el directorio creado, posteriormente tan solo copiaremos los layout de nuestra aplicación en el nuevo directorio.

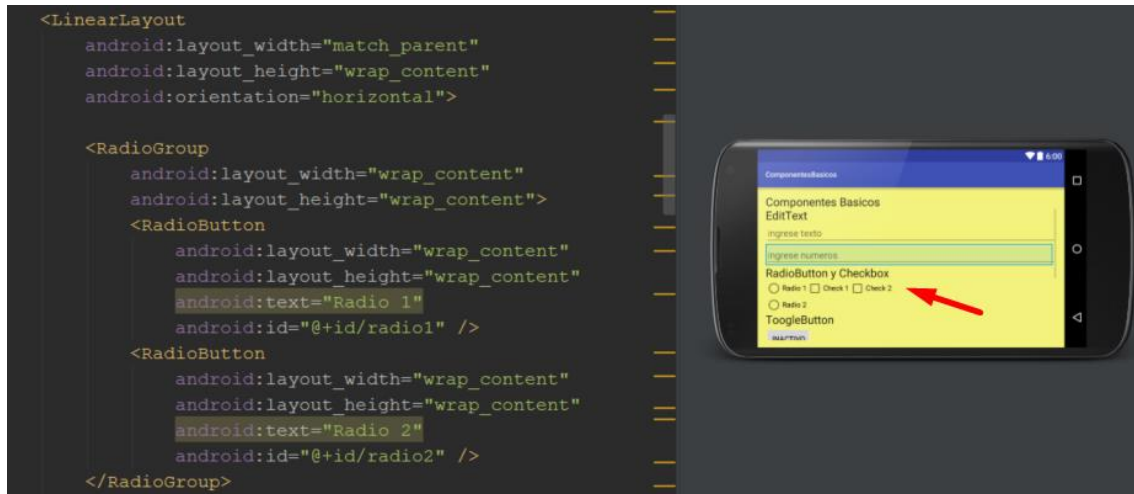


De esa manera ya tenemos listos nuestros layout para las 2 orientaciones, desde aquí podemos empezar a organizarlas para verificar su funcionamiento.

Para esto podemos agregar nuevos componentes, layout y propiedades en general para que se muestre en una vista o en otra.

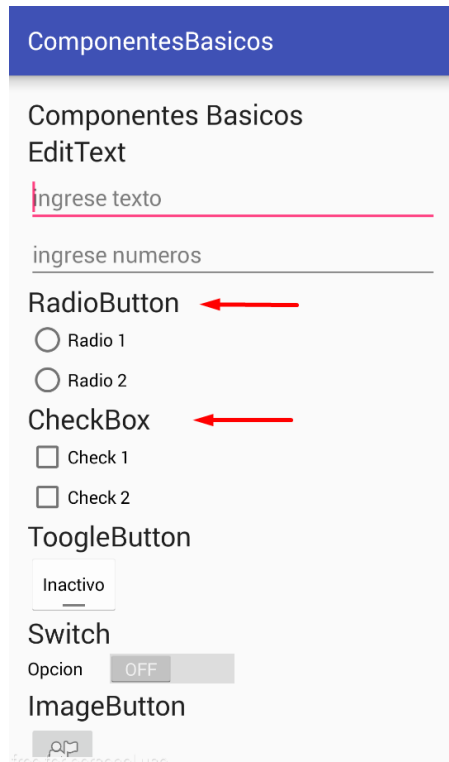
Algo importante a tener en cuenta es que previamente la aplicación fue creada con un ScrollView, si verificamos el funcionamiento en modo landscape nos daremos cuenta que podemos acceder a los otros componentes gracias a esta barra de desplazamiento.

Para este ejemplo modificamos el layout, agregando un color de fondo y uniendo los componentes RadioButton y CheckBox por medio de un LinearLayout

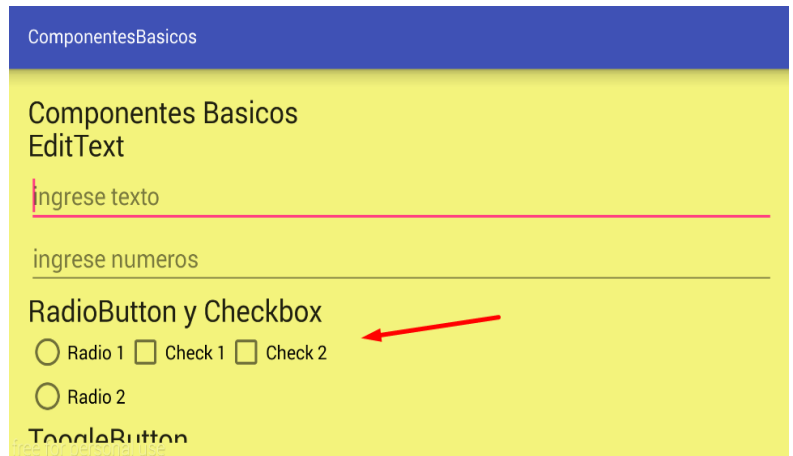


Si ejecutamos la aplicación podemos ver su funcionamiento en cualquiera de las orientaciones.

PORTRAIT



LANDSCAPE



Y de esta manera ya tenemos nuestra aplicación optimizada para ambas vistas.

Soporte para múltiples pantallas.

Android brinda soporte para múltiples pantallas, en los ejemplos que estamos trabajando solo nos enfocamos en la densidad de pantalla dependiendo del dispositivo al que queramos llegar, pero es importante poder cubrir la mayor cantidad de dispositivos, por esta razón se ha hablado de las densidades, imágenes, orientación y ahora tamaño de pantallas, buscando siempre la mejor experiencia de usuario.

Desde la web oficial de android podemos encontrar un apartado sobre esto en http://developer.android.com/intl/es/guide/practices/screens_support.html allí se encuentra una sección donde se habla de la importancia de este soporte y a la vez nos muestra la manera de realizarlo en nuestras aplicaciones.

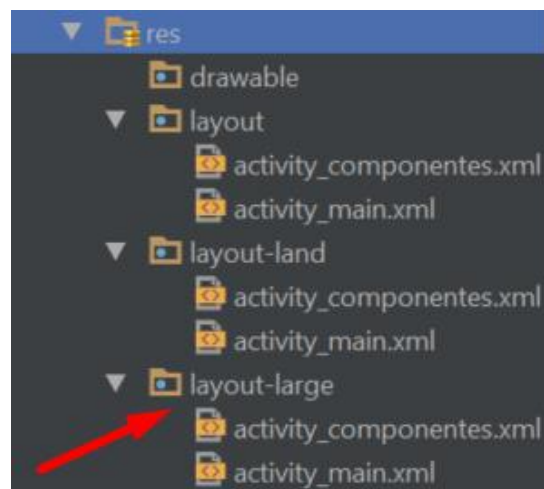
```
res/layout/my_layout.xml // layout for normal screen size ("default")
res/layout-large/my_layout.xml // layout for large screen size
res/layout-xlarge/my_layout.xml // layout for extra-large screen size
res/layout-xlarge-land/my_layout.xml // layout for extra-large in landscape orientation

res/drawable-mdpi/graphic.png // bitmap for medium-density
res/drawable-hdpi/graphic.png // bitmap for high-density
res/drawable-xhdpi/graphic.png // bitmap for extra-high-density
res/drawable-xxhdpi/graphic.png // bitmap for extra-extra-high-density

res/mipmap-mdpi/my_icon.png // launcher icon for medium-density
res/mipmap-hdpi/my_icon.png // launcher icon for high-density
res/mipmap-xhdpi/my_icon.png // launcher icon for extra-high-density
res/mipmap-xxhdpi/my_icon.png // launcher icon for extra-extra-high-density
res/mipmap-xxxhdpi/my_icon.png // launcher icon for extra-extra-extra-high-density
```

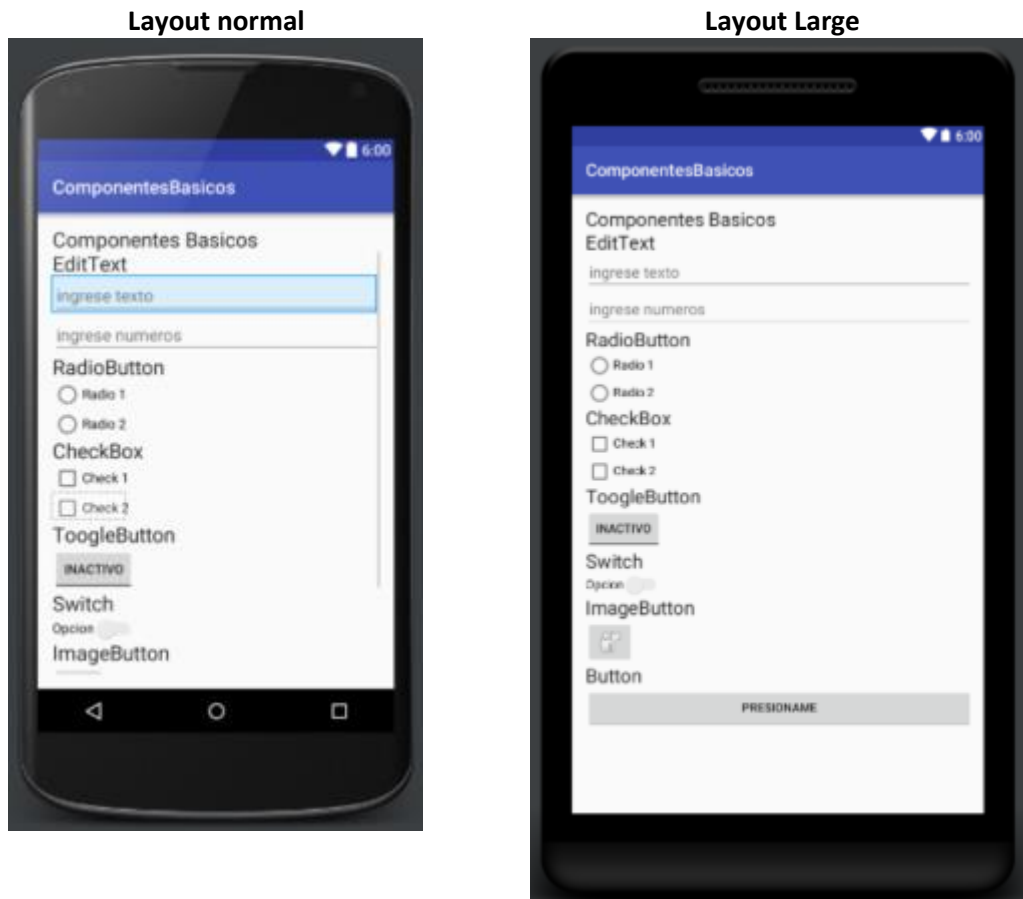
Como se puede ver, se indica la manera de dar el soporte por medio de la creación de directorios con sufijos, en estas guías hasta el momento se ha hablado de drawable y mipmap pero como ya vimos también se puede hacer con el directorio layout, no solo para definir la orientación sino también para definir la densidad de pantalla a la que queremos llegar.

Como ejemplo daremos soporte para pantallas grandes usando el sufijo **large** de la misma manera como dimos soporte a landscape.



Creamos el directorio “layout-large” tal como se muestra en la página oficial de android y allí pegamos los layout de nuestro sistema para poder adaptarlos a la nueva densidad de pantalla.

Si revisamos los layout vemos como automáticamente la interfaz se actualiza dependiendo de la ubicación en la que se encuentre, en large el dispositivo donde se muestra posee una densidad mayor, lo que se puede evidenciar al comparar la densidad original con la nueva, pudiendo ver todos los elementos de nuestra interfaz.



Ya el proceso de cómo queremos que se muestren nuestros componentes es el mismo para cada resolución pues la idea es adaptar u organizar los componentes en las diferentes pantallas.