**Diseño de app con android studio**

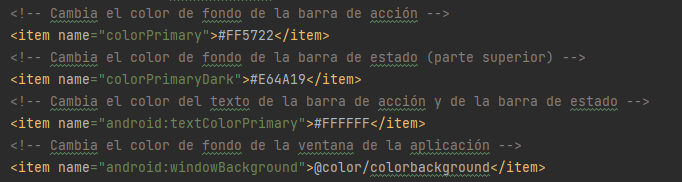
1. Editar el archivo de estilo de la aplicación, este se encuentra en la carpeta res🡪values🡪themes, cambiar el estilo

**Theme.Material3.DayNight.NoActionBar**

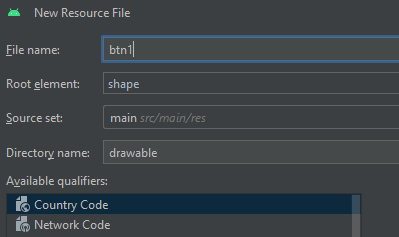
Por

**Theme.AppCompat.Light.DarkActionBar**

1. Adicionar los siguientes estilos para dar una mejor presentación



1. Probamos los cambios realizados
2. Continuando con el diseño creamos un archivo “**Drawable Resource file**” dando click derecho en la carpeta drawable, lo llamaremos **btn1**y Root element**: shape**



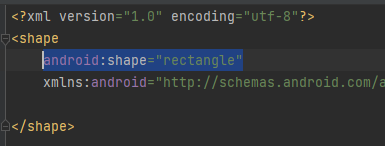
Dentro del archivo se le adicona **android:shape="rectangle"**

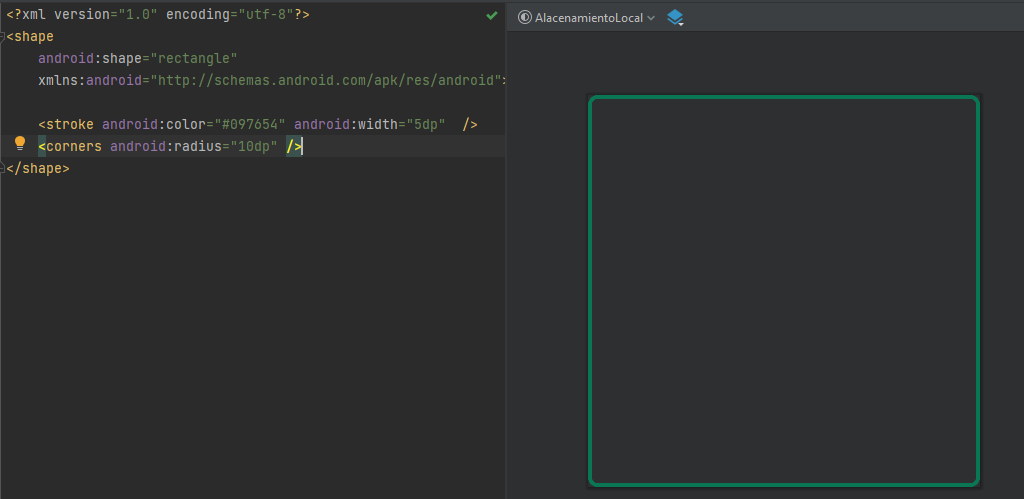
**android:shape** se utiliza para definir la forma de un elemento gráfico en un archivo XML de recursos, como un archivo drawable

Cuando estableces android:shape="rectangle", estás indicando que el elemento debe tener la forma de un rectángulo.

**<stroke> :** **Permite configurar el borde**. Mediante los atributos width y color podemos indicar la anchura y color, y con dashWidth y dashGap podemos crear una línea punteada.

**<corners> :** Permite crear rectángulos con bordes redondeados. <padding> : Define el espaciado interior, desde el borde hasta el contenido

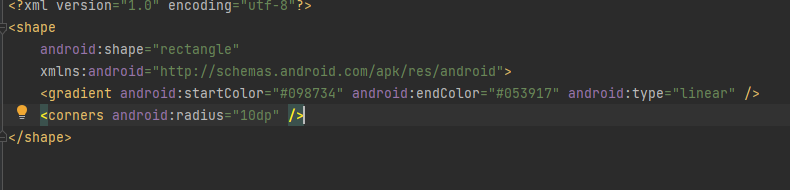




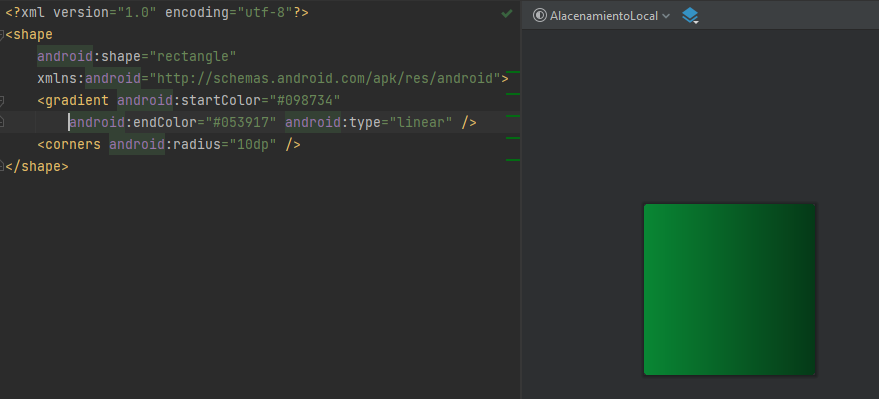
Para adicionar el estilo al botón basta con ir al archivo xml de la vista y adicioanar la propiedad

android:background="@drawable/btn1"

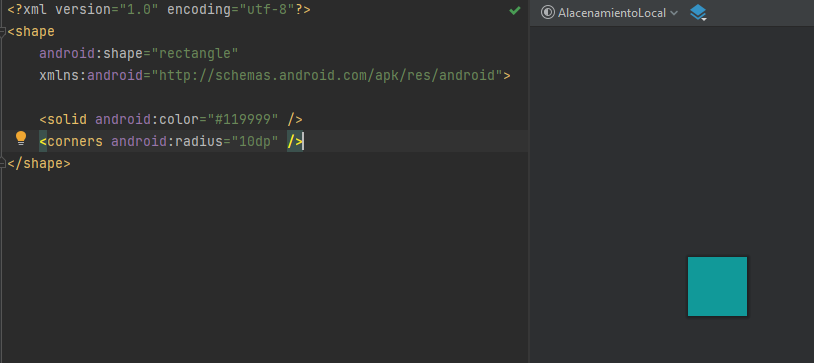
De la misma forma que en el ejercicio anterior se crea un Resource llamado btn2, con la siguiente estructura interna



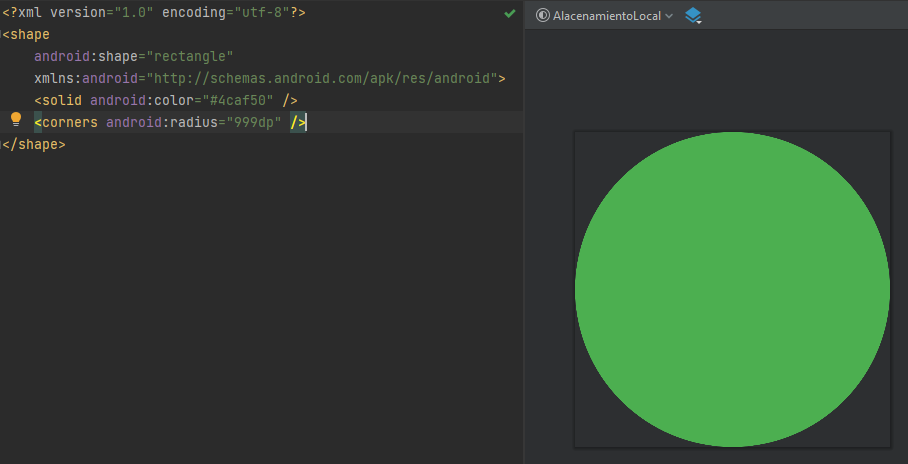
Y se adiciona al siguiente botón.



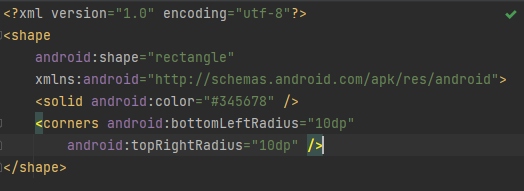
Continuando con los estilos para los botones se crea un tercer Resource llamada btn3



Cuarto archivo xml Resource btn4



Un quinto ejemplo



Como siguiente medida se va a trabajar con elemento llamado **TextInputLayout,** internamente tiene un **TextInputEditText** que es donde se escribe, este campo es mas adecuado para el registro de formularios, login, entre otro. Este campo permite adicionar iconos al inicio, al final, máximo de caracteres, leyenda, adiconal a esto permite que el elemento hint a la hora de colocar el cursor en el campo el contenido se suba realizando un estilo agradable.

Este elemento en su **TextInputLayout** se le puede adicionar:

**IconDrawabe** 🡪 icono

**hint** 🡪 campo de información

**helperText 🡪** text de ayuda en la parte inferior del campo

**counterMaxLength 🡪** Contador de longitud, para que funcione debe tener **counterEnabled**

en true

**boxBackgroundMode 🡪 outline** permite que el texto hint salga

style="@style/MyTextInputLayoutStyle"  
android:hint="Agregue un nombre"  
app:boxBackgroundMode="outline"  
app:counterEnabled="true"  
app:counterMaxLength="20"  
app:helperText="Requerido"  
app:startIconDrawable="@drawable/correo"

Si el campo anterior va a ser para contraseña, se le pude adicionar un icono que muestra la misma

app:endIconMode="password\_toggle"

**TextInputEditText 🡪** permite que se le adicione texto, números direcciones al widget anterior.

**inputType** 🡪 para los tipos de datos

**style 🡪** permite adicionar color al texto con el atributo **android:textColor**