

Tarea 3: agregar recursos de color

Hasta ahora ha aprendido a cambiar los valores de las propiedades.

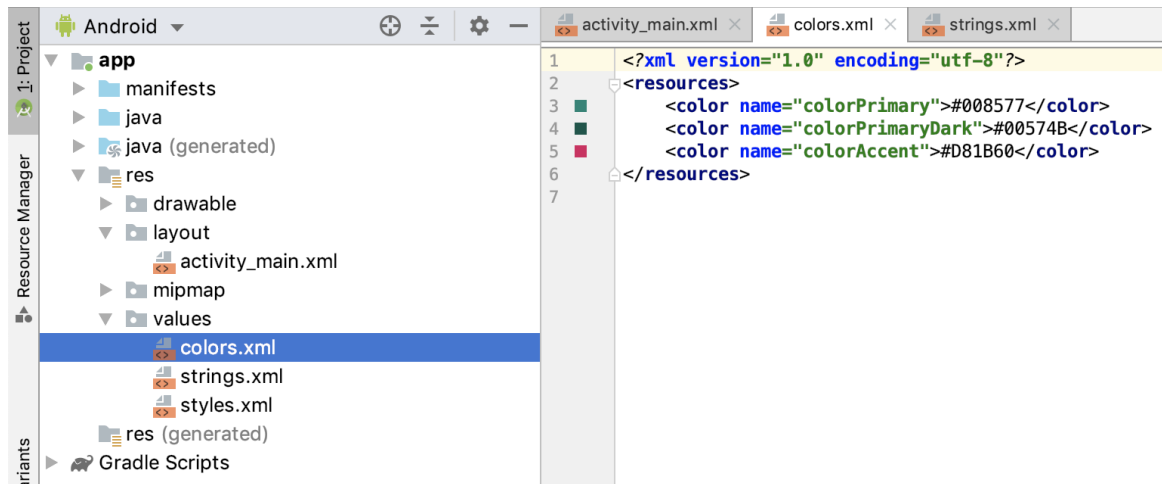
A continuación, aprenderá a crear más recursos como los recursos de cadena con los que trabajó anteriormente.

El uso de recursos le permite usar los mismos valores en varios lugares, o definir valores y hacer que la interfaz de usuario se actualice automáticamente cada vez que se cambie el valor.

Paso 1: Agregar recursos de color

Primero, aprenderá a agregar nuevos recursos de color.

1. En el panel Proyecto de la izquierda, haga doble clic en **res > valores > colores.xml** para abrir el archivo de recursos de color.



```
<resources>
  <color name="colorPrimary">#6200EE</color>
  <color name="colorPrimaryDark">#3700B3</color>
  <color name="colorAccent">#03DAC5</color>
</resources>
```

El archivo **colors.xml** se abre en el editor. Hasta ahora, se han definido tres colores. Estos son los colores que puede ver en el diseño de su aplicación, por ejemplo, púrpura para la barra de la aplicación.

Nota: las diferentes versiones de Android Studio usan diferentes valores para estos colores, por lo que es posible que vea otros colores aquí.

2. Regrese a **fragment_first.xml** para que pueda ver el código XML para el diseño.

3. Agregue una nueva propiedad a TextView llamada **android:background** y comience a escribir para establecer su valor en **@color**. Puede agregar esta propiedad en cualquier lugar dentro del código TextView.

Aparece un menú que ofrece los recursos de color predefinidos:

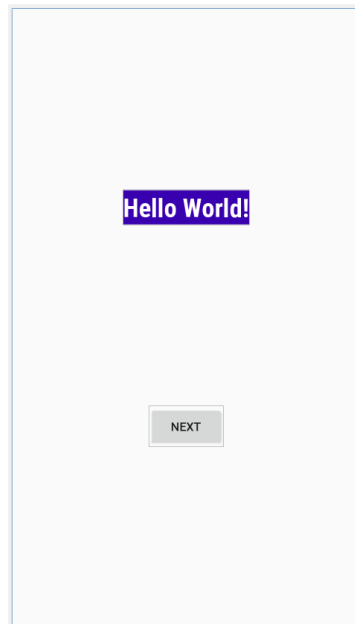
```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:fontFamily="sans-serif-condensed"
  android:text="@string/hello_world_text"
  android:background="@c
  android:textAlignme @color/colorPrimaryDark
  android:textColor="@ @color/colorPrimary
  android:textSize="3 @color/colorAccent
```

4. Elija `@color/colorPrimaryDark`.

5. Cambie la propiedad `android:textColor` y coloque el valor `@android:color/white`.

El marco de trabajo de Android define una gama de colores, incluido el blanco, por lo que no tiene que definir el blanco usted mismo.

6. En el editor de diseño, puede ver que TextView ahora tiene un fondo azul oscuro y el texto se muestra en blanco.



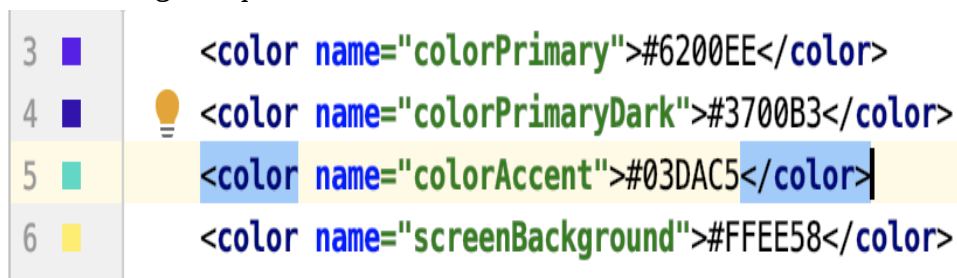
Paso 2: agregue un nuevo color para usar como color de fondo de la pantalla

1. De vuelta en `colors.xml`, cree un nuevo recurso de color llamado `screenBackground`:

```
<color name="fondo de pantalla">#FFEE58</color>
```

Un color se puede definir como 3 números hexadecimales (`#00-#FF` o `0-255`) que representan los componentes rojo, azul y verde (RGB).

El color que acabas de agregar es amarillo. Observe que los colores correspondientes al código se muestran en el margen izquierdo del editor.



Tenga en cuenta que también se puede definir un color que incluya un **valor alfa** (`#00-#FF`) que representa la transparencia (`#00 = 0 % = totalmente transparente`, `#FF = 100 % = totalmente opaco`).

Cuando se incluye, el valor alfa es el primero de 4 números hexadecimales (ARGB).

El valor alfa es una medida de transparencia.

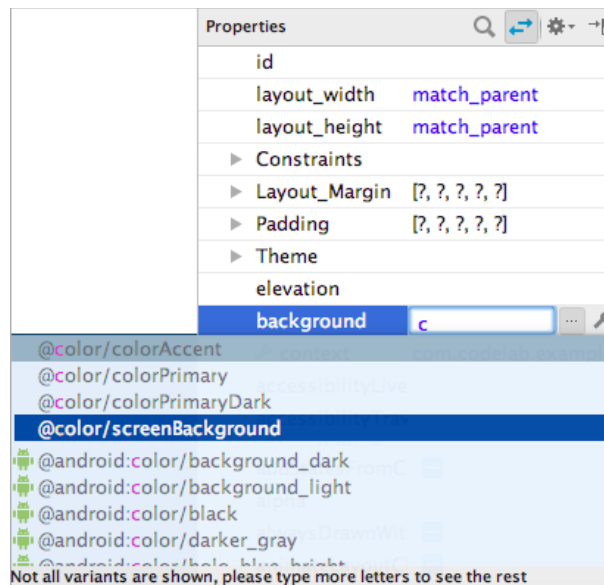
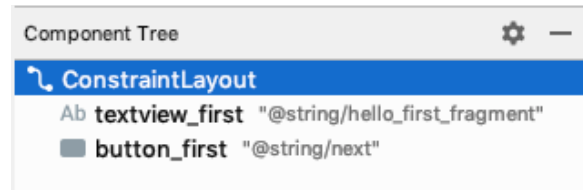
Por ejemplo, `#88FFEE58` hace que el color sea semitransparente, y si usa `#00FFEE58`, es completamente transparente y desaparece de la barra de la izquierda.

2. Vuelva a **fragment_first.xml**.

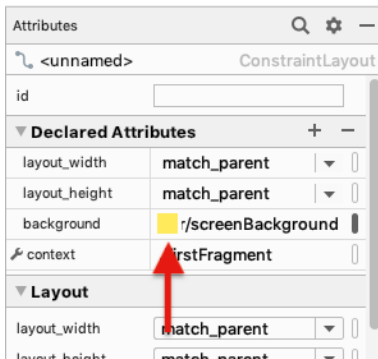
3. En el árbol de componentes, seleccione **ConstraintLayout**.

4. En el panel Atributos, seleccione la propiedad de fondo y presione Entrar. Escriba "c" en el campo que aparece.

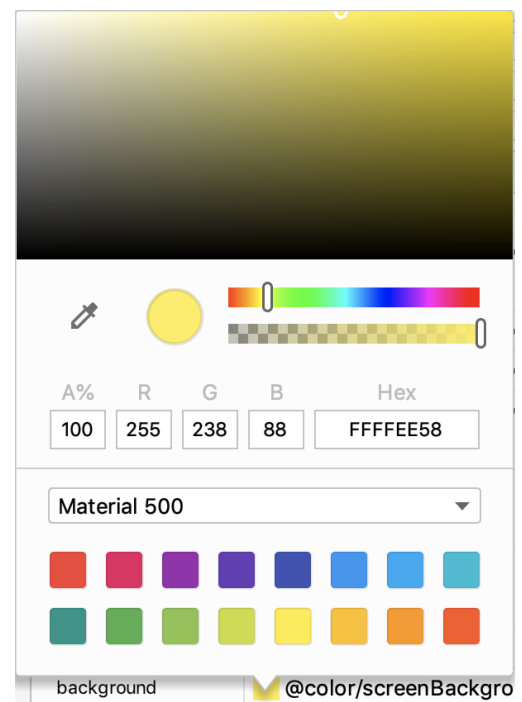
5. En el menú de colores que aparece, selecciona **@color/screenBackground**. Presione Entrar para completar la selección.



6. Haga clic en el parche amarillo a la izquierda del valor del color en el campo de fondo.



Muestra una lista de colores definidos en **colors.xml**. Haga clic en la pestaña **Personalizado (Custom)** para elegir un color personalizado con un selector de color interactivo.

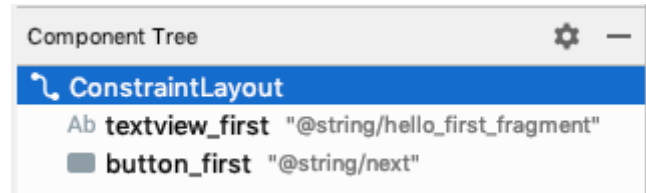


7. No dude en cambiar el valor del **color de fondo de la pantalla**, pero asegúrese de que el color final sea notablemente diferente de los colores **colorPrimary** y **colorPrimaryDark**.

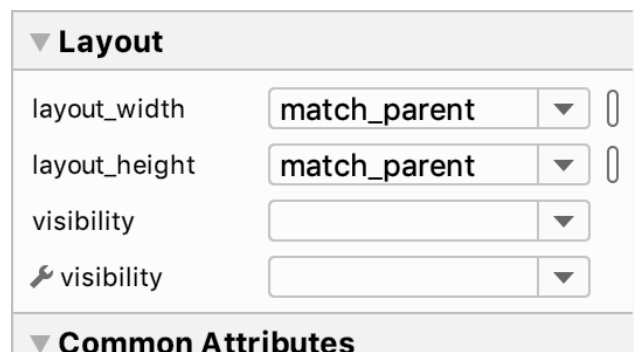
Paso 3: Explore las propiedades de ancho y alto

Ahora que tiene un nuevo color de fondo de pantalla, lo utilizará para explorar los efectos de cambiar las propiedades de ancho y alto de las vistas.

1. En **fragment_first.xml**, en el Árbol de componentes, seleccione **ConstraintLayout**.



2. En el panel Atributos, busque y expanda la sección Diseño.

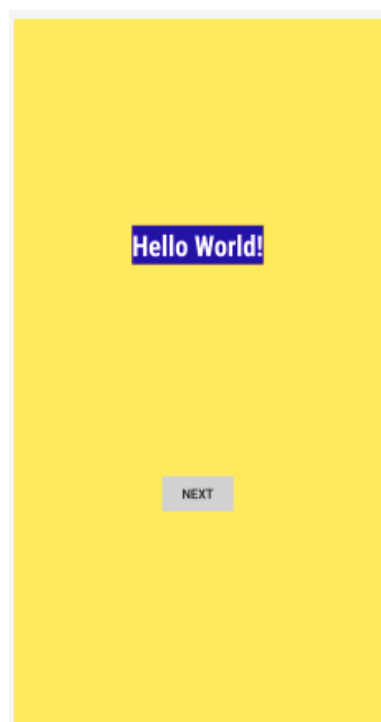


Las propiedades **layout_width** y **layout_height** se establecen en **match_parent**.

El **ConstraintLayout** es la vista raíz de este Fragmento, por lo que el tamaño del diseño "**principal**" es efectivamente el tamaño de su pantalla.

Sugerencia: todas las vistas deben tener las propiedades **layout_width** y **layout_height**.

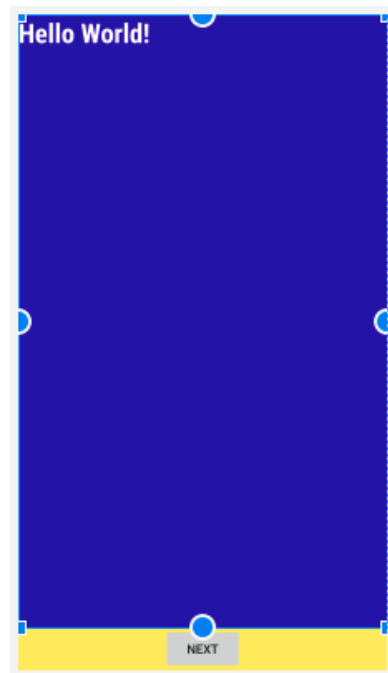
3. Observe que todo el fondo de la pantalla utiliza el color **screenBackground**.



4. Seleccione **textview_first**. Actualmente, el ancho y la altura del diseño son **wrap_content**, lo que le dice a la vista que sea lo suficientemente grande como para encerrar su contenido (más el relleno)
5. Cambie tanto el ancho como la altura del diseño a **match_constraint**, lo que le indica a la vista que sea tan grande como lo que esté restringido.

El ancho y el alto muestran **0dp**, y el texto se mueve hacia la esquina superior izquierda, mientras que **TextView** se expande para coincidir con **ConstraintLayout** excepto por el botón.

El botón y la vista de texto están al mismo nivel en la jerarquía de vistas dentro del diseño de restricciones, por lo que comparten espacio.



6. Explore lo que sucede si el ancho es **match_constraint** y el alto es **wrap_content** y viceversa. También puede cambiar el ancho y el alto de **button_first**.
7. Establezca tanto el ancho como el alto de **TextView** y el **botón** nuevamente en **wrap_content**.