

Tarea 2: explorar el editor de diseño

Generalmente, cada pantalla de su aplicación de Android está asociada con uno o más fragmentos. La pantalla única que muestra "**Hola, primer fragmento**" se crea con un fragmento, llamado **FirstFragment**. Esto se generó para usted cuando creó su nuevo proyecto.

Cada fragmento visible en una aplicación de Android tiene un diseño que define la interfaz de usuario del fragmento. Android Studio tiene un editor de diseño donde puede crear y definir diseños.

Los diseños se definen en XML. El editor de diseño le permite definir y modificar su diseño codificando XML o usando el editor visual interactivo.

Cada elemento en un diseño es una vista.

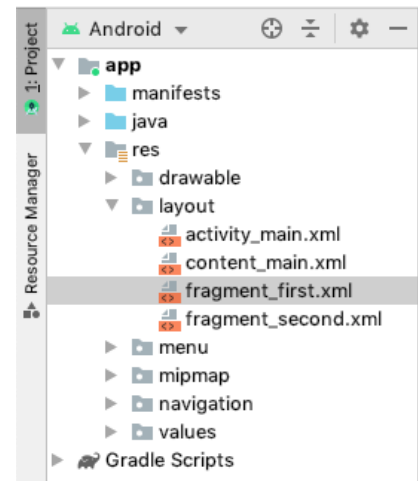
Paso 1: abre el editor de diseño

1. Busque y abra la carpeta de diseño (aplicación > res > diseño) en el lado izquierdo del panel Proyecto.

2. Haga doble clic en **fragment_first.xml**.

Solución de problemas: si no ve el archivo **fragment_first.xml**, confirme que está ejecutando Android Studio 3.6 o posterior, que es necesario para este codelab.

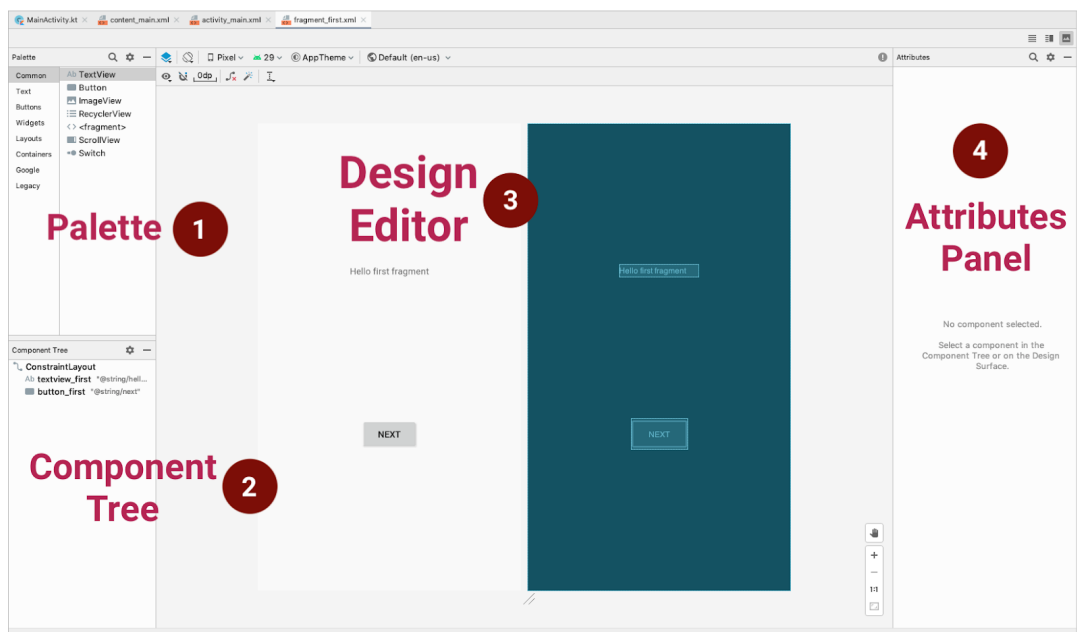
Los paneles a la derecha de la vista Proyecto comprenden el Editor de diseño. Es posible que estén organizados de manera diferente en su versión de Android Studio, pero la función es la misma



A la izquierda hay una Paleta (1) de vistas que puede agregar a su aplicación.

Debajo hay un árbol de componentes (2) que muestra las vistas actualmente en este archivo y cómo están organizadas entre sí.

En el centro está el editor de diseño (3), que muestra una representación visual de cómo se verá el contenido del archivo cuando se compile en una aplicación de Android. Puede ver la representación visual, el código XML o ambos.

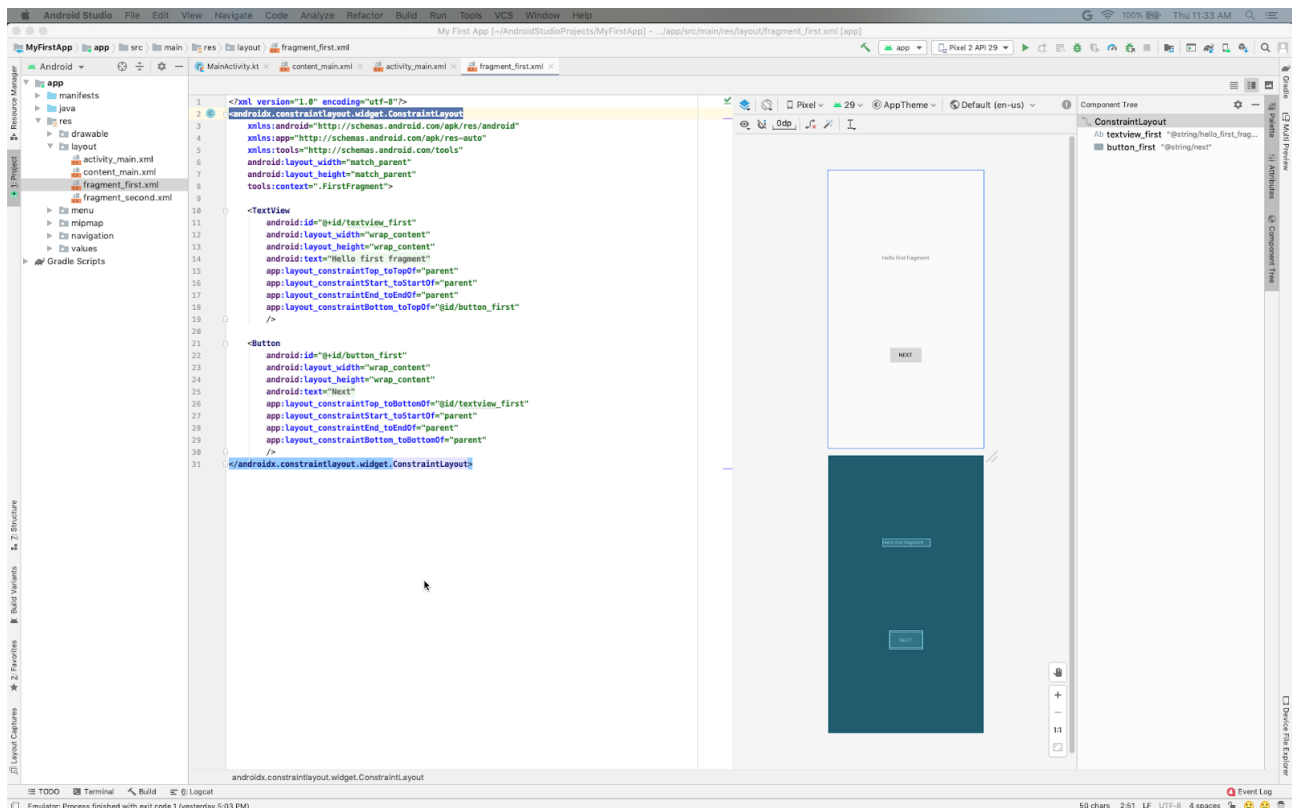


3. En la esquina superior derecha del editor de diseño, arriba de Atributos (4), busque los tres íconos

4. que tienen este aspecto: Representan las vistas Código (solo código), División (código + diseño) y Diseño (solo diseño).



5. Intente seleccionar los diferentes modos. Según el tamaño de la pantalla y el estilo de trabajo, es posible que prefiera cambiar entre Código y Diseño o permanecer en la vista dividida. Si su árbol de componentes desaparece, oculte y muestre la paleta.



Vista dividida:

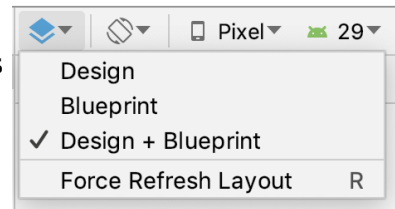
5. En la esquina inferior derecha del editor de diseño, verá los botones + y - para acercar y alejar. Use estos botones para ajustar el tamaño de lo que ve, o haga clic en el botón de zoom para ajustar para que ambos paneles encajen en su pantalla.

El diseño de Diseño a la izquierda muestra cómo aparece su aplicación en el dispositivo. El diseño Blueprint, que se muestra a la derecha, es una vista esquemática del diseño.



El *Design layout* a la izquierda muestra cómo aparece su aplicación en el dispositivo. El diseño Blueprint, que se muestra a la derecha, es una vista esquemática del diseño.

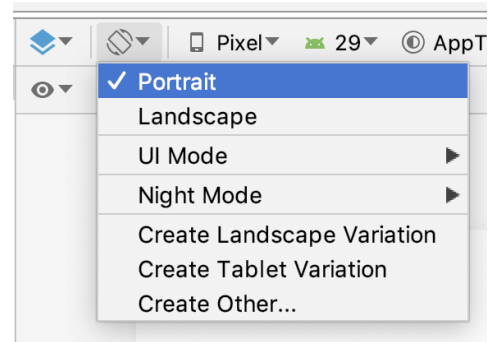
6. Practique usando el menú de diseño en la parte superior izquierda de la barra de herramientas de diseño para mostrar ambas vistas una a lado de la otra.



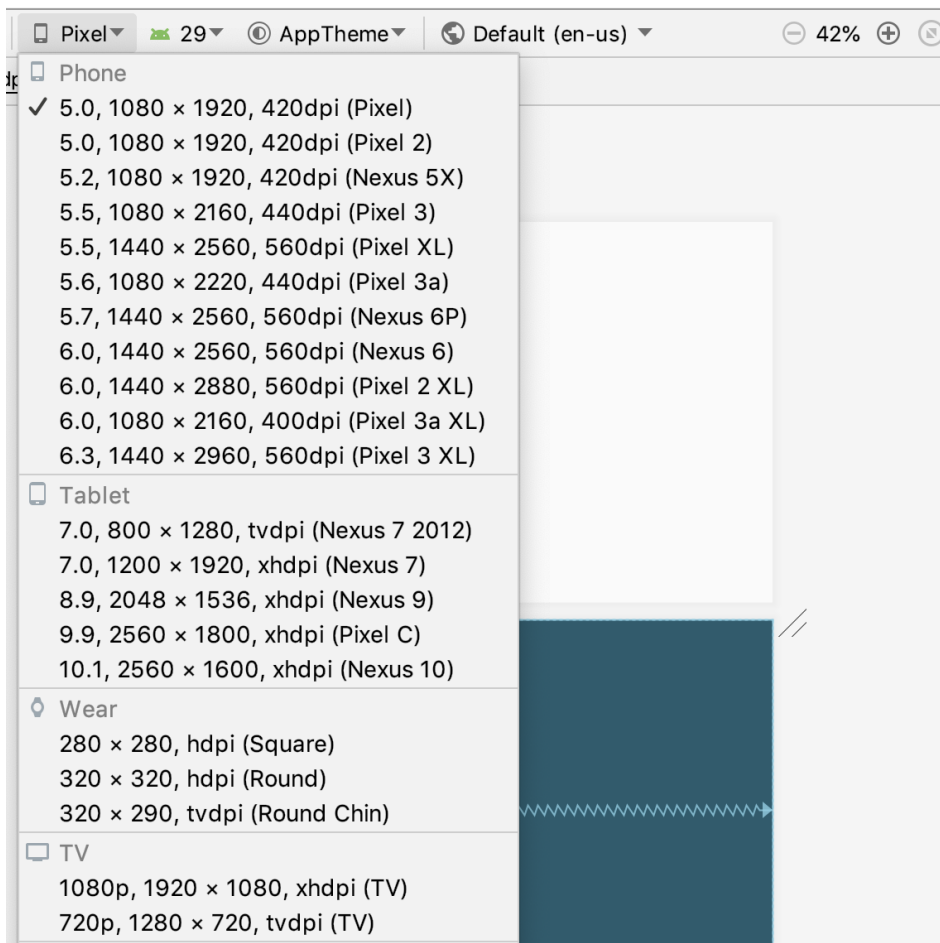
Según el tamaño de su pantalla y su preferencia, es posible que desee mostrar solo la vista Diseño o la vista Blueprint, en lugar de ambas.

7. Utilice el icono de orientación para cambiar la orientación del diseño. Esto le permite probar cómo su diseño se ajustará a los modos vertical y horizontal.

8. Use el menú del dispositivo para ver el diseño en diferentes dispositivos. (¡Esto es extremadamente útil para probar!)

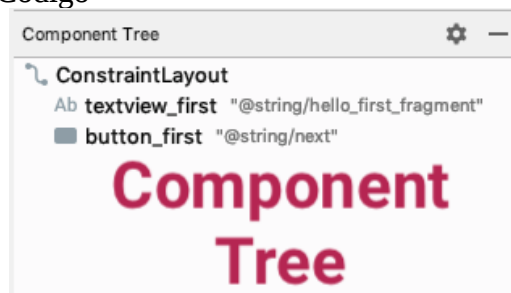


A la derecha está el panel Atributos. Aprenderás sobre eso más tarde.



Paso 2: Explore y cambie el tamaño del árbol de componentes

1. En **fragment_first.xml**, mire el árbol de componentes. Si no se muestra, cambie el modo a Diseño en lugar de Dividir o Código



Este panel muestra la jerarquía de vistas en su diseño, es decir, cómo se organizan las vistas entre sí.

2. Si es necesario, cambie el tamaño del árbol de componentes para que pueda leer al menos parte de las cadenas.

3. Haga clic en el icono Ocultar en la parte superior derecha del árbol de componentes. El árbol de componentes se cierra.

4. Recupere el Árbol de componentes haciendo clic en la etiqueta vertical Árbol de componentes a la izquierda.



Paso 3: Explore las jerarquías de vistas

1. En el árbol de componentes, observe que la raíz de la jerarquía de vistas es una **vista ConstraintLayout**.

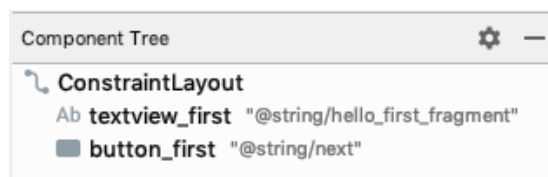
Cada diseño debe tener una vista raíz que contenga todas las demás vistas. La vista raíz siempre es un grupo de vistas, que es una vista que contiene otras vistas.

Un ConstraintLayout es un ejemplo de un grupo de vista.

2. Observe que **ConstraintLayout** contiene un TextView, llamado **textview_first** y un Button, llamado **button_first**.

3. Si no se muestra el código, cambie a Código o Vista dividida usando los íconos en la esquina superior derecha.

4. En el código XML, observe que el elemento raíz es `<androidx.constraintlayout.widget.ConstraintLayout>`. El elemento raíz contiene un elemento `<TextView>` y un elemento `<Button>`.



```
<androidx.constraintlayout.widget.ConstraintLayout
... >
  <TextView
    ... />
  <Button
    ... />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Paso 4: cambiar los valores de propiedad

1. En el editor de código, examine las propiedades del elemento TextView.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello first fragment"
    ... />
```

2. Haga clic en la cadena en la propiedad de texto y notará que se refiere a un recurso de cadena, **hello_first_fragment**.

```
android:text="@string/hello_first_fragment"
```

3. Haga clic derecho en la propiedad y haga clic en **Ir a > Declaración o usos Declaración o Usos value/strings.xml** se abre con la cadena resaltada.

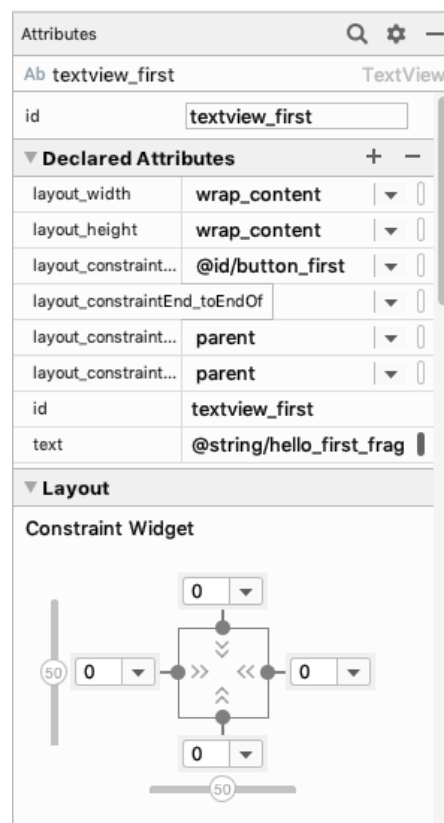
```
<string name="hello_first_fragment">Hello first fragment</string>
```

4. Cambie el valor de la propiedad de la cadena a **Hello World!**.

5. Vuelva a **fragment_first.xml**.

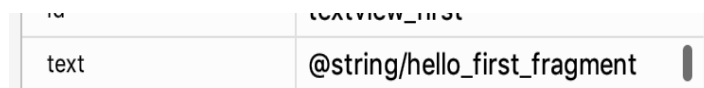
6. Seleccione **textview_first** en el árbol de componentes

7. Mire el panel Atributos a la derecha y abra la sección Atributos declarados si es necesario.



Solución de problemas de este paso:

Si el panel Atributos no está visible, haga clic en la etiqueta Atributos vertical en la parte superior derecha.



9. In the **text** field of the **TextView** in **Attributes**, notice it still refers to the string resource **@string/hello_first_fragment**. Having the strings in a resource file has several advantages. You can change the value of string without having to change any other code. This simplifies translating your app to other languages, because your translators don't have to know anything about the app code.

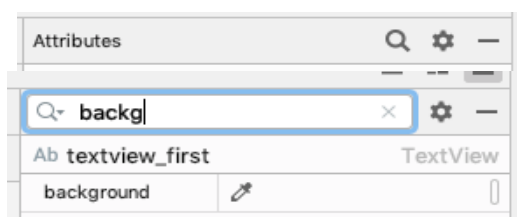
Tip: To find a property in the list of all the properties, click on the magnifying glass icon to the right of **Attributes**, and begin typing the name of the property. Android Studio will show just the properties that contain that string.

9. En el campo de texto de TextView en Attributes, observe que todavía se refiere al recurso de cadena **@string/hello_first_fragment**.

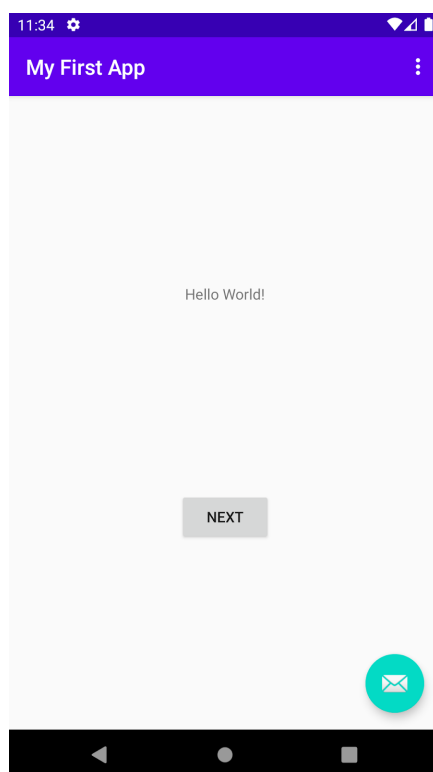
Tener las cadenas en un archivo de recursos tiene varias ventajas.

- Puede cambiar el valor de la cadena sin tener que cambiar ningún otro código.
- Esto simplifica la traducción de su aplicación a otros idiomas, porque sus traductores no tienen que saber nada sobre el código de la aplicación.

Sugerencia: Para encontrar una propiedad en la lista de todas las propiedades, haga clic en el icono de lupa a la derecha de Atributos y comience a escribir el nombre de la propiedad. Android Studio mostrará solo las propiedades que contienen esa cadena.

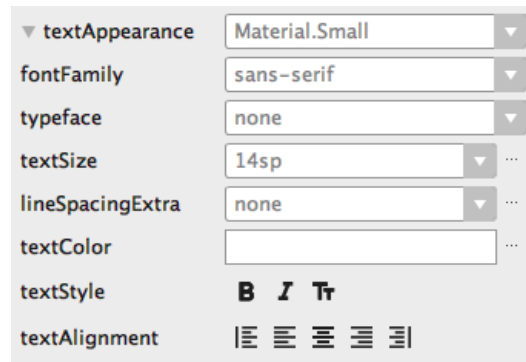


10. Ejecute la aplicación para ver el cambio que realizó en strings.xml. Su aplicación ahora muestra "¡Hola mundo!".



Paso 5: cambiar las propiedades de visualización de texto

1. Con **textview_first** todavía seleccionado en el árbol de componentes, en el editor de diseño, en la lista de atributos, en **Atributos comunes**, expanda el campo **textAppearance**. (Es posible que deba desplazarse hacia abajo para encontrarlo).



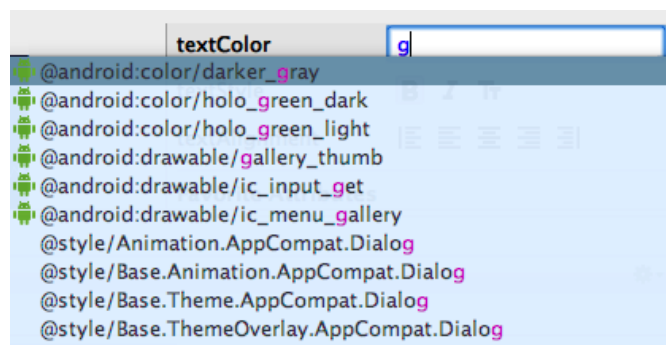
2. Cambie algunas de las propiedades de apariencia del texto.

Por ejemplo, cambie la familia de fuentes, aumente el tamaño del texto y seleccione el estilo de negrita. (Es posible que deba desplazarse por el panel para ver todos los campos).

3. Cambia el color del texto.

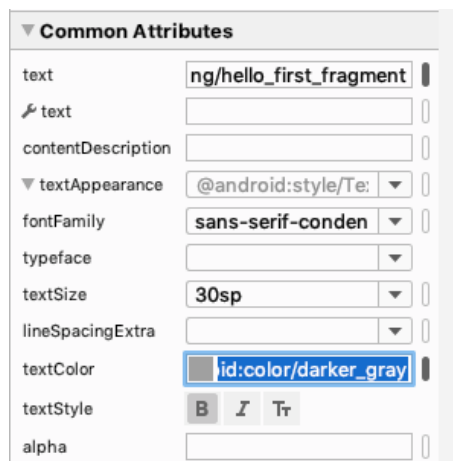
Haga clic en el campo **textColor** e ingrese **g**.

Aparece un menú con posibles valores de finalización que contienen la letra g. Esta lista incluye colores predefinidos.



4. Seleccione **@android:color/darker_gray** y presione Entrar.

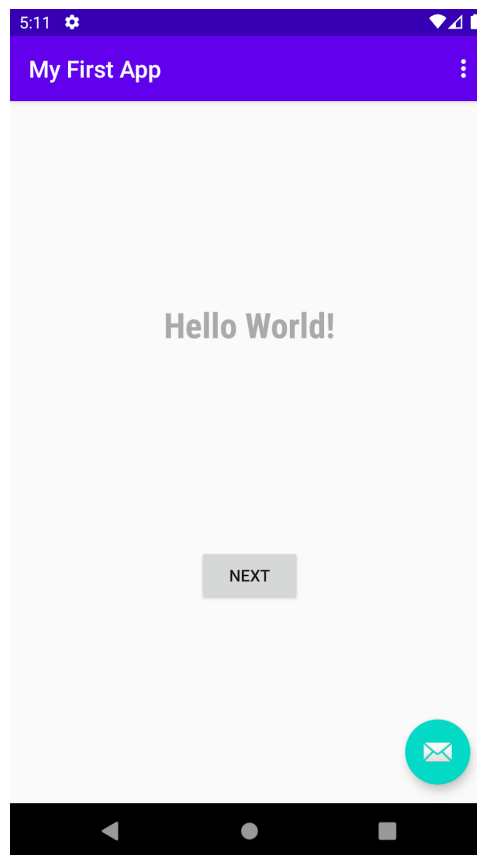
A continuación se muestra un ejemplo de los atributos textAppearance después de realizar algunos cambios.



5. Mire el XML para TextView. Verá que se han agregado las nuevas propiedades.










```
<TextView
    android:id="@+id/textview_first"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif-condensed"
    android:text="@string/hello_first_fragment"
    android:textColor="@android:color/darker_gray"
    android:textSize="30sp"
    android:textStyle="bold"
```

6. Vuelva a ejecutar su aplicación y vea los cambios aplicados a su Hello World! cadena



Paso 6: Mostrar todos los atributos

1. En el panel Atributos, desplácese hacia abajo hasta encontrar **Todos los atributos**. Si no ve ningún atributo en el panel Atributos, asegúrese de que **textview_first** aún esté seleccionado en el Árbol de componentes.
2. Scroll through the list to get an idea of the attributes you could set for a **TextView**.

Attributes			  
Ab <unnamed>			TextView
▼ All Attributes			
accessibilityHeading			0
accessibilityLiveRegion		▼	0
accessibilityPaneTitle			0
accessibilityTraversalAfter		▼	0
accessibilityTraversalBefore		▼	0
allowUndo			0
alpha			0
▶ autoLink			0
autoSizeMaxTextSize			0
autoSizeMinTextSize			0
autoSizePresetSizes			0
autoSizeStepGranularity			0
autoSizeTextType		▼	
autoText			0
autofillHints			0
background			0
backgroundTint			0
backgroundTintMode		▼	