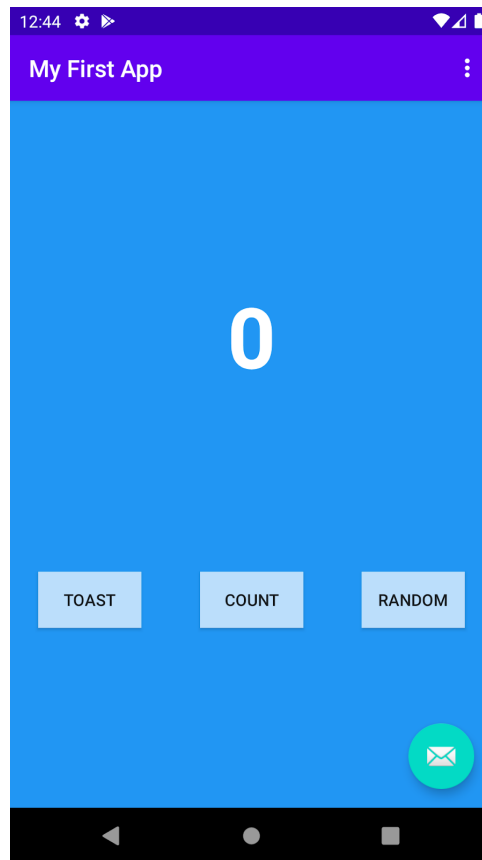


## Tarea 4: agregar vistas y restricciones

En esta tarea, agregará dos botones más a su interfaz de usuario y actualizará el botón existente, como se muestra a continuación.



### Paso 1: ver las propiedades de las restricciones

1. En `fragment_first.xml`, mire las propiedades de restricción para `TextView`.

```
app:layout_constraintBottom_toTopOf="@id/button_first"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
```

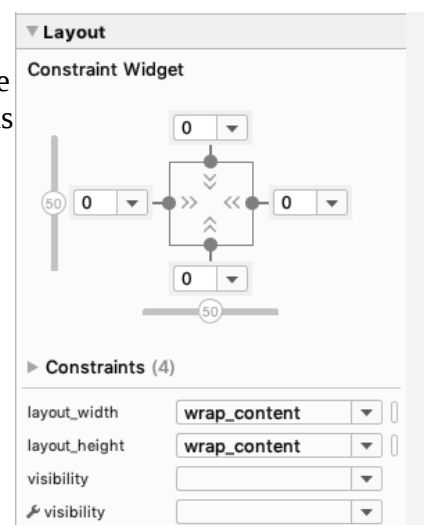
Estas propiedades definen la posición de **TextView**.

Puede restringir la parte superior, inferior, izquierda y derecha de una vista a la parte superior, inferior, izquierda y derecha de otras vistas.

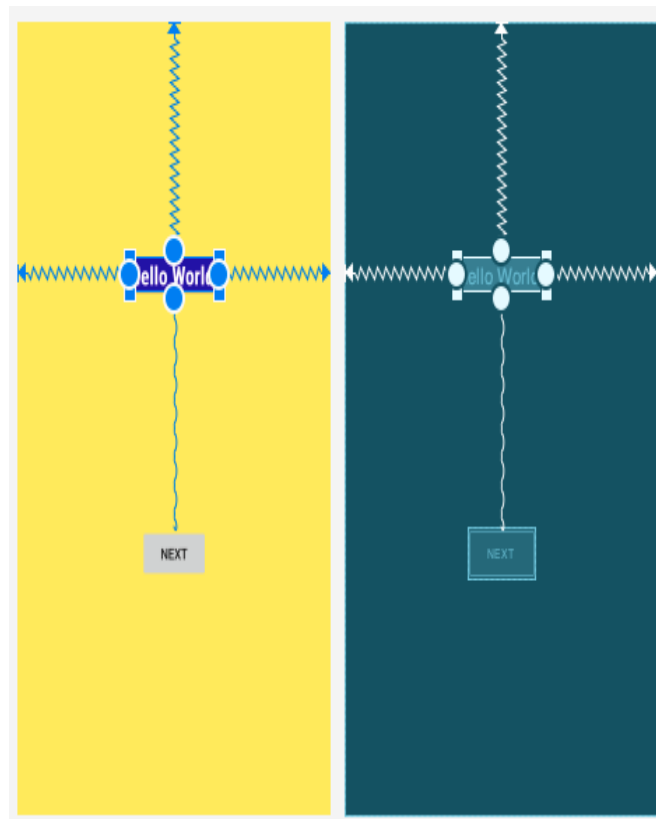
2. Seleccione **textview\_first** en el árbol de componentes y mire el widget de restricción en el panel de atributos.

El cuadrado representa la vista seleccionada.

Cada uno de los puntos grises representa una restricción, arriba, abajo, izquierda y derecha; para este ejemplo, desde `TextView` a su padre, `ConstraintLayout`, o al botón `Siguiente` para la restricción inferior.



3. Observe que las vistas de proyecto y de diseño también muestran las restricciones cuando se selecciona una vista en particular. Algunas de las restricciones son líneas irregulares, pero la del botón Siguiente es un garabato, porque es un poco diferente.

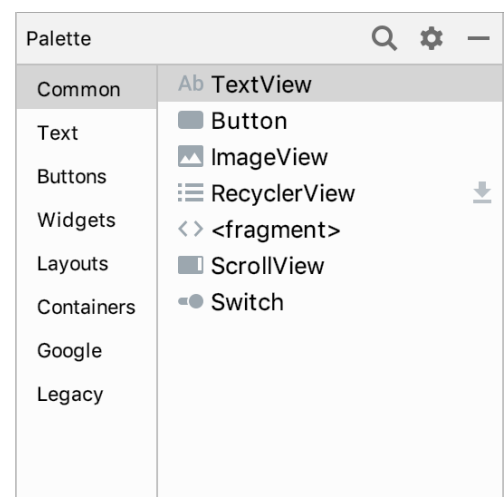


## Paso 2: Agrega botones y restringe sus posiciones

Usar restricciones para conectar las posiciones de las vistas entre sí, agregará botones al diseño.

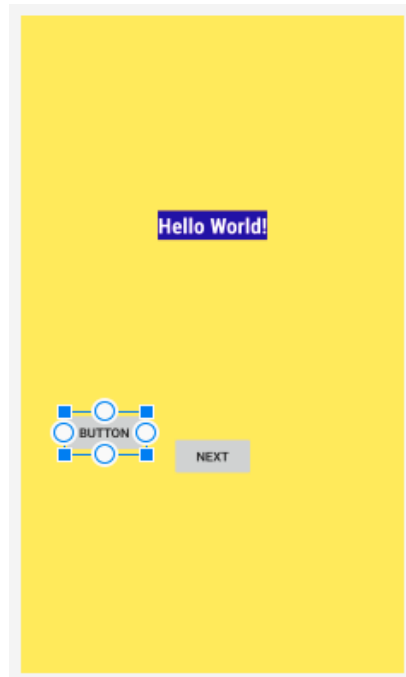
Su primer objetivo es agregar un botón y algunas restricciones, y cambiar las restricciones en el botón Siguiente.

1. Observe la Paleta en la parte superior izquierda del editor de diseño.  
Mueva los lados si es necesario, para que pueda ver muchos de los elementos de la paleta.



2. Haga clic en algunas de las categorías y desplácese por los elementos de la lista si es necesario para tener una idea de lo que está disponible.
3. Seleccione **Botón**, que está cerca de la parte superior, y arrástrelo y suéltelo en la vista de diseño, colocándolo debajo de **TextView** cerca del otro botón.

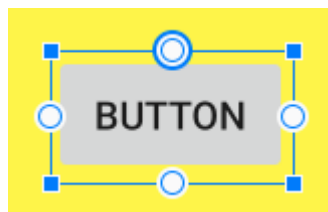
Observe que se ha agregado un **botón** al árbol de componentes en **ConstraintLayout**.



### Paso 3: agregue una restricción al nuevo botón

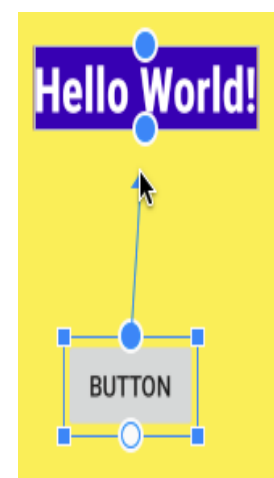
Ahora restringirá la parte superior del botón a la parte inferior de **TextView**.

1. Mueva el cursor sobre el círculo en la parte superior del **Botón**.



2. Haga clic y arrastre el círculo en la parte superior del **Botón** hacia el círculo en la parte inferior de **TextView**.

El **botón** se mueve hacia arriba para sentarse justo debajo de **TextView** porque la parte superior del botón ahora está restringida a la parte inferior de **TextView**.



4. Eche un vistazo al widget de restricción en el panel Diseño del panel Atributos. Muestra algunas restricciones para el botón, incluido **Top -> BottomOf textView**.

5. Eche un vistazo al código XML del botón. Ahora incluye el atributo que restringe la parte superior del botón a la parte inferior de **TextView**.

```
aplicación:layout_constraintTop_toBottomOf="@+id/textview_first"
```

6. Es posible que vea una advertencia, "**Sin restricciones horizontales**". Para solucionar esto, agregue una restricción desde el lado izquierdo del botón al lado izquierdo de la pantalla.

7. Agregue también una restricción para restringir la parte inferior del botón a la parte inferior de la pantalla.

Antes de agregar otro botón, vuelva a etiquetar este botón para que las cosas sean un poco más claras sobre qué botón es cuál.

1. Haga clic en el botón que acaba de agregar en el diseño.

2. Mire el panel Atributos a la derecha y observe el campo de identificación.

3. Cambie la identificación de button a toast\_button.

#### Paso 4: Ajuste el botón Siguiente

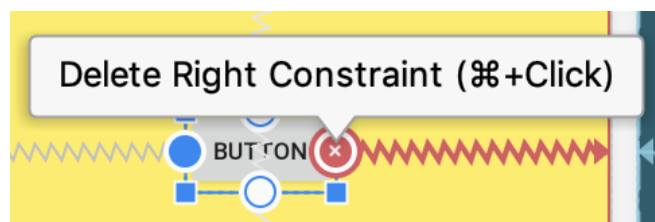
Ajustará el botón con la etiqueta Siguiente, que Android Studio creó para usted cuando creó el proyecto. La restricción entre él y TextView se ve un poco diferente, una línea ondulada en lugar de una irregular, sin flecha.

Esto indica una cadena, donde las restricciones vinculan dos o más objetos entre sí, en lugar de solo uno con otro.

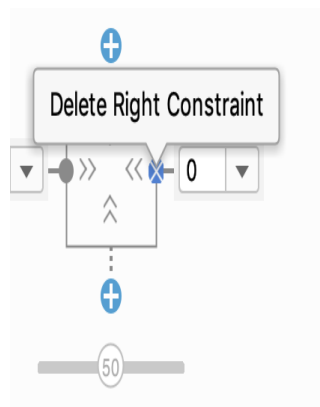
Por ahora, eliminará las restricciones encadenadas y las reemplazará con restricciones regulares.

#### Para eliminar una restricción:

En la vista de diseño o vista de plano:



- mantenga presionada la tecla Ctrl y mueva el cursor sobre el círculo para la restricción hasta que el círculo se resalte, luego haga clic en el círculo.
- O haga clic en una de las vistas restringidas, luego haga clic con el botón derecho en la restricción y seleccione Eliminar en el menú.
- O en el panel Atributos, mueva el cursor sobre el círculo de la restricción hasta que muestre una x, luego haga clic en él.



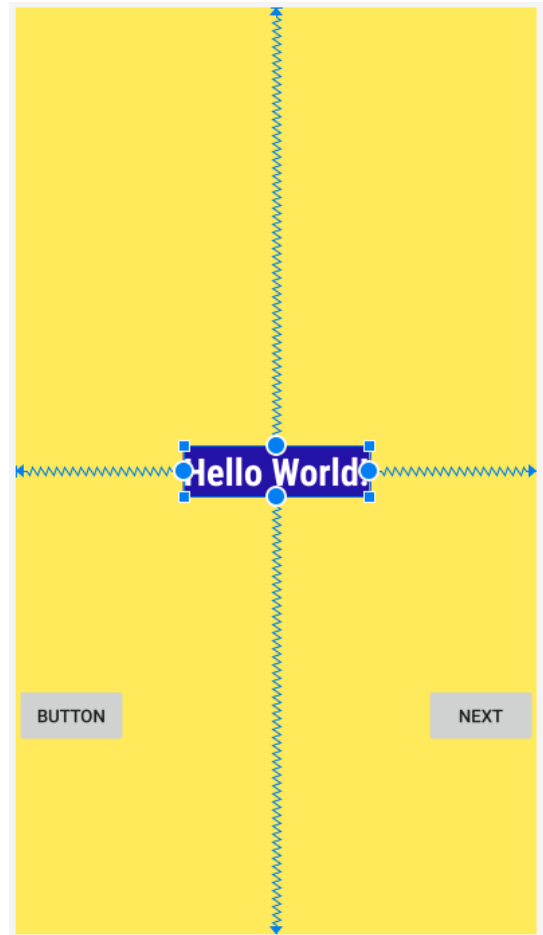
Si elimina una restricción y desea recuperarla, deshaga la acción o cree una nueva restricción.

### Paso 5: elimine las restricciones de la cadena

1. Haga clic en el **botón Siguiente** y luego elimine la restricción desde la parte superior del botón hasta **TextView**.
2. Haga clic en **TextView** y luego elimine la restricción desde la parte inferior del texto hasta el botón Siguiente.

### Paso 6: Agregar nuevas restricciones

1. Restrinja el lado derecho del botón Siguiente a la derecha de la pantalla si aún no lo está.
  2. Elimine la restricción en el lado izquierdo del botón Siguiente.
  3. Ahora restrinja la parte superior e inferior del botón Siguiente para que la parte superior del botón esté restringida a la parte inferior de TextView y la parte inferior esté restringida a la parte inferior de la pantalla. El lado derecho del botón está restringido al lado derecho de la pantalla.
  4. Restrinja también TextView a la parte inferior de la pantalla.
- Puede parecer que las vistas saltan mucho, pero eso es normal a medida que agrega y elimina restricciones. Su diseño ahora debería verse así.

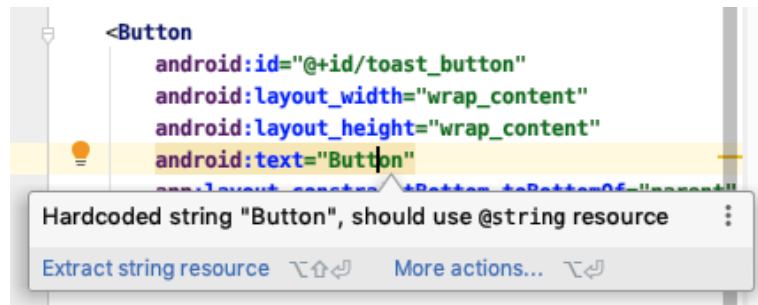


### Paso 7: extraer recursos de cadena

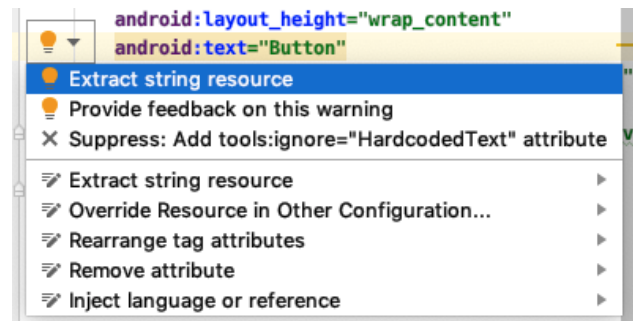
1. En el archivo de diseño **fragment\_first.xml**, busque la propiedad de texto para el botón **toast\_button**.

```
<Button
    android:id="@+id/toast_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
```

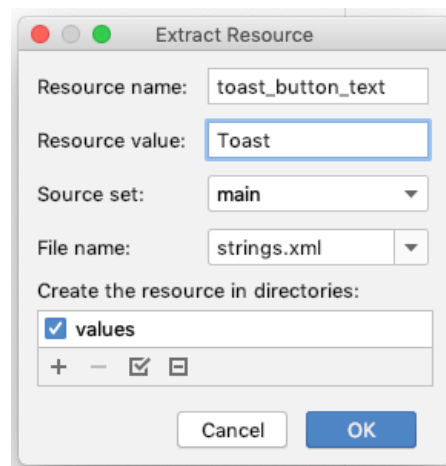
2. Observe que el texto "Botón" está directamente en el campo de diseño, en lugar de hacer referencia a un recurso de cadena como lo hace TextView. Esto hará que sea más difícil traducir su aplicación a otros idiomas.
3. Para solucionar esto, haga clic en el código resaltado. Aparece una bombilla a la izquierda.



4. Haga clic en la bombilla. En el menú que aparece, seleccione **Extraer recurso de cadena**.



4. En el cuadro de diálogo que aparece, cambie el nombre del recurso a **toast\_button\_text** y el valor del recurso a **Toast** y haga clic en **Aceptar**.



6. Observe que el valor de la propiedad **android:text** ha cambiado a **@string/toast\_button\_text**.

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/toast_button_text"
```

7. Vaya al archivo **res > valores > cadenas.xml**. Tenga en cuenta que se ha agregado un nuevo recurso de cadena, llamado **toast\_button\_text**.

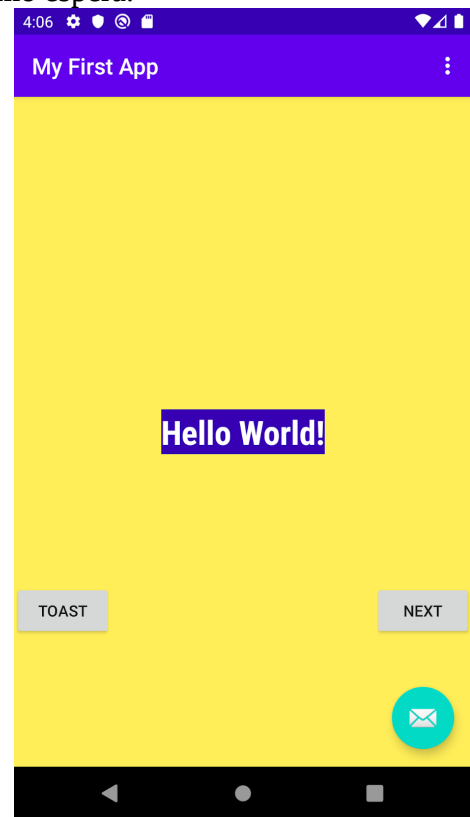
```
<resources>
...
<string name="toast_button_text">Toast</string>
</resources>
```

8. Ejecute la aplicación para asegurarse de que se muestre como espera.

Ahora sabe cómo crear nuevos recursos de cadena extrayéndolos de los valores de campo existentes. (También puede agregar nuevos recursos al archivo **strings.xml** manualmente). Y sabe cómo cambiar la identificación de una vista.

**Nota:** la identificación de una vista lo ayuda a identificar esa vista claramente de otras vistas.

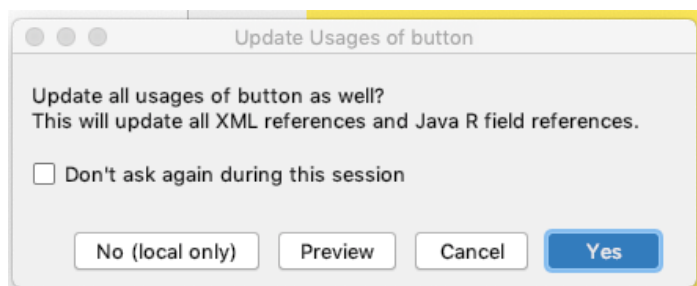
Usará esto más adelante para encontrar vistas particulares usando el método `findViewById()` en su código Java.



### Paso 8: actualice el botón Siguiente

El botón Siguiente ya tiene su texto en un **recurso de cadena**, pero realizará algunos cambios en el botón para que coincida con su nueva función, que será generar y mostrar un número aleatorio.

1. Al igual que hizo con el **botón Toast**, cambie la identificación del botón Siguiente de **button\_first** a **random\_button** en el panel Atributos.
2. Si aparece un cuadro de diálogo que solicita actualizar todos los usos del botón, haga clic en Sí. Esto arreglará cualquier otra referencia al botón en el código del proyecto.



3. En **strings.xml**, haga clic con el botón derecho en el siguiente recurso de cadena.
4. Seleccione **Refactorizar > Cambiar nombre...** y cambie el nombre a **random\_button\_text**.
5. Haga clic en Refactorizar para cambiar el nombre de su cadena y cerrar el cuadro de diálogo.
6. Cambie el valor de la cadena de **Siguiente** a **Aleatorio**.

7. Si lo desea, mueva **random\_button\_text** debajo de **toast\_button\_text**.

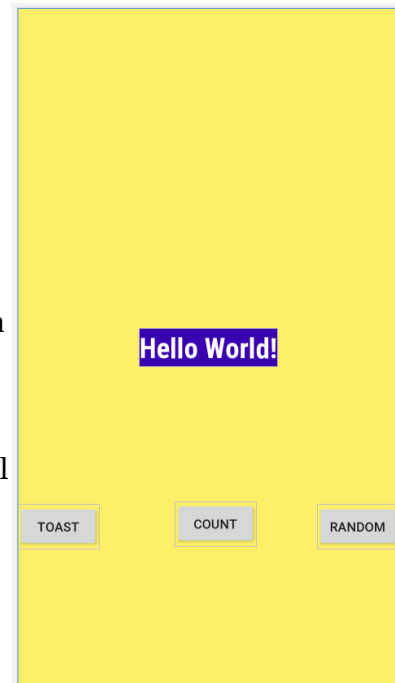
### Paso 9: Agrega un tercer botón

Su diseño final tendrá tres botones, restringidos verticalmente de la misma manera y espaciados uniformemente entre sí.

1. En **fragment\_first.xml**, agregue otro botón al diseño y suéltelo en algún lugar entre el **botón Toast** y el **botón Random**, debajo de **TextView**.

2. Agregue restricciones verticales al igual que los otros dos botones. Restrinja la parte superior del tercer botón a la parte inferior de **TextView**; restrinja la parte inferior del tercer botón a la parte inferior de la pantalla.

3. Agregue restricciones horizontales desde el tercer botón a los otros botones. Restrinja el lado izquierdo del tercer botón al lado derecho del **botón Toast**; restrinja el lado derecho del tercer botón al lado izquierdo del **botón Aleatorio**. Su diseño debería ser algo como esto:



5. Examine el código XML de **fragment\_first.xml**.

¿Alguno de los botones tiene el atributo `app:layout_constraintVertical_bias`? Está bien si no ve esa restricción.

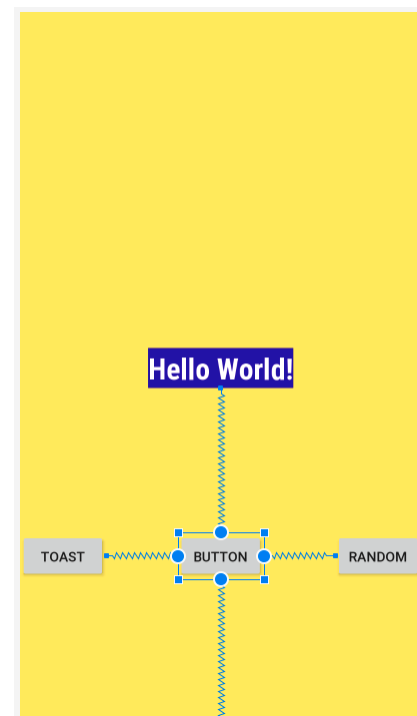
Las restricciones de "sesgo" le permiten ajustar la posición de una vista para que esté más en un lado que en el otro cuando ambos lados están restringidos en direcciones opuestas.

Por ejemplo, si los lados superior e inferior de una vista están restringidos a la parte superior e inferior de la pantalla, puede usar un sesgo vertical para colocar la vista más hacia la parte superior que hacia la parte inferior.

Aquí está el código XML para el diseño terminado.

Su diseño puede tener diferentes márgenes y quizás algunas restricciones de sesgo verticales u horizontales diferentes.

Los valores exactos de los atributos para la apariencia de TextView pueden ser diferentes para su aplicación.





```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout <?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/screenBackground"
    tools:context=".FirstFragment">
    <TextView
        android:id="@+id/textview_first"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimaryDark"
        android:fontFamily="sans-serif-condensed"
        android:text="@string/hello_first_fragment"
        android:textColor="@android:color/white"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/random_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/random_button_text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textview_first" />
    <Button
        android:id="@+id/toast_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/toast_button_text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textview_first" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
```

```

        app:layout_constraintEnd_toStartOf="@+id/random_button"
        app:layout_constraintStart_toEndOf="@+id/toast_button"
        app:layout_constraintTop_toBottomOf="@+id/textview_first" />
</androidx.constraintlayout.widget.ConstraintLayout>

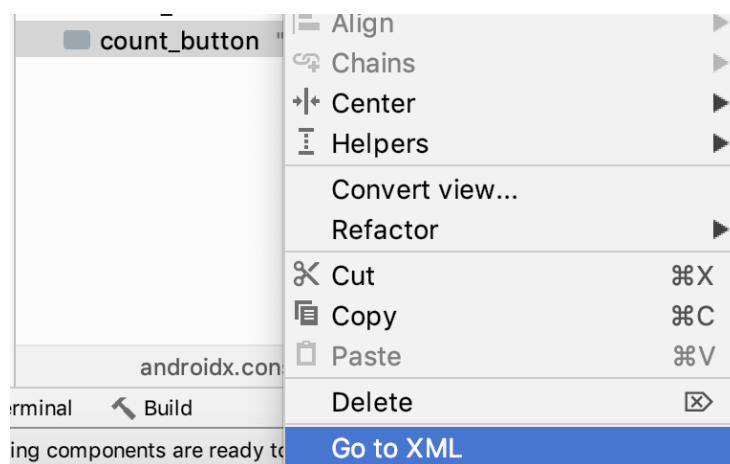
```

### Paso 10: prepare su interfaz de usuario para la siguiente tarea

La siguiente tarea es hacer que los botones hagan algo cuando se presionan.

Primero, debe preparar la interfaz de usuario.

1. Cambie el texto de TextView para mostrar 0 (el número cero).
2. Cambie la identificación del último botón que agregó, **button2**, a **count\_button** en el panel Atributos en el editor de diseño.
3. En el XML, extraiga el recurso de cadena en **count\_button\_text** y establezca el valor en **Count**.



4. Los botones ahora deberían tener el siguiente texto e identificadores:

Button	text	id
Left button	Toast	@+id/toast_button
Middle button	Count	@+id/count_button
Right button	Random	@+id/random_button

5. Ejecute la aplicación

### Paso 11: Corrija los errores si es necesario

Si editó el XML para el diseño directamente, es posible que vea algunos errores.

### Step 11: Fix errors if necessary

```

app:layout_constraintEnd_toStartOf="@+id/button"
app:layout_constraintStart_toEndOf="@+id/button2"

```

Si editó el XML para el diseño directamente, es posible que vea algunos errores.

Los errores ocurren porque los botones han cambiado su identificación y ahora estas restricciones hacen referencia a vistas inexistentes.

- Si tiene estos errores, corríjalos actualizando la identificación de los botones en las restricciones que están subrayadas en rojo.

```
app:layout_constraintEnd_toStartOf="@+id/random_button"  
app:layout_constraintStart_toEndOf="@+id/toast_button"
```