

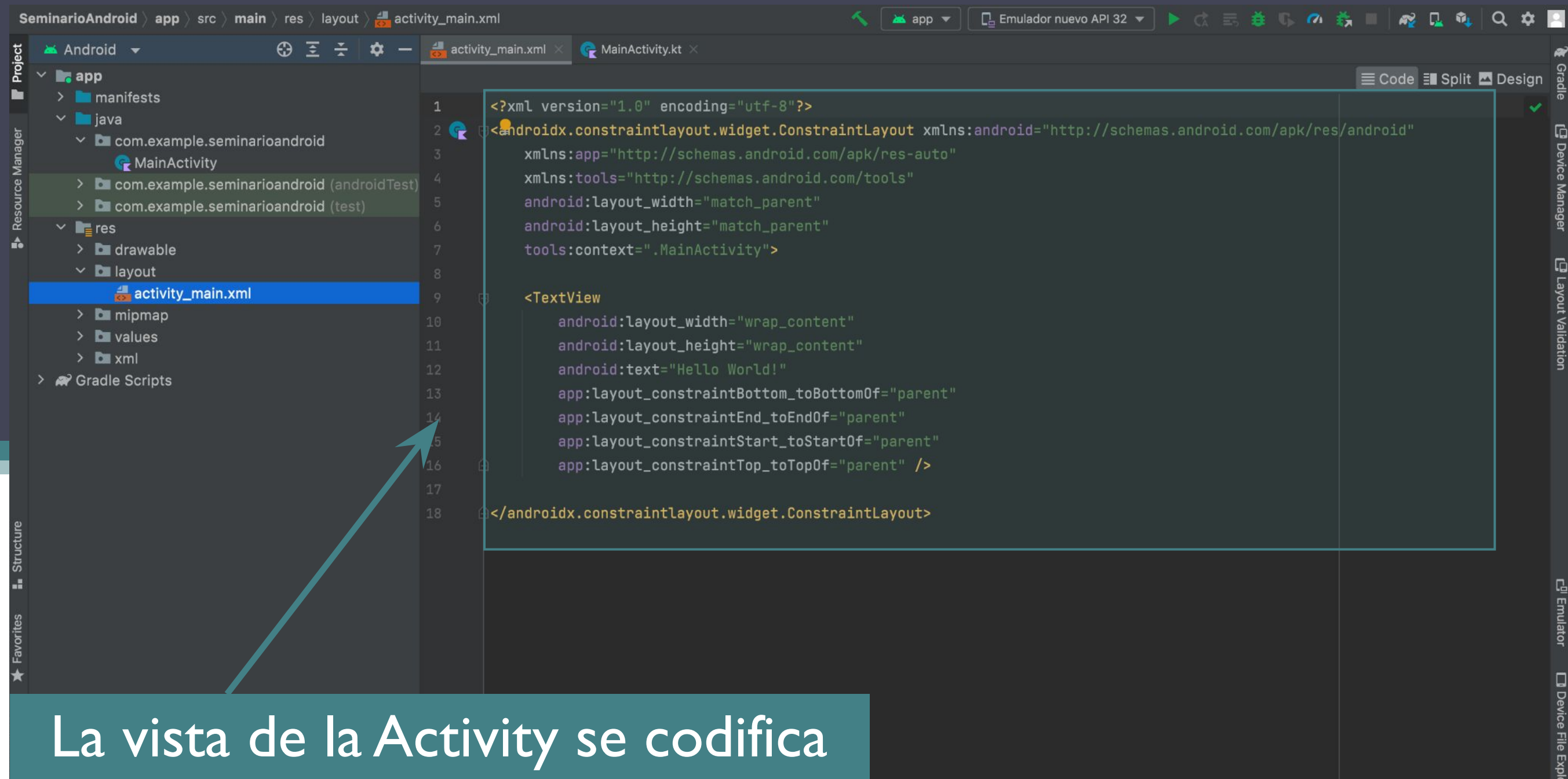
SEMINARIO DE LENGUAJES

OPCIÓN ANDROID + KOTLIN



Introducción a las Views

Esp. Fernández Sosa Juan Francisco



Qué es XML?

```
<?xml versión = "1.0"?>
```

```
<autores>
```

```
  <autor ID = "1">
```

```
    <nombre> Perez </nombre>
```

```
  </autor>
```

```
  <!-- Este es un comentario. -->
```

```
</autores>
```

Es un lenguaje de marcas con el que se pueden describir estructuras de datos

Qué es XML?

```
<?xml versión = "1.0"?>
```

```
<autores>
```

```
<autor ID = "1">
```

```
<nombre> Perez </nombre>
```

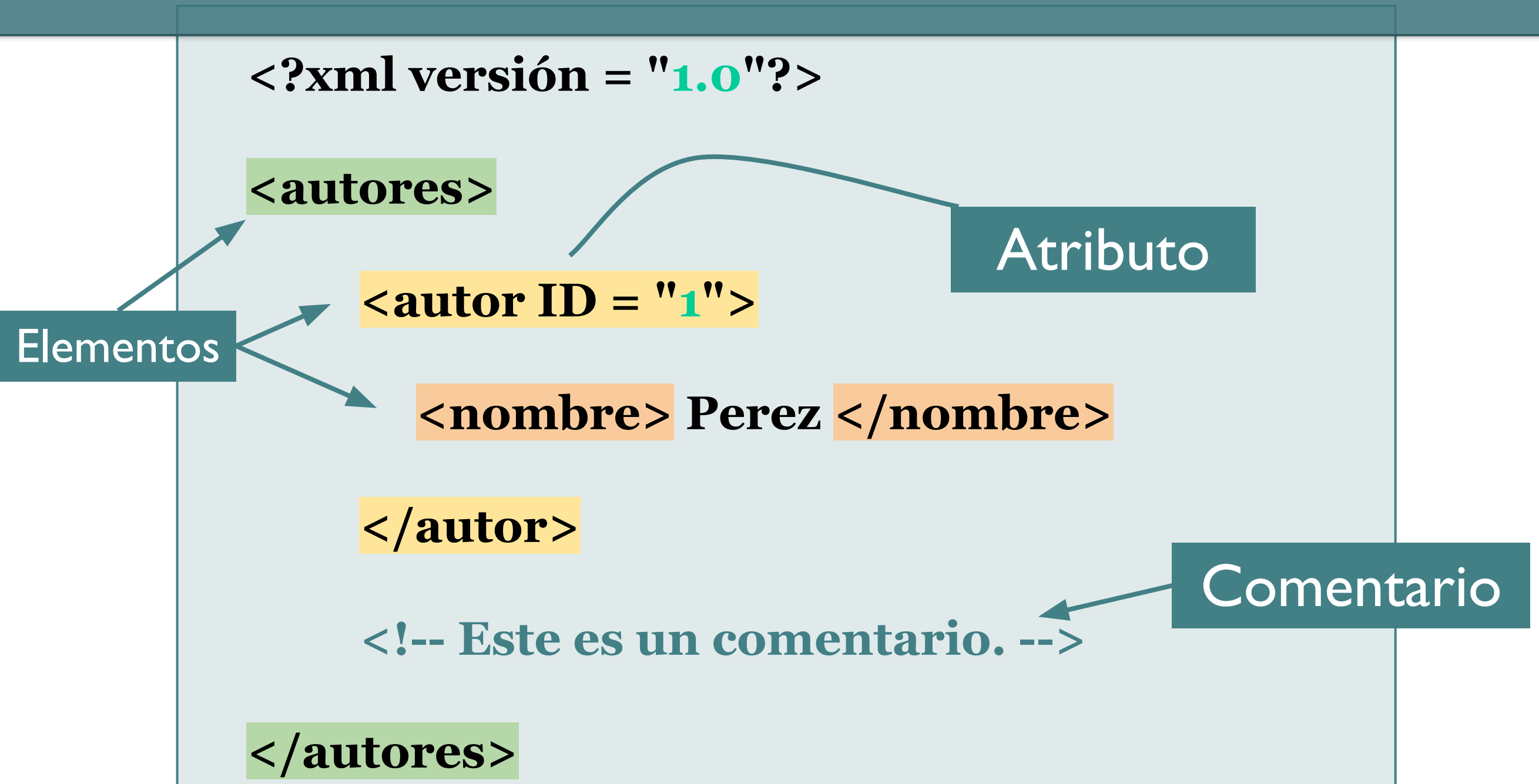
```
</autor>
```

```
<!-- Este es un comentario. -->
```

```
</autores>
```

Las marcas se organizan de a pares: Una de **apertura** y la correspondiente de **cierre**

Qué es XML?



Las marcas (Tags) representan **elementos**

Los elementos pueden tener otros **elementos anidados** y **atributos**

Qué es XML?

- Un elemento usualmente consiste de un **tag de inicio** y un **tag de cierre**.

<Boton> </Boton>

- Cualquier elemento puede contener **atributos** que se declaran en el **tag de inicio**

<Boton ancho="80" alto="10"></Boton>



El valor de un atributo siempre va entre comillas (pueden ser simples o dobles)

Qué es XML?

- Un **elemento vacío**, es decir que no contiene elementos anidados **puede abreviarse con un solo tag** de la siguiente manera

<Boton ancho="80" alto="10"></Boton>



<Boton ancho="80" alto="10" />

- Un documento XML admite un **único elemento raíz** (estructura de árbol)

XML bien formado

```
<Empleados>  
  <Empleado id="01">  
    <Nombre> Juan Pérez </Nombre>  
    <Cargo> Programador </Cargo>  
  </Empleado>  
  
  <Empleado id="02">  
    <Nombre> Emilia García </Nombre>  
    <Cargo> Project Manager </Cargo>  
  </Empleado>  
</Empleados>
```


XML bien formado?

<Empleados>

<Empleado id="01">

<Nombre> Juan Pérez </Nombre>

<Cargo> Programador </Cargo>

<Empleado id="02">

<Nombre> Emilia García </Nombre>

<Cargo> Project Manager </Cargo>

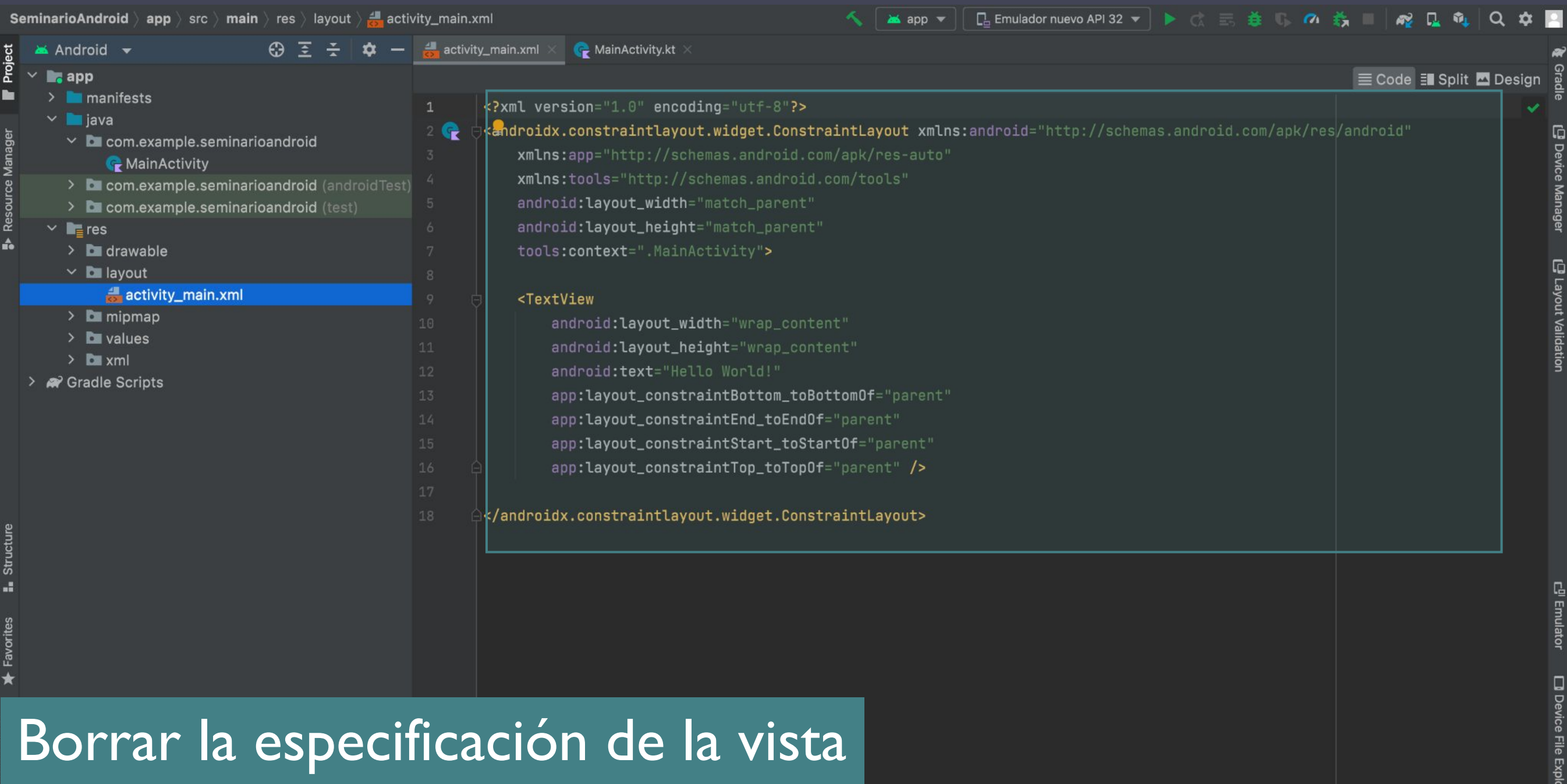
</Empleado>

</Empleados>

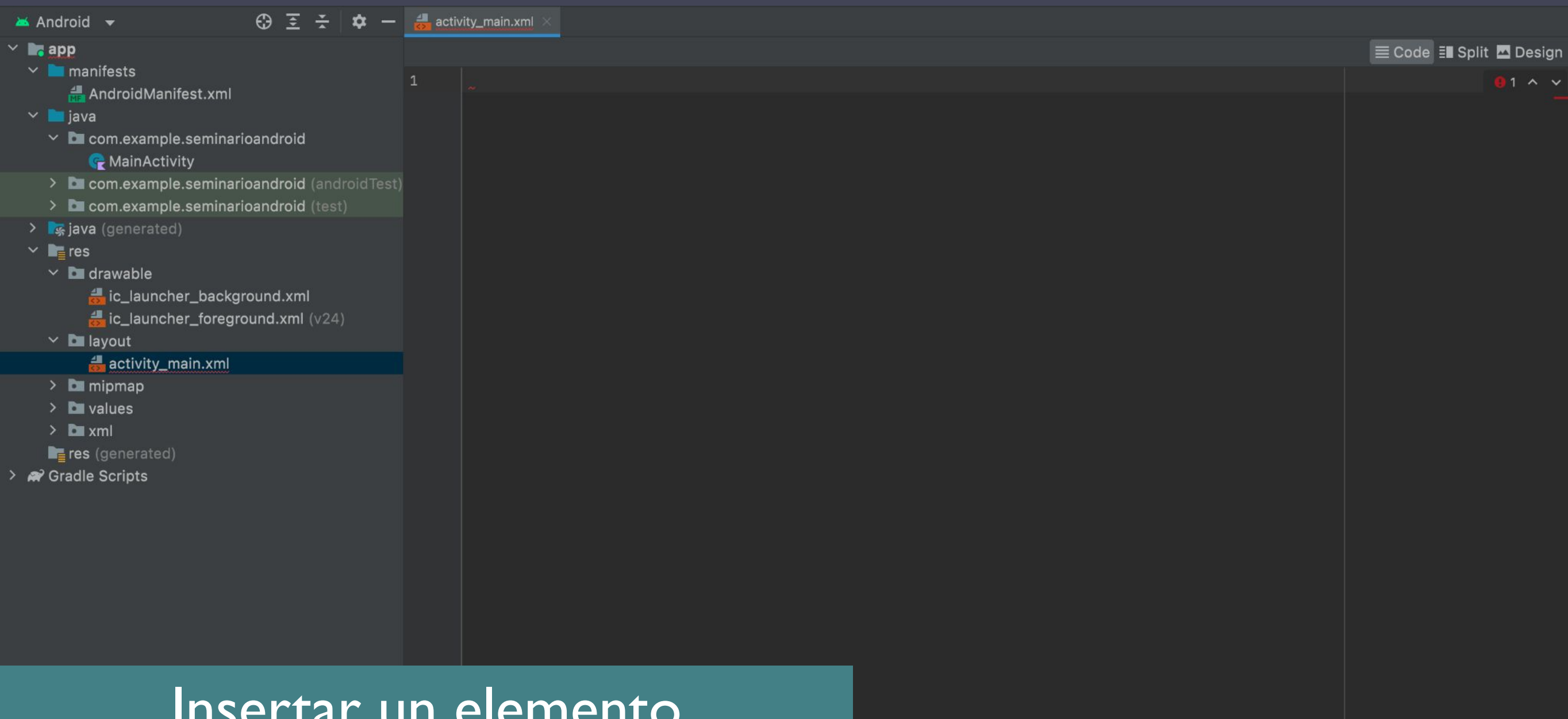
XML bien formado?

```
<Empleado id="01">  
  <Nombre> Juan Pérez </Nombre>  
  <Cargo> Programador </Cargo>  
</Empleado>
```

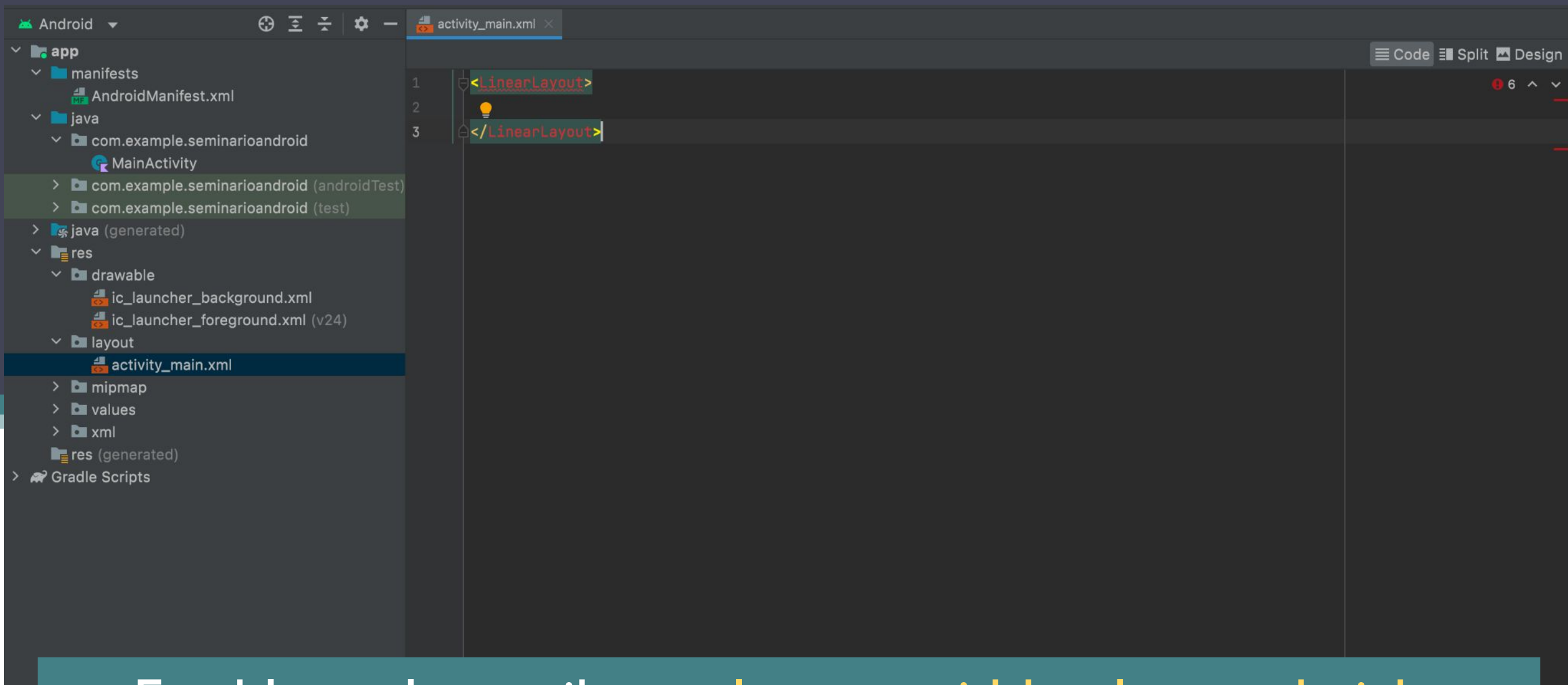
```
<Empleado id="02">  
  <Nombre> Emilia García </Nombre>  
  <Cargo> Project Manager </Cargo>  
</Empleado>
```



Borrar la especificación de la vista completamente

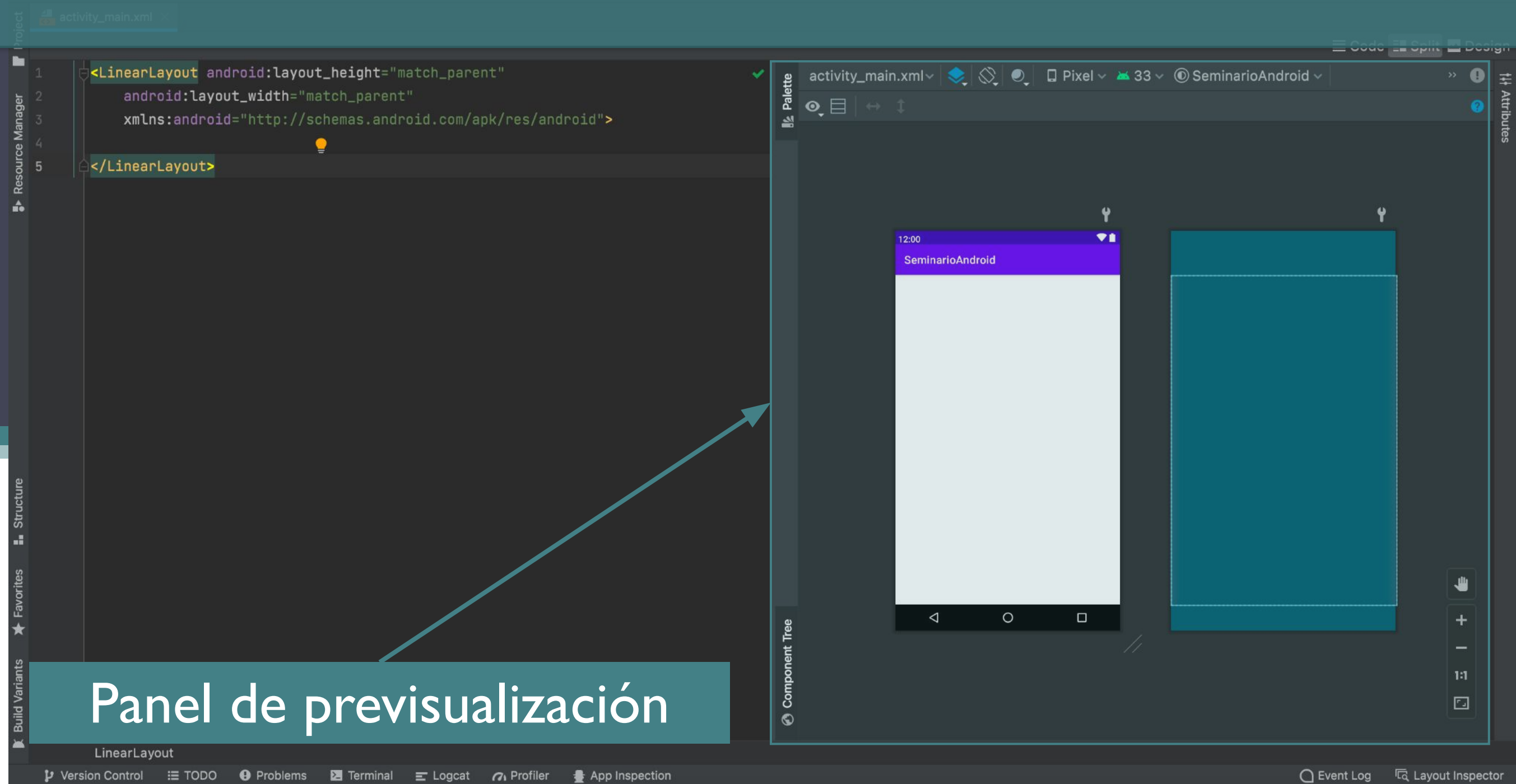


Insertar un elemento
<LinearLayout>



Establecer los atributos `layout_width` y `layout_height` ambos con valor “`match_parent`”

Presionando sobre la pestaña “Split” aparece o se oculta el panel de previsualización a la derecha.

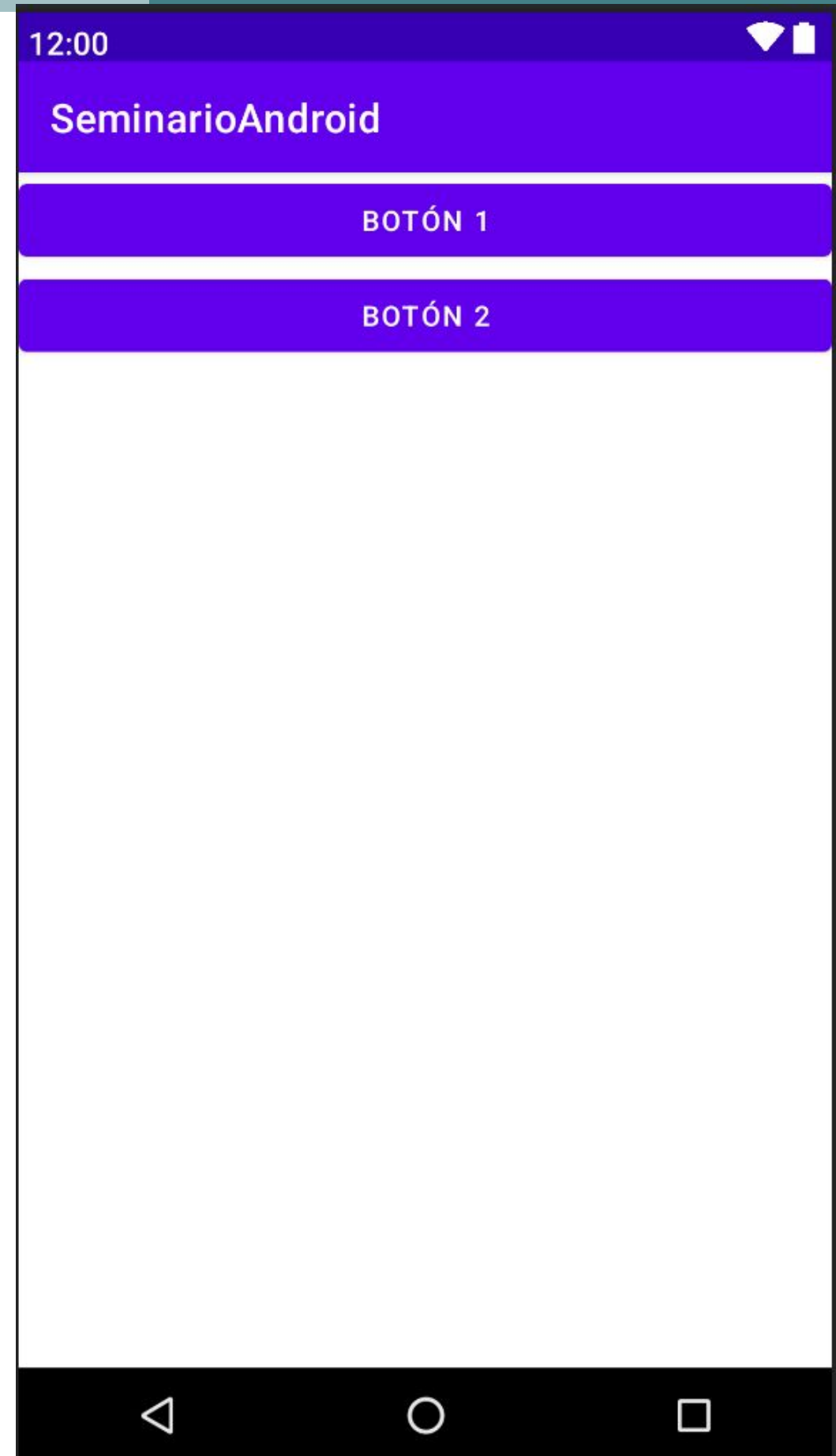


Agregar dos botones al Layout y ejecutar la aplicación

```
<LinearLayout android:layout_height="match_parent"
  android:layout_width="match_parent"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="Botón 1" />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="Botón 2" />
</LinearLayout>
```

Intentar responder: ¿Qué efecto tienen los valores
match_parent y **wrap_content**?

Utilizar el atributo **orientation** del elemento **LinearLayout** para que la aplicación se vea de esta forma



Solución

```
<LinearLayout android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Botón 1" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Botón 2" />
</LinearLayout>
```

Agregar los siguientes tres botones al **Layout**

<Button

```
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:text="Botón 3" />
```

<Button

```
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:layout_gravity="right"  
    android:text="Botón 4" />
```

<Button

```
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:text="Botón 5" />
```

Intentar responder: ¿Para qué se utiliza el atributo
layout_gravity?

Probar con el atributo **gravity**

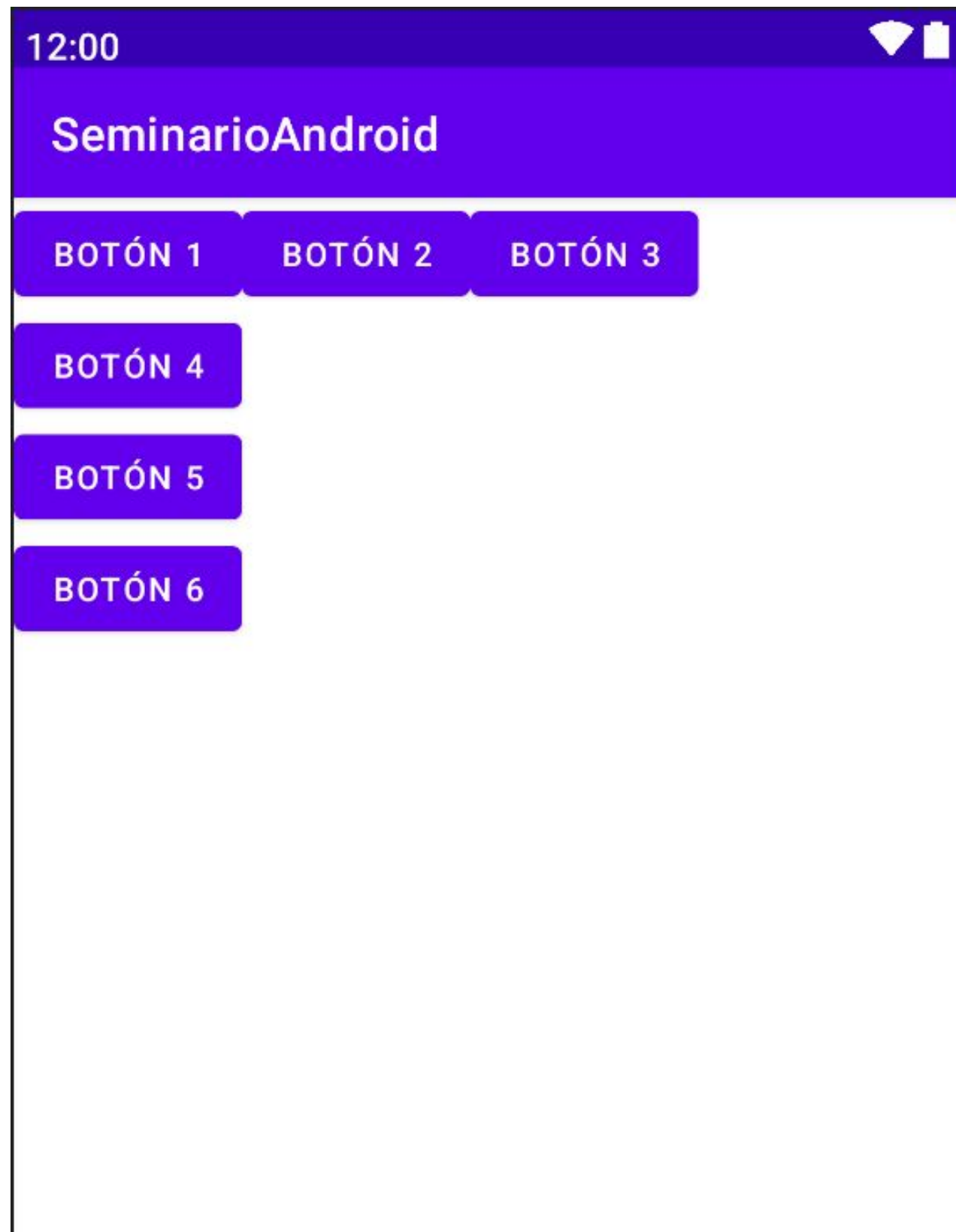
¿Cuál es la diferencia con el atributo **layout_gravity**?

Probar también combinaciones de valores, por ejemplo
gravity="right|bottom"

¿Cómo se podría definir la siguiente interfaz?



Utilizando **LinearLayout** anidados...



Posible solución

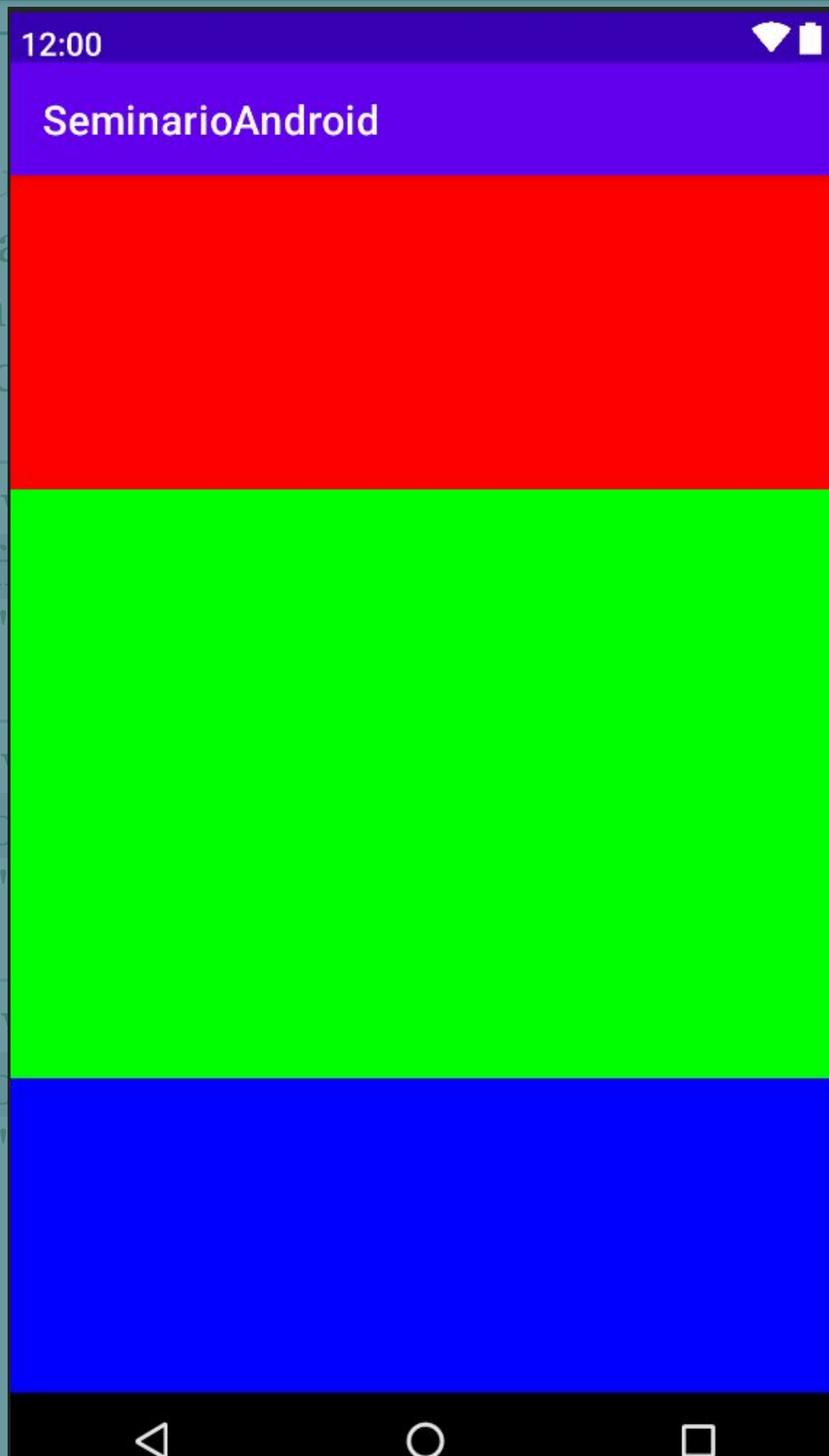
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <LinearLayout android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <Button android:text="Botón 1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <Button android:text="Botón 2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <Button android:text="Botón 3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
    <Button android:text="Botón 4"
        android:layout_width="wrap_content" android:layout_height="wrap_content"/>
    <Button android:text="Botón 5"
        android:layout_width="wrap_content" android:layout_height="wrap_content"/>
    <Button android:text="Botón 6"
        android:layout_width="wrap_content" android:layout_height="wrap_content"/>
</LinearLayout>
```

Definir la siguiente vista

```
<LinearLayout android:layout_width="match_parent"
  android:layout_height="match_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical">
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#ff0000"
    android:layout_weight="1" />
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00ff00"
    android:layout_weight="2" />
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#0000ff"
    android:layout_weight="1" />
</LinearLayout>
```

El color se expresa por medio de un hexadecimal (dos dígitos para cada componente RGB)

Resultado



El color se expresa por medio de un hexadecimal (dos dígitos para cada componente RGB)

```
<LinearLayout android:layout_
android:layout_height="ma
xmlns:android="http://sch
android:orientation="vertic
<TextView android:layout_
android:layout_height="
android:background="#f
android:layout_weight="
<TextView android:layout_
android:layout_height="
android:background="#c
android:layout_weight="
<TextView android:layout_
android:layout_height="
android:background="#c
android:layout_weight="
</LinearLayout>
```


Definir la siguiente vista

```
<LinearLayout android:layout_width="match_parent"
  android:layout_height="match_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical">
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#ff0000"
    android:layout_weight="1" />
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00ff00"
    android:layout_weight="2" />
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#0000ff"
    android:layout_weight="1" />
</LinearLayout>
```

Experimentar
con
distintos valores
para
layout_weight

¿Cómo funciona
esta propiedad?

Definir la siguiente vista

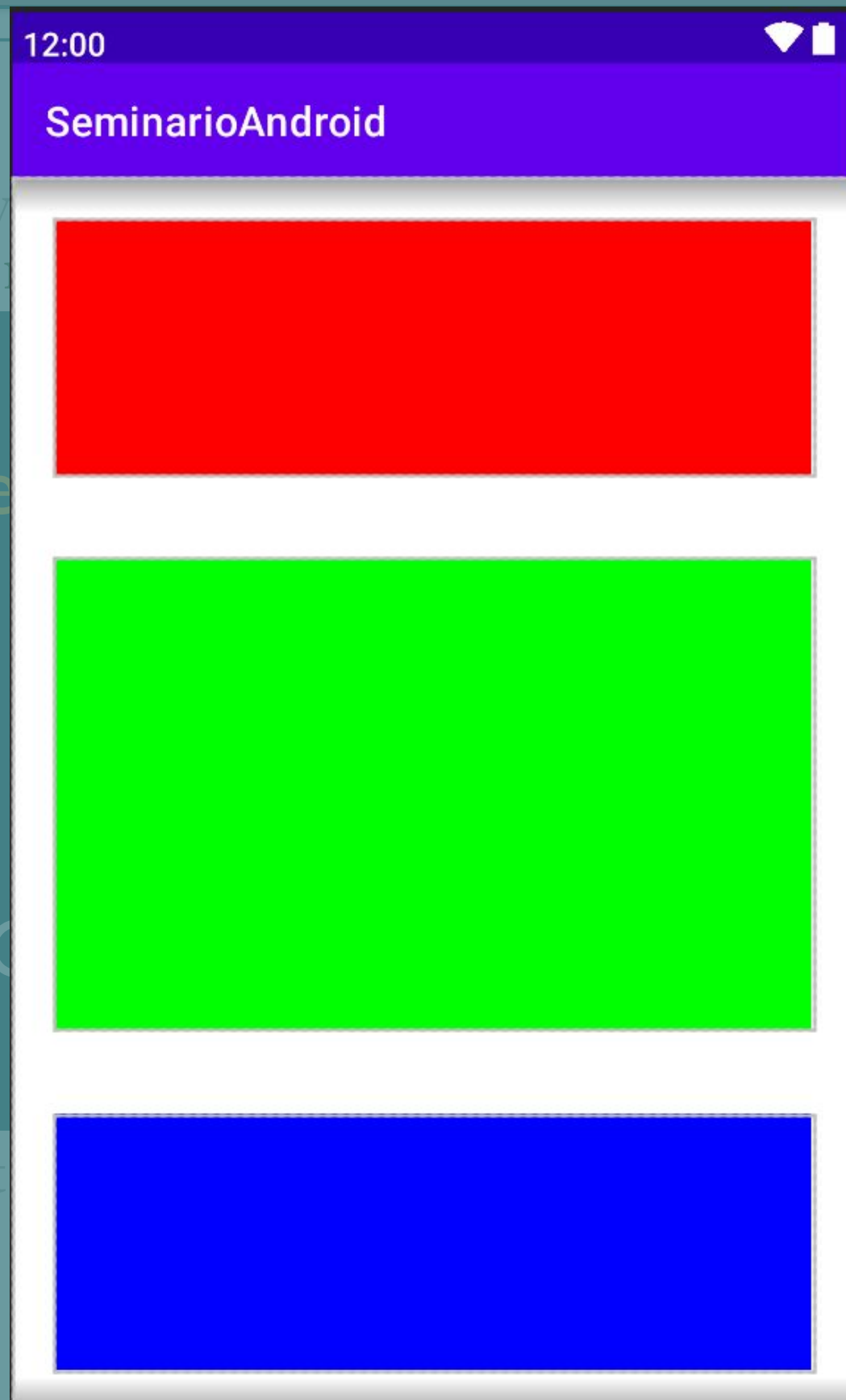
```
<LinearLayout android:layout_width="match_parent"
  android:layout_height="match_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical">
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:layout_margin="20dp"
    android:text="Hello World" />
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:layout_margin="20dp"
    android:text="Hello World" />
  <TextView android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:layout_margin="20dp"
    android:layout_weight="1" />
</LinearLayout>
```

Agregar a los elementos
TextView el siguiente atributo:

layout_margin="20dp"

¿Cuál es su comportamiento?

Resultado



Probar también con otros tipos de márgenes

`layout_marginLeft`

`layout_marginRight`

`layout_marginTop`

`layout_marginBottom`