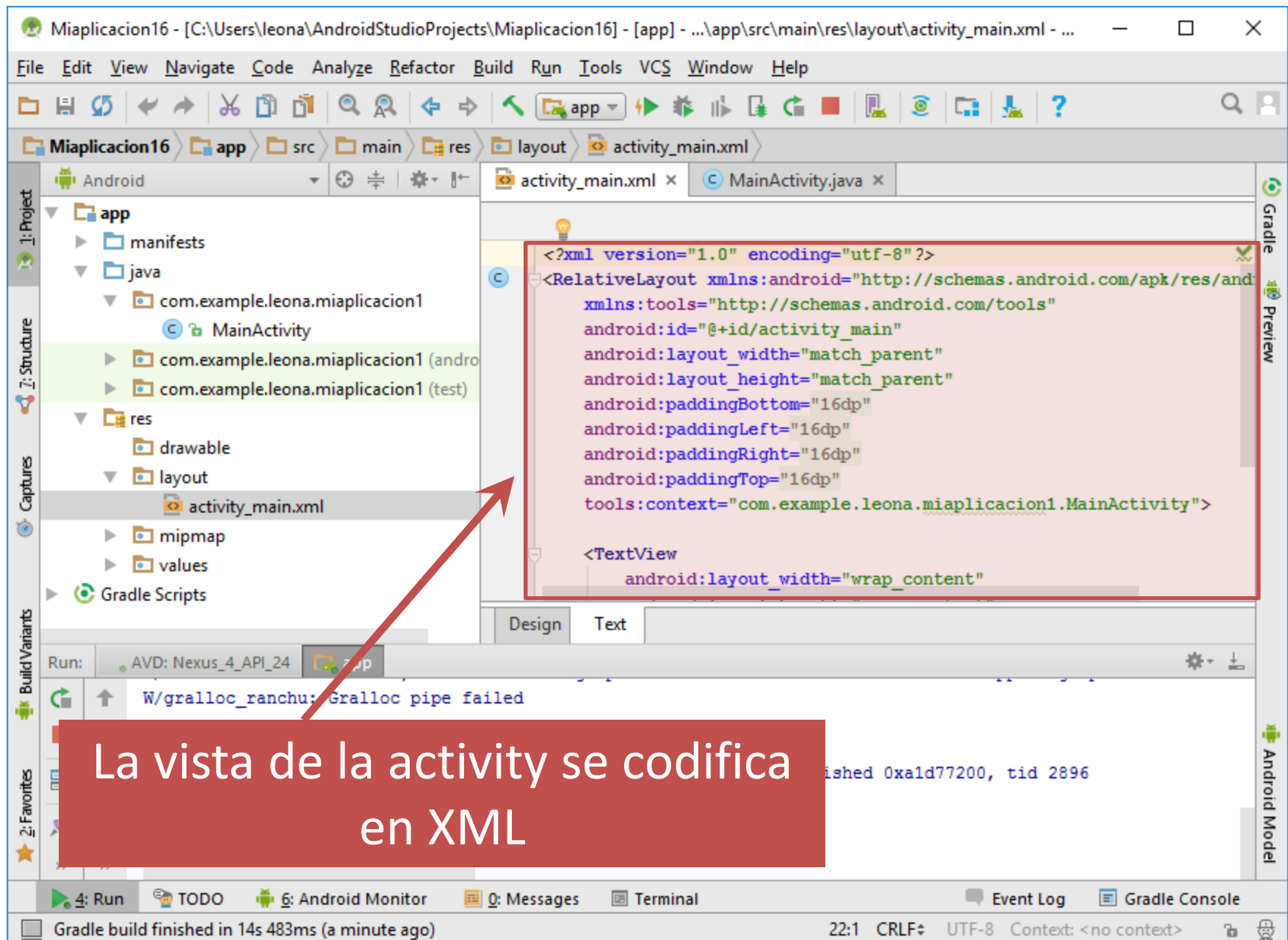


SEMINARIO DE LENGUAJES OPCIÓN ANDROID



Introducción a las Views

Mg. Corbalán Leonardo, Esp. Delía Lisandro



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?

```
<?xml versión = "1.0"?>
```

```
<autores>
```

Atributo

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

Elementos

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

```
</autor>
```

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

```
</autores>
```

Comentario

Las marcas (Tags) representan elementos

Los elementos pueden tener elementos anidados y atributos


XML

Un elemento usualmente consiste en 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)

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

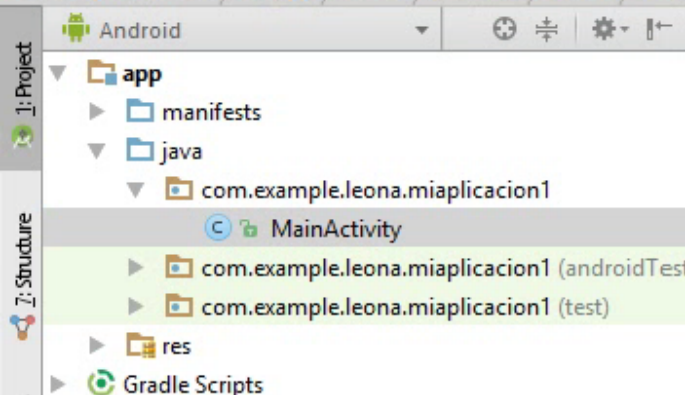
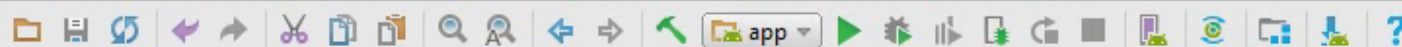
```
<empleados>
  <empleado id = "01">
    <nombre> Stuart Munson </nombre>
    <cargo> Programmer </cargo>
  </empleado>
  <empleado id = "02">
    <nombre> Robert Brown </nombre>
    <cargo> Tester </cargo>
  </empleado>
</empleados>
```


XML ¿ bien formado ?

```
<empleados>
  <empleado id = "01">
    <nombre> Stuart Munson </nombre>
    <cargo> Programmer </cargo>
  <empleado id = "02">
    <nombre> Robert Brown </nombre>
    <cargo> Tester </cargo>
  </empleado>
</empleados>
```

XML ¿ bien formado ?

```
<empleado id = "01">  
  <nombre> Stuart Munson </nombre>  
  <cargo> Programmer </cargo>  
</empleado>  
  
<empleado id = "02">  
  <nombre> Robert Brown </nombre>  
  <cargo> Tester </cargo>  
</empleado>
```

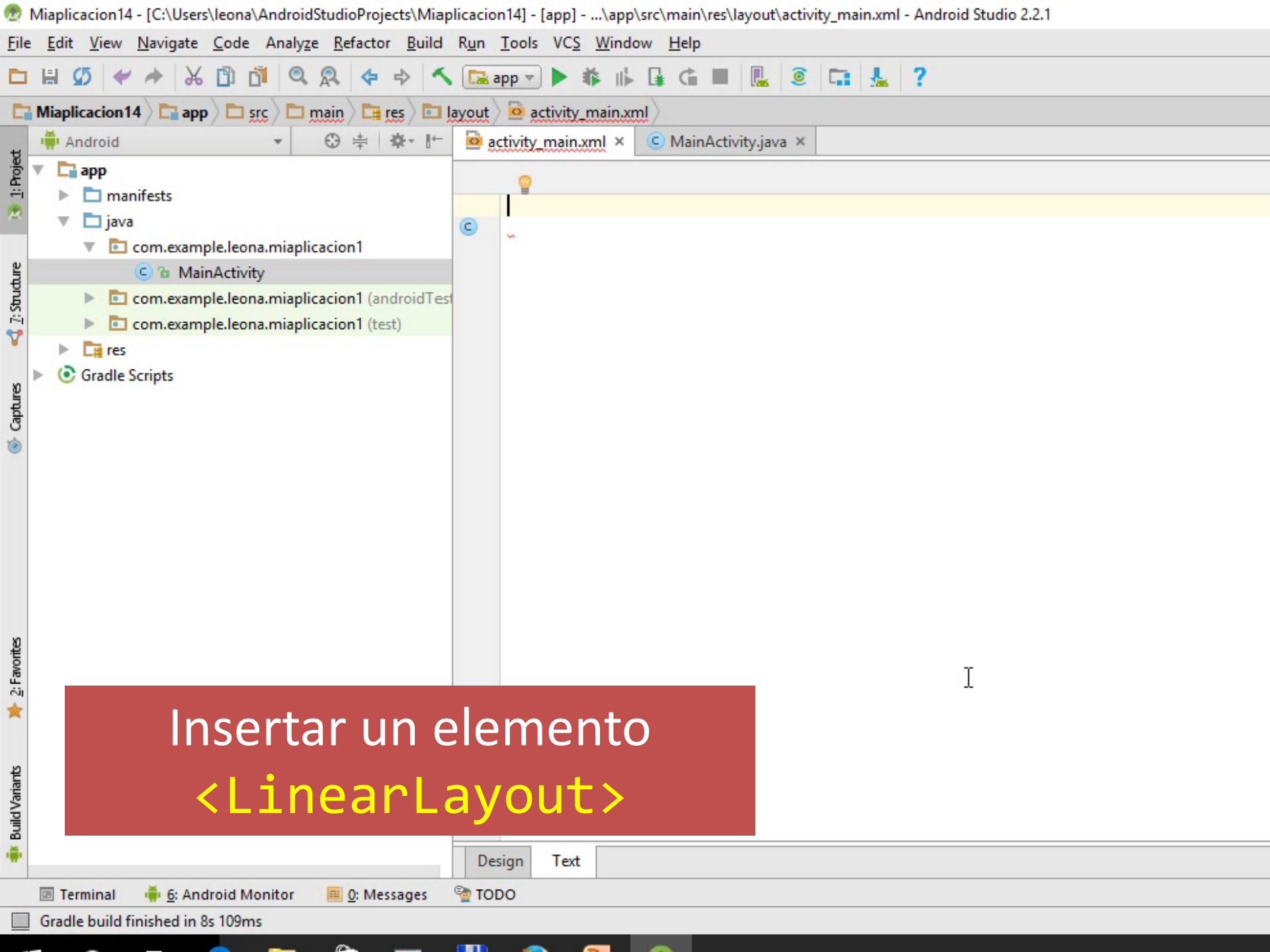


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.leona.miaplicacion1.MainActivity">

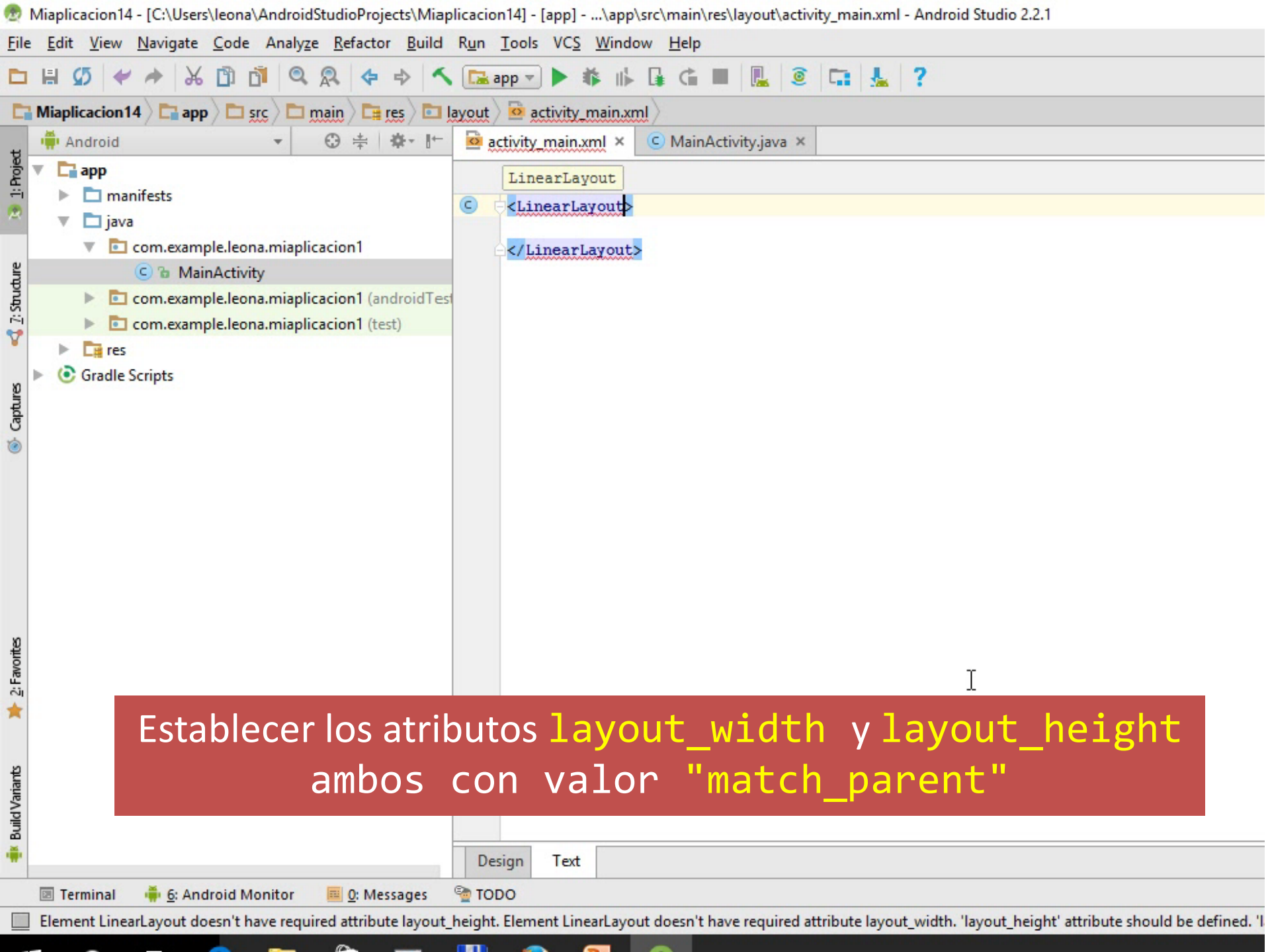
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```

Borrar la especificación de la
vista completamente

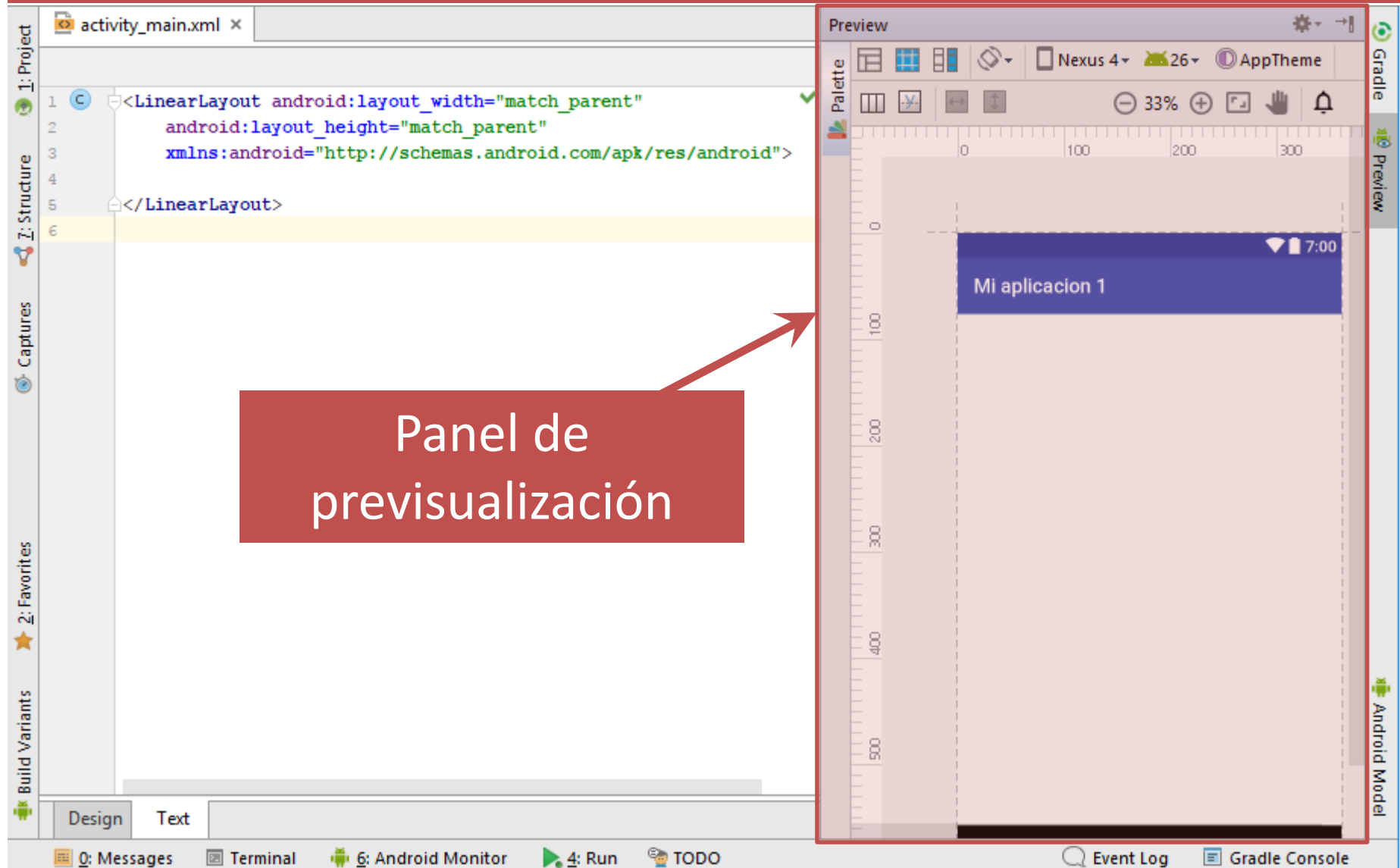


Insertar un elemento
<LinearLayout>



Establecer los atributos `layout_width` y `layout_height` ambos con valor `"match_parent"`

Presionando sobre la pestaña "Preview" aparece o se oculta el panel de previsualización a la derecha



Agregar dos botones al **Layout** y ejecutar la aplicación

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="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="Este es el 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_width="match_parent"
    android:layout_height="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="Este es el 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"

Utilizar **LinearLayout** anidados para obtener este resultado



Posible solución

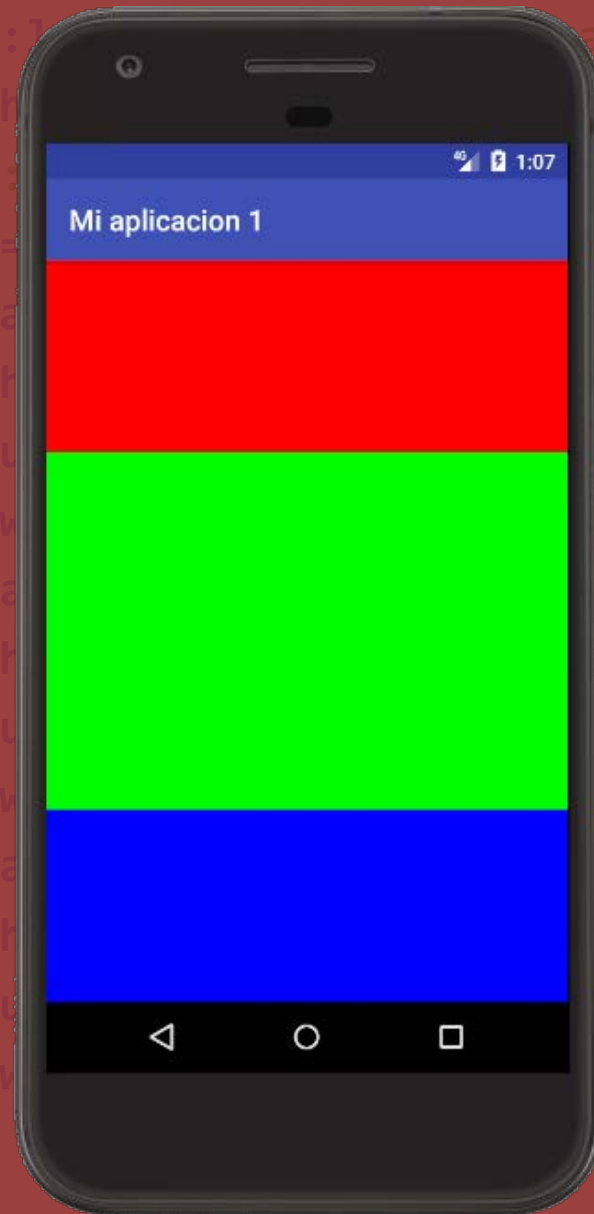
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content">
        <Button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



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
```

```
<TextView
```

```
  an
```

```
  an
```

```
  an
```

```
<TextView
```

```
  an
```

```
  an
```

```
  an
```

```
<TextView
```

```
  an
```

```
  android:background="#0000ff"
```

```
  android:layout_weight="1"/>
```

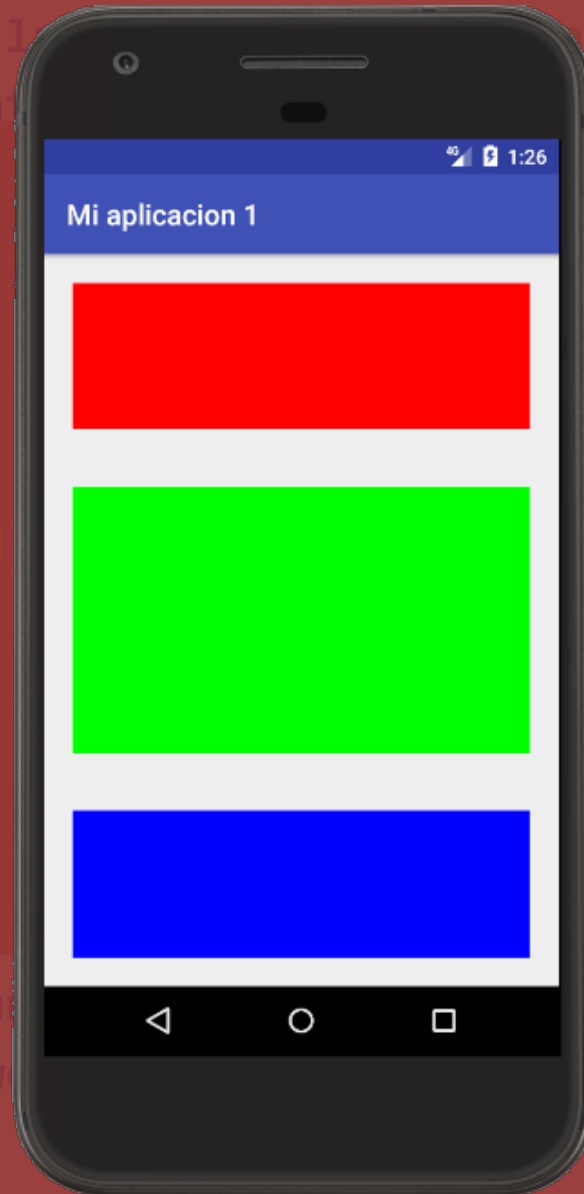
```
</LinearLayout>
```

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`