

## Creación de proyecto en Android Studio

### Paso 1

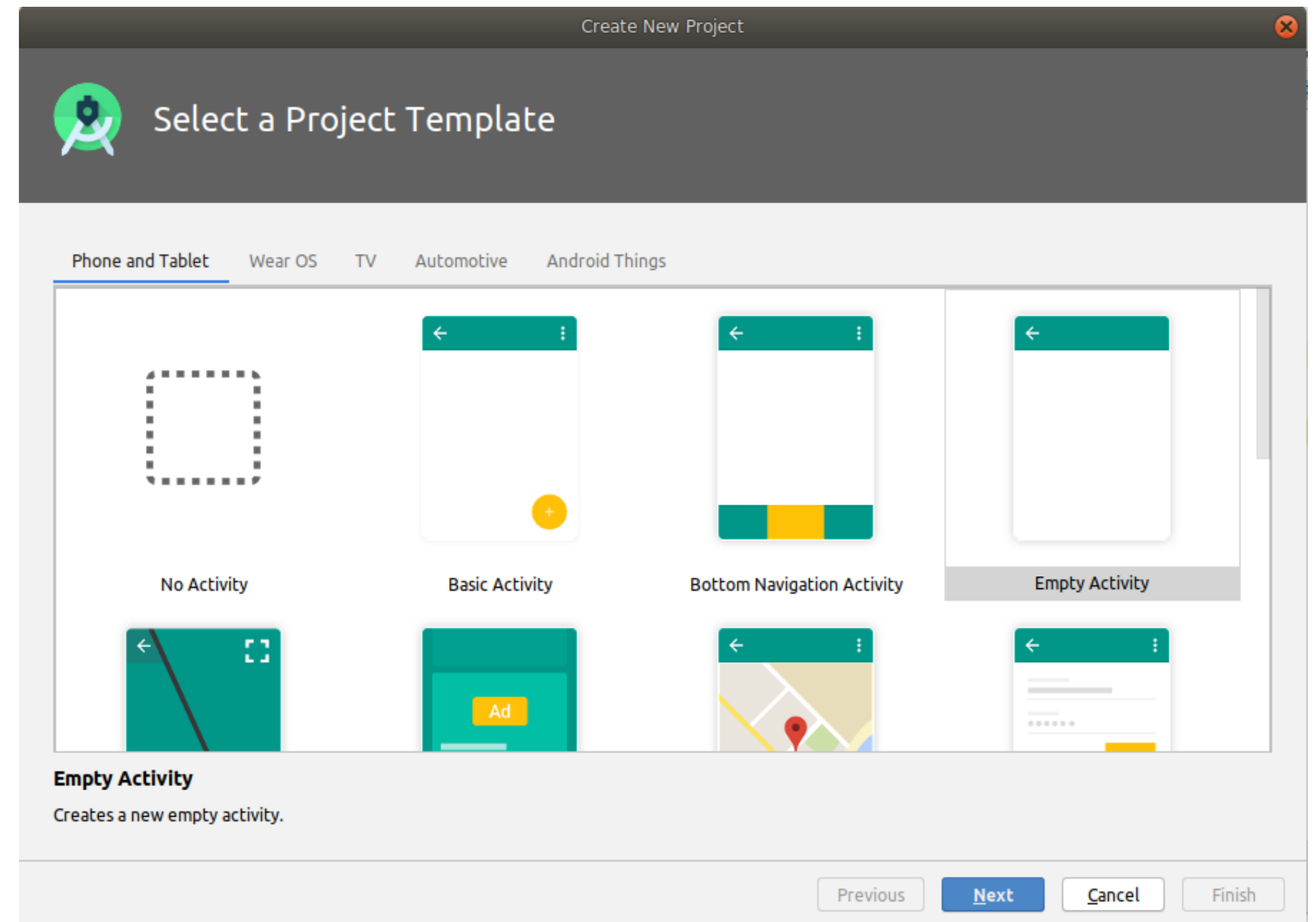
- Abrir el programa Android-Studio



# Creación de proyecto en Android Studio

## Paso 2

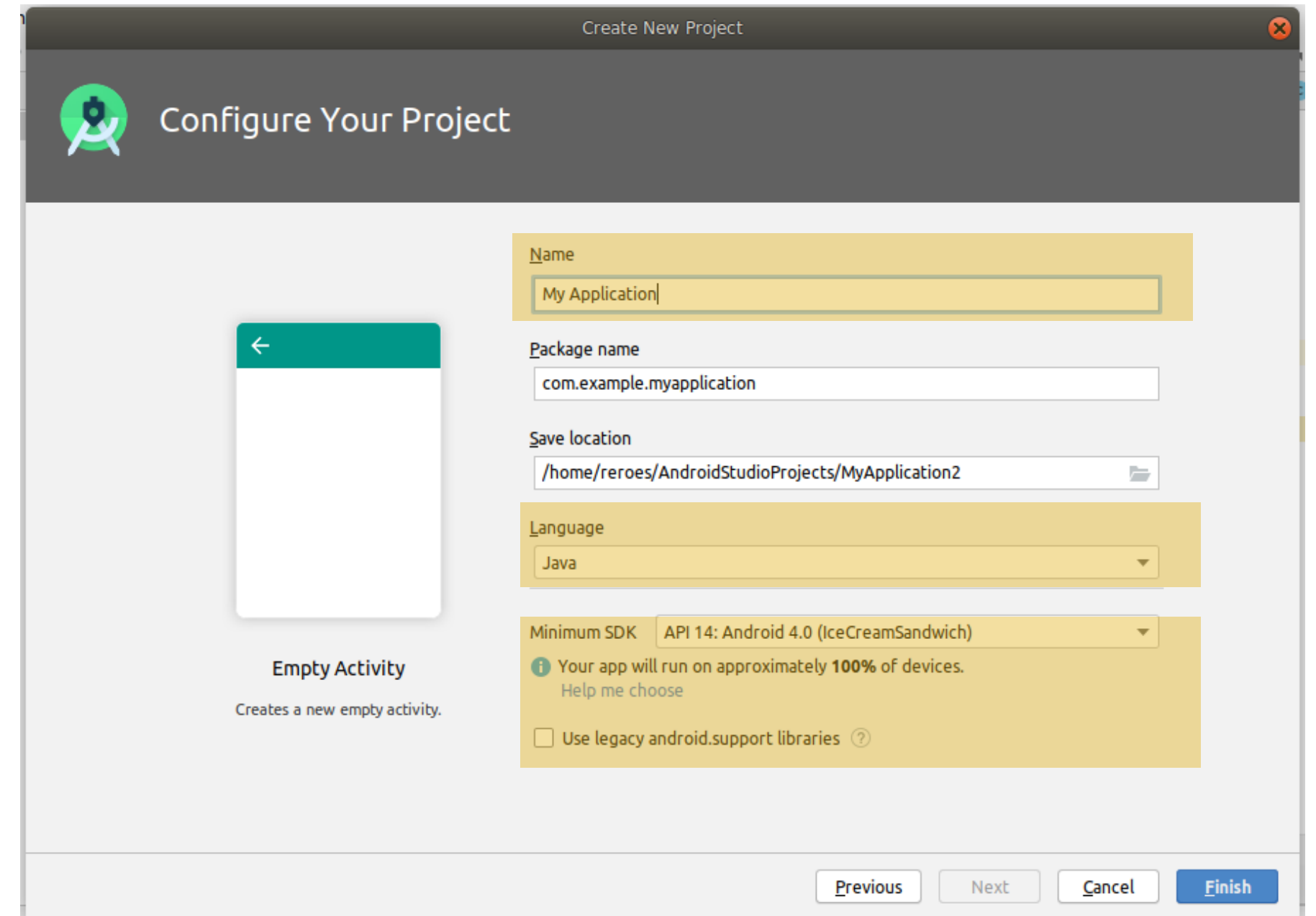
- Se crea el proyecto, a través del menú File>New>New Project
- Se debe visualizar la siguiente pantalla para seleccionar la plantilla.
  - Seleccionamos la opción **Empty Activity**
- Pulsamos la opción **Next**



# Creación de proyecto en Android Studio

## Paso 3

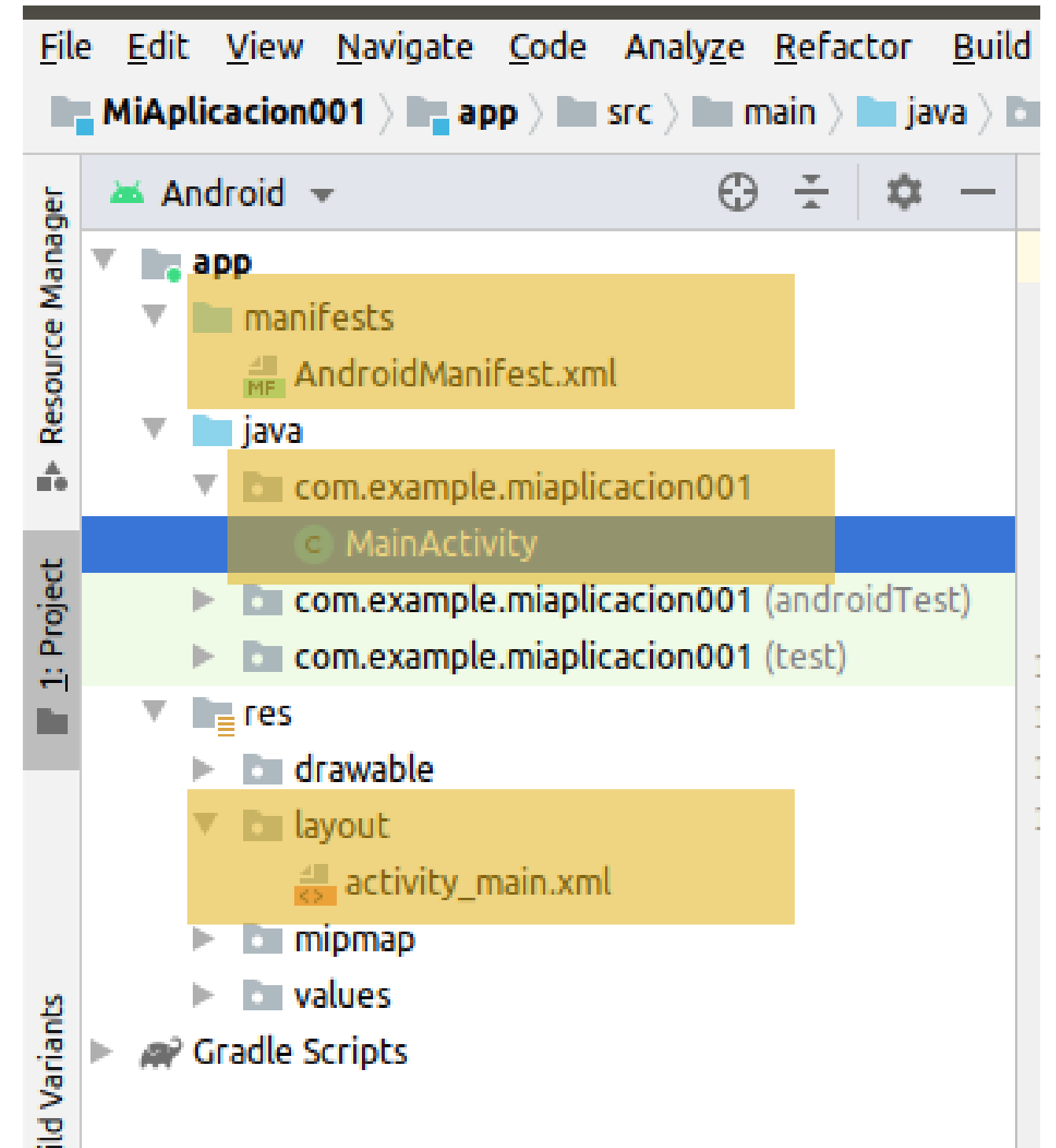
- Se ingresan datos básicos para configurar el proyecto
  - Nombre de la aplicación [**Name**]
  - El lenguaje de programación a usar. Para el ejemplo, se usa Java [**Language**]
  - Selección de la compatibilidad del SDK mínima
  - Se da un visto en la opción [**Use legacy android support libraries**]
- Se pulsa el botón **Finish**



## Creación de proyecto en Android Studio

### Paso 4

- Verificar la estructura de carpetas y archivos que han sido generados por Android-Studio
  - Se debe poner atención a los siguientes archivos:
    - AndroidManifest.xml
    - MainActivity
    - activity\_main.xml



## Creación de proyecto en Android Studio

### Paso 5

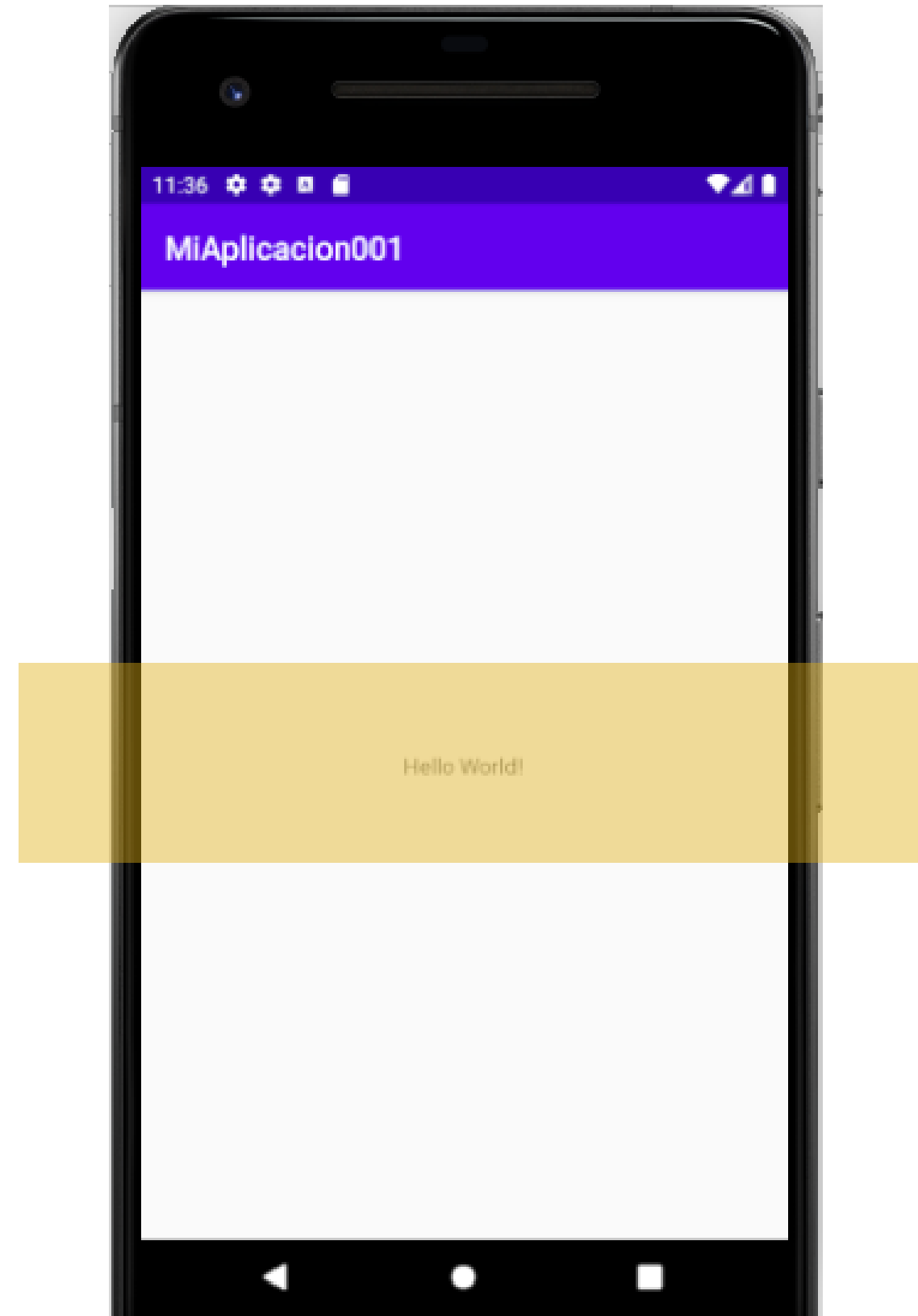
- Ejecutar la primer versión de la aplicación
  - Revisar el proceso de ejecutar la aplicación mediante un emulador:  
<https://developer.android.com/training/basics/firstapp/running-app?hl=es-419>
- Hacer clic en la opción Run del menú superior



## Creación de proyecto en Android Studio

### Paso 6

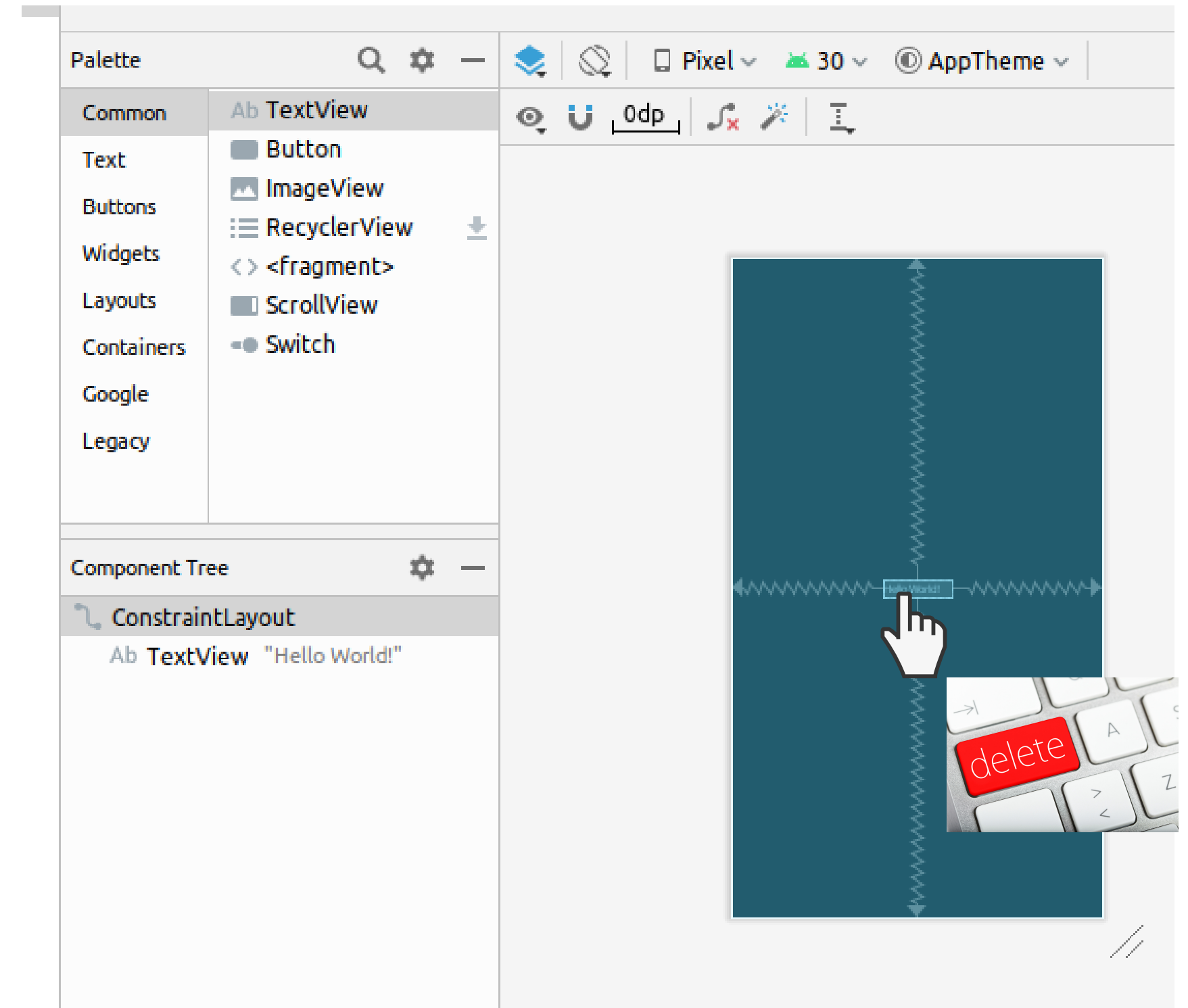
- Verificar que la aplicación se inicie en el emulador



# Creación de proyecto en Android Studio

## Paso 7

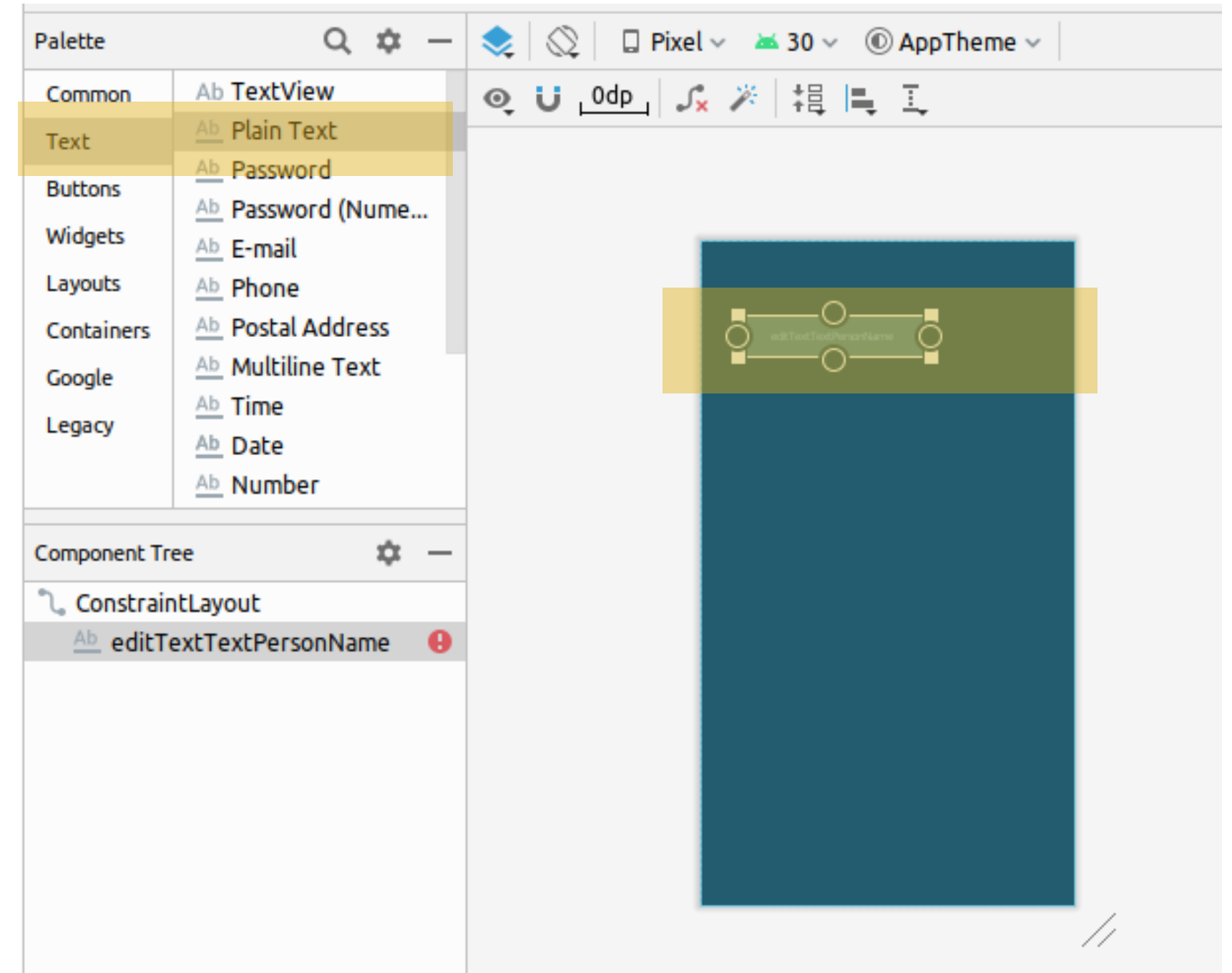
- Agregar un diseño personalizado a la aplicación. Que se permita ingresar un texto, se pulse un botón y que se redirija la aplicación a una nueva pantalla que muestre el mensaje ingresado como texto
  - **Eliminar el texto "Hello World!!"**
  - Agregar un cuadro de texto
  - Agregar un botón



## Creación de proyecto en Android Studio

### Paso 8

- Agregar un diseño personalizado a la aplicación. Que se permita ingresar un texto, se pulse un botón y que se redirija la aplicación a una nueva pantalla que muestre el mensaje ingresado como texto
  - Eliminar el texto "Hello World!!"
  - **Agregar un cuadro de texto**
  - Agregar un botón

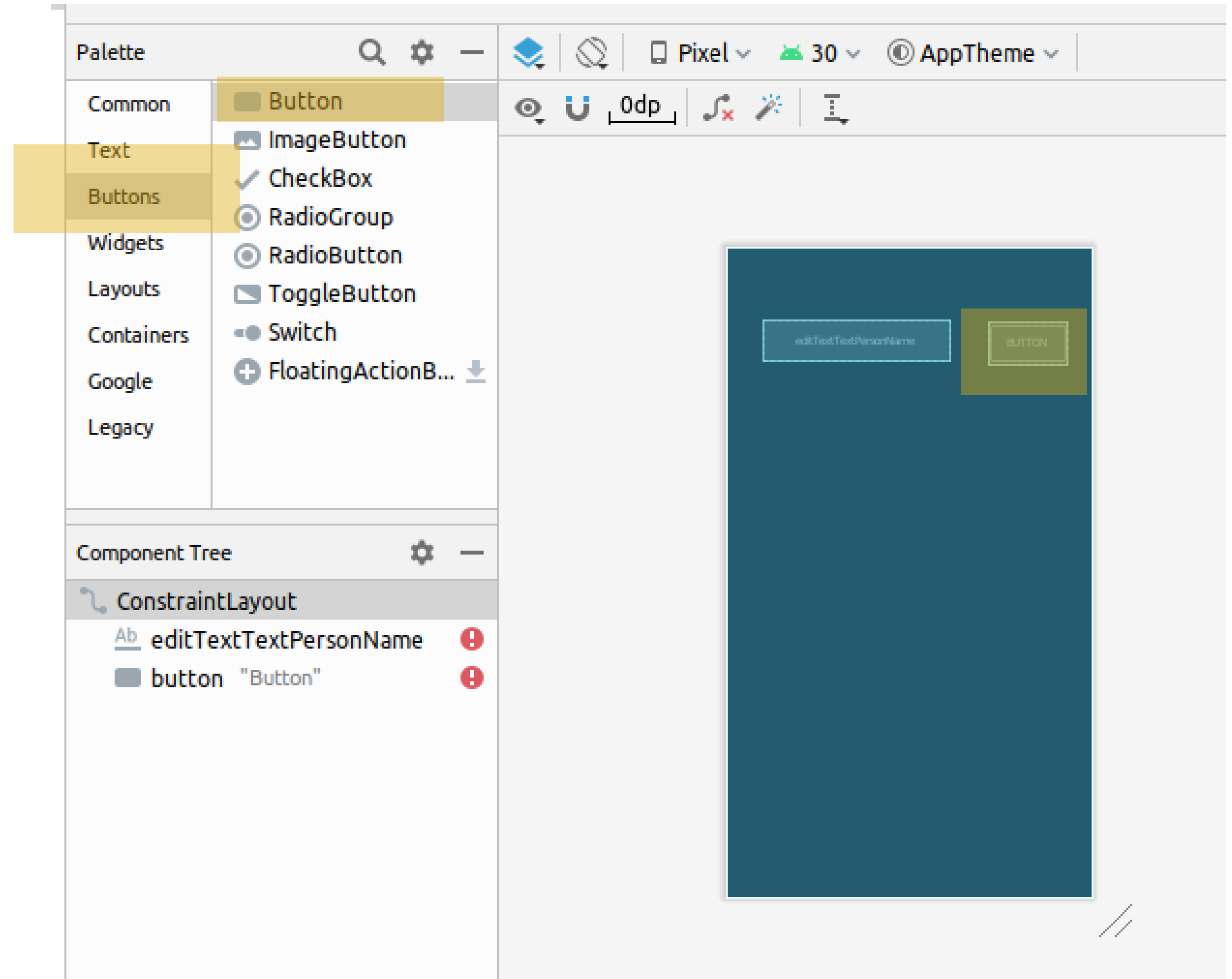




## Creación de proyecto en Android Studio

### Paso 9

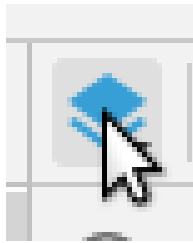
- Agregar un diseño personalizado a la aplicación. Que se permita ingresar un texto, se pulse un botón y que se redirija la aplicación a una nueva pantalla que muestre el mensaje ingresado como texto
  - Eliminar el texto "Hello World!!"
  - Agregar un cuadro de texto
  - **Agregar un botón**



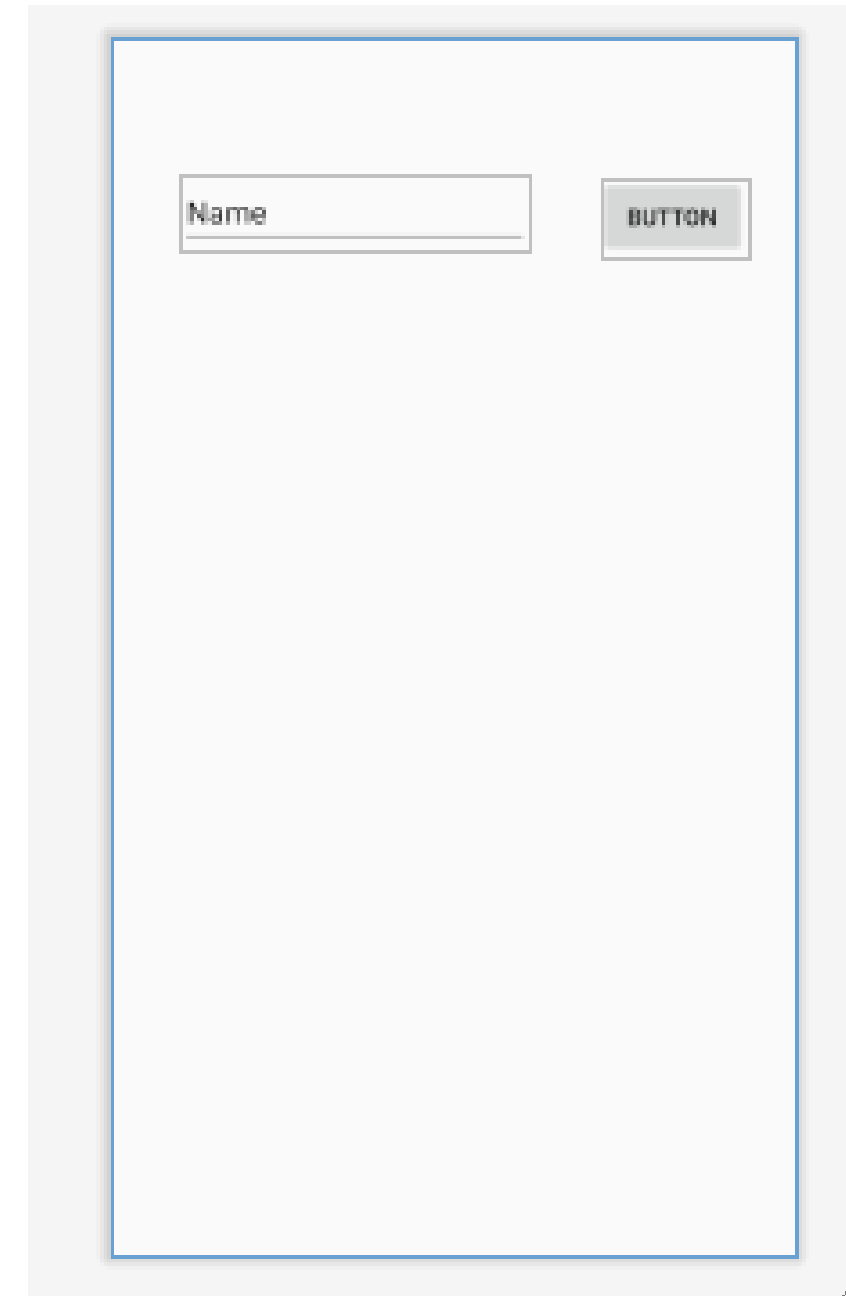
## Creación de proyecto en Android Studio

### Paso 10

- Verificar el diseño de la aplicación
  - En la opción (Select Design Surface), seleccionar Design



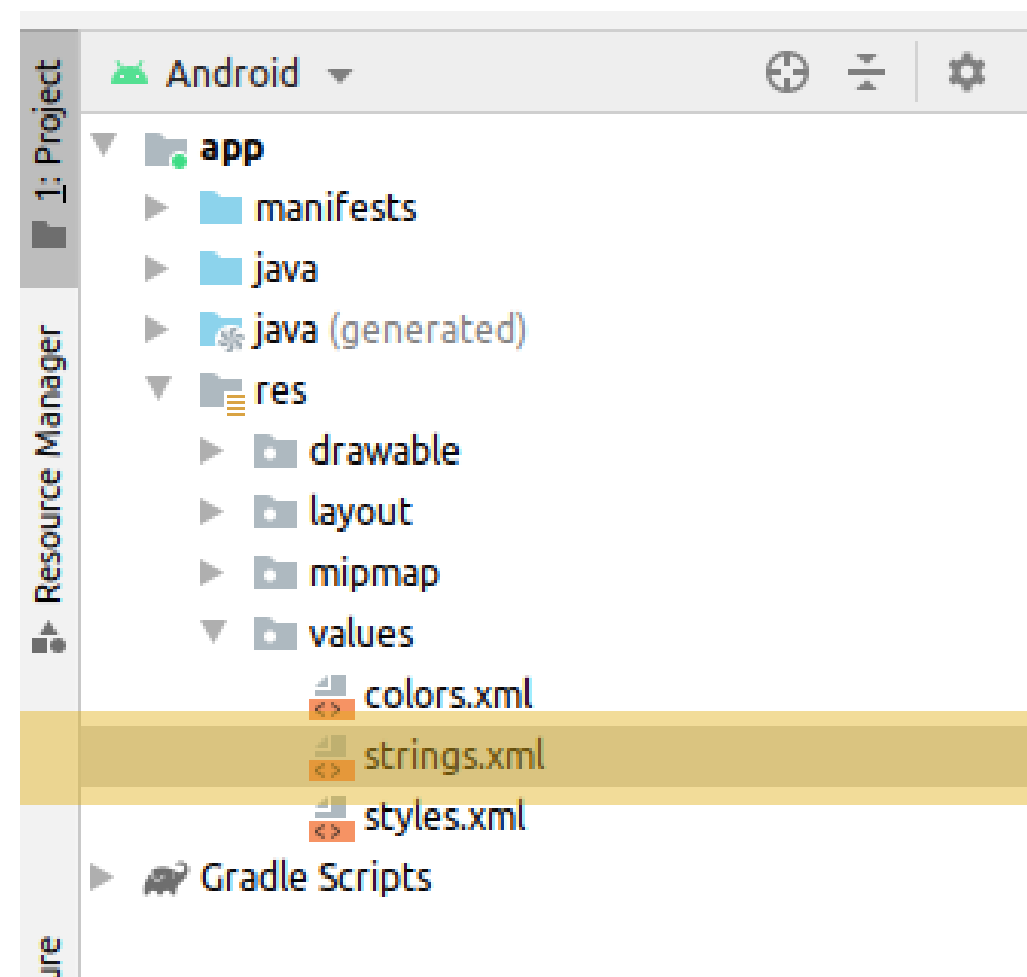
- Se debe visualizar lo siguiente



## Creación de proyecto en Android Studio

### Paso 11

- Agregar información en el archivo strings.xml, que está en la carpeta



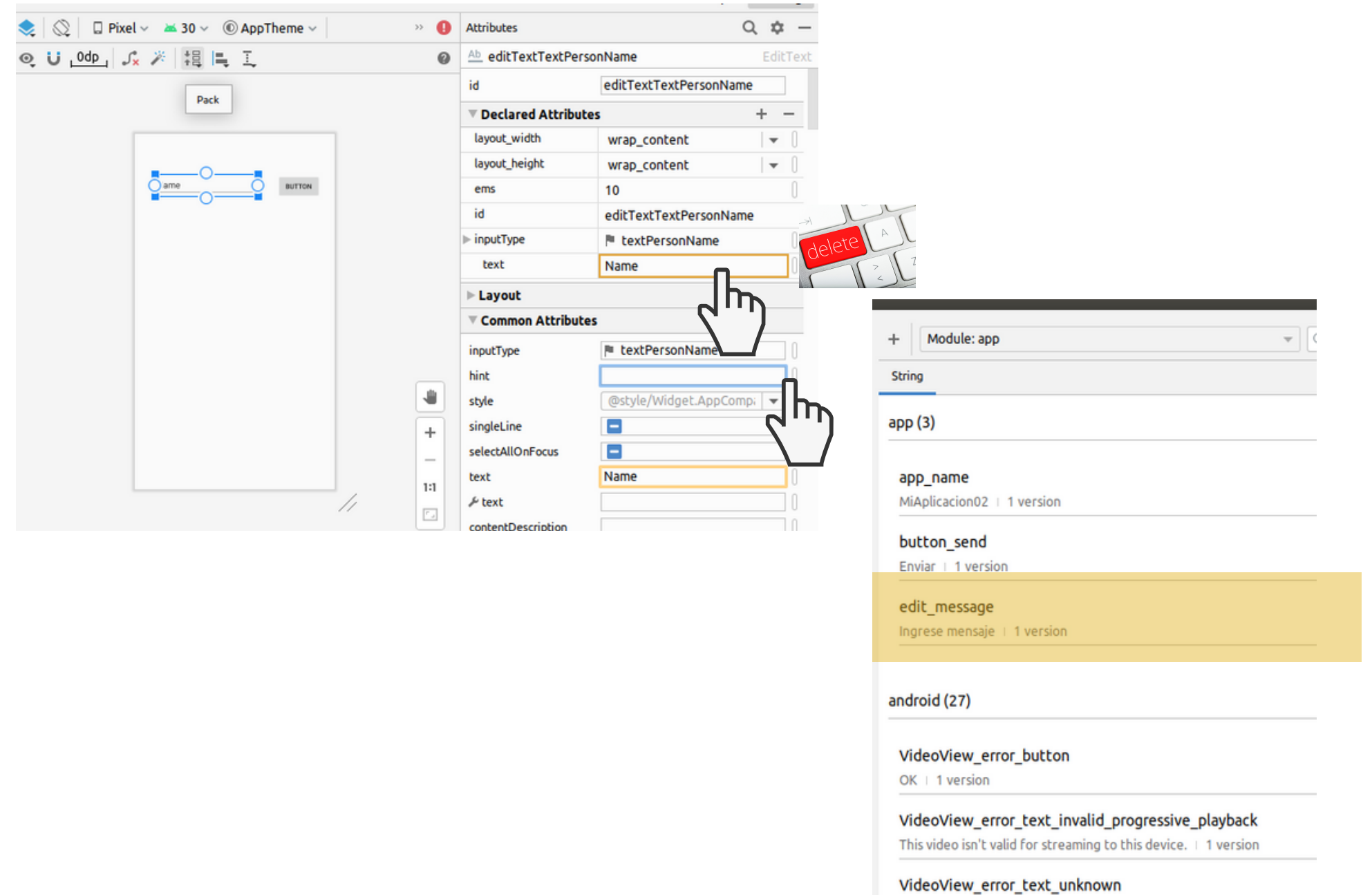
- Dejar el archivo como se muestra a continuación

```
1 <resources>
2     <string name="app_name">MiAplicacion02</string>
3     <string name="edit_message">Ingrese mensaje</string>
4     <string name="button_send">Enviar</string>
5 </resources>
```

# Creación de proyecto en Android Studio

## Paso 12

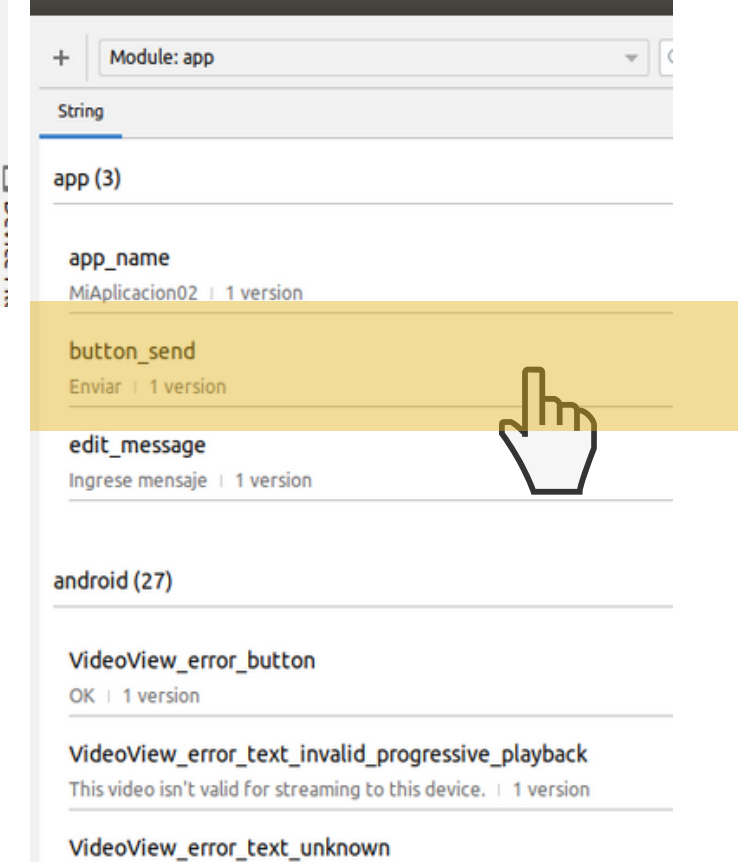
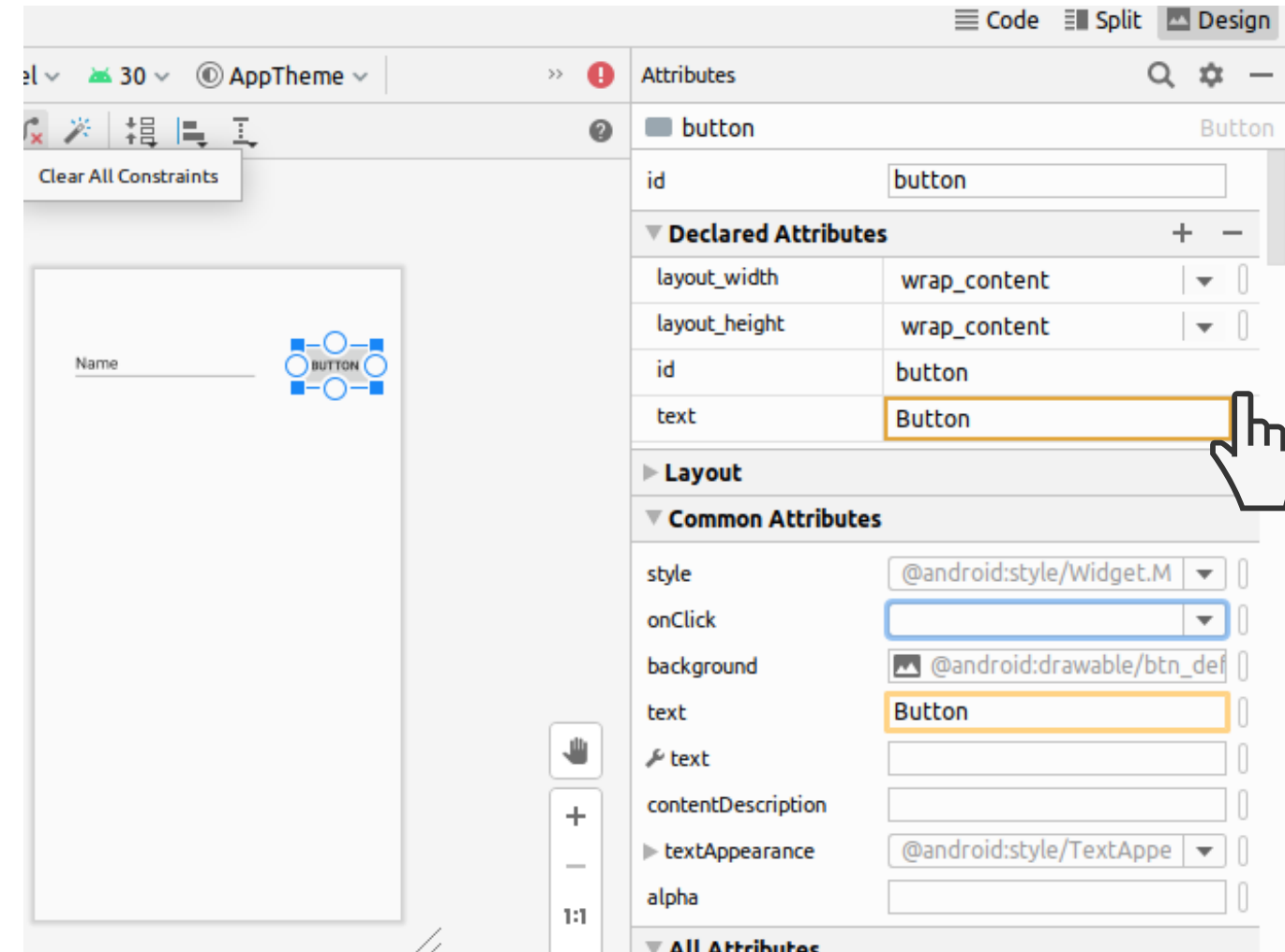
- Agregar las cadenas generadas cuadro de texto y el botón



## Paso 13

- Agregar las cadenas generadas cuadro de texto y el botón

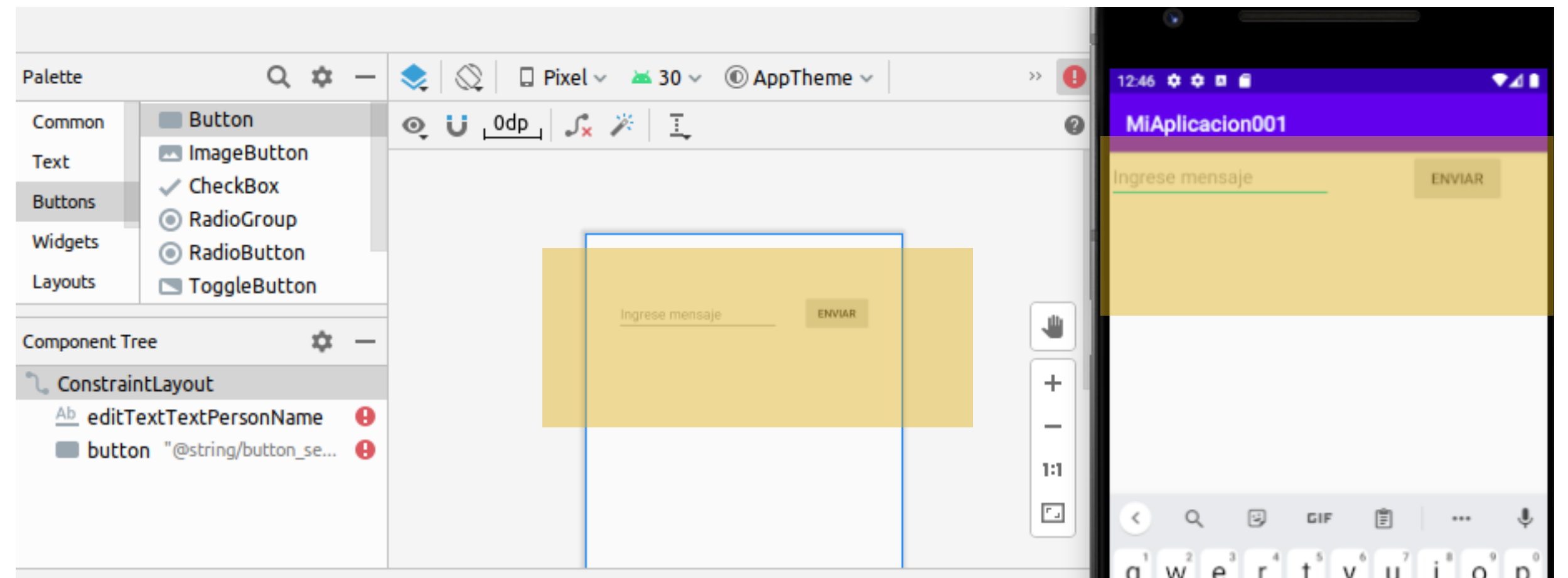
## Creación de proyecto en Android Studio



# Creación de proyecto en Android Studio

## Paso 14

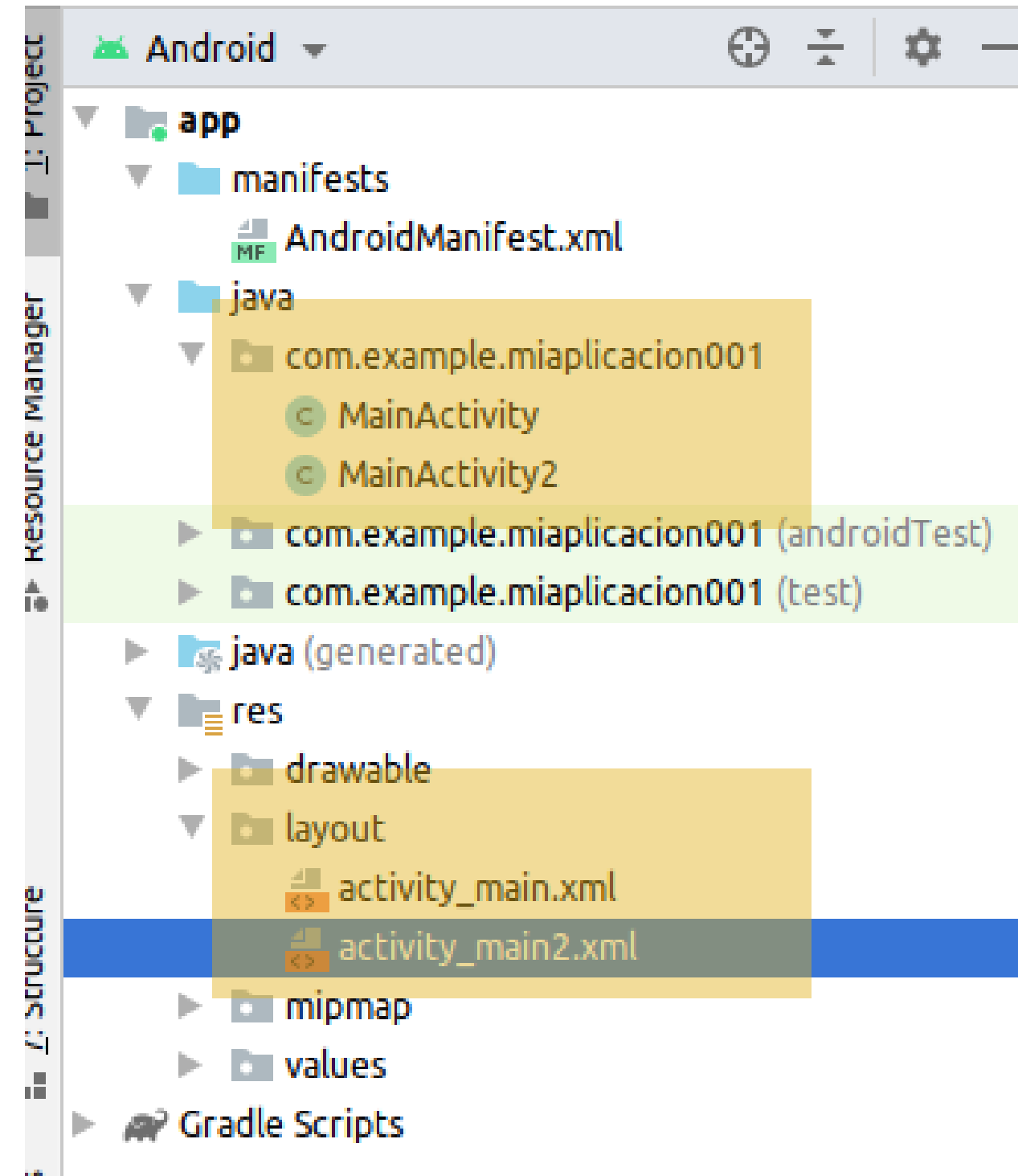
- Verificar el estado de la interfaz



## Creación de proyecto en Android Studio

### Paso 15

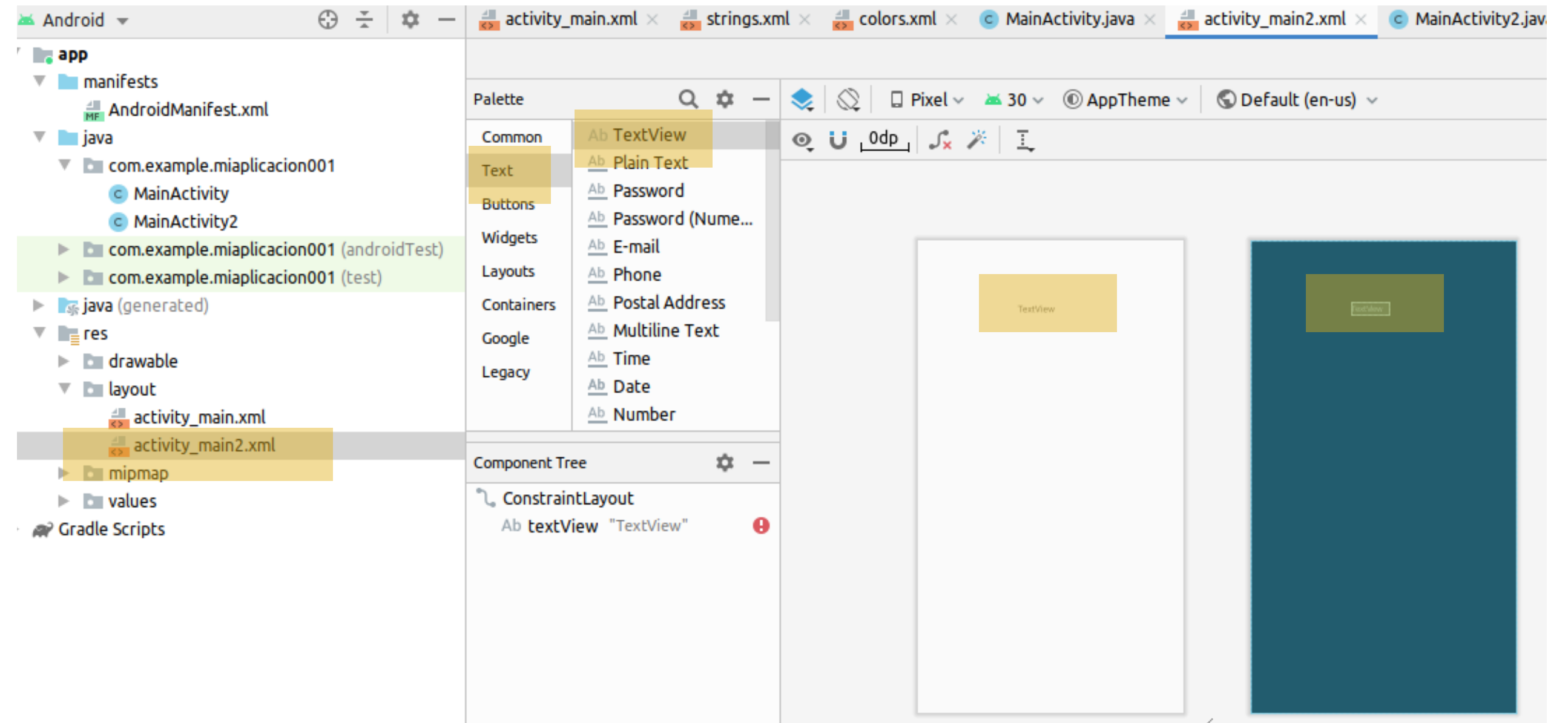
- Agregar una nueva actividad llamada MainActivity2
  - La creación de la actividad, asume la creación de archivo activity\_main2.xml



# Creación de proyecto en Android Studio

## Paso 16

- En el archivo activity\_main2.xml agregar un TextView





## Creación de proyecto en Android Studio

### Paso 17

- Modificación de archivos
  - El archivo **MainActivity**, debe tener la siguiente estructura

```
1 package com.example.miaplicacion001;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.EditText;
8
9 public class MainActivity extends AppCompatActivity {
10     public static final String EXTRA_MESSAGE = "com.example.myfirstapp.MESSAGE";
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16     }
17
18     public void sendMessage(View view) {
19         // Do something in response to button
20         Intent intent = new Intent( packageContext: this, MainActivity2.class);
21         EditText editText = (EditText) findViewById(R.id.editTextTextPersonName);
22         String message = editText.getText().toString();
23         intent.putExtra(EXTRA_MESSAGE, message);
24         startActivity(intent);
25     }
26 }
```

## Creación de proyecto en Android Studio

### Paso 18

- Modificación de archivos
  - El archivo **MainActivity2**, debe tener la siguiente estructura

```
1 package com.example.miaplicacion001;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 public class MainActivity2 extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main2);
14
15         // Get the Intent that started this activity and extract the string
16         Intent intent = getIntent();
17         String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
18
19         // Capture the layout's TextView and set the string as its text
20         TextView textView = findViewById(R.id.textView);
21         textView.setText(message);
22     }
23 }
```

## Paso 19

- Modificación de archivos
  - El archivo **AndroidManifest.xml**, debe tener la siguiente estructura

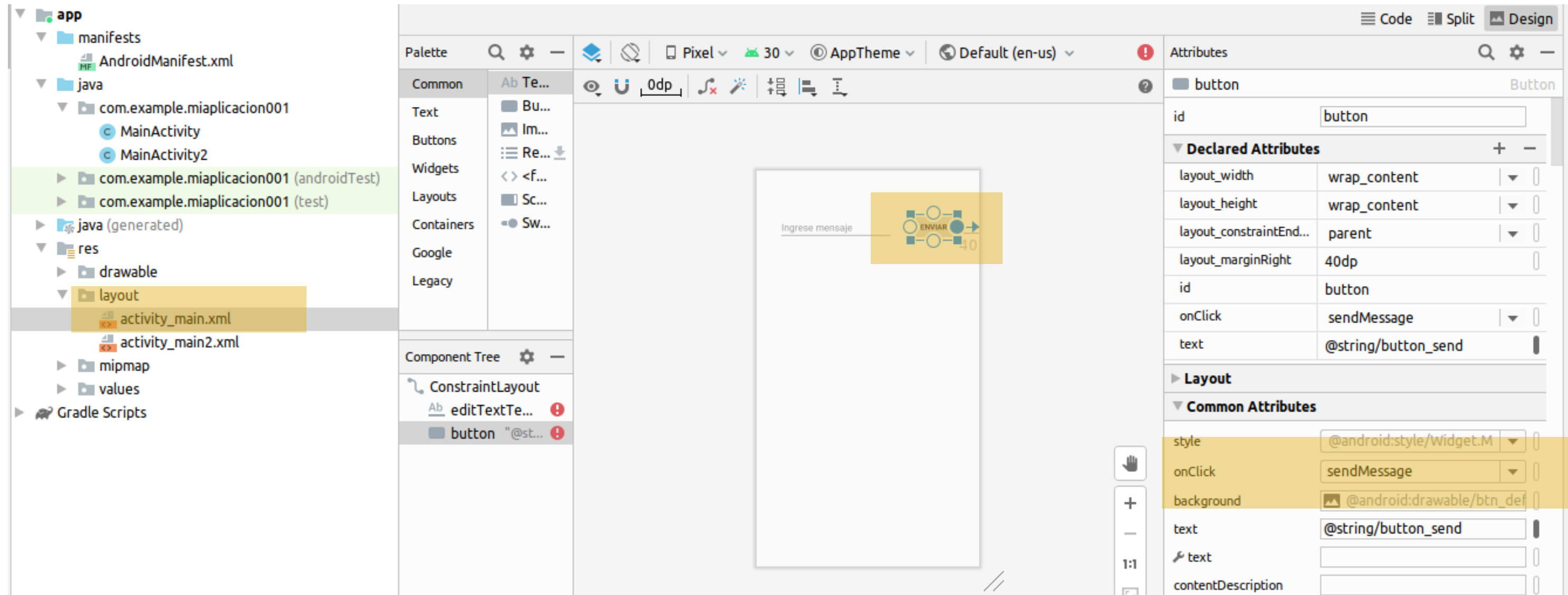
## Creación de proyecto en Android Studio

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3         package="com.example.miaplicacion001">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="MiAplicacion001"
9          android:roundIcon="@mipmap/ic_launcher_round"
10         android:supportRtl="true"
11         android:theme="@style/AppTheme">
12
13         <activity android:name=".MainActivity2">
14             android:parentActivityName=".MainActivity">
15                 <!-- The meta-data tag is required if you support API level 15 and lower -->
16                 <meta-data
17                     android:name="android.support.PARENT_ACTIVITY"
18                     android:value=".MainActivity" />
19             </activity>
20             <activity android:name=".MainActivity">
21                 <intent-filter>
22                     <action android:name="android.intent.action.MAIN" />
23
24                     <category android:name="android.intent.category.LAUNCHER" />
25                 </intent-filter>
26             </activity>
27         </application>
28     </manifest>
```

# Creación de proyecto en Android Studio

## Paso 20

- Agregar la acción al botón, accediendo a la pestaña Design del archivo **activity\_main.xml**



## Creación de proyecto en Android Studio

### Paso 21

- Ejecutar la aplicación y comprobar el funcionamiento.
  - Ingresar una cadena de texto en la pantalla 1, pulsar **Enviar** y esperar la presentación de la cadena en una nueva pantalla

