



Asignatura: Desarrollo de Aplicaciones Móviles Nativas.
Temas: Programación gráfica de componentes.

El IDE de Android Studio permite crear elementos gráficos para la interfaz de una aplicación sin la necesidad de utilizar el código. Esta función ayuda a diseñar el aspecto visual de las aplicaciones en una forma sencilla.

Los elementos de la paleta de Android Studio son un aspecto esencial de la aplicación con los cuales el usuario tiene mayor interacción, habiendo de distintos tipos, por ejemplo, los siguientes:

Layouts, RadioButton y CheckBox, Switch y Toggle Buttons, Buttons, Tipos Text, TextFields, entre otros.

Además, también se incluyen varios elementos para expertos; algunos son los siguientes:

Espacio, TextView de chequeo, TextView de autocompletado, Botones de Zoom (+ y -), Controlador de Media, Vista de gestos de pantalla, y otros.

PARTE I.

Diseño de una interface gráfica móvil con Android Studio.

1. Crear un nuevo proyecto seleccionando New Project. En la siguiente ventana, seleccionar EmptyViewsActivity.

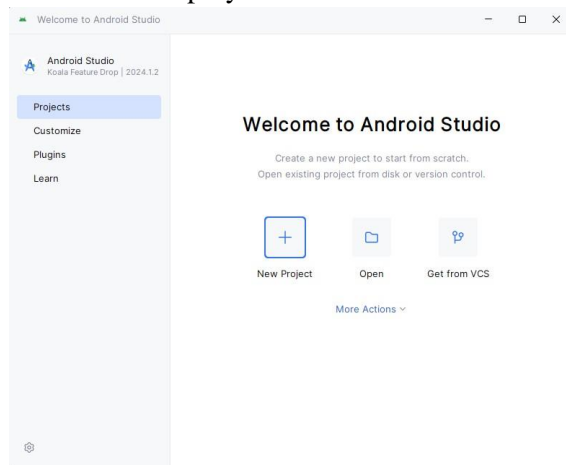


Figura 1. Inicio del IDE de Android Studio.

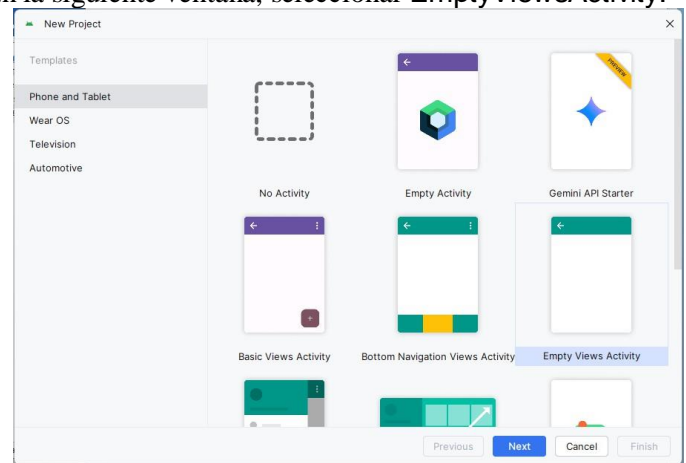


Figura 2. Selección de Empty Views Activity.

2. Asignar un nombre al proyecto, por ejemplo, con ProgramacionGrafica. Seleccionar el lenguaje Java. En la opción Minimum SDK seleccionar la API 31, como se indica en la siguiente figura 3. Digitar el botón Finish.

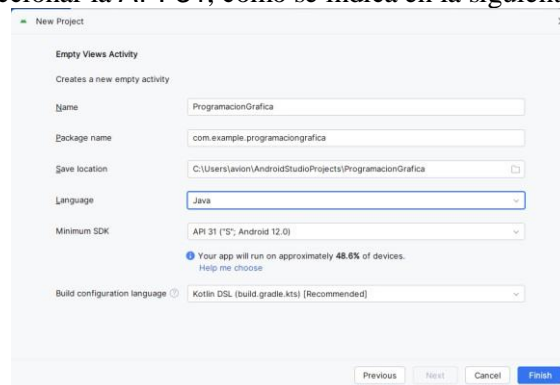


Figura 3. Configuración del proyecto,

3. Enseguida, se muestra la ventana del IDE de Android Studio. Expandir las carpetas del repositorio para que la ventana sea similar a la imagen de la figura 4.

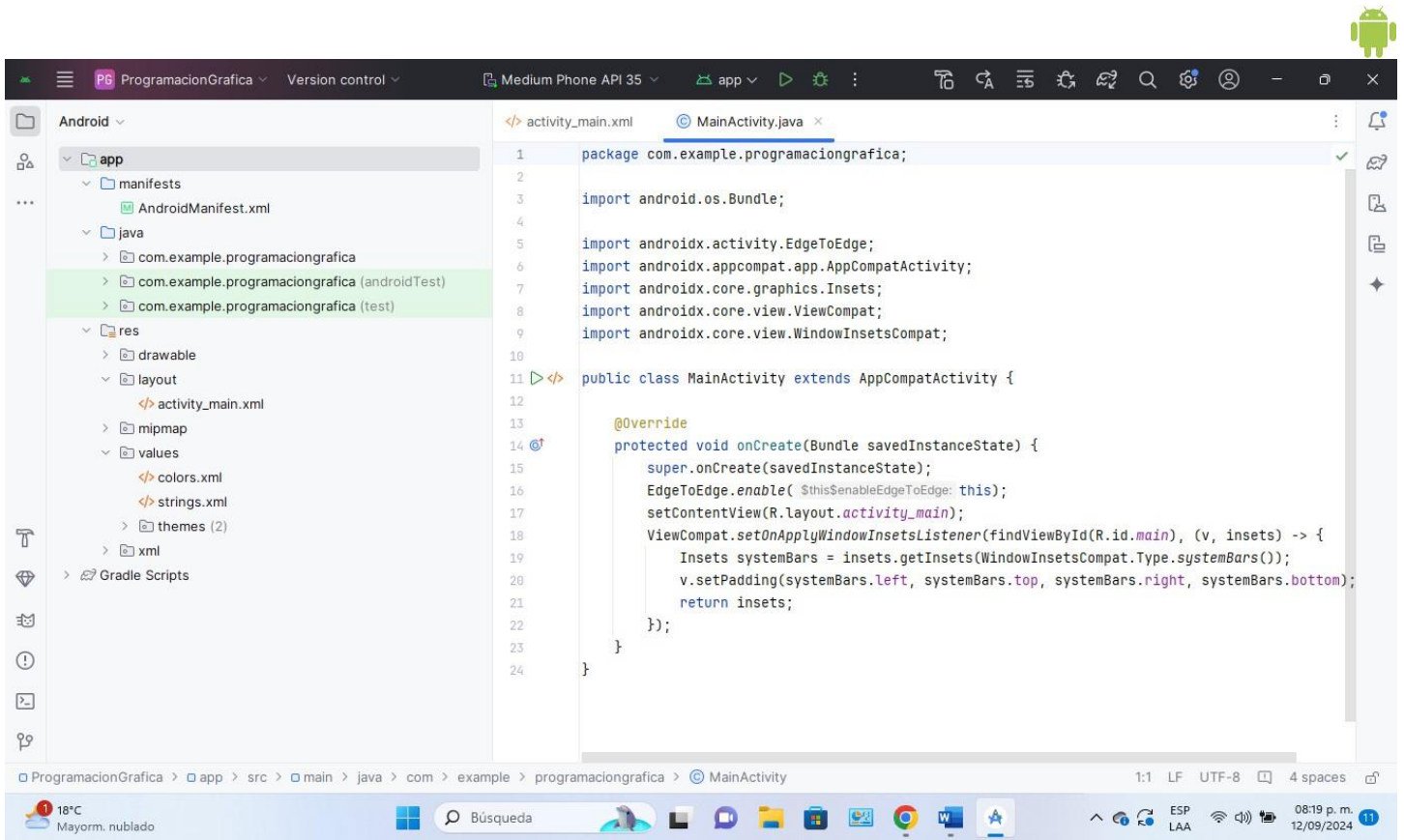


Figura 4. El IDE de Android Studio con el repositorio y el editor de código.

4. Seleccionar y abrir el archivo *activity_main.xml*. En las pestañas correspondientes **Code Split Design**, digitar en la pestaña **Design**.

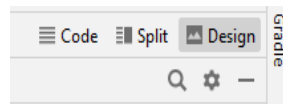


Figura 5. Las pestañas de la región designada para el archivo *activity_main.xml*.

5. En la ventana de componentes, seleccionar el **TextView** mostrado y borrarlo para limpiar la plantilla predeterminada.

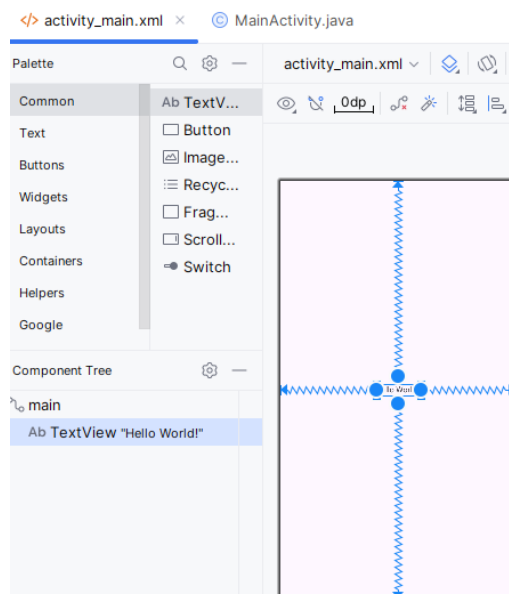


Figura 6. La paleta de componentes y el componente TextView seleccionado.



Todos los componentes gráficos poseen nodos, o pequeños círculos laterales para posicionar los componentes en las plantillas.

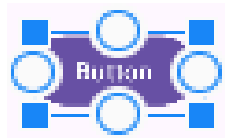


Figura 7. Los nodos de un componente.

6. En la paleta, seleccionar y arrastrar un componente **Button** hacia la plantilla. Seleccionar los nodos, izquierdo, derecho e inferior, para centrar el componente en la parte inferior de la pantalla. Ver la figura 8.

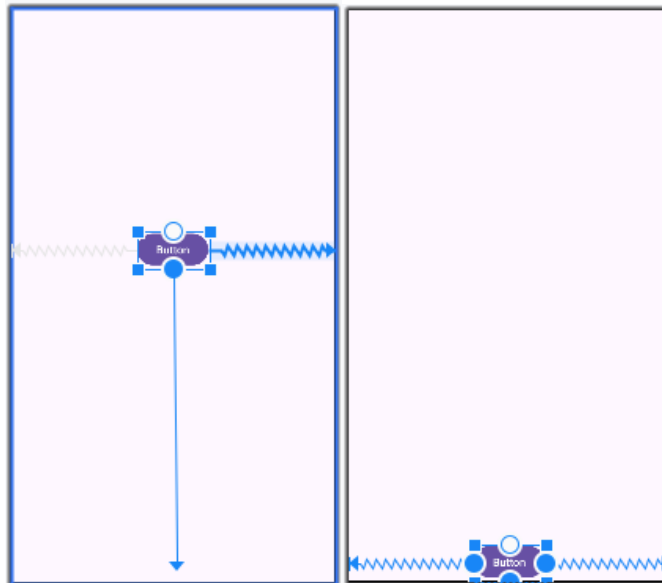


Figura 8. Selección de un botón y su posición en la plantilla.

7. En la sección derecha **Attributes** del IDE, en el atributo `layout_width` seleccionar la opción `0dp` (`match constraint`). El componente **Button** ahora ocupará el ancho de la pantalla como se indica en las siguientes figuras.

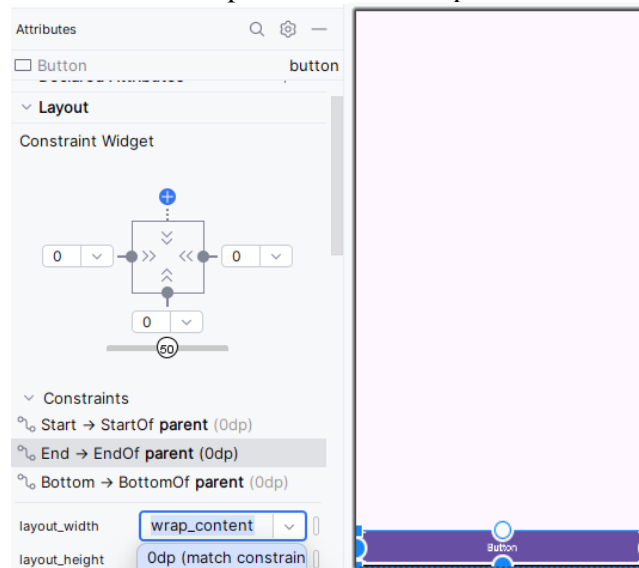


Figura 9. Los atributos del componente **Button**.

8. En la paleta de componentes **Layouts**, seleccionar y arrastrar **LinearLayout** (vertical) al centro de la plantilla.

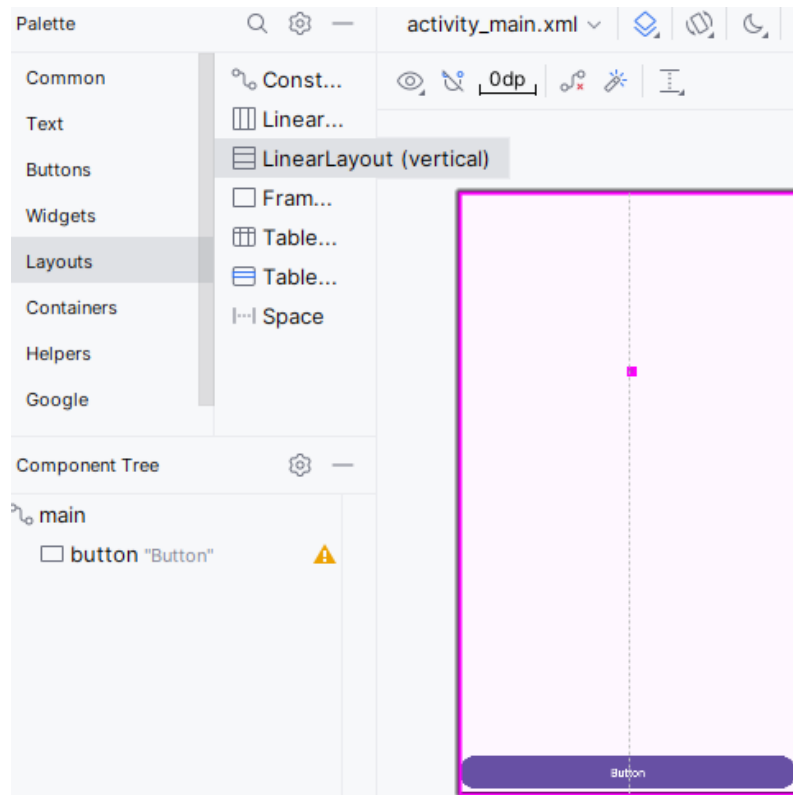


Figura 9. Selección de LinearLayout (vertical).

Enseguida, redimensionar la plantilla **LinearLayout** y posicionarla con sus cuatro nodos laterales a la plantilla principal. En la sección **Attributes** y en los atributos **layout_width** y **layout_height**, seleccionar **0dp**. Ver la figura 10.

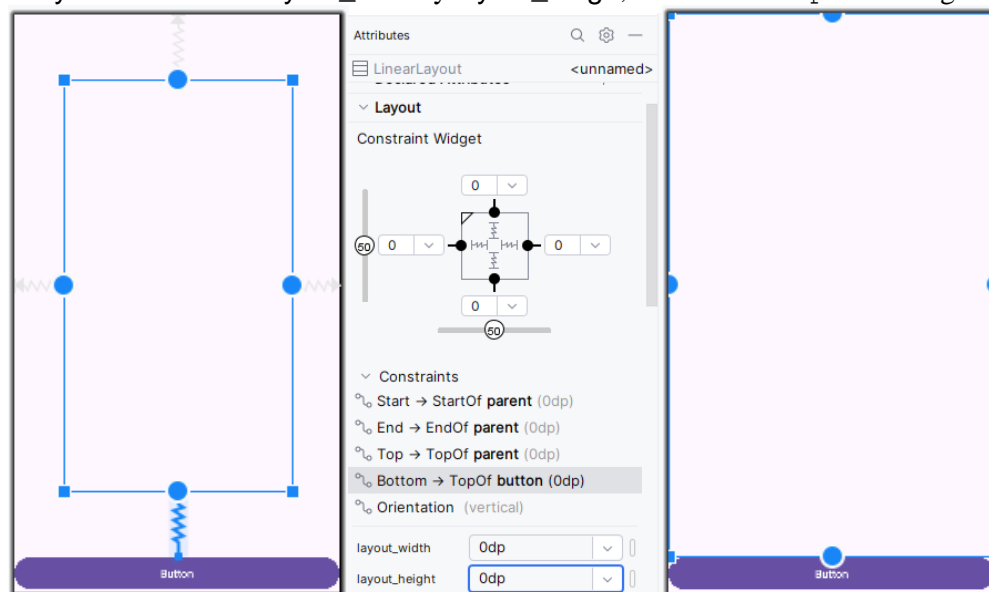


Figura 10. Redimensión del componente LinearLayout.

9. En la paleta de componentes, seleccionar **Buttons** y el componente **Button** arrastrarlo hacia la parte inferior de la **paleta** en la sección **Component Tree**, justo abajo del componente **LinearLayout**. El botón se acomodará automáticamente en la plantilla gráfica.

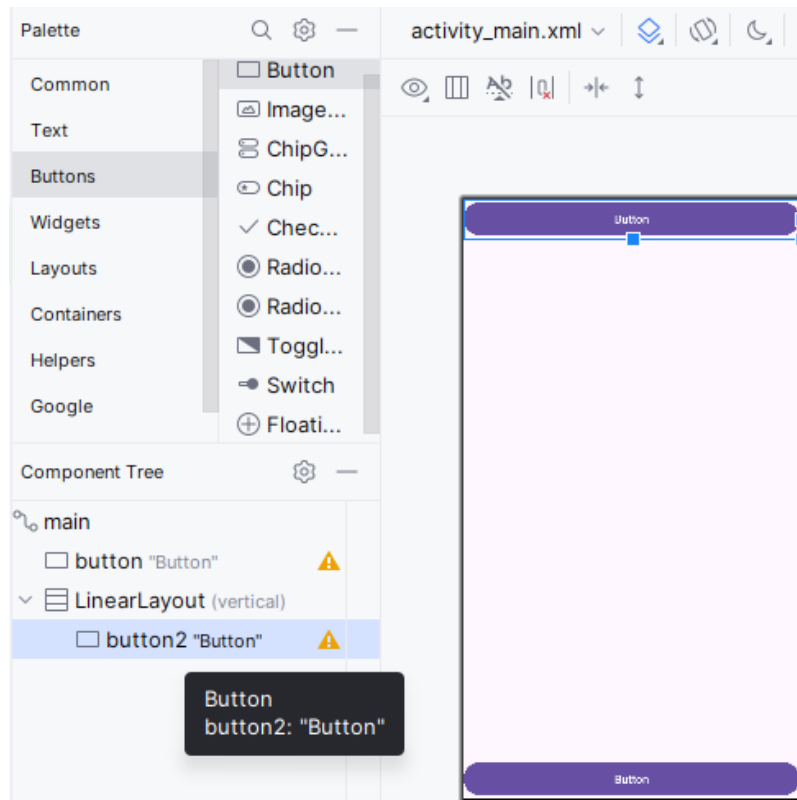


Figura 11. Agregar otro componente Button.

10. En la paleta, seleccionar **Containers** y **Spinner**. El componente **Spinner** es una lista desplegable/contraíble. Arrastrar el componente **Spinner** hacia la parte inferior en **Component Tree**, justo debajo del botón 2. El **Spinner** se acomodará automáticamente en la plantilla gráfica. Ver las figuras 12a y 12b.

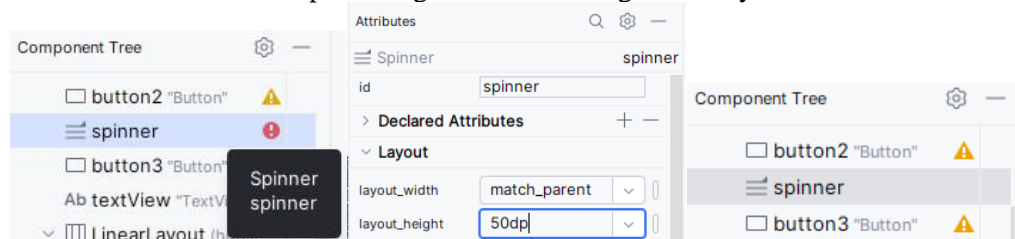


Figura 12a. Selección del componente Spinner.

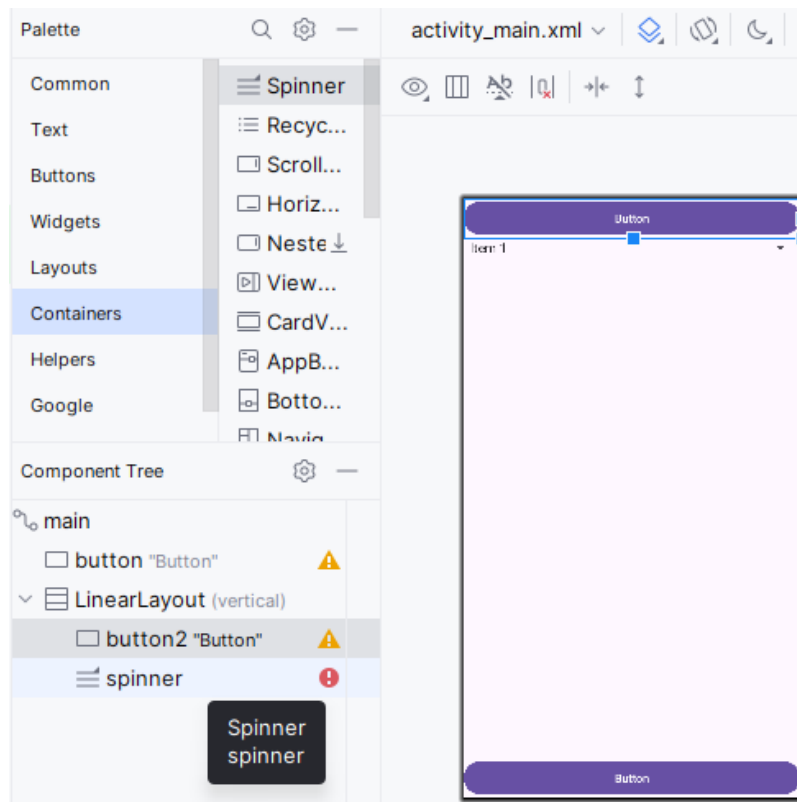


Figura 12b. Agregar el componente Spinner.

11. En la paleta de componentes, seleccionar **Buttons** y el componente **Button** y arrastrarlo hacia la parte inferior de la paleta en la sección Component Tree, justo abajo del componente Spinner. El botón se acomodará automáticamente en la plantilla gráfica.

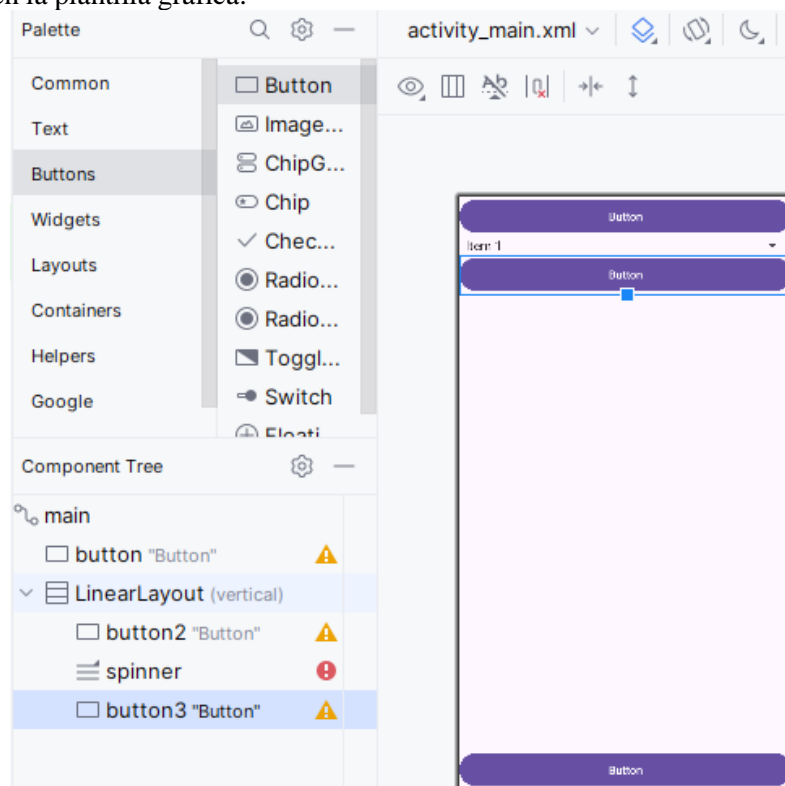


Figura 13. Agregar otro Button.



- En la paleta de componentes, seleccionar **Text** y el componente **TextView** y arrastrarlo hacia la parte inferior de la paleta en la sección **Component Tree**, justo abajo del componente **button3**. El **TextView** se acomodará automáticamente en la plantilla gráfica.

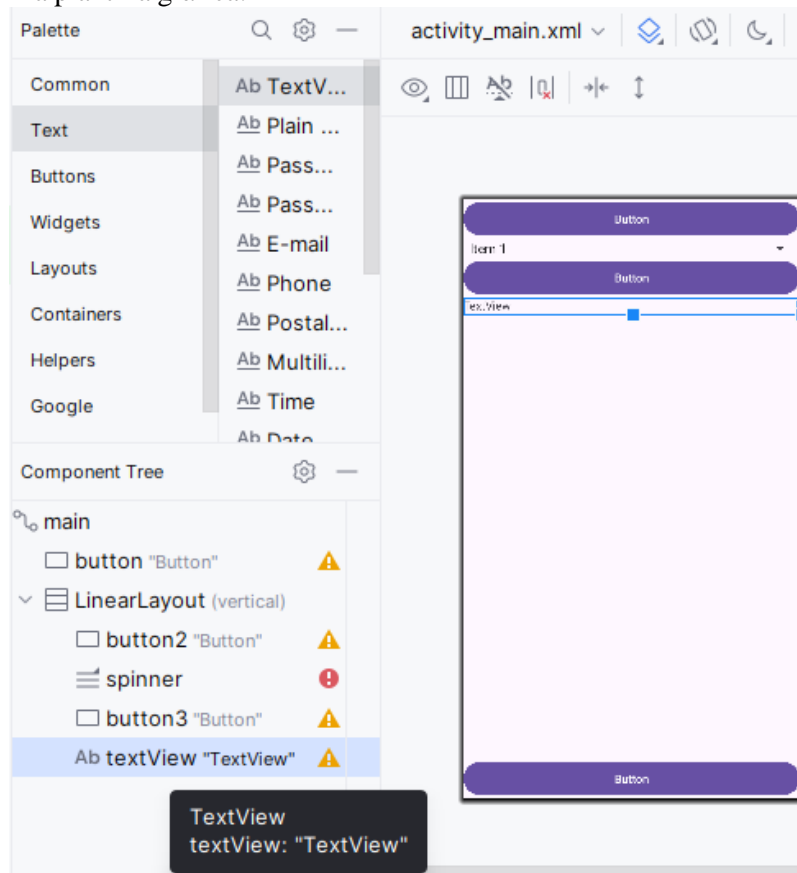


Figura 14. Agregar el componente **TextView**.

- En la paleta de componentes, seleccionar **Layouts** y el componente **LinearLayout** (horizontal) y arrastrarlo hacia la parte inferior de la paleta en la sección **Component Tree**, justo abajo del componente **TextView**. Las plantillas **Layout** se acomodarán automáticamente en la plantilla gráfica.

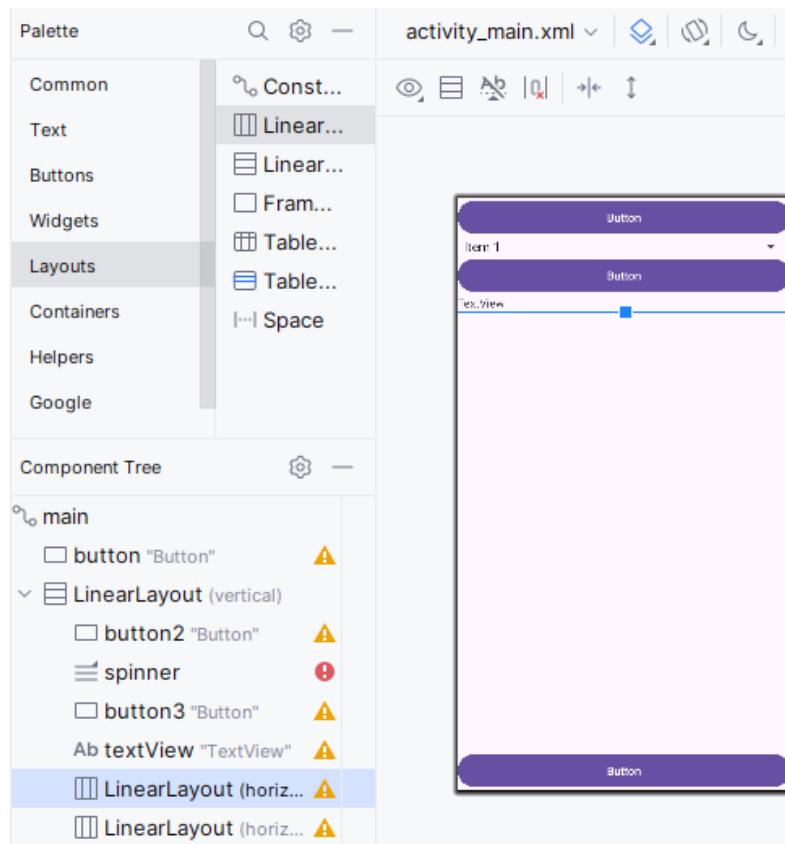


Figura 15. Agregar el componente LinearLayout (horizontal).

Enseguida, en la plantilla gráfica, seleccionar los dos componentes **LinearLayout** recién insertados y en la región de Attributes en el atributo `layout_height` seleccionar la opción `wrap_content`.

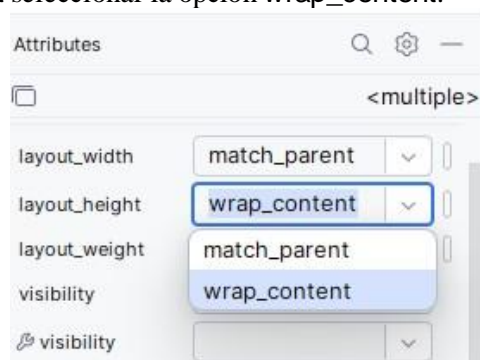


Figura 16. Los atributos de LinearLayout.

14. En la sección **Componente Tree**, seleccionar el primer componente **LinearLayout** recién insertado. Enseguida, agregar dos botones arrastrándolos hacia esta misma sección y depositarlos, uno por uno, en esta plantilla.

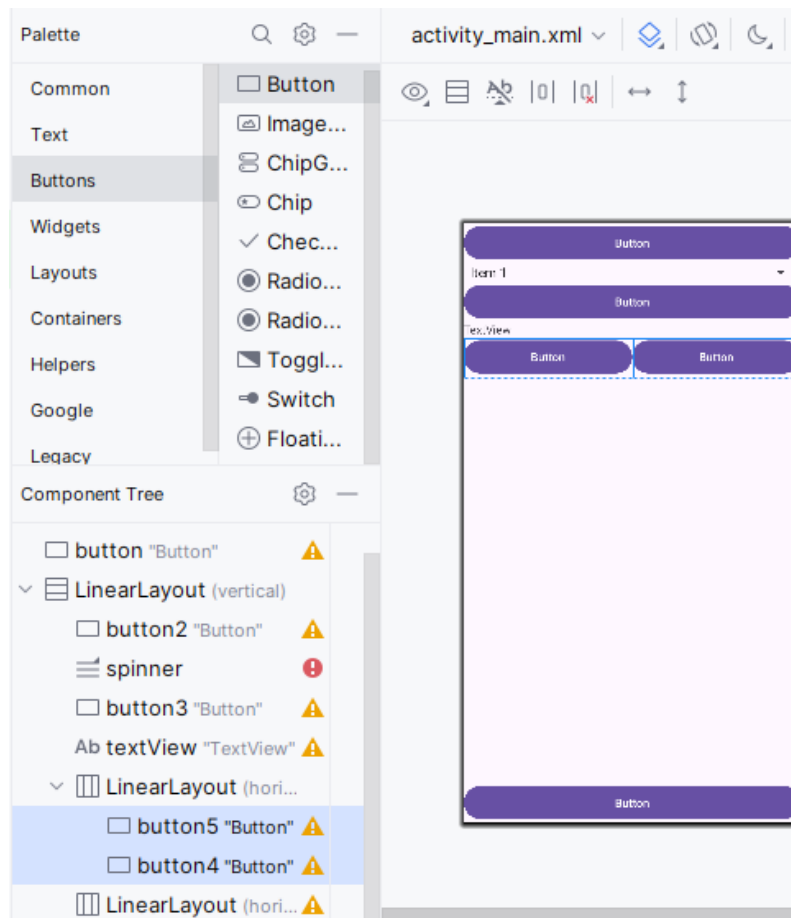


Figura 16. Agregar dos botones a la plantilla LinearLayout (horizontal).

15. Para terminar la inserción de componentes, otra vez, en la sección **Componente Tree**, seleccionar el segundo componente **LinearLayout** recién insertado. Enseguida, agregar dos botones arrastrándolos hacia esta misma sección y depositarlos, uno por uno, en esta segunda plantilla. En la figura 17 se muestra la versión parcial de esta etapa de acomodo de los componentes en una pantalla principal.

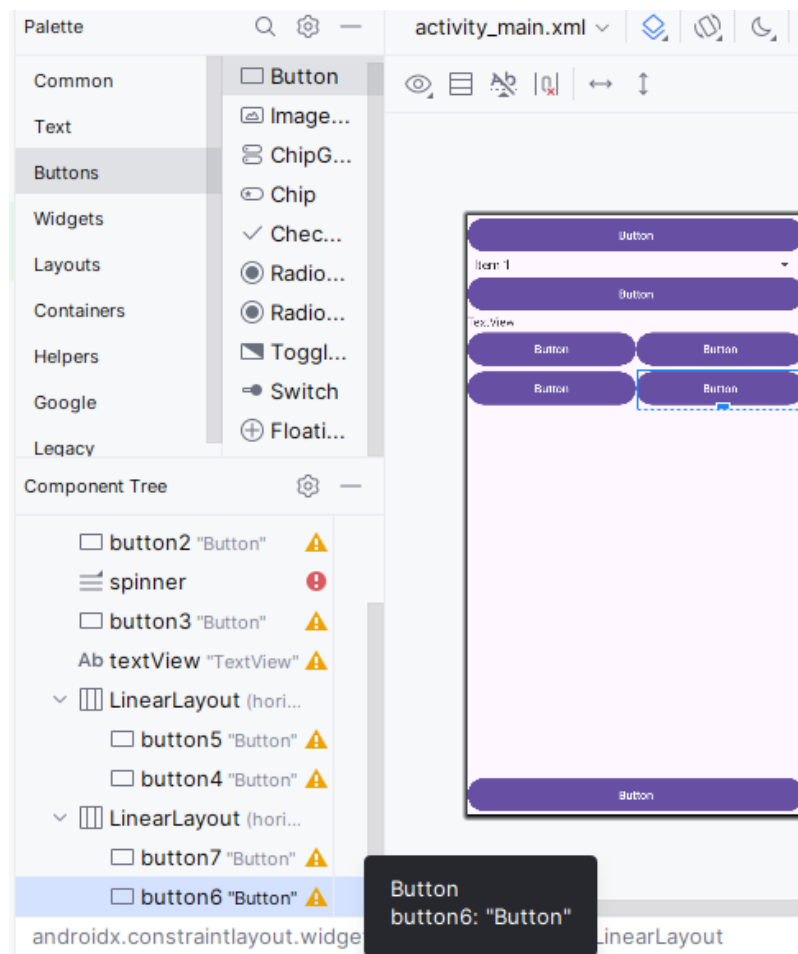


Figura 17. Agregar otros dos botones más a la plantilla LinearLayout (horizontal).

PARTE II.

Una vez que se han insertado los componentes gráficos, se asigna el contenido textual con el nombre de cada componente.

16. Seleccionar el **botón2** superior. En la región **Attributes**, en el atributo **text** ingresar el texto **Buscar** y digitar **Enter**. En el atributo **id** ingresar el identificador **IdBtnBuscar**. En el diálogo **Rename** digitar **Refactor**.

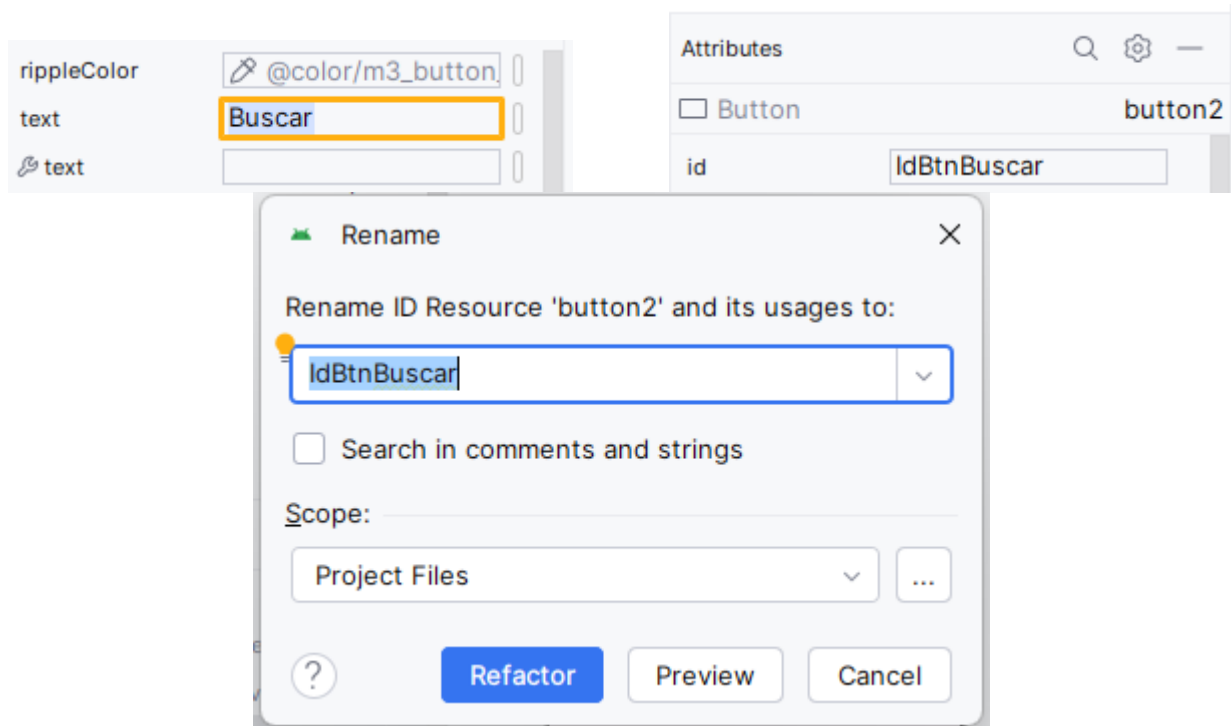


Figura 18. Asignación del nombre Buscar del botón.

17. Seleccionar el **botón3**. En la región Attributes, en el atributo text ingresar el texto Conectar y digitar Enter. En el atributo id ingresar el identificador IdBtnConectar. En el diálogo Rename digitar Refactor.

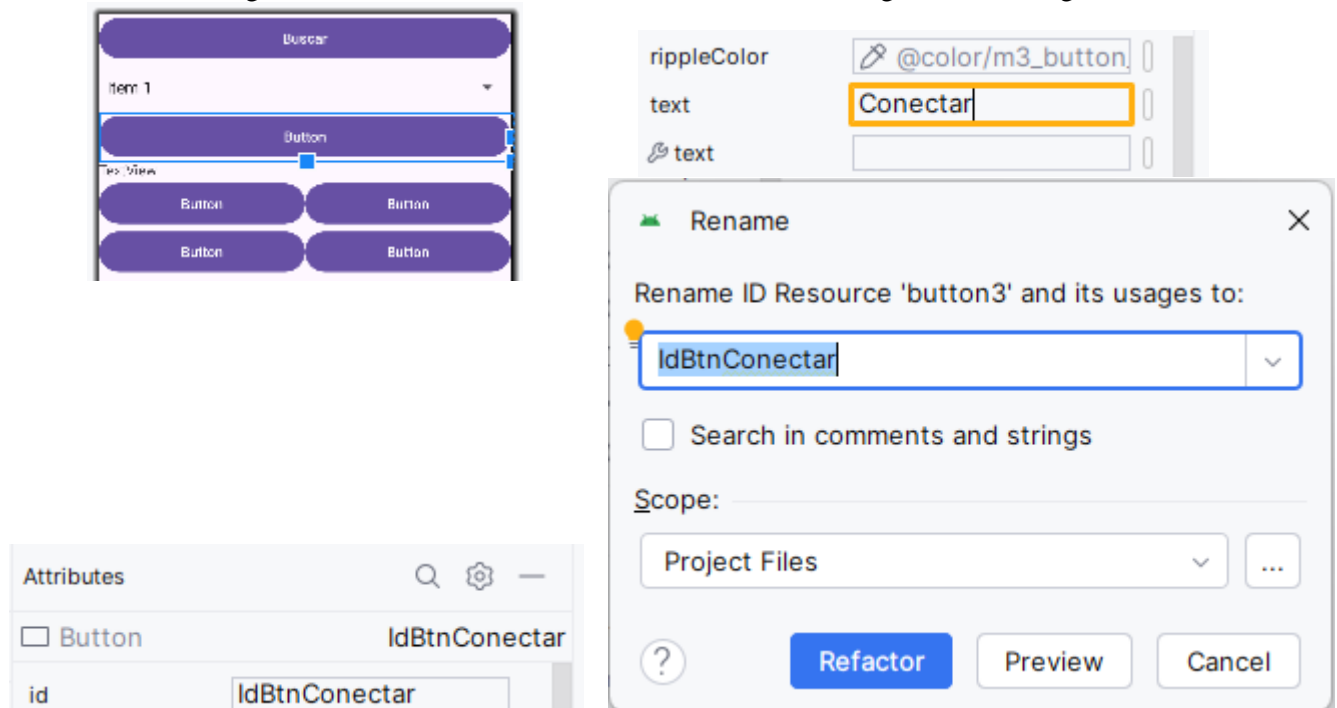


Figura 19. Asignación del nombre Conectar al botón.



18. Seleccionar la etiqueta **TextView**. En la región **Attributes**, en el atributo **text** ingresar el texto **CONTROL LEDS** y digitar **Enter**. En el atributo **textSize** seleccionar el tamaño **16sp**. En el atributo **textStyle** habilitar la selección **bold**.

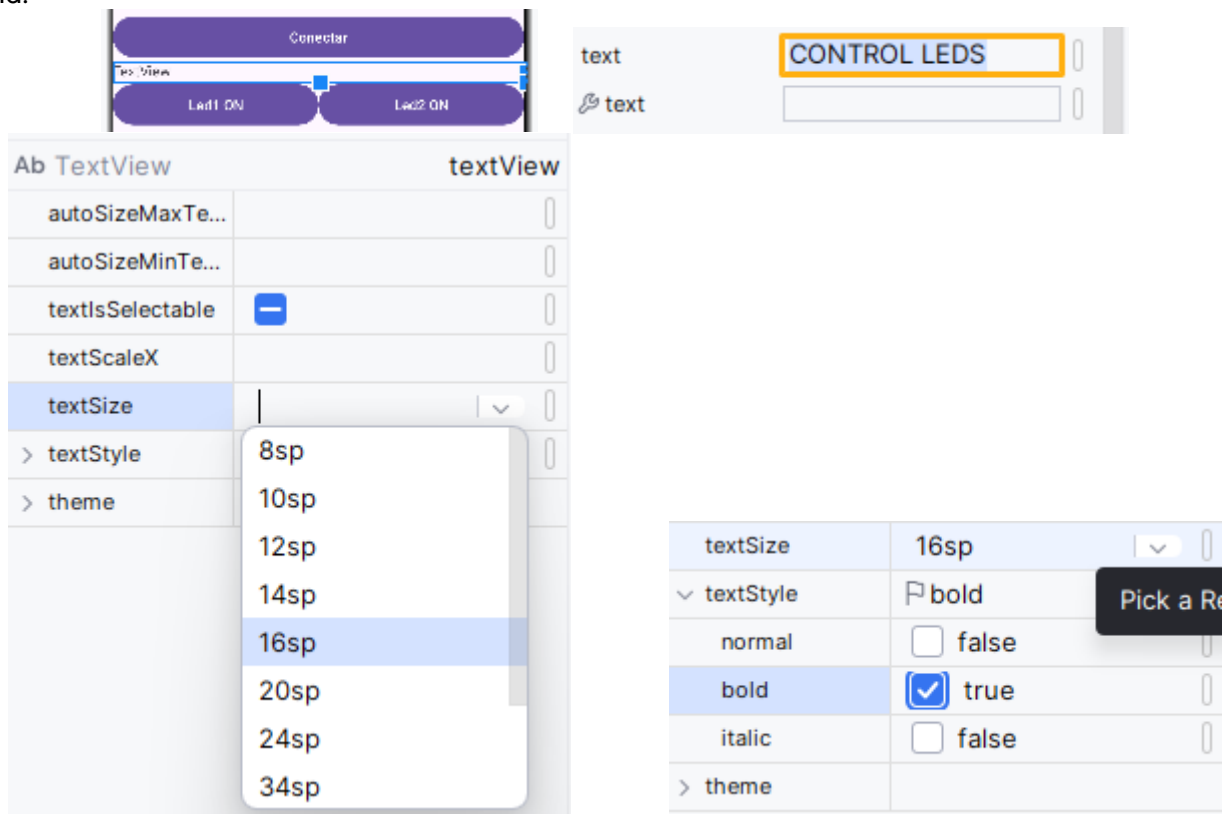


Figura 20. Cambio de los atributos TextView.

19. Seleccionar el **botón5**. En la región **Attributes**, en el atributo **text** ingresar el texto **Led1 ON** y digitar **Enter**. En el atributo **id** ingresar el identificador **IdBtnLed1on**. En el diálogo **Rename** digitar **Refactor**.

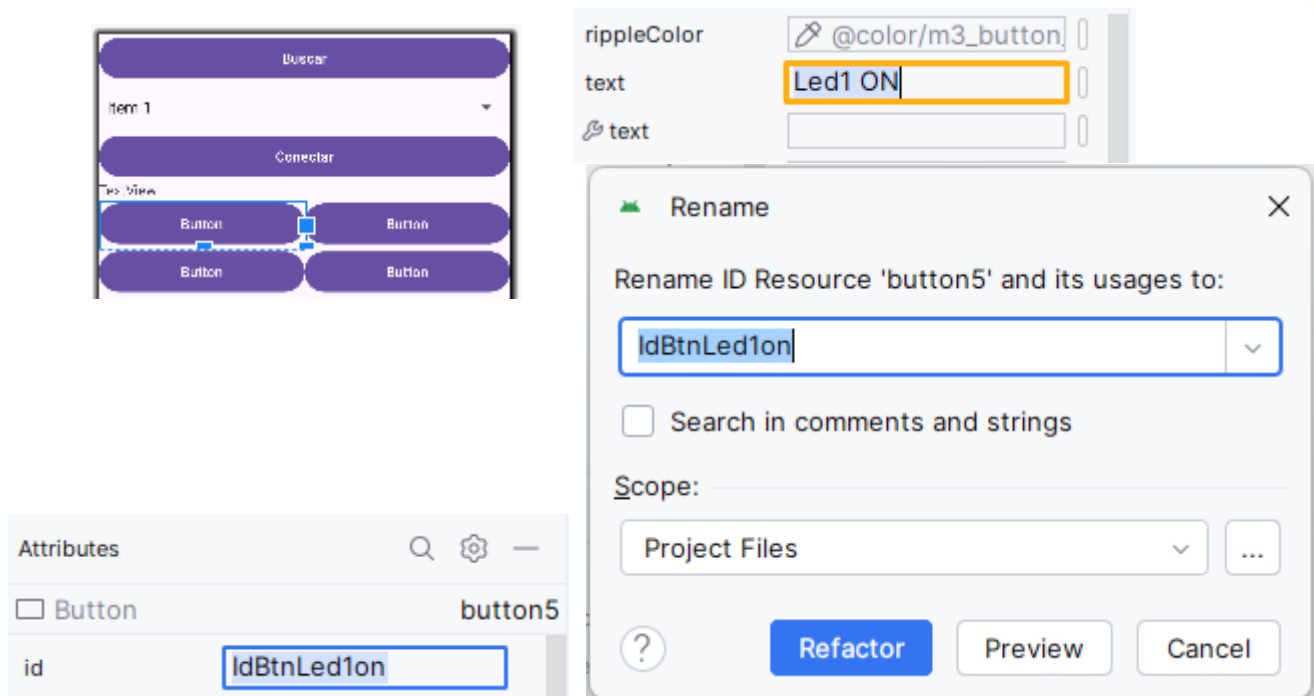


Figura 21. Asignación de atributos del botón5.

20. Repitiendo el paso anterior, seleccionar el **botón7.**, en la región Attributes, en el atributo text ingresar el texto Led1 OFF y digitar Enter. En atributo id ingresar el identificador IdBtnLed1 off. En el diálogo Rename digitar Refactor.
21. Seleccionar el **botón4.** En la región Attributes, en el atributo text ingresar el texto Led2 ON y digitar Enter. En atributo id ingresar el identificador IdBtnLed1off. En el diálogo Rename digitar Refactor

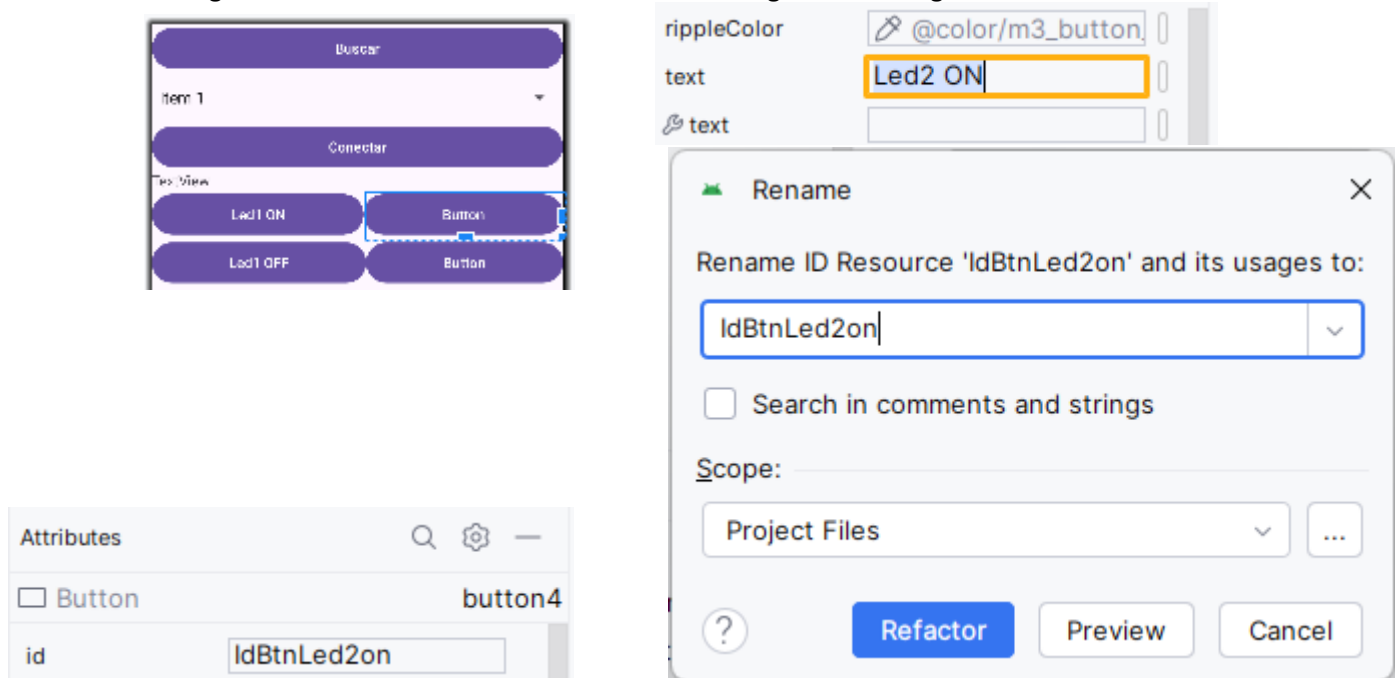


Figura 22. Asignación de atributos del botón4.

22. Seleccionar el botón6. En la región Attributes, en el atributo text ingresar el texto Led2 OFF y digitar Enter. En atributo id ingresar el identificador IdBtnLed2off. En el diálogo Rename digitar Refactor.

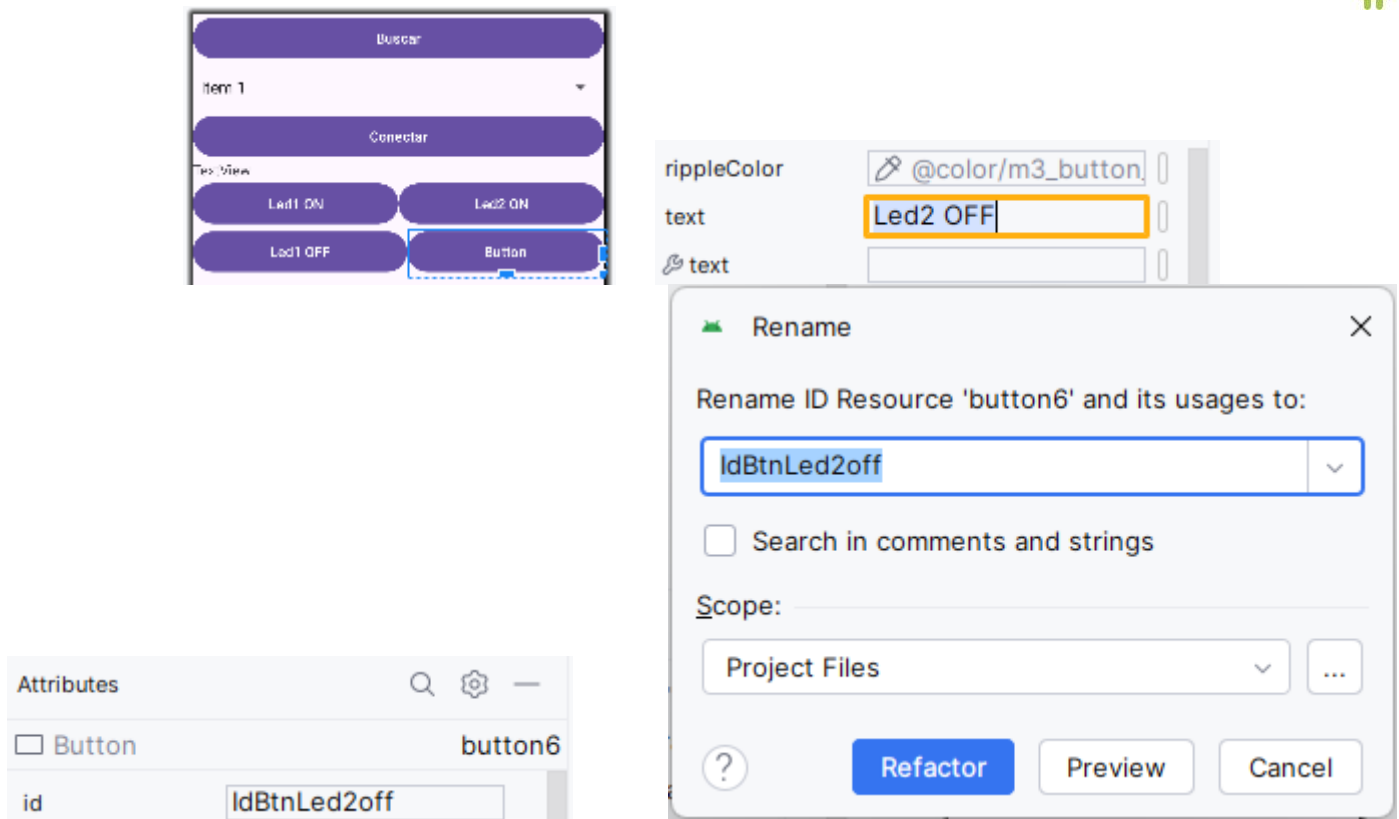


Figura 23. Asignación de atributos del botón6.

23. Seleccionar el botón inferior. En la región **Attributes**, en el atributo **text** ingresar el texto **Desconectar** y digitar **Enter**. En atributo **id** ingresar el identificador **IdBtnLed1off**. En el diálogo **Rename** digitar **Refactor**.

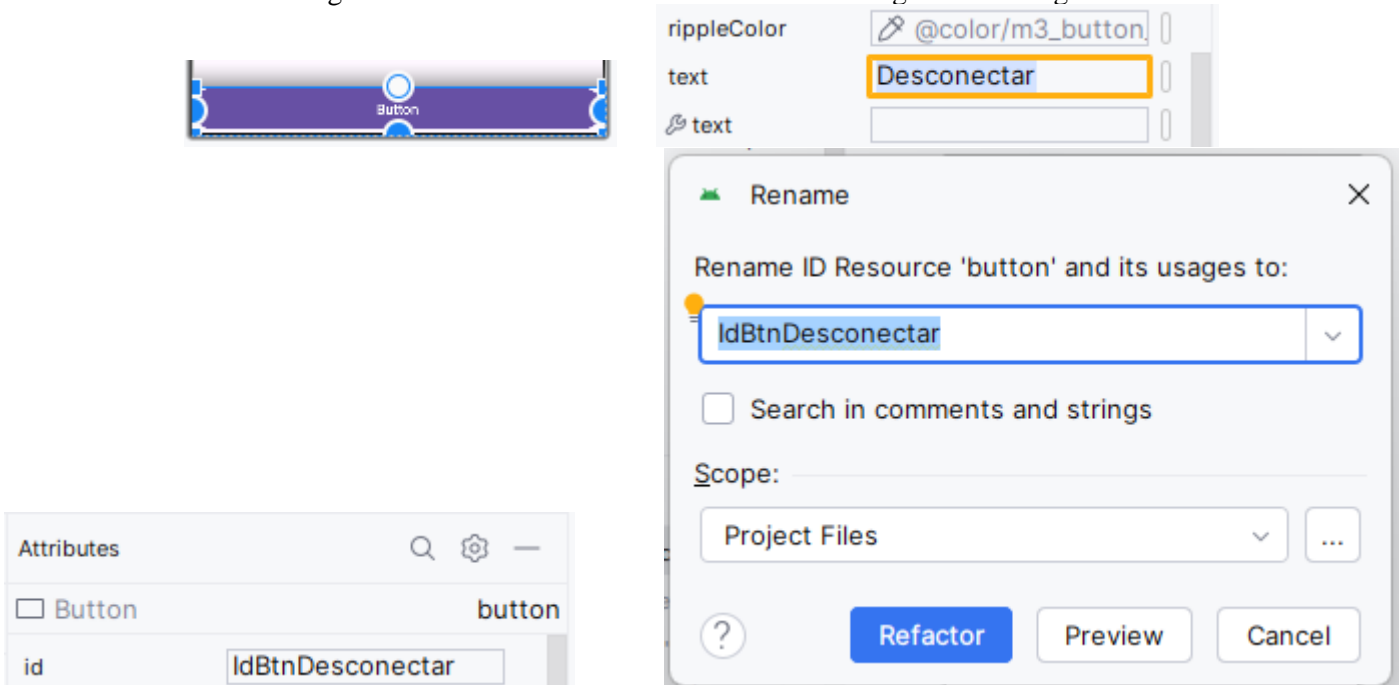


Figura 24. Asignación de atributos del botón Desconectar.

24. Para terminar de asignar los identificadores faltantes, seleccionar el componente **Spinner** y en el atributo **id** ingresar **IdDisEncontrados**. En el cuadro de diálogo **Rename** digitar **Refactor**.

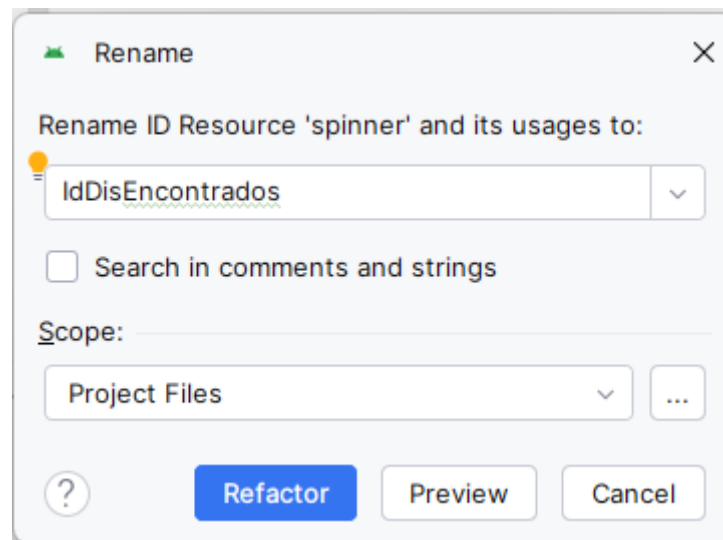


Figura 25. El identificador del Spinner.

La apariencia hasta este momento debe ser similar a la imagen de la siguiente figura 27.

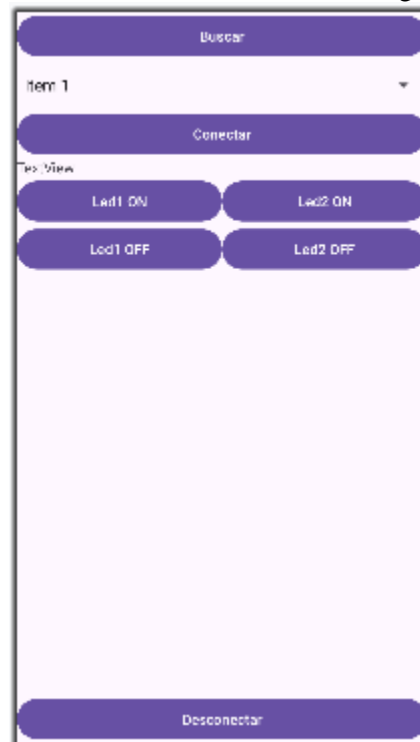


Figura 26. Apariencia final de los componentes.

PARTE III.

El código final de la aplicación de los archivos **MainActivity.java** y **activity_main.xml** se indica enseguida.

<!-- El archivo activity_main.xml. -->

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
```



```
android:layout_height="match_parent"
tools:context=".MainActivity">
<Button
    android:id="@+id/IdBtnDesconectar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Desconectar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
<LinearLayout
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toTopOf="@+id/IdBtnDesconectar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
    <Button
        android:id="@+id/IdBtnBuscar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Buscar" />
    <Spinner
        android:id="@+id/IdDisEncontrados"
        android:layout_width="match_parent"
        android:layout_height="50dp" />
    <Button
        android:id="@+id/IdBtnConectar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Conectar" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="CONTROL LEDS"
        android:textSize="16sp"
        android:textStyle="bold" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/IdBtnLed1on"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Led1 ON" />
        <Button
            android:id="@+id/IdBtnLed2on"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Led2 ON" />
    </LinearLayout>
</LinearLayout>
```




```
        android:orientation="horizontal">
        <Button
            android:id="@+id/IdBtnLed1off"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Led1 OFF" />
        <Button
            android:id="@+id/IdBtnLed2off"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Led2 OFF" />
    </LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

// El archivo MainActivity.java

```
import android.os.Bundle;
import android.widget.Button;
import android.widget.Spinner;
import android.view.View;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    Button IdBtnBuscar, IdBtnConectar, IdBtnLed1on, IdBtnLed1off, IdBtnLed2on,
    IdBtnLed2off, IdBtnDesconectar;
    Spinner IdDisEncontrados;
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        IdBtnBuscar = findViewById(R.id.IdBtnBuscar);
        IdBtnConectar = findViewById(R.id.IdBtnConectar);
        IdBtnLed1on = findViewById(R.id.IdBtnLed1on);
        IdBtnLed1off = findViewById(R.id.IdBtnLed1off);
        IdBtnLed2on = findViewById(R.id.IdBtnLed2on);
        IdBtnLed2off = findViewById(R.id.IdBtnLed2off);
        IdBtnDesconectar = findViewById(R.id.IdBtnDesconectar);
        IdDisEncontrados = findViewById(R.id.IdDisEncontrados);
        IdBtnBuscar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // acciones
            }
        });
        IdBtnConectar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // acciones
            }
        });
        IdBtnLed1on.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "Se digitó el botón Led1 ON",
                Toast.LENGTH_SHORT).show();
            }
        });
        IdBtnLed1off.setOnClickListener(new View.OnClickListener() {
```



```
        @Override
        public void onClick(View v){
            // acciones
        }
    });
    IdBtnLed2on.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v){
            // acciones
        }
    });
    IdBtnLed2off.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v){
            // acciones
        }
    });
    IdBtnDesconectar.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v){
            // acciones
        }
    });
}
```

25. Probar la ejecución de la aplicación. Digitar el botón Led1 ON para visualizar el mensaje Toast. Ver la figura 28.

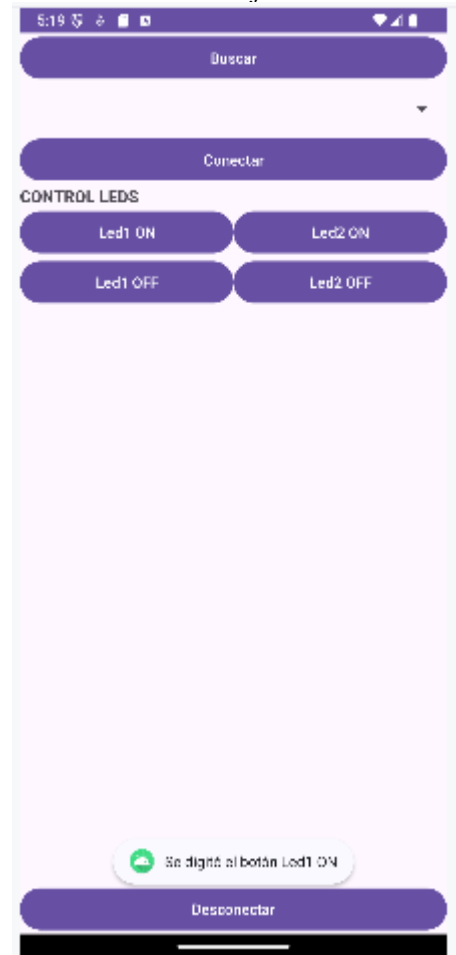
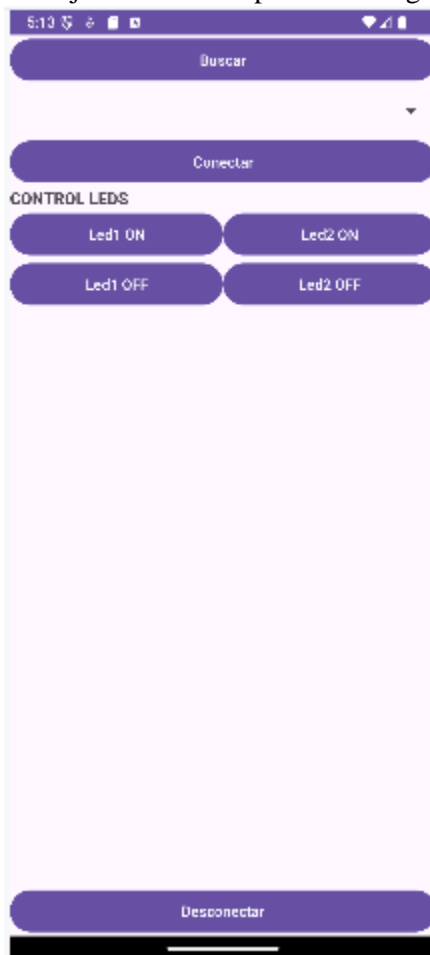




Figura 28. La imagen final de los componentes.

NOTA. Generar un reporte con las imágenes obtenidas y guardarlas en un archivo `AlumnoPrograficaGrupo.doc`. Enviarlos al sitio indicado por el profesor.