

Darle permisos a la aplicación

Se realiza en el manifest y son los siguientes:

- **INTERNET** → conexión a internet
- **READ_CONTACTS** → leer en la lista de contactos
- **WRITE_CONTACTS** → escribir en la lista de contactos
- **SEND_SMS** → enviar SMS
- **ACCESS_COARSE_LOCATION** → ubicación mediante señales del WIFI
- **ACCESS_FINE_LOCATION** → ubicación mediante GPS
- **BLUETOOTH** → permite el uso de Bluetooth

Ejemplo:

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Métodos del ciclo de vida

On Create()

Inicialización (se ejecuta una vez cuando la actividad es creada)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); // Inflar el layout de la actividad

    // Inicializar componentes de la UI o variables
    Log.d("CicloDeVida", "onCreate llamado");
}
```

On Start()

La actividad se hace visible

On Resume()

La actividad está en primer plano, interactiva.

On Pause()

La actividad deja de estar interactiva (a punto de ir a segundo plano).

On Stop()

La actividad ya no es visible.

On Restart()

La actividad vuelve a estar visible después de haber sido detenida.

On Destroy()

La actividad está siendo destruida (ya sea por el usuario o el sistema).

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("CicloDeVida", "onCreate llamado");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d("CicloDeVida", "onStart llamado");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d("CicloDeVida", "onResume llamado");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d("CicloDeVida", "onPause llamado");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d("CicloDeVida", "onStop llamado");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("CicloDeVida", "onRestart llamado");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("CicloDeVida", "onDestroy llamado");
    }
}
```

Intent

Es como una "nota" que le pasas al sistema diciendo algo que quieres hacer.

Ejemplo: (asignando un intent a un botón)

Para ello necesitamos:

- Añadir en el Manifest el SecondActivity
- Crear 2 xml (uno con el botón en el layout principal y otro con el layout que abre el botón)
- Crear 2 java (uno con el botón y Oncreate y el otro con el Oncreate que abre el xml_2)

JAVA 1

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Buscar el botón en la interfaz
        Button boton = findViewById(R.id.button_open_second);

        // Configurar el evento onClick
        boton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Aquí se crea y lanza el Intent para abrir SecondActivity
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

JAVA 2

```
public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }
}
```

XML 1

```
<?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/button_open_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Abrir Segunda Pantalla"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

XML 2

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Esto es el Second Activity"
        android:textSize="18sp"
        android:layout_margin="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

SetAction()

Darle instrucciones claras al sistema sobre qué hacer, como abrir un enlace, hacer una llamada o tomar una foto.

Ejemplos: (añadirle a un botón que abra Intent con setAction)

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button boton2 = findViewById(R.id.button_open_second);
```

```
boton2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(Intent.ACTION_VIEW);  
        intent.setData(Uri.parse("https://www.google.com"));  
        startActivity(intent);  
    }  
});  
}
```

Hay diferentes tipos:

- Abrir un enlace (ejemplo):

```
Intent intent = new Intent(Intent.ACTION_VIEW);
```

- Abrir teléfono con número (sin llamar):

```
Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:+623157362"));
```

- Realizar llamada:

```
Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:+623157362"));
```

- Enviar Correo Electrónico:

```
Intent intent = new Intent(Intent.ACTION_SENDTO, Uri.parse("mailto:manuel@example.com"));
```

- Abrir cámara:

```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

- Abrir ajustes:

```
Intent intent = new Intent(Settings.ACTION_SETTINGS);
```

- Buscar algo:

```
Intent intent = new Intent(Intent.ACTION_SEARCH)  
intent.putExtra(SearchManager.QUERY, "texto a buscar");
```

- Mostrar un mapa:

```
Intent intent = new Intent(Intent.ACTION_VIEW,  
Uri.parse("geo:0,0?q=Sagrada+Familia,+Barcelona,+Spain"));  
startActivity(intent);
```

- Enviar mensaje o compartir algo:

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(Intent.EXTRA_TEXT, "Este es el texto a compartir");  
startActivity(Intent.createChooser(intent, "Compartir vía"));
```

Layouts

Atributos

<i>Atributos de posicionamiento</i>	layout_width	ancho
	layout_height	alto
<i>Atributos para los márgenes</i>	layout_margin	cuatro márgenes
	layout_marginBottom	margen inferior
	layout_marginLeft	margen izquierdo
	layout_marginRight	margen derecho
	layout_marginTop	margen superior
<i>Atributos para el espaciado</i>	android:padding	espaciado a los cuatro lados
	android:paddingTop	espaciado superior
	android:paddingBottom	espaciado inferior
	android:paddingLeft	espaciado izquierdo
	android:paddingRight	espaciado derecho

LinearLayout

Permite organizar los elementos de la interfaz de usuario en una sola dirección, ya sea de forma vertical (uno debajo del otro) u horizontal (uno al lado del otro). Esto lo hace muy útil para crear layouts simples y ordenados.

Ejemplos:

- Formulario de contacto:

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <!-- Título -->
  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Formulario de Registro"
    android:textSize="24sp"
    android:layout_gravity="center"
    android:layout_marginBottom="24dp" />

  <!-- Campo de Nombre -->
  <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Nombre"
    android:layout_marginBottom="16dp"
    android:inputType="text"/>

  <!-- Campo de Correo Electrónico -->
  <EditText
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Correo Electrónico"
        android:layout_marginBottom="16dp"
        android:inputType="textEmailAddress" />

<!-- Botón de Registro -->
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Registrarse"
    android:id="@+id/bt1"/>
</LinearLayout>
```

Constraint Layout

Permite posicionar y dimensionar los elementos UI basándose en restricciones y relaciones entre ellos, lo que facilita la creación de diseños complejos y adaptables a diferentes tamaños de pantalla.

Ejemplo:

- Diseño de tarjeta (usando Imagen View)

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:background="@android:color/white"
    android:elevation="4dp">

    <!-- Imagen -->
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="150dp"
        android:src="@drawable/sample_image" <!-- Reemplaza con tu imagen -->
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/titleTextView"
        android:scaleType="centerCrop"
        android:layout_marginBottom="8dp" />

    <!-- Título -->
    <TextView
        android:id="@+id/titleTextView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Título de la Tarjeta"
        android:textSize="18sp"
        android:textColor="@android:color/black"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginBottom="4dp" />

    <!-- Descripción -->
    <TextView
```

```
        android:id="@+id/descriptionTextView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Esta es una descripción corta de la tarjeta. Aquí puedes agregar más
información."
        android:textSize="14sp"
        android:textColor="@android:color/darker_gray"
        app:layout_constraintTop_toBottomOf="@+id/titleTextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Relative Layout

Permite posicionar sus elementos hijos en relación entre sí o en relación a su contenedor padre. Este layout es útil cuando necesitas un diseño donde los elementos deben estar alineados o posicionados de forma específica entre sí.

Ejemplo:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Texto superior"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"/>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botón inferior"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"/>

</RelativeLayout>
```


Frame Layout

Puede considerarse un marco en el que se colocan las vistas. Es uno de los layouts más simples y básicos disponibles en Android.

Ejemplo de clase:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF">

    <!-- Primera fila: Pegados arriba -->
    <FrameLayout
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="left|top"
        android:background="#FF0000">

    </FrameLayout>

    <FrameLayout
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal|top"
        android:background="#00FF00">

    </FrameLayout>

    <FrameLayout
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="right|top"
        android:background="#0000FF">

    </FrameLayout>

    <!-- Segunda fila: En el medio -->
    <FrameLayout
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_vertical|left"
        android:layout_marginEnd="16dp"
        android:background="#FF00FF">

    </FrameLayout>

    <FrameLayout
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:background="#00FFFF">

    </FrameLayout>

    <FrameLayout
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_vertical|right"
        android:background="#FFFF00">

    </FrameLayout>

    <!-- Tercera fila: Pegados abajo -->
```

```
<FrameLayout
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="left|bottom"
    android:layout_marginEnd="16dp"
    android:background="#FF8800">

</FrameLayout>

<FrameLayout
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="center_horizontal|bottom"
    android:background="#8800FF">

</FrameLayout>

<FrameLayout
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="right|bottom"
    android:background="#00FF88">

</FrameLayout>
</FrameLayout>
```

Table Layout

Es un componente de la interfaz de usuario en Android que permite organizar diferentes vistas en pestañas (tabs). Es una forma fácil y visualmente atractiva de presentar contenido que puede ser dividido en secciones, facilitando la navegación entre ellas.

Ejemplo:

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

    <TableLayout
        android:id="@+id/tableLayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:stretchColumns="1"> <!-- Estira la segunda columna -->

        <!-- Fila 1 -->
        <TableRow>
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Botón 1"
                android:onClick="onButtonClick1" />
        </TableRow>

        <!-- Fila 2 -->
        <TableRow>
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```

```
        android:text="Botón 2"
        android:onClick="onButtonClick2" />
    </TableRow>

    <!-- Fila 3 -->
    <TableRow>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 3"
            android:onClick="onButtonClick3" />
    </TableRow>

</TableLayout>

</LinearLayout>
```

Absolute Layout

Define un contenedor donde se pueden colocar vistas en posiciones fijas, independientemente del tamaño de la pantalla o resolución.

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botón 1"
        android:layout_x="50dp"
        android:layout_y="50dp"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botón 2"
        android:layout_x="150dp"
        android:layout_y="150dp"/>

</AbsoluteLayout>
```

Animaciones

Descargamos la carpeta animaciones, en la carpeta res creamos una carpeta animaciones y metemos todos los códigos de las animaciones.

En el xml añadimos la imagen y todos los botones que realizarán una acción cada uno:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@drawable/sevilla" />

    <Button
        android:id="@+id/btnAlpha1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alfa 1" />

    <Button
        android:id="@+id/btnAlpha2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alfa 2" />

    <Button
        android:id="@+id/btnScale1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Escala 1" />

    <Button
        android:id="@+id/btnScale2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Escala 2" />

    <Button
        android:id="@+id/btnMove1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mueve 1" />

    <Button
        android:id="@+id/btnMove2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mueve 2" />

    <Button
        android:id="@+id/btnRotate1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Rota 1" />
```

```
<Button
    android:id="@+id/btnRotate2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Rota 2" />

<!-- Añade más botones según los archivos de animaciones disponibles -->

</LinearLayout>
```

Botones

Xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <!-- Botón de Registro -->
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Boton 1"
        android:id="@+id/bt1"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Boton 2"
        android:id="@+id/bt2"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Boton 3"
        android:id="@+id/bt3"/>

</LinearLayout>
```

java

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button boton1 = findViewById(R.id.bt1);
        boton1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });

        Button boton2 = findViewById(R.id.bt2);
```

```
        boton2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });

        Button boton3 = findViewById(R.id.bt3);
        boton3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Estilos

Para añadir estilos creamos en la carpeta drawable un xml styles en el que definimos los estilos que vamos a añadir

XML styles:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">

    <!-- Definir el tamaño del borde -->
    <stroke
        android:width="2dp"
        android:color="#000000" />

    <!-- Definir el color de fondo -->
    <solid android:color="#008DFF" />

    <!-- Definir el radio de las esquinas a 0 para que no haya redondeo -->
    <corners android:radius="0dp" />
</shape>
```

XML activity:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">
    <!-- Botón 1 -->
    <Button
        android:id="@+id/button1"
        android:layout_width="180dp"
```

```
        android:layout_height="wrap_content"
        android:text="Botón 1"
        android:minWidth="100dp"
        android:background="@drawable/styles"
        android:layout_marginTop="16dp"
        app:backgroundTint="#FF0000"
    />
<!-- Botón 2 -->
<Button
    android:id="@+id/button2"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:text="Botón 2"
    android:minWidth="100dp"
    android:background="@drawable/styles"
    android:layout_marginTop="16dp"
    app:backgroundTint="#FF0000"
/>
<!-- Botón 3 -->
<Button
    android:id="@+id/button3"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:text="Botón 3"
    android:minWidth="100dp"
    android:background="@drawable/styles"
    android:layout_marginTop="16dp"/>
<!-- Botón 4 -->
<Button
    android:id="@+id/button4"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Botón 4"
    app:layout_constraintHorizontal_bias="0.498"
    android:background="@drawable/styles"
/>
</LinearLayout>
```

Añadimos en android:background el archivo styles, si queremos añadir un color y bordes cuadrados debemos añadir el color en este xml con:
app:backgroundTint="#FF0000"

Toggle Button

Toggle Button + Toast

java

```
package com.example.repaso;

import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.ToggleButton;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Asegúrate de que este sea el nombre de tu layout

        // Encontrar el ToggleButton por su ID
        ToggleButton toggleButton = findViewById(R.id.toggle_button);

        // Configurar un listener para el ToggleButton
        toggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                // Mostrar un Toast según el estado del ToggleButton
                if (isChecked) {
                    Toast.makeText(MainActivity.this, "Encendido", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Apagado", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <!-- ToggleButton centrado -->
    <ToggleButton
        android:id="@+id/toggle_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOff="Apagar"
        android:textOn="Encender"
        android:checked="false" />

</LinearLayout>
```


Edit Text

Tipos de inputType:

android:inputType="textPersonName"

Texto Simple

- **text**: Texto normal.
- **textCapSentences**: Texto normal con capitalización de oraciones.
- **textCapWords**: Capitaliza cada palabra.
- **textCapCharacters**: Capitaliza todas las letras.
- **textMultiLine**: Permite múltiples líneas de texto.

Contraseñas

- **textPassword**: Para ingresar contraseñas (oculta el texto).
- **textVisiblePassword**: Para mostrar una contraseña visible.
- **textEmailAddress**: Para ingresar direcciones de correo electrónico.
- **textPersonName**: Para ingresar nombres de personas.

Números

- **number**: Solo permite números.
- **numberSigned**: Permite números con signo (positivo o negativo).
- **numberDecimal**: Permite números decimales.

Teléfonos y Contactos

- **phone**: Para ingresar números de teléfono.
- **phonePad**: Para ingresar números de teléfono en formato de teclado.

Fechas y Horas

- **datetime**: Para ingresar fecha y hora.
- **date**: Para ingresar solo fechas.
- **time**: Para ingresar solo horas.

Direcciones de Correo

- **textEmailAddress**: Para ingresar direcciones de correo electrónico (habilita validación de formato).

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Introduce tu nombre"
    android:inputType="textPersonName"
    android:layout_marginBottom="16dp" />
```

OnDestroy

Abre otra activity cuando se cierra este activity

```
@Override
protected void onDestroy() {
    super.onDestroy();
    // Iniciar la segunda actividad al destruir la actual
    Intent main2 = new Intent(this, MainActivity2.class);
    startActivity(main2);
}
}
```

Toast

java_1

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Con esto se el layout definido en activity_main.xml
        setContentView(R.layout.activity_main);

        // Estoy buscando el botón en el layout que tiene el ID btnEnviar
        Button btnEnviar = findViewById(R.id.btnEnviar);

        // Ahora le voy a añadir un listener para que me escuche cuando lo presionen
        btnEnviar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Aquí estoy creando los datos que quiero enviar a la siguiente actividad
                String cadena = "Hola desde MainActivity"; // Este es un mensaje de texto
                double decimal = 2.21; // Este es un número decimal

                // Estoy creando un Intent para iniciar la Actividad2 y enviarle los datos
                Intent actividad2 = new Intent(MainActivity.this, Actividad2.class);

                // Estoy añadiendo los datos al Intent para que los reciba la siguiente actividad
                actividad2.putExtra("cadena", cadena);
                actividad2.putExtra("decimal", decimal);

                // Finalmente, estoy iniciando la Actividad2
                startActivity(actividad2);
            }
        });
    }
}
```

java_2

```
public class Actividad2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        // Recibo los datos enviados
        String cadena = getIntent().getStringExtra("cadena");
        double decimal = getIntent().getDoubleExtra("decimal", 0.0);

        // Muestro los datos con Toast
        String mensaje = "Cadena: " + cadena + ", Decimal: " + decimal;
        Toast.makeText(this, mensaje, Toast.LENGTH_LONG).show();
    }
}
```

xml_1

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <Button
        android:id="@+id/btnEnviar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enviar Datos" />
</LinearLayout>
```

xml_2

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bienvenido a Actividad 2"
        android:textSize="24sp" />
</LinearLayout>
```

EJEMPLOS HECHOS EN CLASE

4.1. Realiza con LinearLayout la siguiente estructura de layouts anidados.



4.2. Realiza una aplicación con una actividad cuyo Framelayout principal tenga nueve Framelayouts interiores y muestren la disposición de la figura 4.4.

4.1

xml_1

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Creo un linear layout en horizontal para meter dentro 3 views con sus respectivos colores-->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center">

    <!-- Línea Vertical Azul Izquierda -->
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#002CFF" />

    <!-- Línea Vertical Blanca -->
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@android:color/white" />

    <!-- Línea Vertical Azul Derecha -->
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#002CFF" />

</LinearLayout>
```

xml_2

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center">

    <!-- Línea Vertical Azul Izquierda -->
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#26FB2B" />

    <!-- Línea Vertical Azul Derecha -->
    <LinearLayout
```

```
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="5"
        android:background="@android:color/white" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#26FB2B" />

</LinearLayout>
```

xml_3

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- Parte Superior (layout en horizontal) -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FFFFFF" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FF0404" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FFFFFF" />
    </LinearLayout>

    <!-- Parte Inferior (layout en horizontal) -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FF0404" />

        <LinearLayout
```

```

        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#FFFFFF" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FF0404" />
    </LinearLayout>
</LinearLayout>

```

xml_4

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- Parte Superior (layout en horizontal) -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FFFFFF" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FF0404" />

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#FFFFFF" />
    </LinearLayout>

    <!-- Parte Inferior (layout en vertical con colores azul, blanco, azul) -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:background="#0000FF" /> <!-- Azul -->

        <LinearLayout
            android:layout_width="match_parent"

```

```

        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FFFFFF" /> <!-- Blanco -->

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:background="#0000FF" /> <!-- Azul -->
        </LinearLayout>
    </LinearLayout>

```

java_1

```

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        // Iniciar la segunda actividad al destruir la actual
        Intent main2 = new Intent(this, MainActivity2.class);
        startActivity(main2);
    }
}

```

Este código habría que repetirlo para abrir los demás xml.

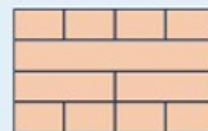
4.2

Ejemplo del Frame Layout

Actividad propuesta 4.4



Realiza una aplicación con una actividad cuya vista contenga un `TableLayout` con la estructura mostrada a la derecha. Puedes usar los botones como objetos incluidos en el interior de la tabla (siguiendo el ejemplo anterior).



4.4

java

```

public class MainActivity extends AppCompatActivity {

    @Override

```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable(this);  
    setContentView(R.layout.activity_main);  
}  
}
```

xml

```
<TableLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:useDefaultMargins="true">  
  
    <TableRow>  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:backgroundTint="@color/black"/>  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:backgroundTint="#FF0000" />  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:backgroundTint="#2196F3" />  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:backgroundTint="#E4D000" />  
    </TableRow>  
  
    <TableRow>  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:layout_span="4"  
            android:backgroundTint="#F88686" />  
    </TableRow>  
  
    <TableRow>  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:layout_span="2"  
            android:backgroundTint="#D603FF" />  
  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:layout_span="2"  
            android:backgroundTint="#7F9E9E" />  
    </TableRow>  
</TableLayout>
```



```

</TableRow>

<TableRow>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@color/black" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF0000" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#2196F3" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#E40000" />
</TableRow>

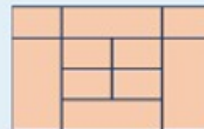
</TableLayout>

```

Actividad propuesta 4.5



Realiza una aplicación con una actividad cuya vista contenga un GridLayout con la estructura mostrada a la derecha. Puedes usar los botones como objetos incluidos en el interior de la tabla (siguiendo el ejemplo anterior).



4.5

java

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }
}

```

xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:columnCount="4"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#000000" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#ABFF0A"
        android:layout_columnSpan="2"
        android:layout_gravity="fill" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#864AED" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="fill"
        android:layout_rowSpan="3"
        android:backgroundTint="#FB9D2B"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@android:color/holo_purple" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@android:color/holo_blue_light" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="fill"
        android:layout_rowSpan="3"
        android:backgroundTint="@android:color/holo_red_light" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@android:color/holo_green_light" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="#FFA48B" />

    <Button
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_columnSpan="2"
        android:layout_gravity="fill"
        android:backgroundTint="#F88AAE" />

</GridLayout>

```

5.1



imagen_1

- Crear xml y archivo .java que abra ese xml

xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Texto construido desde el XML"
        android:textSize="20dp"
        android:textColor="#FF0000"
        android:textStyle="bold"/>

</LinearLayout>

```

java:

```

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class Activity1 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity1);
    }
}

```

imagen_2

- Crear vista directamente en archivo .java

java:

```
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class Activity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Crear un TextView dinámicamente
        TextView textView = new TextView(this);

        // Establecer el texto
        textView.setText("Texto creado en Java con tamaño 20dp, italic y color azul");

        // Establecer el tamaño del texto (20dp)
        textView.setTextSize(20);

        // Establecer el estilo del texto (italic)
        textView.setTypeface(null, Typeface.ITALIC);

        // Establecer el color del texto (azul)
        textView.setTextColor(Color.BLUE);

        // Establecer el TextView como la vista principal
        setContentView(textView);
    }
}
```

imagen_3

- Crear xml y archivo .java que abra ese xml

xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Texto construido desde el XML"
        android:textSize="20dp"
        android:textColor="#FF0000"
        android:textStyle="bold"/>
</LinearLayout>
```

java:

```
import android.graphics.Color;
import android.os.Bundle;
```

```
import android.text.Spannable;
import android.text.SpannableString;
import android.text.style.ForegroundColorSpan;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class Activity3 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity3);

        // Obtener referencia al TextView desde el XML
        TextView textView = findViewById(R.id.textView);

        // Establecer el color del texto azul
        textView.setTextColor(Color.BLUE);

        // Añadir texto en azul usando append()
        textView.append("\n y texto añadido desde append");
    }
}
```

imagen_4

- Crear xml y archivo .java que abra ese xml
- Descargar fuente Umbrella y meterla en la carpeta res/font (hay que crear la carpeta font)

xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Texto escrito fuente Umbrella"
        android:textSize="20dp"
        android:textColor="#FF0000"
        android:fontFamily="@font/umbrella"
        android:gravity="center"/>
</LinearLayout>
```

java:

```
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class Activity4 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity4);
    }
}
```

}

Botones en el centro

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

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:rowCount="3"
        android:columnCount="3"
        android:alignmentMode="alignMargins"
        android:layout_gravity="center"
        android:padding="16dp">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 1"
            android:layout_row="0"
            android:layout_column="0" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 2"
            android:layout_row="0"
            android:layout_column="1" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_row="0"
            android:layout_column="2"
            android:text="Botón 3" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 4"
            android:layout_row="1"
            android:layout_column="0" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 5"
            android:layout_row="1"
            android:layout_column="1" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 6"
            android:layout_row="1"
            android:layout_column="2" />

        <Button
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botón 7"
        android:layout_row="2"
        android:layout_column="0" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 8"
            android:layout_row="2"
            android:layout_column="1" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Botón 9"
            android:layout_row="2"
            android:layout_column="2" />

    </GridLayout>

</LinearLayout>
```

Edit Text en el centro

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <!-- EditText para el nombre -->
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Introduce tu nombre"
        android:inputType="textPersonName"
        android:layout_marginBottom="16dp" /> <!-- Margen inferior -->

    <!-- EditText para el email -->
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Introduce tu email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="16dp" /> <!-- Margen inferior -->

    <!-- EditText para la contraseña -->
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Introduce tu contraseña"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp" /> <!-- Margen inferior -->

</LinearLayout>
```