

# **FUNDAMENTOS DE ACTIVIDADES EN ANDROID**

# Ciclo de vida de una actividad

Hay tres ciclos anidados:

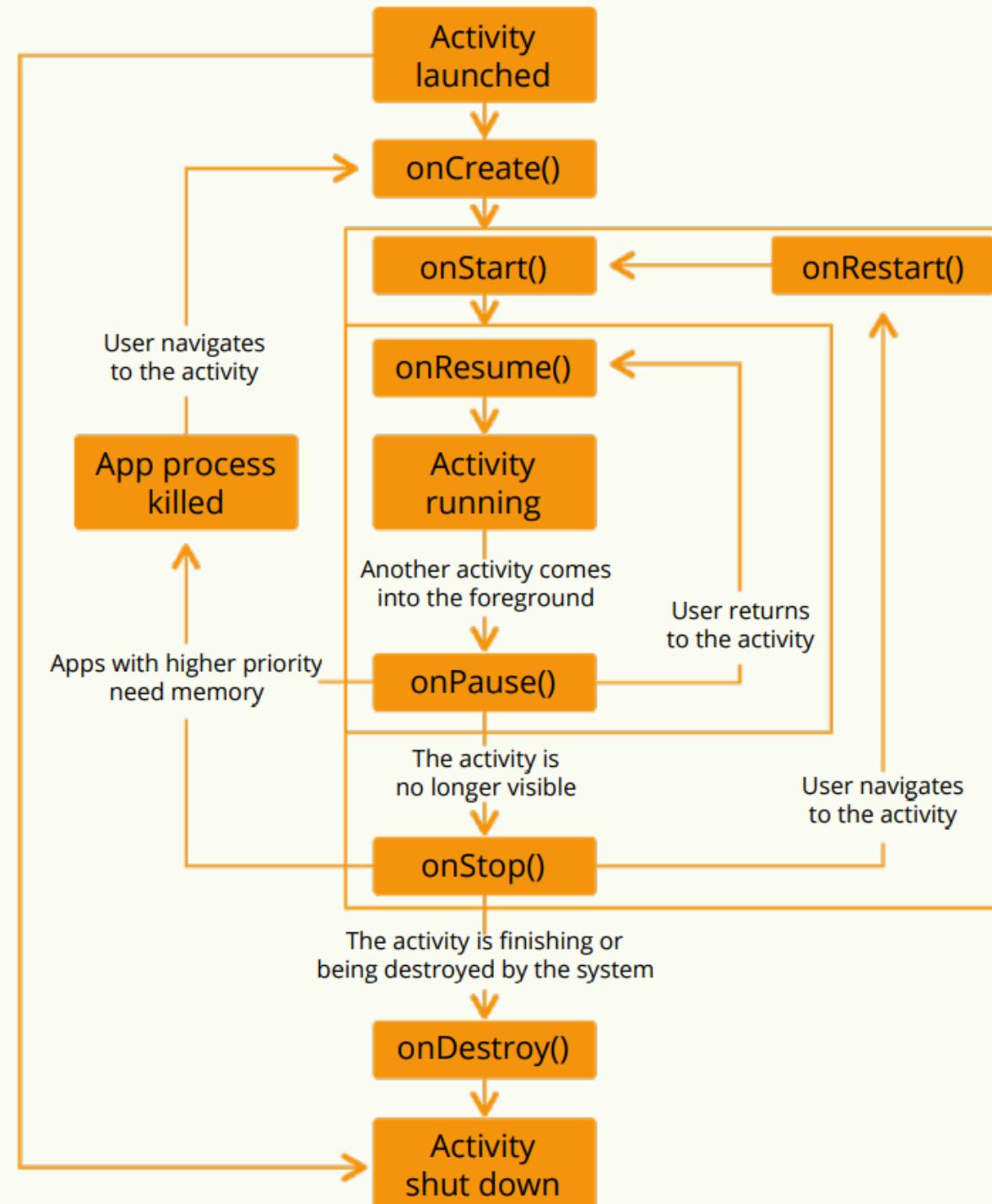
1. Completo

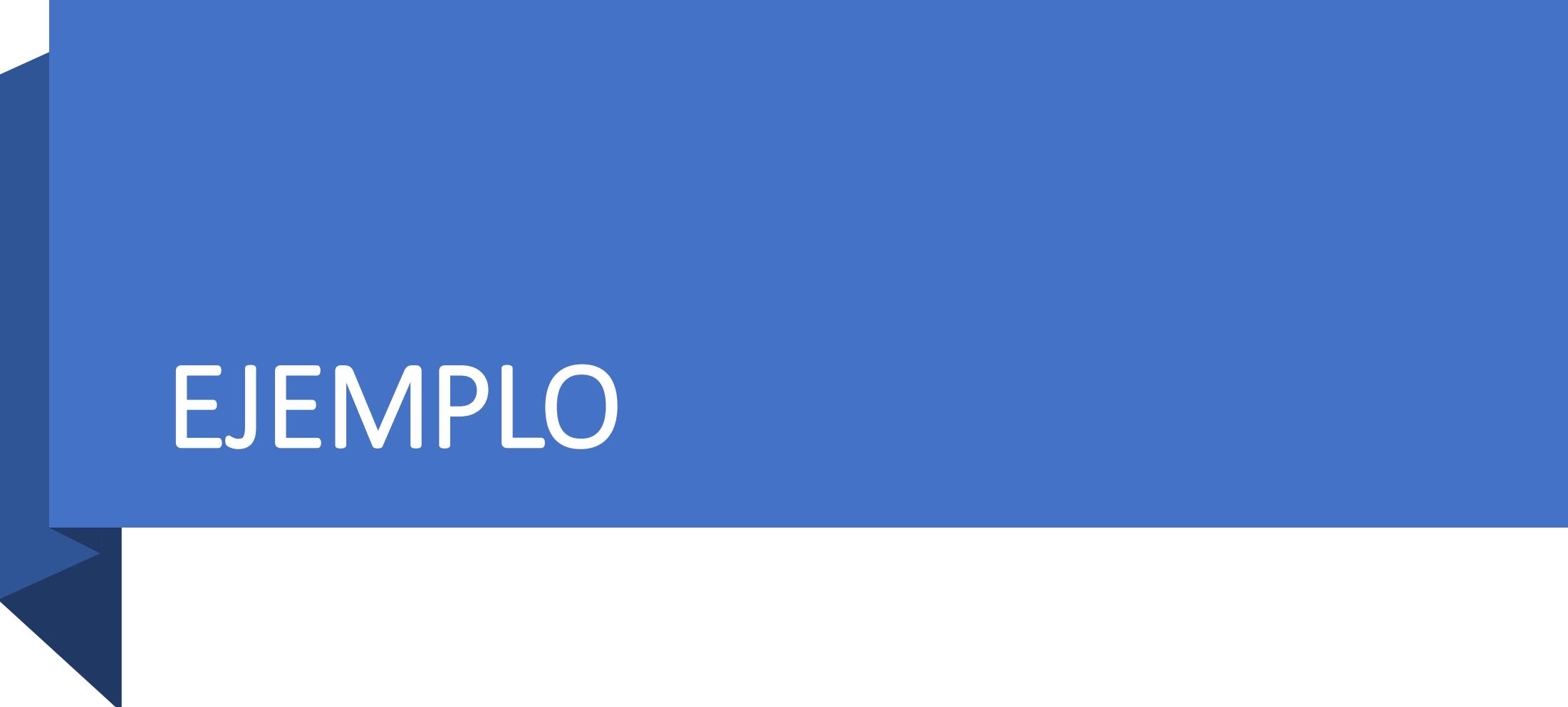
2. Visible

De onStart() a onStop()

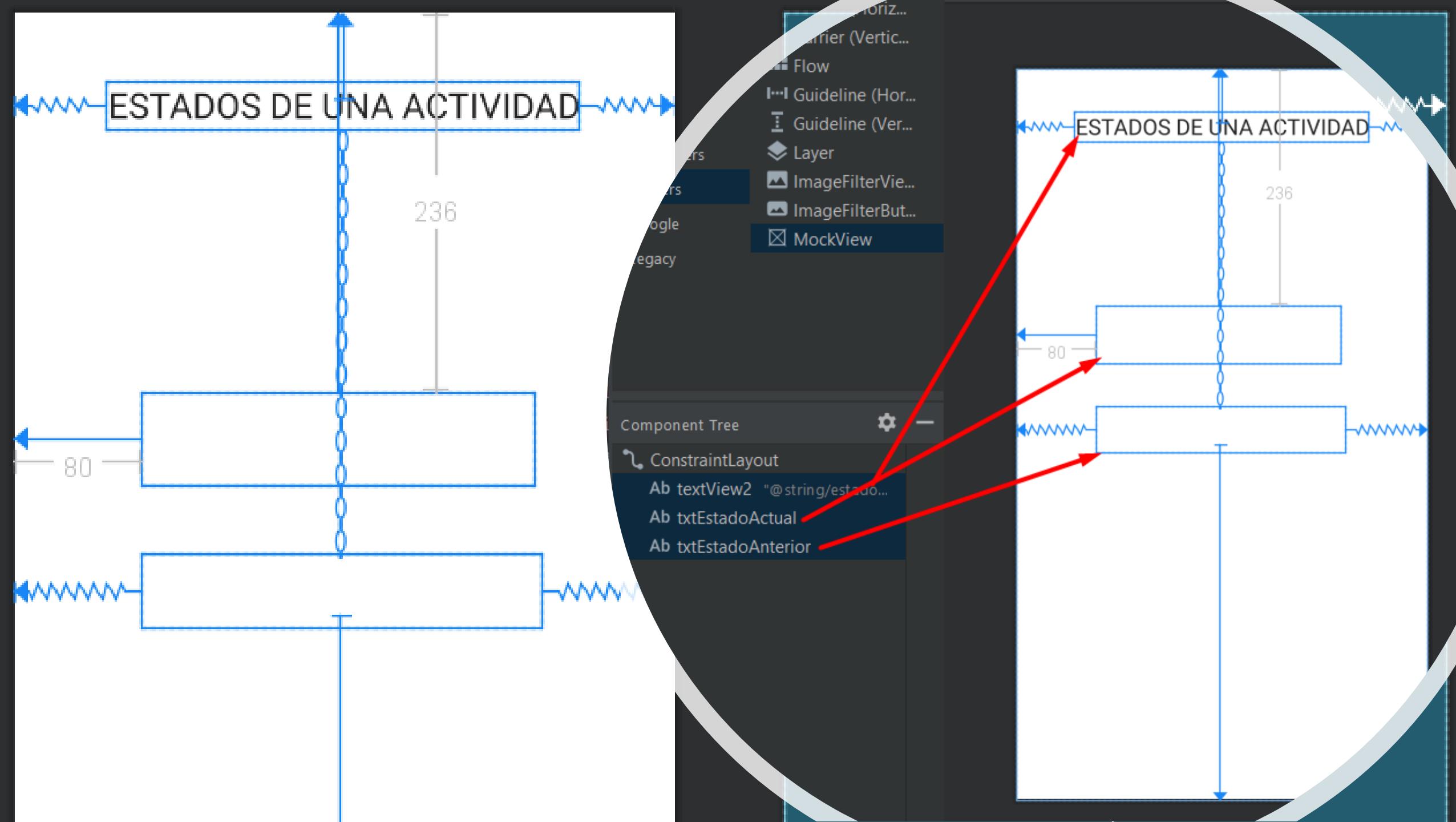
3. Primer plano

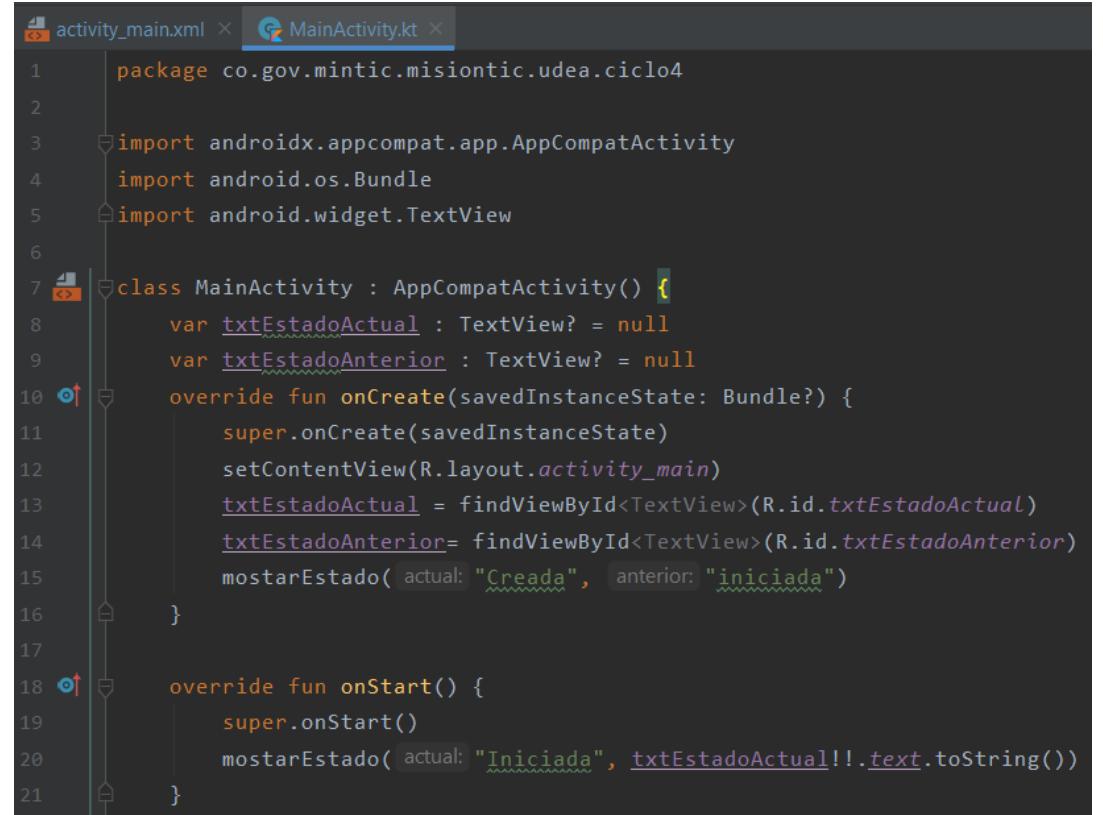
De onResume() a onPause()





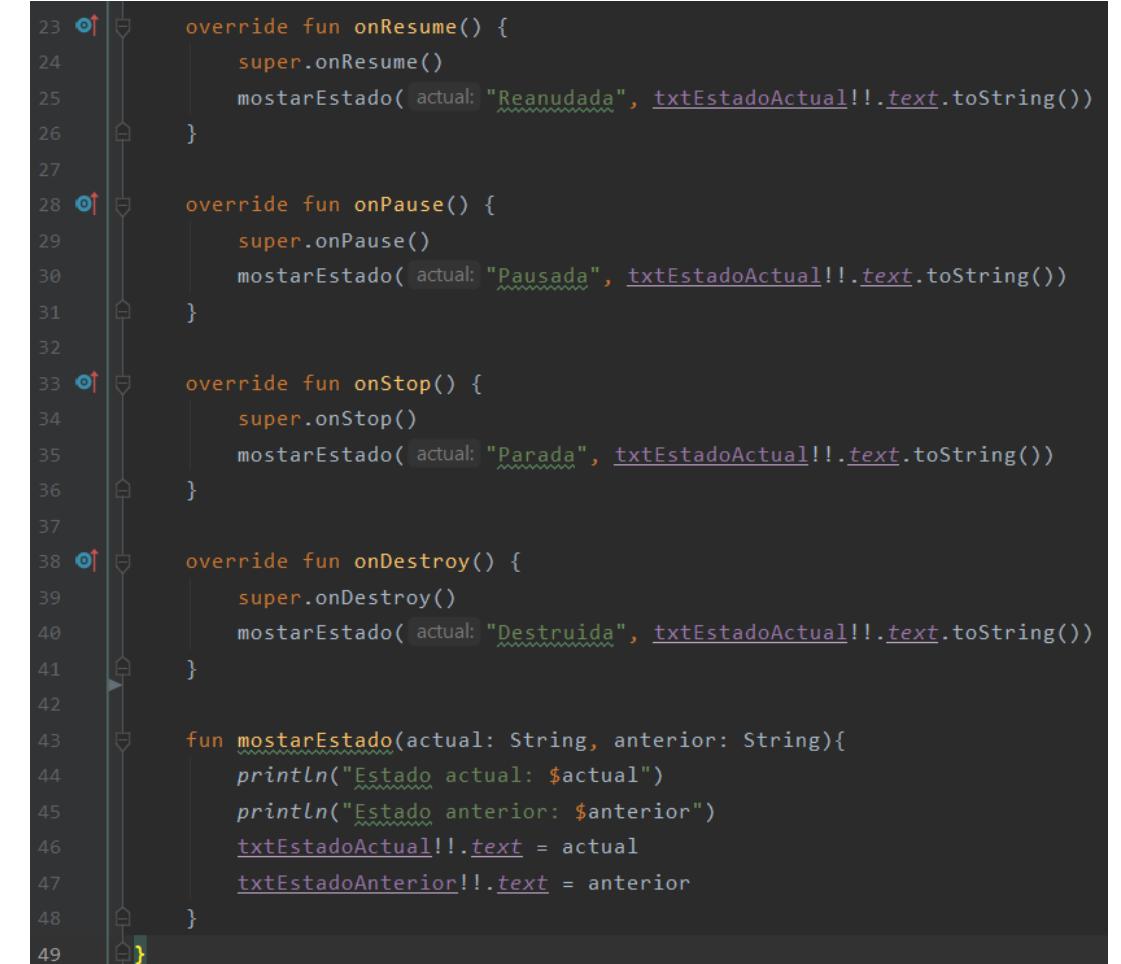
EJEMPLO





```
activity_main.xml x MainActivity.kt x

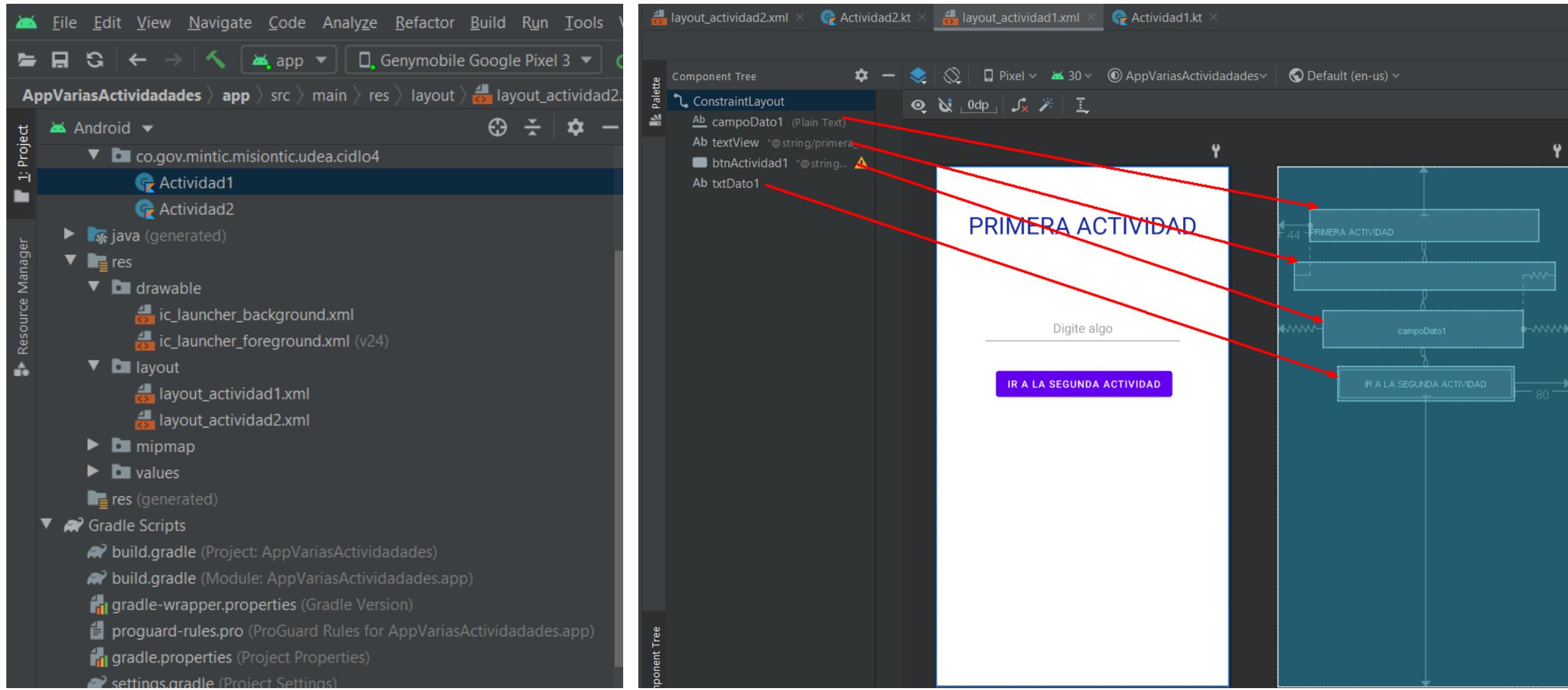
1 package co.gov.mintic.misiontic.udea.ciclo4
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.widget.TextView
6
7 class MainActivity : AppCompatActivity() {
8     var txtEstadoActual : TextView? = null
9     var txtEstadoAnterior : TextView? = null
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13        txtEstadoActual = findViewById<TextView>(R.id.txtEstadoActual)
14        txtEstadoAnterior= findViewById<TextView>(R.id.txtEstadoAnterior)
15        mostarEstado( actual: "Creada", anterior: "iniciada")
16    }
17
18    override fun onStart() {
19        super.onStart()
20        mostarEstado( actual: "Iniciada", txtEstadoActual!!.text.toString())
21    }
}
```

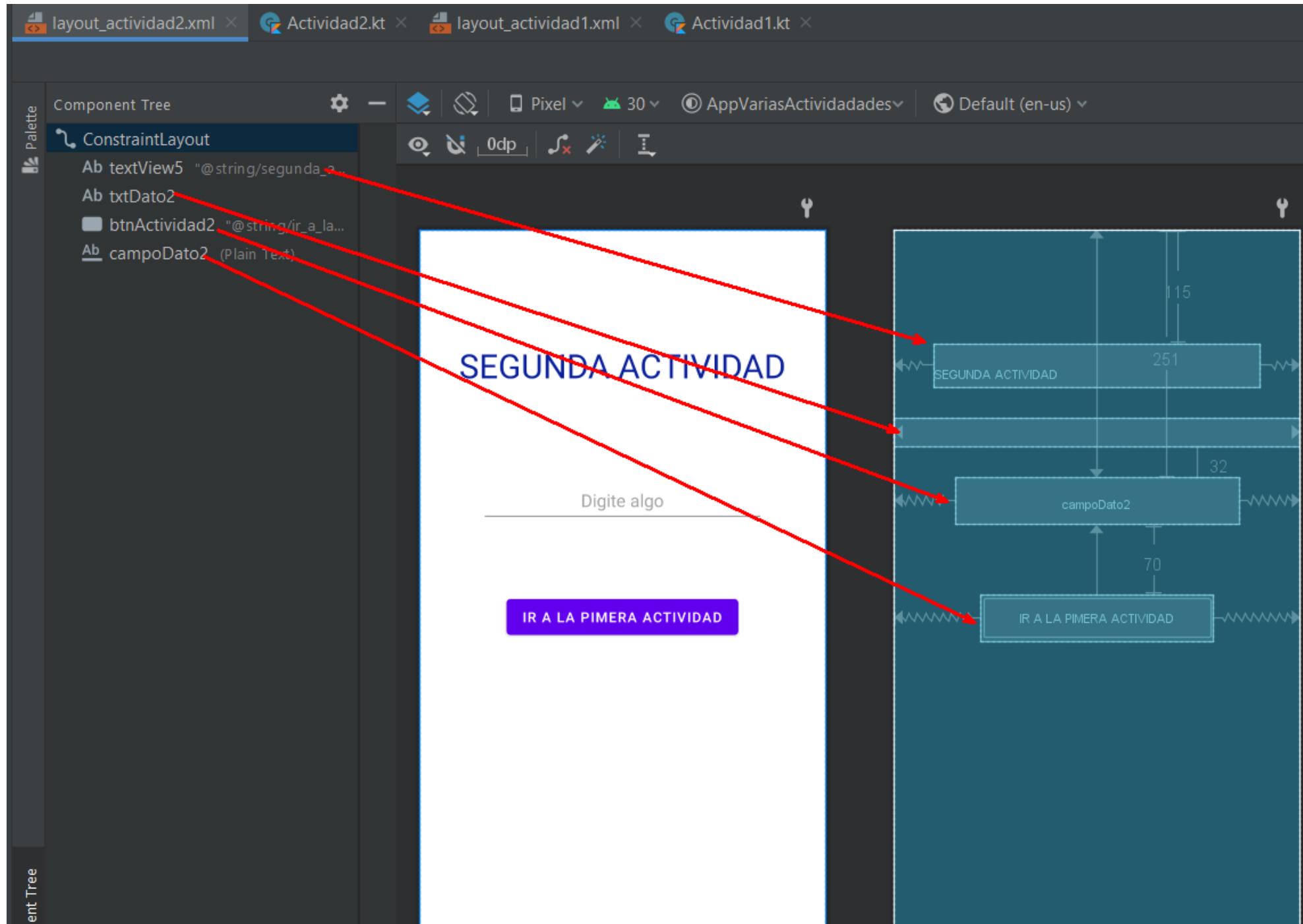


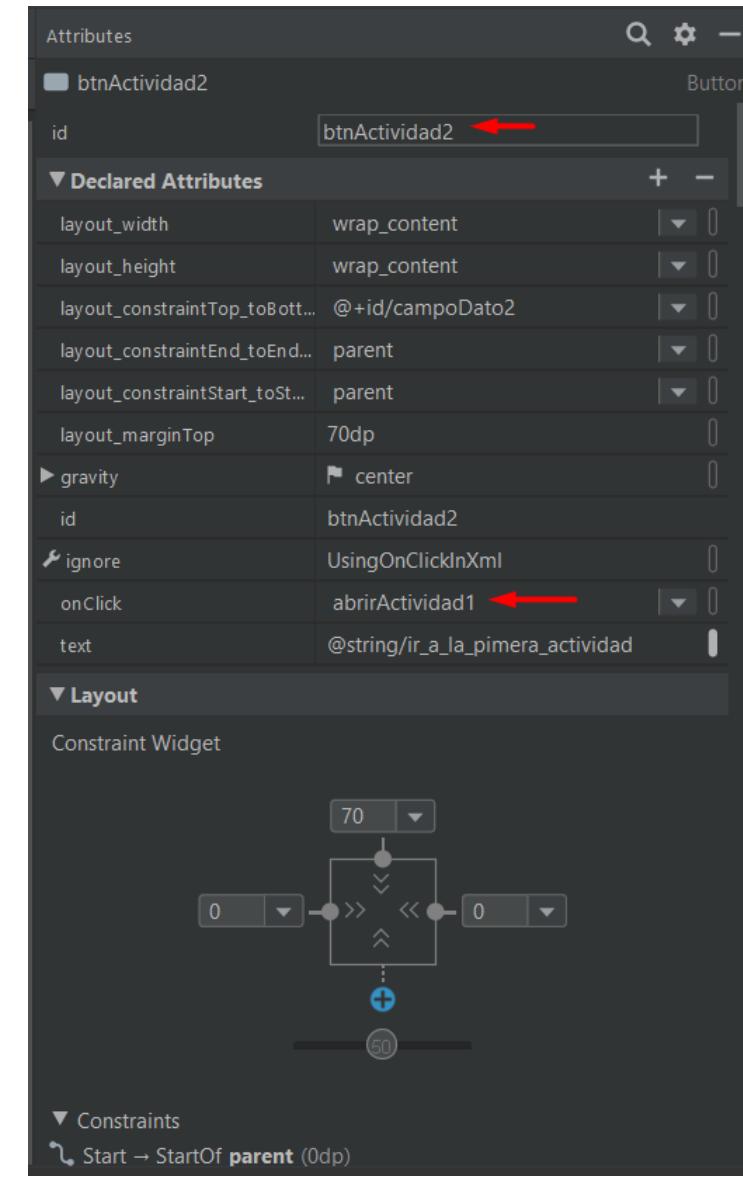
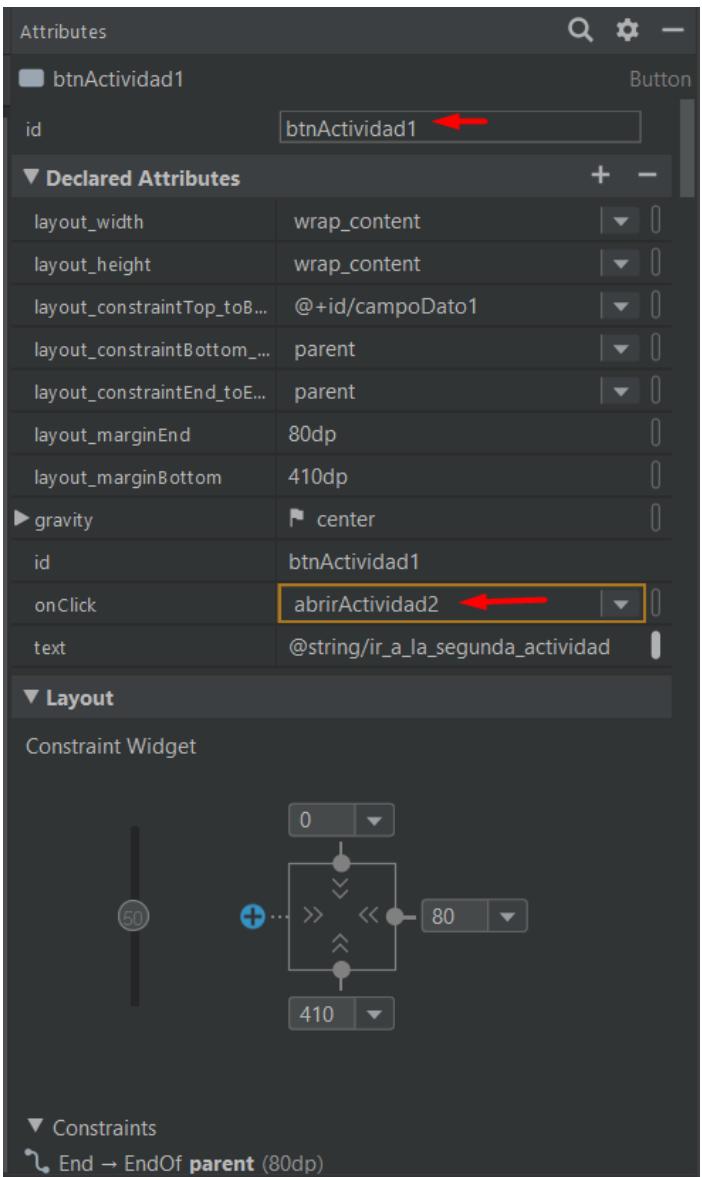
```
23    override fun onResume() {
24        super.onResume()
25        mostarEstado( actual: "Reanudada", txtEstadoActual!!.text.toString())
26    }
27
28    override fun onPause() {
29        super.onPause()
30        mostarEstado( actual: "Pausada", txtEstadoActual!!.text.toString())
31    }
32
33    override fun onStop() {
34        super.onStop()
35        mostarEstado( actual: "Parada", txtEstadoActual!!.text.toString())
36    }
37
38    override fun onDestroy() {
39        super.onDestroy()
40        mostarEstado( actual: "Destruida", txtEstadoActual!!.text.toString())
41    }
42
43    fun mostarEstado(actual: String, anterior: String){
44        println("Estado actual: $actual")
45        println("Estado anterior: $anterior")
46        txtEstadoActual!!.text = actual
47        txtEstadoAnterior!!.text = anterior
48    }
49}
```

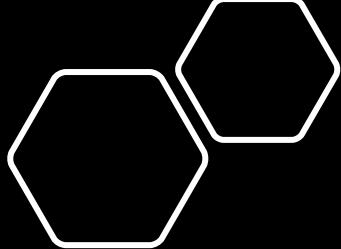
## COMUNICACIÓN ENTRE ACTIVIDADES





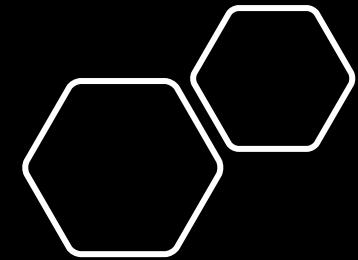






```
layout_actividad2.xml x Actividad2.kt x layout_actividad1.xml x Actividad1.kt x
1 package co.gov.mintic.misiontic.udea.cidlo4
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.view.View
7 import android.widget.EditText
8 import android.widget.TextView
9 import co.gov.mintic.misiontic.udea.appvariasactividades.R
10
11 class Actividad1 : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.layout_actividad1)
15     }
16
17     override fun onResume() {
18         super.onResume()
19         findViewById<TextView>(R.id.txtDato1).text = intent.getStringExtra("dato2")
20     }
21
22     fun abrirActividad2(v : View){
23         var dato1 = findViewById<EditText>(R.id.campoDato1).text.toString()
24         val intento = Intent(packageContext, Actividad2::class.java)
25         intento.putExtra("dato1", dato1)
26         startActivity(intento)
27     }
28 }
```

```
fun abrirActividad2(v : View){  
    var dato1 = findViewById<EditText>(R.id.campoDato1).text.toString()  
    // FORMA # 2  
    val intento = Intent( packageContext: this, Actividad2::class.java).apply { this: Intent  
        putExtra( name: "dato1", dato1)  
    }  
    startActivity(intento)  
}
```



```
//findViewById<TextView>(R.id.txtDato1).text = intent.getStringExtra("dato2")  
// FORMA ALTERNATIVA DE APLICAR CAMBIOS A UN COMPONENTE  
  
val txtDato1 = findViewById<TextView>(R.id.txtDato1).apply { this: TextView!  
    text = intent.getStringExtra( name: "dato2")  
}
```

```
layout_actividad2.xml x Actividad2.kt x layout_actividad1.xml x Actividad1.kt x
    package co.gov.mintic.misiontic.udea.cidlo4

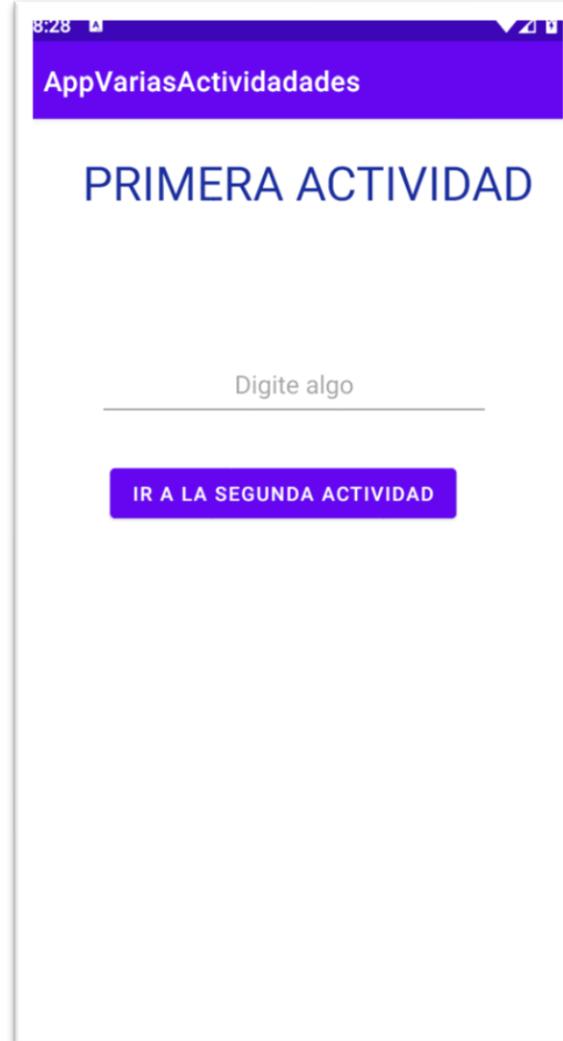
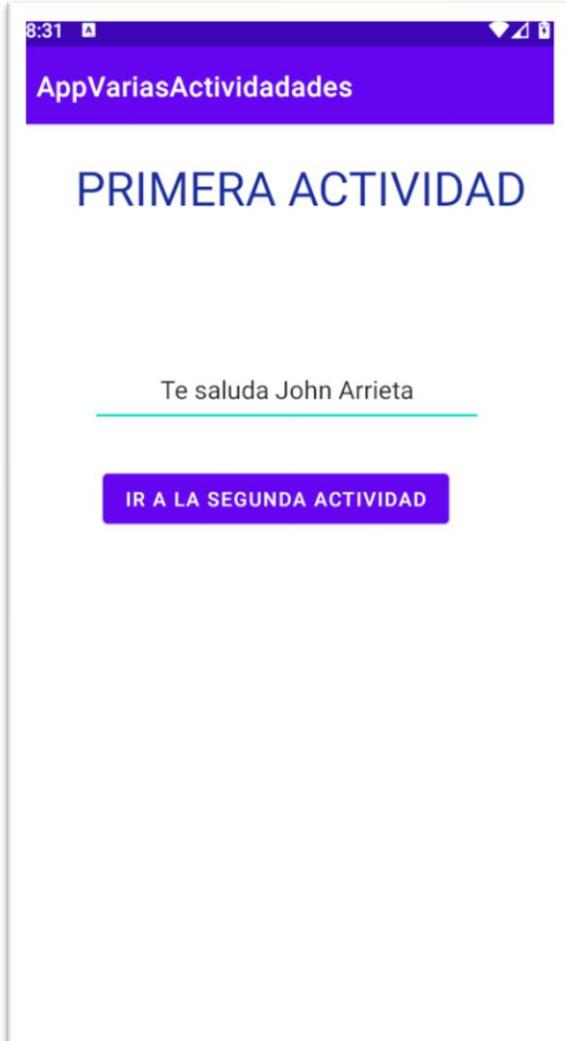
    import android.content.Intent
    import androidx.appcompat.app.AppCompatActivity
    import android.os.Bundle
    import android.view.View
    import android.widget.EditText
    import android.widget.TextView
    import co.gov.mintic.misiontic.udea.appvariasactividades.R

    class Actividad2 : AppCompatActivity() {
        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)
            setContentView(R.layout.layout_actividad2)
        }

        override fun onResume() {
            super.onResume()
            findViewById<TextView>(R.id.txtDato2).text = intent.getStringExtra("dato1")
        }

        fun abrirActividad1(v : View){

            var dato2 = findViewById<EditText>(R.id.campoDato2).text.toString()
            val intento = Intent(packageContext, Actividad1::class.java)
            intento.putExtra("dato2", dato2)
            startActivity(intento)
        }
    }
}
```



# FUNDAMENTOS DE FRAGMENTOS



# ¿Qué es un fragment?

- Un **fragmento** es una sección “**modular**” de interfaz de usuario embebida dentro de una **actividad anfitriona**.
- Permite versatilidad y optimización de diseño
- Se trata de **miniactividades** contenidas dentro de una actividad anfitriona.
- Tiene su propio diseño (un **recurso layout** propio) y su **ciclo de vida**.

# Origen de los Fragments

- La necesidad de usar **fragments** nace con la **versión 3.0 (API 11) de Android** debido a los múltiples tamaños de pantalla que estaban apareciendo en el mercado y a la capacidad de orientación de la interfaz (**Landscape y Portrait**).
- Estas nuevas características requieren dotar a las aplicaciones **Android** de la capacidad para adaptarse y responder a la interfaz de usuario sin importar el dispositivo.

# Fragments

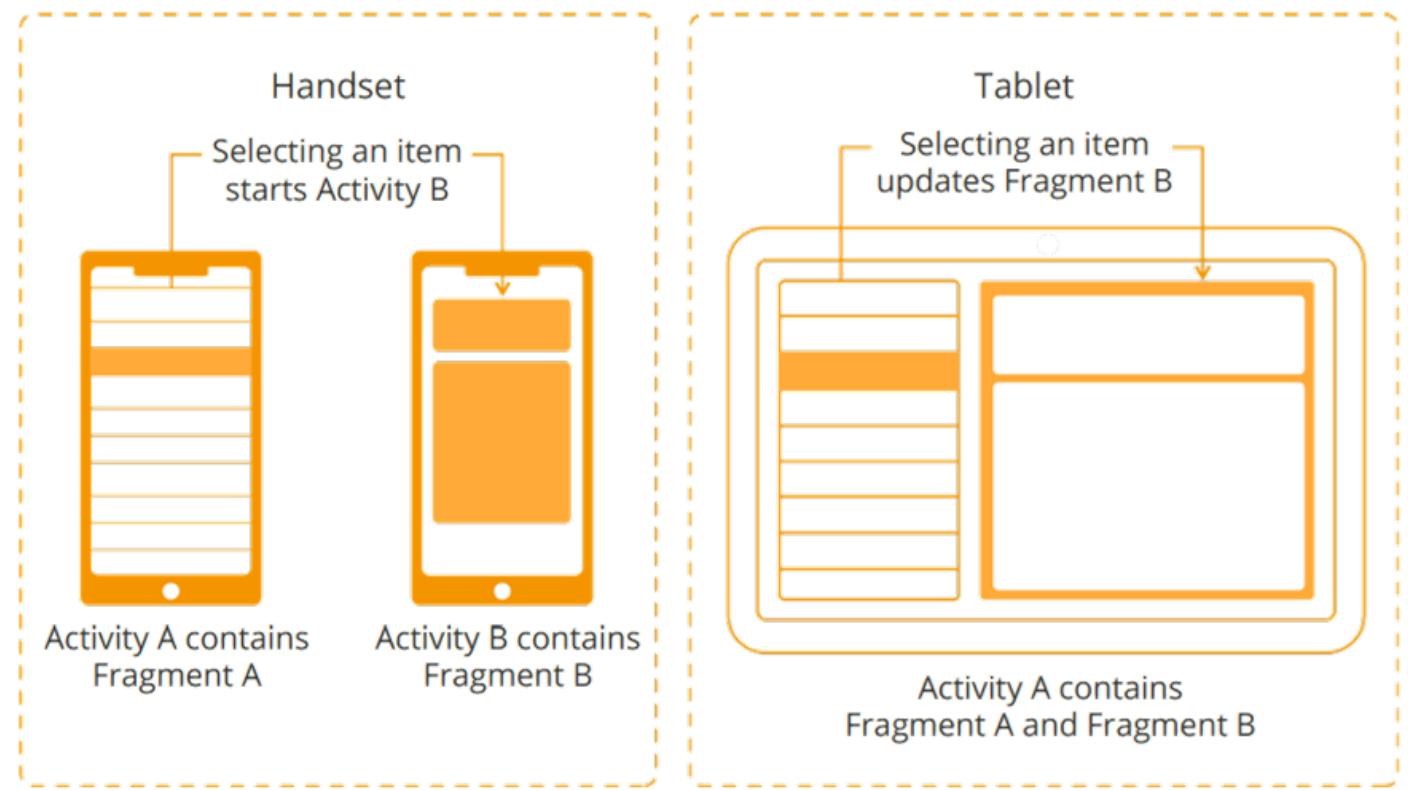
- Los **Fragments** descomponen la funcionalidad de una aplicación y la interfaz de usuario en módulos reusables.
- Se pueden añadir múltiples fragmentos a una Actividad (screen) para evitar estar cambiando entre actividades.
- Los **Fragments** tienen su propio ciclo de vida, estado y stack.
- Requieren **Android 3.0** (API level 11) o superior.

# Fragments

- Desde **HoneyComb (Android 3.0)**, en la implementación de las actividades se pueden usar los **Fragments** para:
  - Modularizar mejor el código.
  - Construir interfaces de usuario de manera más sofisticada para pantallas más grandes.
  - Ayudar a escalar la aplicación entre pantallas pequeñas y grandes.

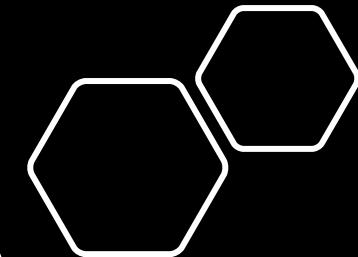
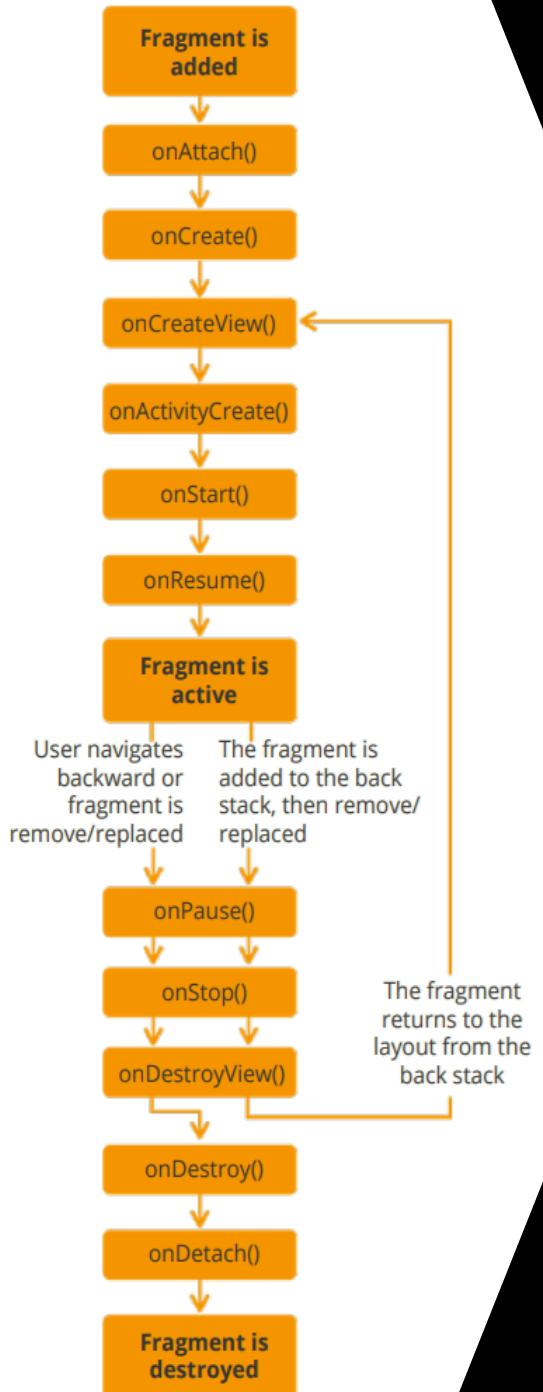
# Actividades: Fragments

Ejemplo de la manera en que dos módulos de la IU definidos por fragmentos se pueden combinar en una actividad para un diseño de tablet y se presentan por separado para un diseño de teléfono.



# Ciclo de vida de un Fragment

- El ciclo de vida de un fragmento depende del ciclo de vida de una actividad



# Ciclo de vida de un Fragment

La administración del ciclo de vida de un fragmento se parece mucho a la del ciclo de vida de una actividad. Al igual que una actividad, un fragmento puede tener tres estados:

## **Reanudado**

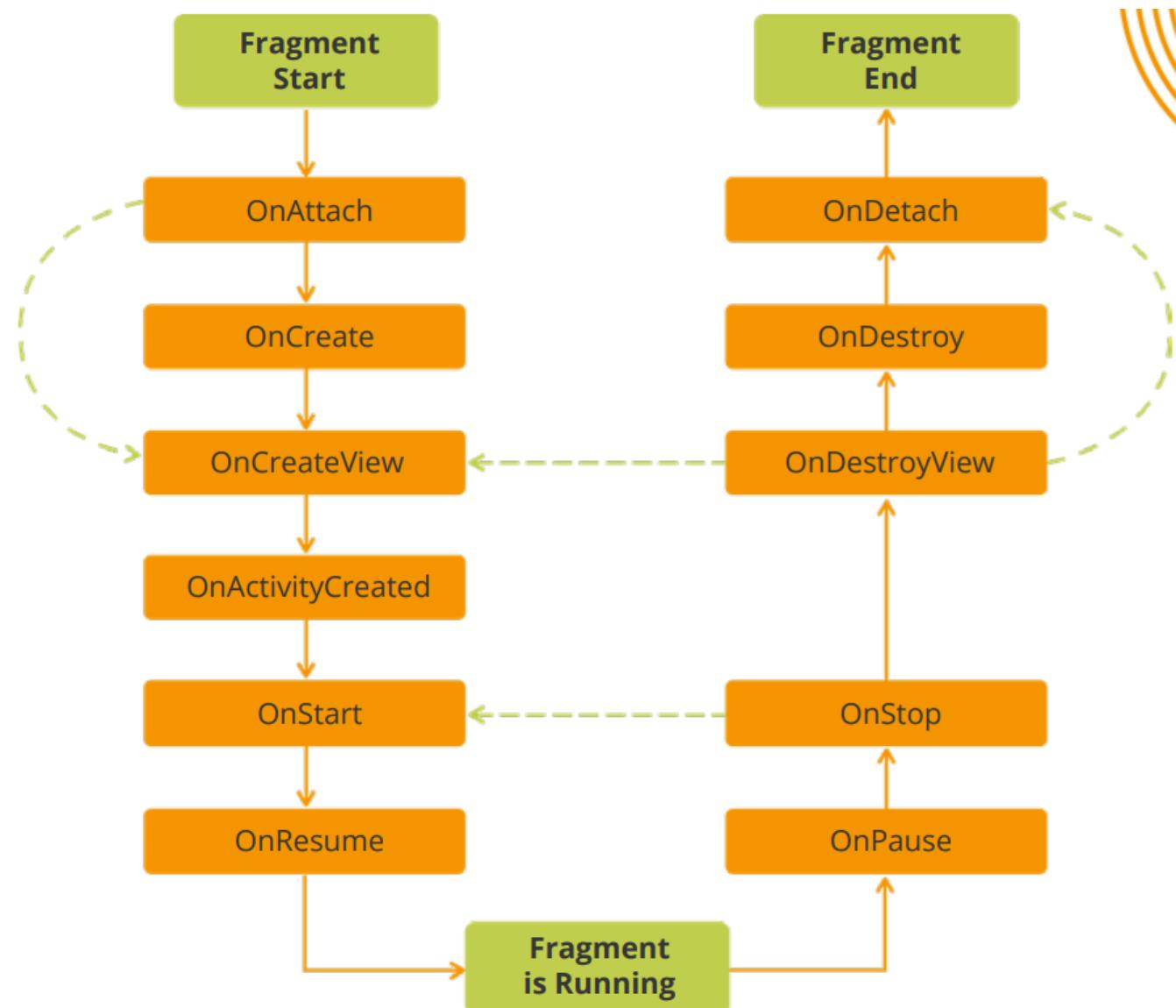
El fragmento está visible en la actividad que se está ejecutando.

## Pausado

Otra actividad se encuentra en primer plano y tiene el foco, pero la actividad en la que reside este fragmento aún está visible (la actividad en segundo plano es parcialmente transparente o no cubre toda la pantalla).

## Detenido

El fragmento no es visible. O bien se detuvo la actividad anfitriona, o bien se quitó el fragmento de la actividad, pero se agregó a la pila de actividades. Un fragmento detenido aún está activo (el sistema conserva el estado y la información de miembro). No obstante, ya no está visible para el usuario y se cerrará si finaliza la actividad.



# Métodos

- **onAttach()**. Es invocado cuando el fragmento ha sido asociado a la actividad anfitriona.
- **onCreate()**. Este método es llamado cuando el fragmento se está creando.
  - En él se pueden inicializar todos los componentes que se desee guardar si el fragmento fue pausado o detenido, luego reiniciado.
- **onCreateView()**. El sistema llama a este método cuando es el momento de crear la UI por primera vez; debe retornar un **view**.
- **onActivityCreated()**. Se ejecuta cuando la actividad anfitriona ya ha terminado la ejecución de su método **onCreate()**.

# Métodos

•**onStart()**. Se llama cuando el fragmento está visible ante el usuario. Obviamente depende del método onStart() de la actividad para saber si la actividad se está mostrando.

•**onResume()**. Es ejecutado cuando el fragmento está activo e interactuando con el usuario.

-Esta situación depende de que la actividad anfitriona esté primero en su estado Resumed.

•**onPause()**. Al igual que las actividades, onPause se ejecuta cuando se detecta que el usuario dirigió el foco por fuera del fragmento.

-Aquí es donde se debe enviar cualquier cambio que deseé persistir más allá de la sesión del usuario actual, ya que este podría no volver.

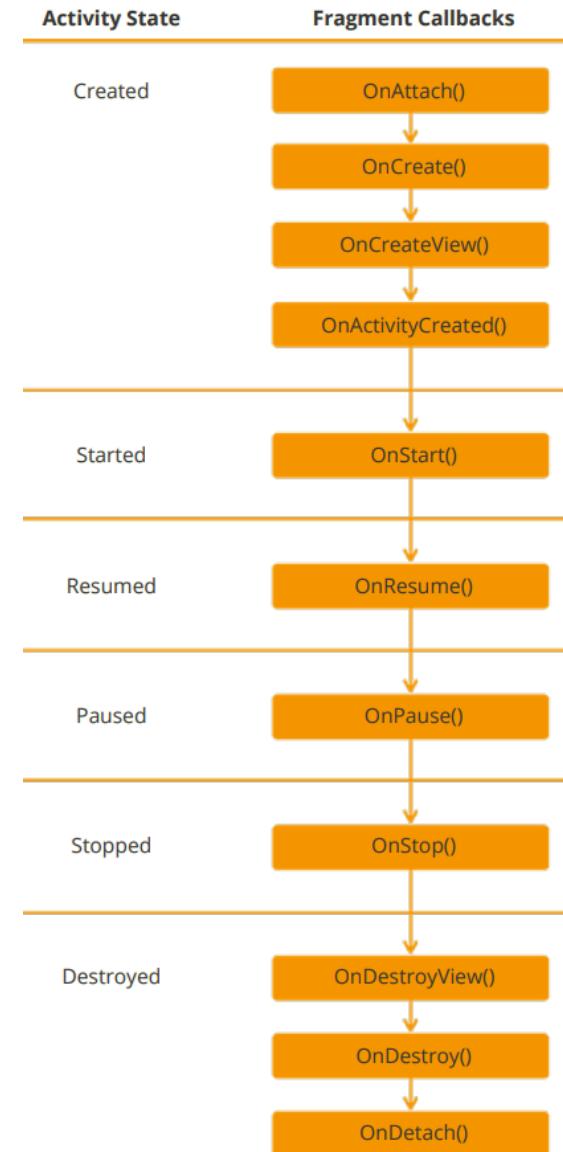
# Métodos

- **onStop()**. Se llama cuando un fragmento ya no es visible para el usuario
  - La actividad anfitriona está detenida.
  - Dentro de la actividad se está gestionando una operación de fragmentos.
- **onDestroyView()**. Este método es llamado cuando la jerarquía de views a la cual ha sido asociado el fragmento ha sido destruida.
- **onDetach()**. Se llama cuando el fragmento ya no está asociado a la actividad anfitriona.

# Ciclo de vida de un Fragment

El efecto del ciclo de vida de la actividad anfitriona en el ciclo de vida del fragment

Un fragmento siempre debe estar alojado en una actividad y el ciclo de vida del fragmento se ve afectado directamente por el ciclo de vida de la actividad anfitriona. Por ejemplo, cuando la actividad está pausada, también lo están todos sus fragmentos, y cuando la actividad se destruye, lo mismo ocurre con todos los fragmentos.



# ¿Cómo usar un Fragment?

## Uso de un fragment

Un **Fragment**, como una actividad tiene:

- Un **XML** con un **layout** para su **interfaz**
- Una clase **Java/Kotlin** que representa el controlador del **Fragment**

Para crear un **fragment** se debe heredar de la clase **Fragment**, e implementar como mínimo los siguientes métodos:

- **onCreate()**
- **onCreateView()**

## Uso de un Fragment

Ejemplo del **XML** con el archivo layout con el nombre **fragment\_a.xml**.

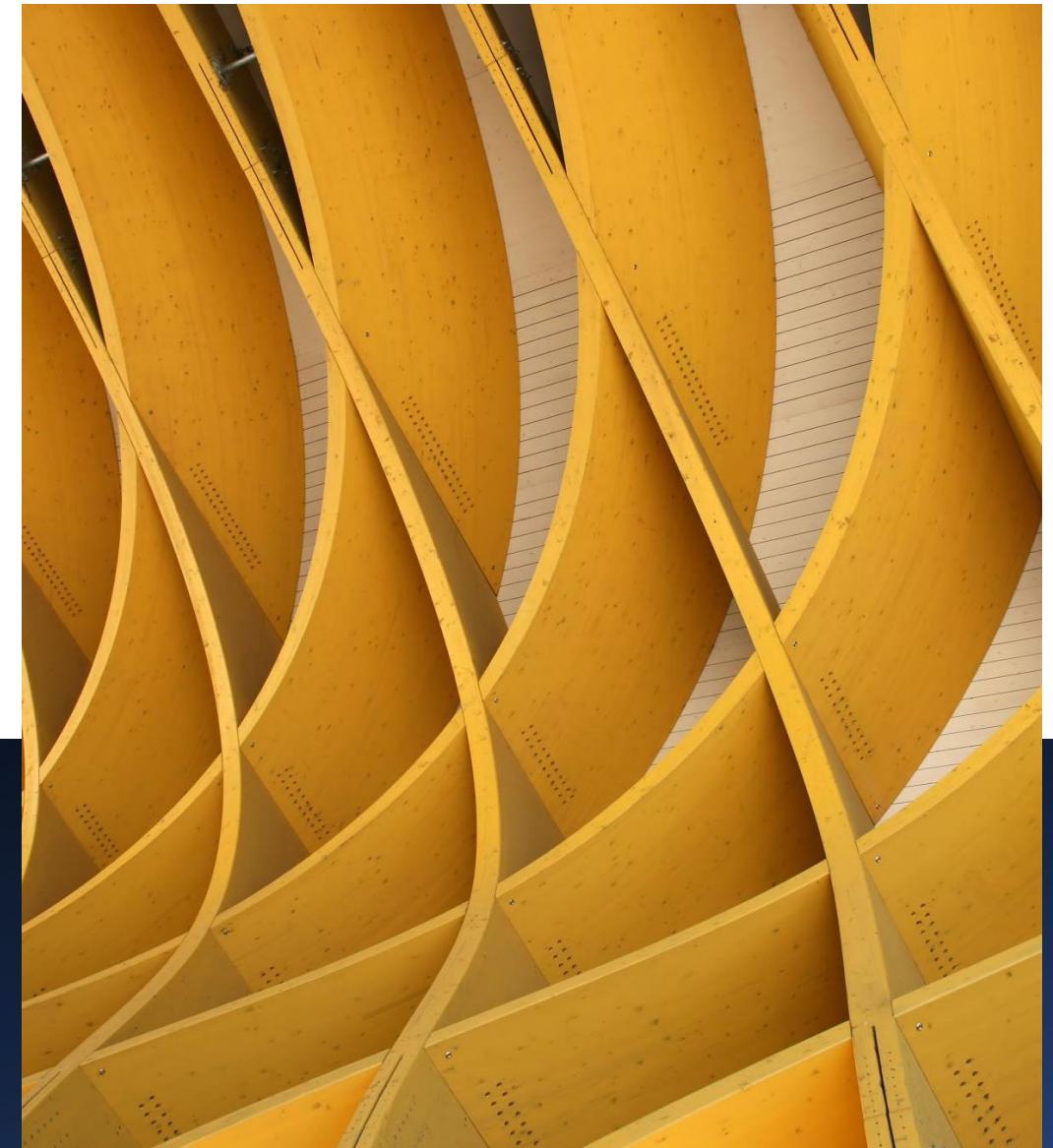
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

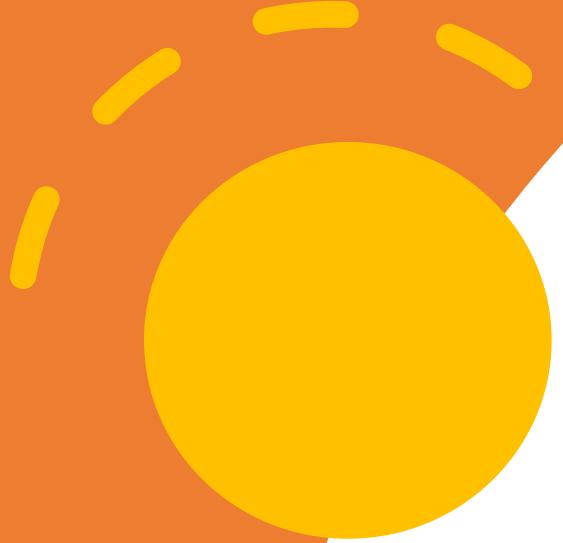
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />

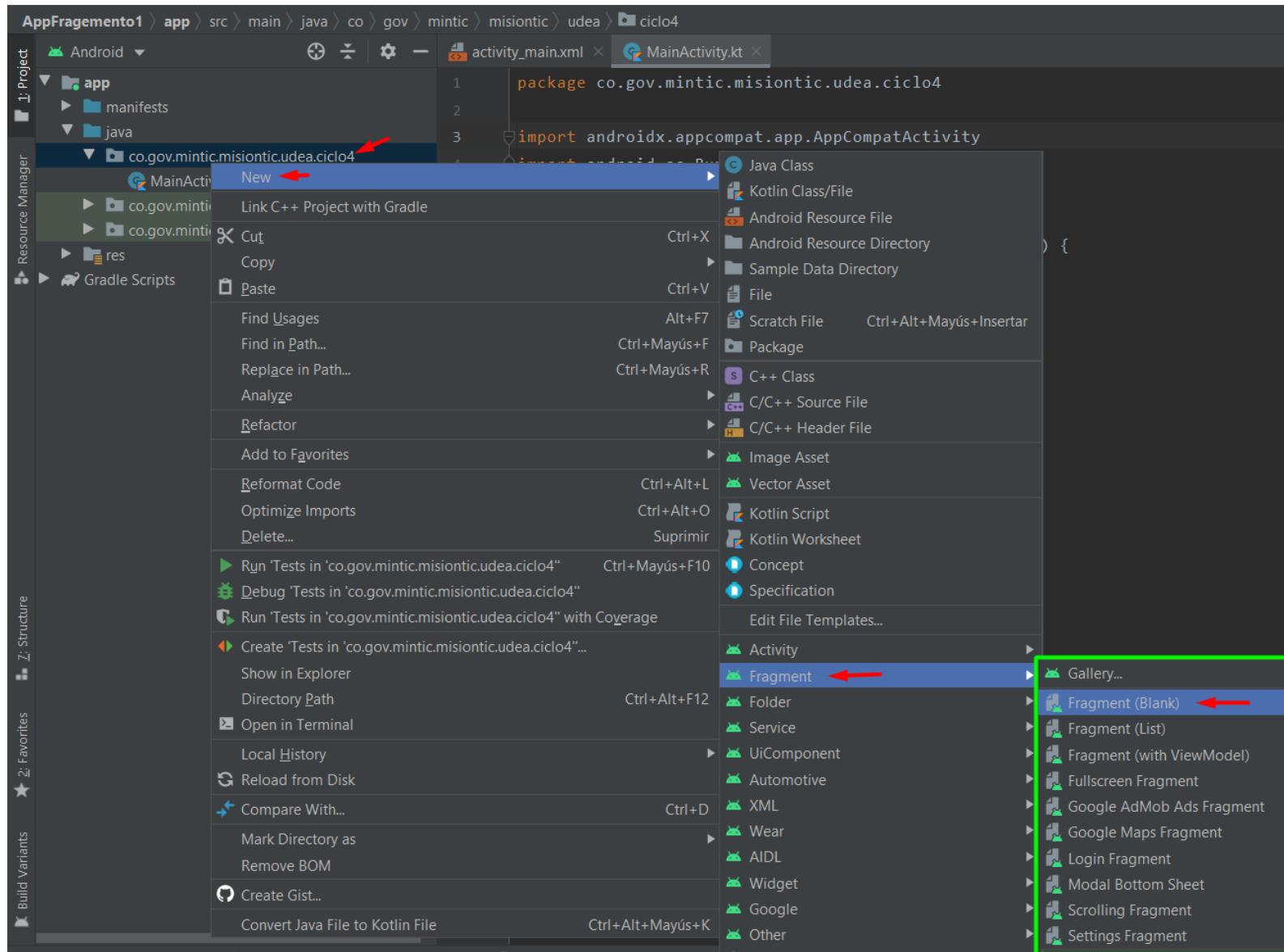
</LinearLayout>
```

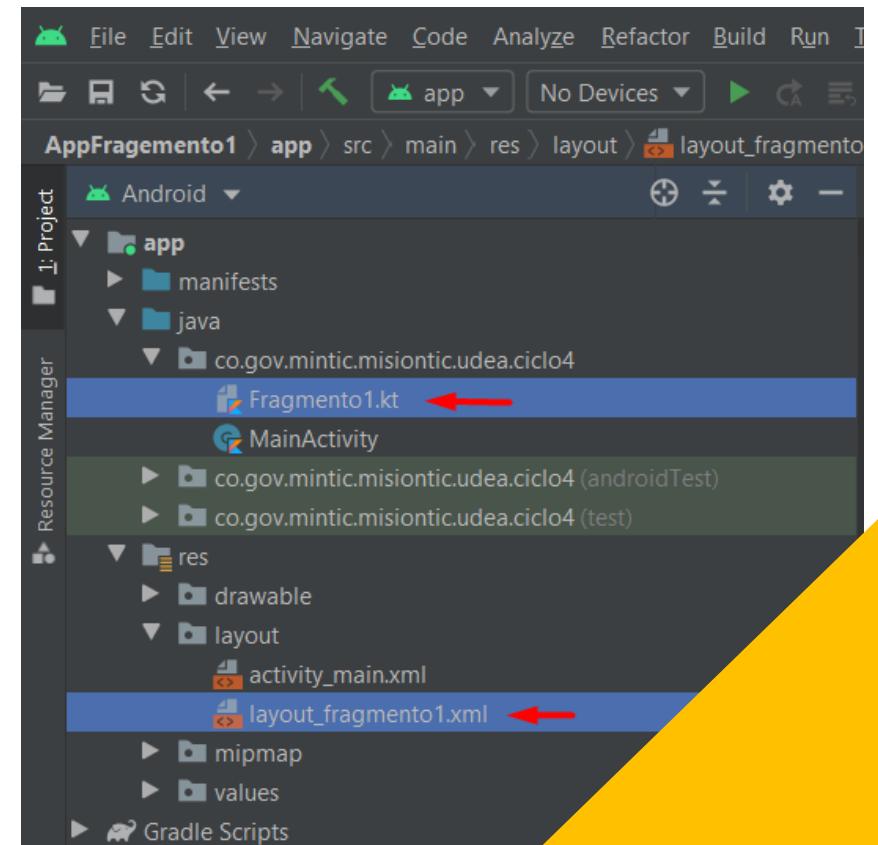
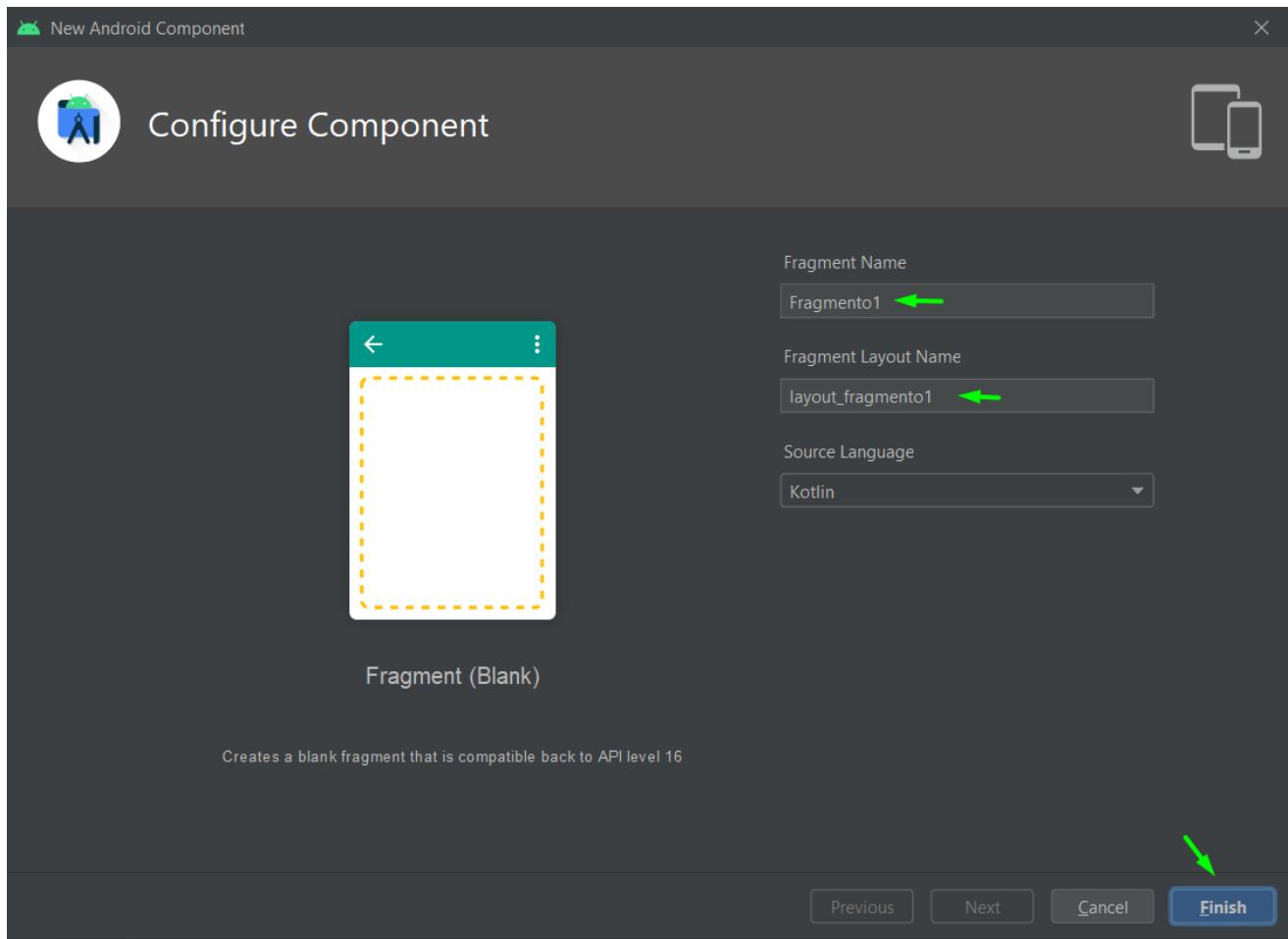
# EJEMPLO





Crear y definir un  
Actividad de tipo  
Fragmento





```
1  <?xml version="1.0" encoding="utf-8"?>
2  <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".Fragmento1">
7
8      <!-- TODO: Update blank fragment layout -->
9      <TextView
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         android:text="Estes un Fragmento en Blanco" />
13
14  </FrameLayout>
15
16  |
```

Estes un Fragmento en Blanco

```
activity_main.xml × MainActivity.kt × layout_fragmento1.xml × Fragmento1.kt ×  
1 package co.gov.mintic.misiontic.udea.ciclo4  
2  
3 import android.os.Bundle  
4 import androidx.fragment.app.Fragment  
5 import android.view.LayoutInflater  
6 import android.view.View  
7 import android.view.ViewGroup  
8  
9 // TODO: Puedes cambiar el nombre de los parametros  
10 // Los parametros de inicializacion del Fragmento  
11 private const val ARG_PARAM1 = "param1"  
12 private const val ARG_PARAM2 = "param2"  
13  
14 /**  
 * Esta es subclase de [Fragment].  
 * Utiliza el método [Fragmento1.newInstance] para  
 * crear una instancia de este fragmento.  
 */  
15 class Fragmento1 : Fragment() {  
16     // TODO: Puedes cambiar los nombres y tipos de los parámetros  
17     private var param1: String? = null  
18     private var param2: String? = null  
19  
20     override fun onCreate(savedInstanceState: Bundle?) {  
21         super.onCreate(savedInstanceState)  
22         arguments?.let { it: Bundle  
23             param1 = it.getString(ARG_PARAM1)  
24             param2 = it.getString(ARG_PARAM2)  
25         }  
26     }  
27  
28     override fun onCreateView(savedInstanceState: Bundle?, container: ViewGroup?,  
29     savedInstanceState: Bundle?): View? {  
30         return inflater.inflate(R.layout.layout_fragmento1, container, false)  
31     }  
32 }
```

```
34 override fun onCreateView(  
35     inflater: LayoutInflater, container: ViewGroup?,  
36     savedInstanceState: Bundle?  
37 ): View? {  
38     // Inflar el diseño de este fragmento  
39     return inflater.inflate(R.layout.layout_fragmento1, container, attachToRoot: false)  
40 }  
41  
42 companion object {  
43     /**  
 * Utiliza este método para crear una nueva instancia de  
 * este fragmento y pasale los parametros necesarios.  
 * @param param1 Parameter 1.  
 * @param param2 Parameter 2.  
 * @return Una nueva instancia del fragmento Fragmento1  
 */  
44     // TODO: Puedes cambiar los nombres y tipos de los parámetros  
45     @JvmStatic  
46     fun newInstance(param1: String, param2: String) =  
47         Fragmento1().apply { this:Fragmento1  
48             arguments = Bundle().apply { this: Bundle  
49                 putString(ARG_PARAM1, param1)  
50                 putString(ARG_PARAM2, param2)  
51             }  
52         }  
53     }  
54 }
```

```
activity_main.xml × MainActivity.kt × layout_fragmento1.xml × Fragmento1.kt ×
1 package co.gov.mintic.misiontic.udea.ciclo4
2
3 import android.os.Bundle
4 import androidx.fragment.app.Fragment
5 import android.view.LayoutInflater
6 import android.view.View
7 import android.view.ViewGroup
8
9 /**
10 * Esta es una subclase de [Fragment]
11 */
12 class Fragmento1 : Fragment() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15     }
16
17     override fun onCreateView(
18         inflater: LayoutInflater, container: ViewGroup?,
19         savedInstanceState: Bundle?
20     ): View? {
21         // Inflar el diseño de este fragmento
22         return inflater.inflate(R.layout.layout_fragmento1, container, attachToRoot: false)
23     }
24 }
```

# Uso de un Fragment

Ejemplo del código Java  
del controller

```
public class FragmentA extends Fragment {  
  
    // The onCreateView method is called when Fragment should  
    // create its View object hierarchy,  
    // either dynamically or via XML layout inflation.  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup  
parent, Bundle savedInstanceState) {  
        // Defines the xml file for the fragment  
        return inflater.inflate(R.layout.fragment_a, parent, false);  
    }  
  
    // This event is triggered soon after onCreateView().  
    // Any view setup should occur here.  
    //E.g., view lookups and attaching view listeners.  
    @Override  
    public void onViewCreated(View view, Bundle  
savedInstanceState) {  
        // Setup any handles to view objects here  
        // EditText etFoo = (EditText) view.findViewById(R.id.etFoo);  
    }  
}
```

# Fragment estático

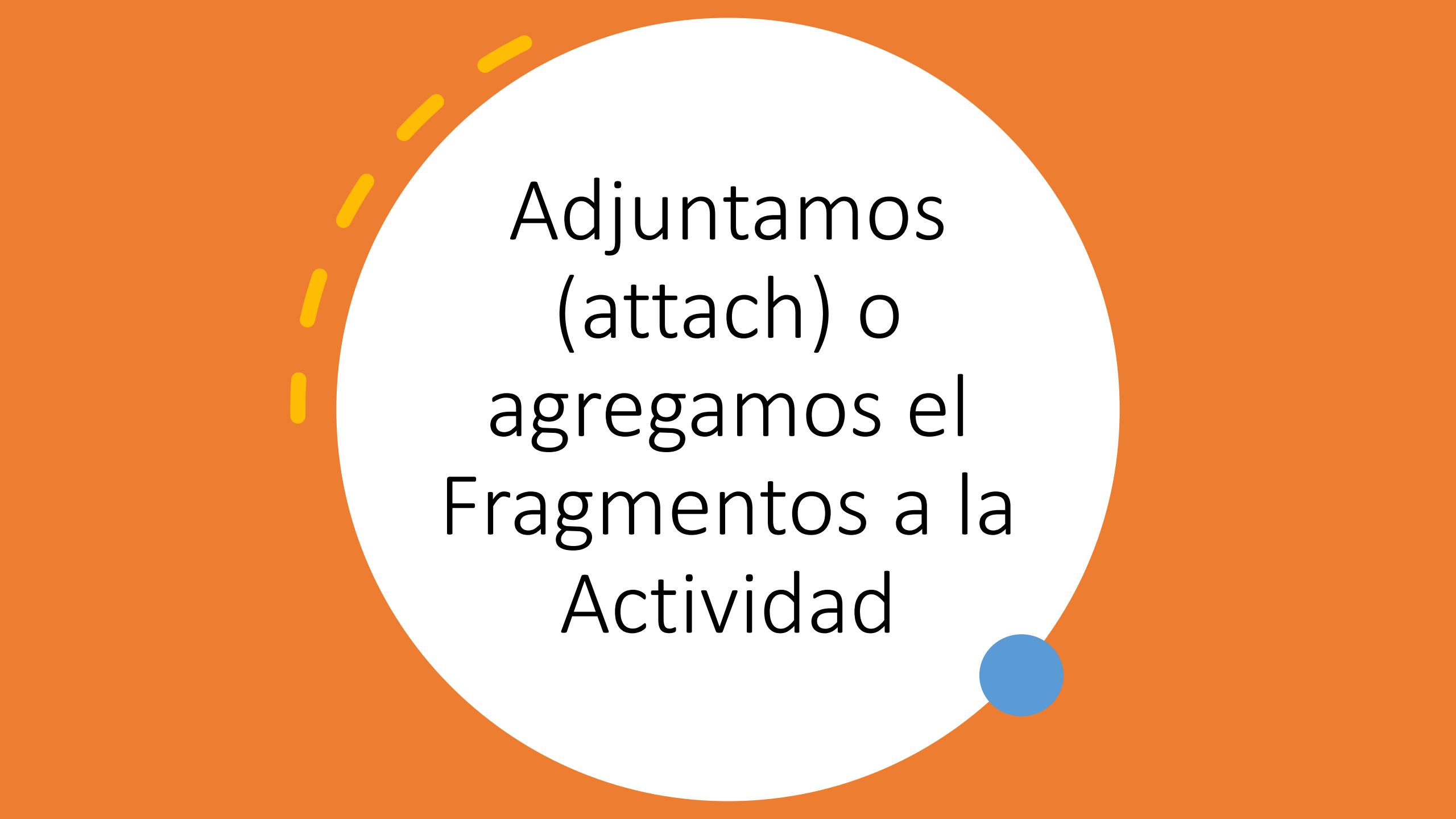
- Para usar un fragment estático simplemente lo ponemos dentro del **xml** del **layout de la actividad** anfitriona.
- No se puede reemplazar un fragment definido **estáticamente**.

## activity\_main.xml

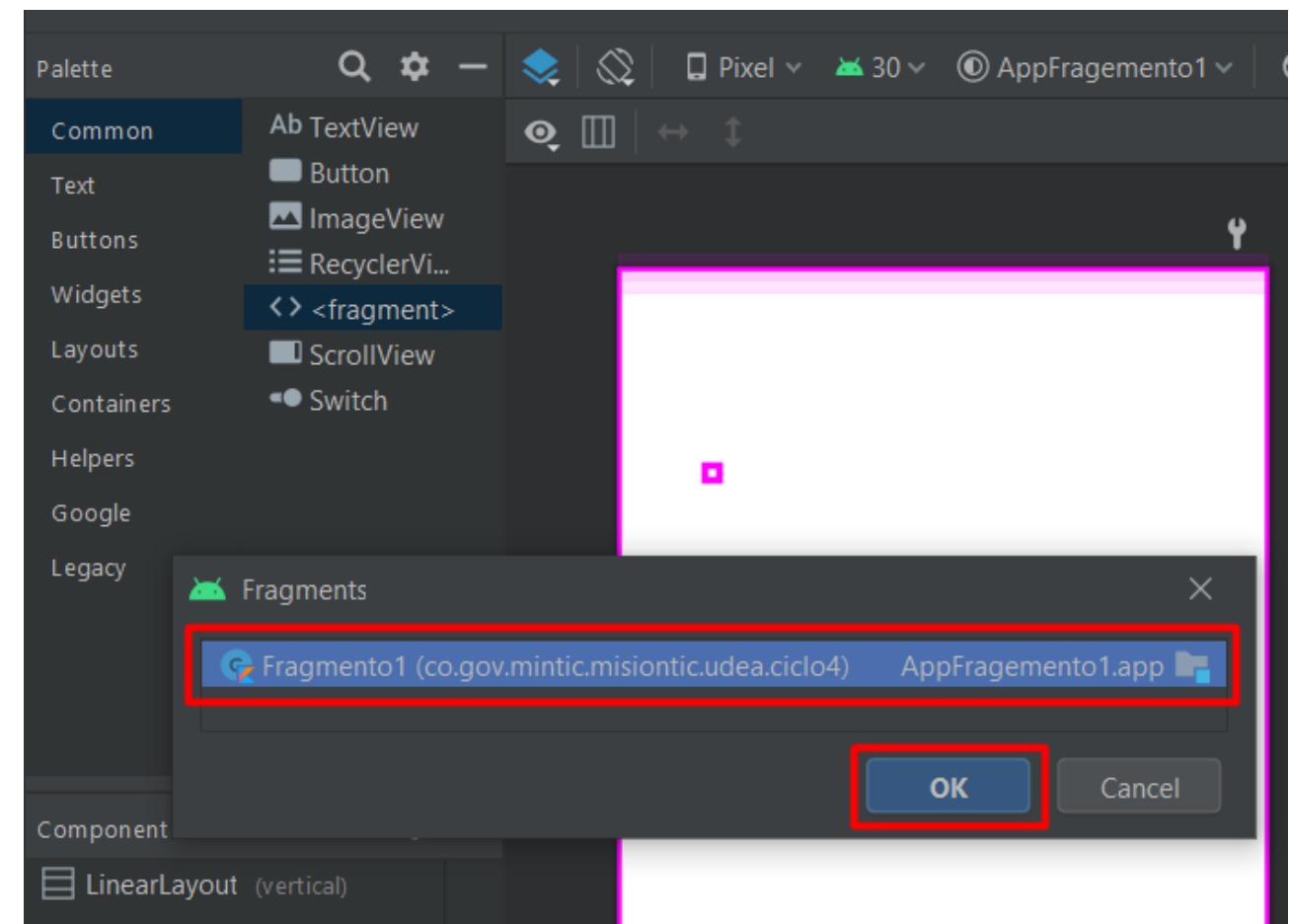
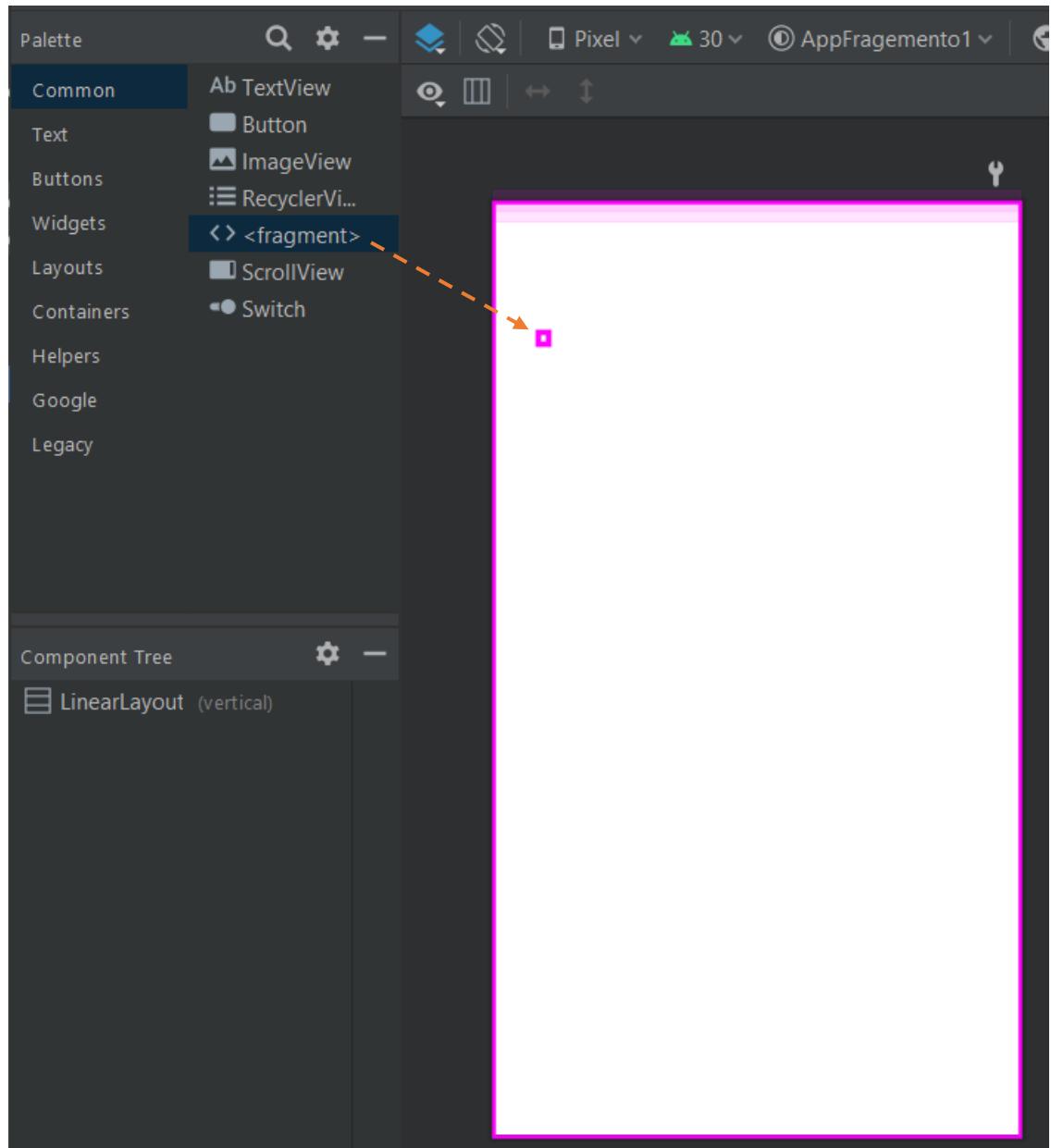
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <fragment
        android:name="com.example.android.FragmentA"
        android:id="@+id/FragmentA"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

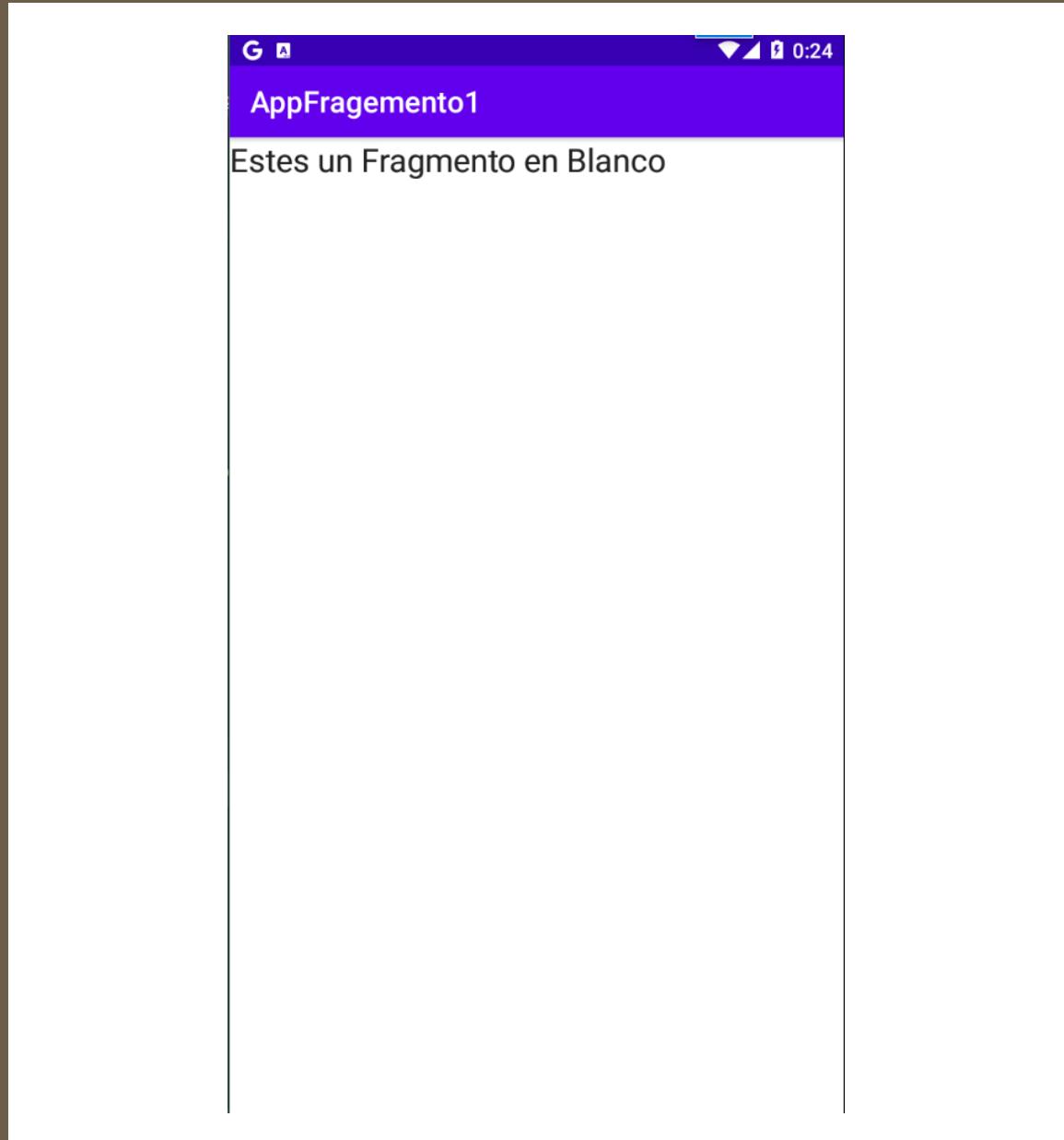
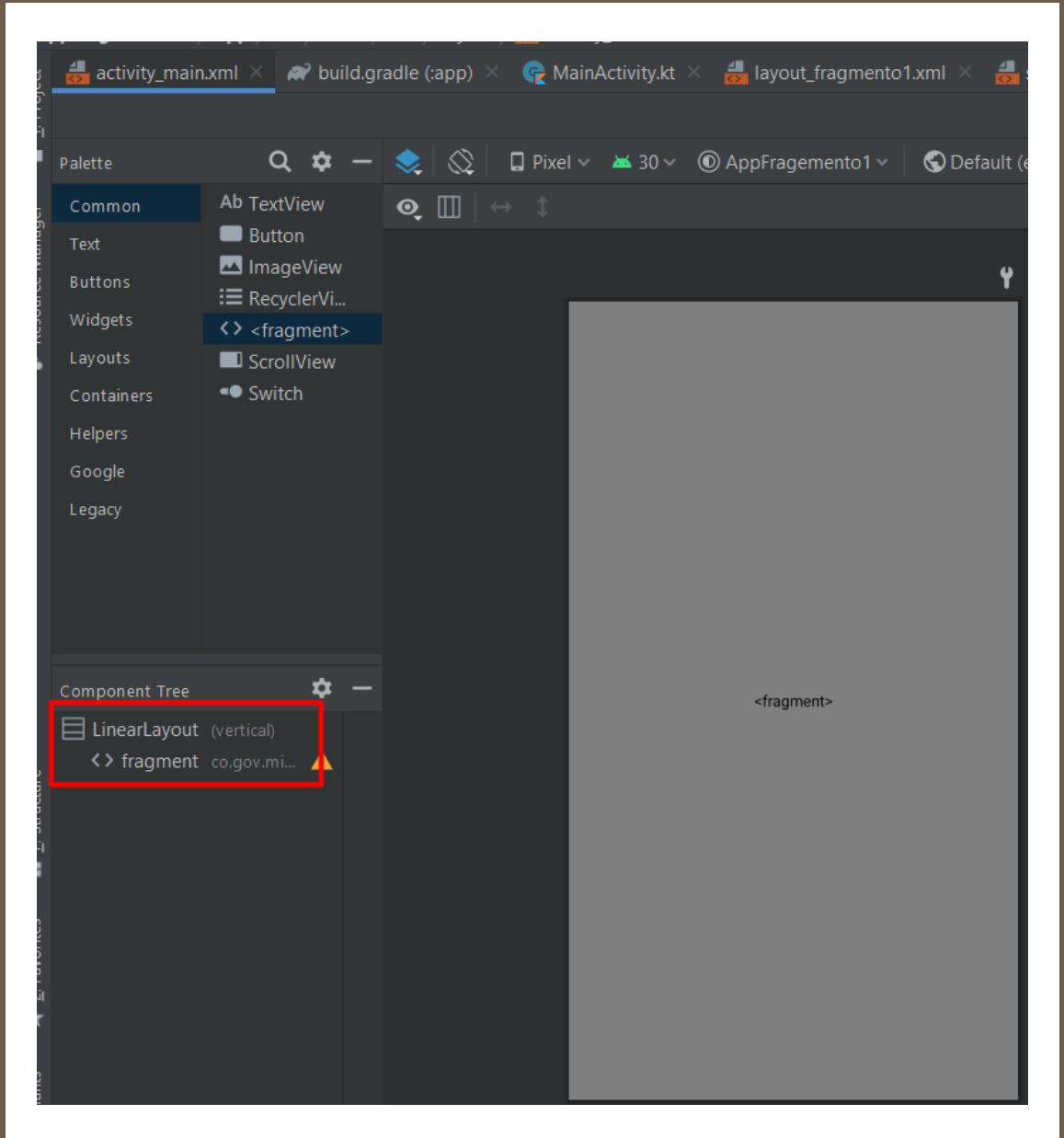
</LinearLayout>
```



Adjuntamos  
(attach) o  
agregamos el  
Fragmentos a la  
Actividad



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity">
9
10 <fragment
11     android:id="@+id/fragment"
12     android:name="co.gov.mintic.misiontic.udea.ciclo4.Fragmento1"
13     android:layout_width="match_parent"
14     android:layout_height="match_parent" />
15 </LinearLayout>
16
17
```





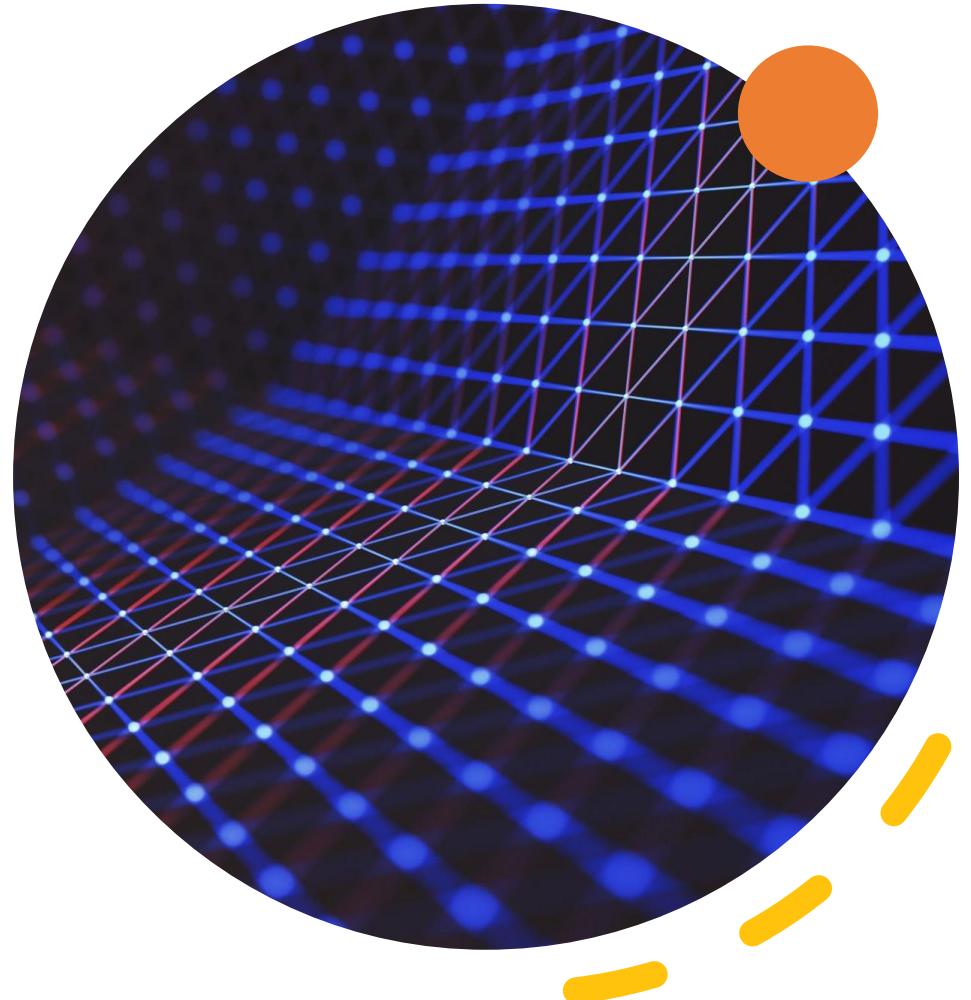
## EJEMPLO DE FRAGMENTOS DINAMICOS

CREAMOS UN PROYECTO

CREAMOS AL MENOS 2  
ACTIVIDADES DE TIPO  
FRAGMENTO

REALIZAMOS EL DISEÑO DE CADA  
UNA

ASIGNAMOS UN ID A CADA  
LAYOUT VINCULADO A CADA  
FRAGMENTO



### layout\_fragmento1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
layout:_fragmento1.xml
-->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout_fragmento1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00BCD4"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="Primer Fragmento"
        android:textAppearance="@style/TextAppe"
        android:textColor="#132CB6" />

</LinearLayout>
```

### layout\_fragmento2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
layout:_fragmento2.xml
-->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout_fragmento2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#4CAF50"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Segundo Fragmento"
        android:textAppearance="@style/TextAppe"
        android:textColor="#132CB6" />

</LinearLayout>
```

La clase de cada fragmento debe heredar de **Fragment** o de alguna de sus clases hijas

Debemos sobrescribir el metodo **onCreate**

Sobre escribir el metodo **onCreateView**

Retorna view el fragmento

Crear una instancia del fragmento deseado dentro de la actividad anfitriona





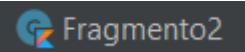
Fragmento1

```
package co.gov.mintic.misiontic.udea.ciclo4

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment

class Fragmento1 : Fragment() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.layout_fragmento1, container, false)
    }
}
```



```
package co.gov.mintic.misiontic.udea.ciclo4

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment

class Fragmento1 : Fragment() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.layout_fragmento1, container, attachToRoot: false)
    }
}
```

Definimos el diseño de la actividad anfitriona

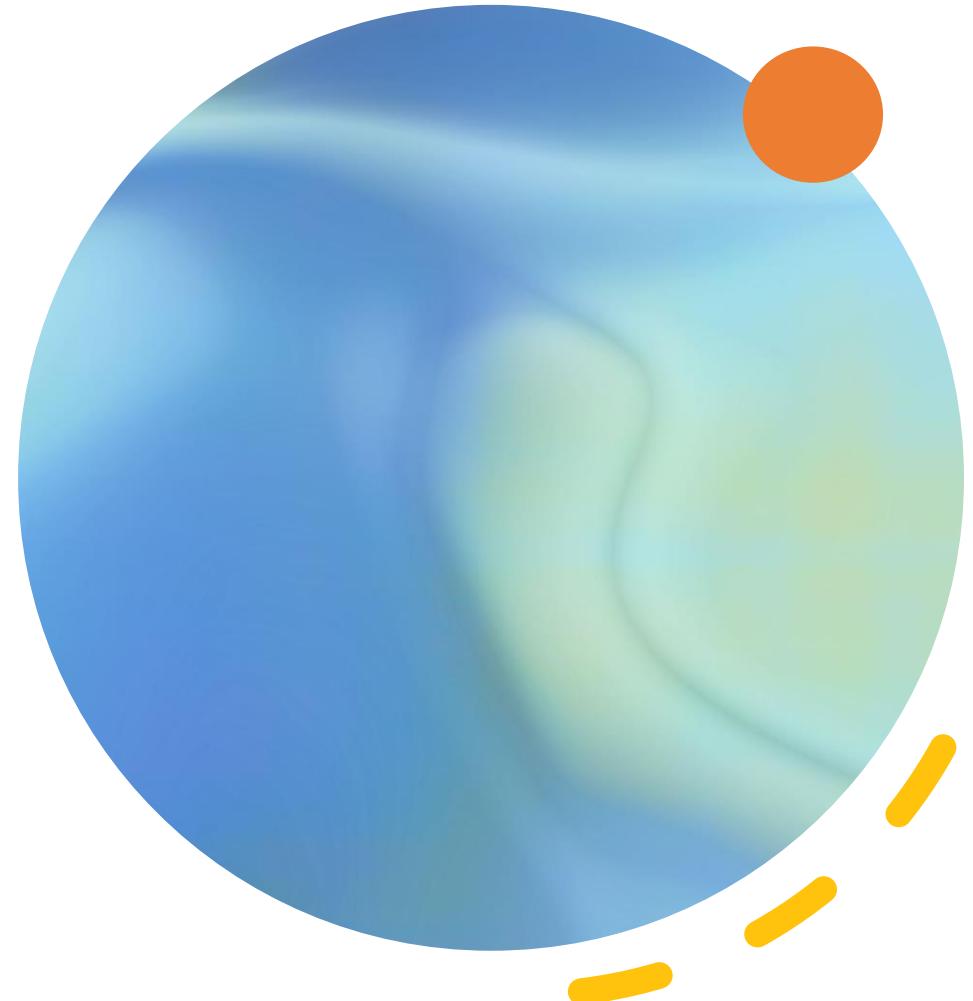
Por ejemplo:

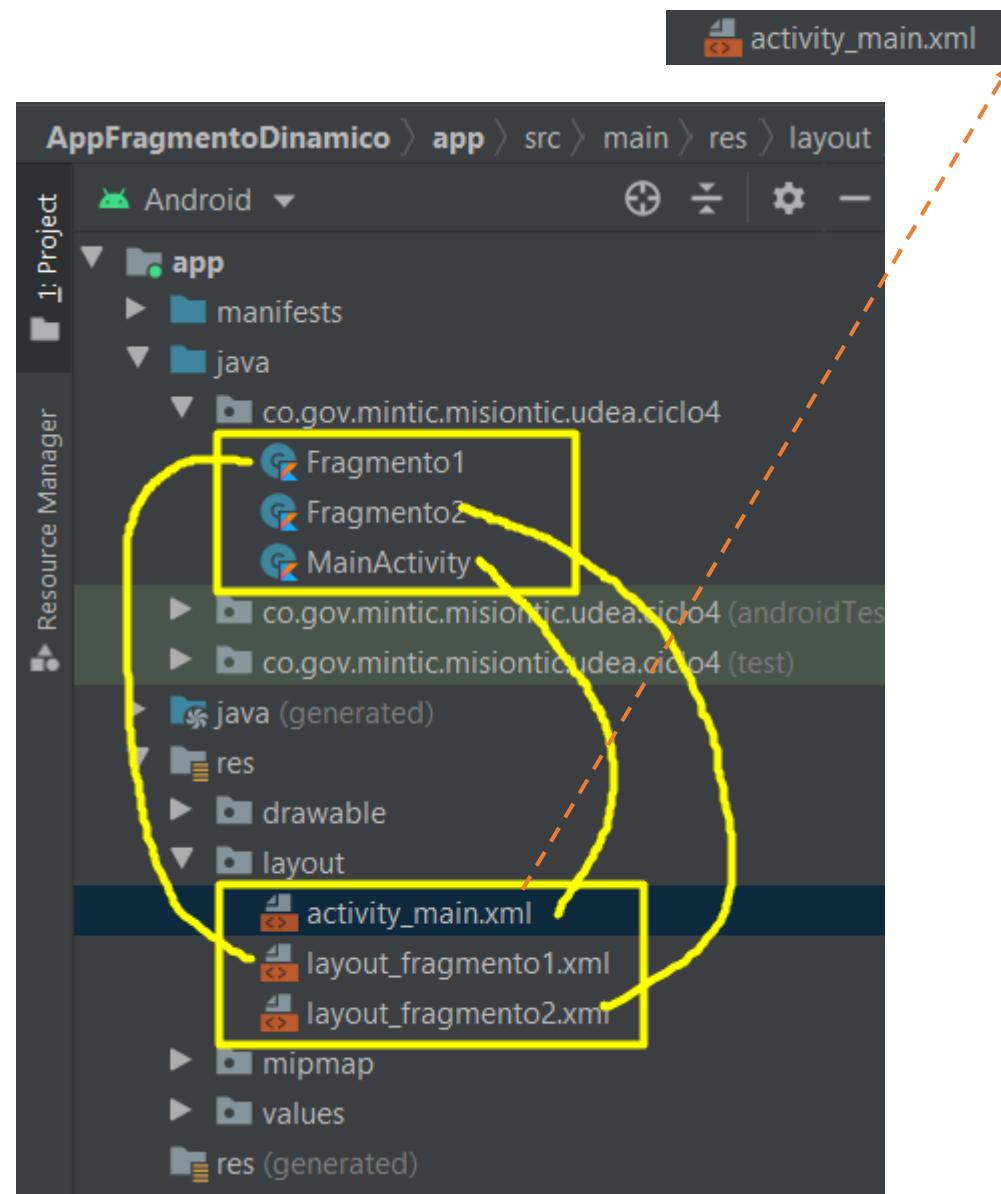
Un boton (**<Button>**) para abrir agregar un Fragmentos

Por ejemplo:

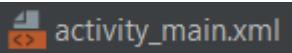
Un boton (**<Button>**) para abrir agregar otro Fragmentos

Agregar un contedor (**Layout**) para colocar los Fragmentos





```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:layout_width="match_parent"    android:layout_height="match_parent"
    android:orientation="vertical"        tools:context=".MainActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"      android:gravity="center"
        android:text="ACTIVIDAD PRINCIPAL"   android:textColor="#ED2C1E"
        android:textSize="30sp" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/btnFragmento1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="30dp"
            android:layout_weight="5"
            android:onClick="mostrarFragmento1"
            android:text="Freamento 1" />
        <Button
            android:id="@+id/btnFragmento2"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="30dp"
            android:layout_marginRight="30dp"
            android:layout_weight="5"
            android:onClick="mostrarFragmento2"
            android:text="Fragmento 2" />
    </LinearLayout>
    <LinearLayout
        android:id="@+id/contenedor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"/>
</LinearLayout>
```



## ACTIVIDAD PRINCIPAL

FREAGMENTO 1

FRAGMENTO 2



Primer Fragmento



Segundo Fragmento

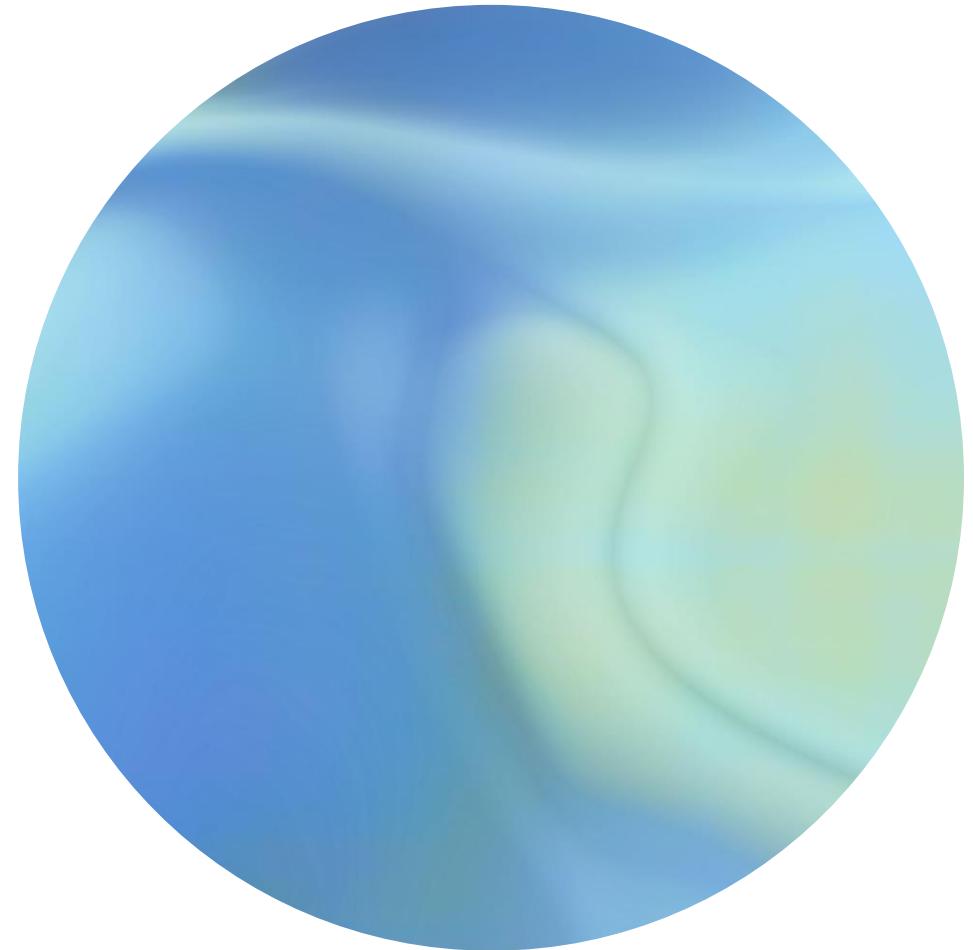
En la actividad anfitriona obtenemos  
un objeto de la clase  
**FragmentManager**

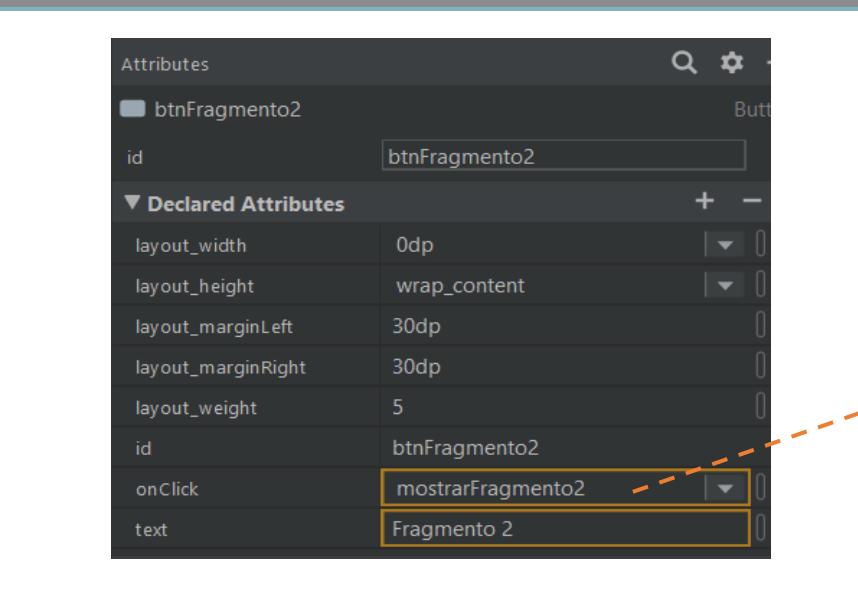
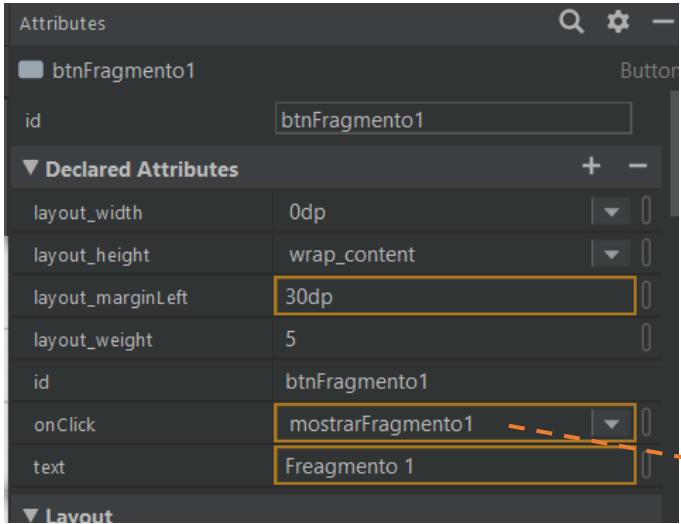
Con el manejador obtenemos un  
**FragmentTransaction**

Creamos un objeto de tipo Fragmento

Agregamos el Fragmento en el layout  
de la actividad anfitriona

Confirmamos la transaccion





```
package co.gov.mintic.misiontic.udea.ciclo4

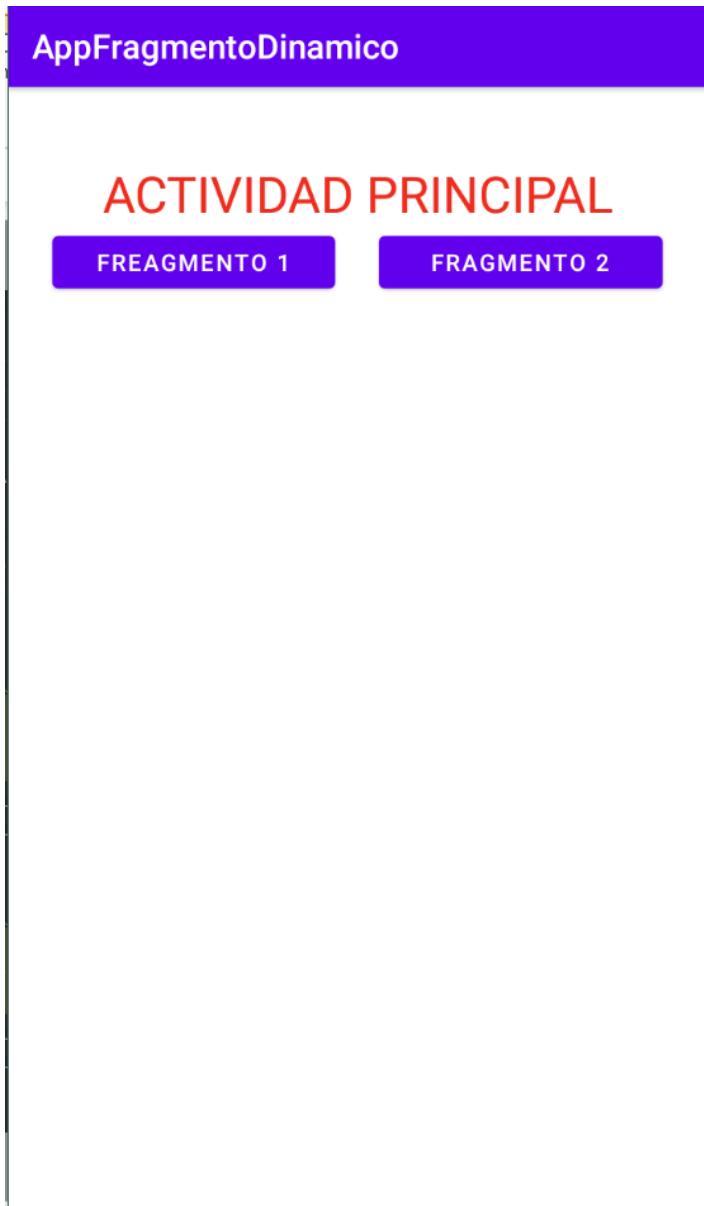
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View

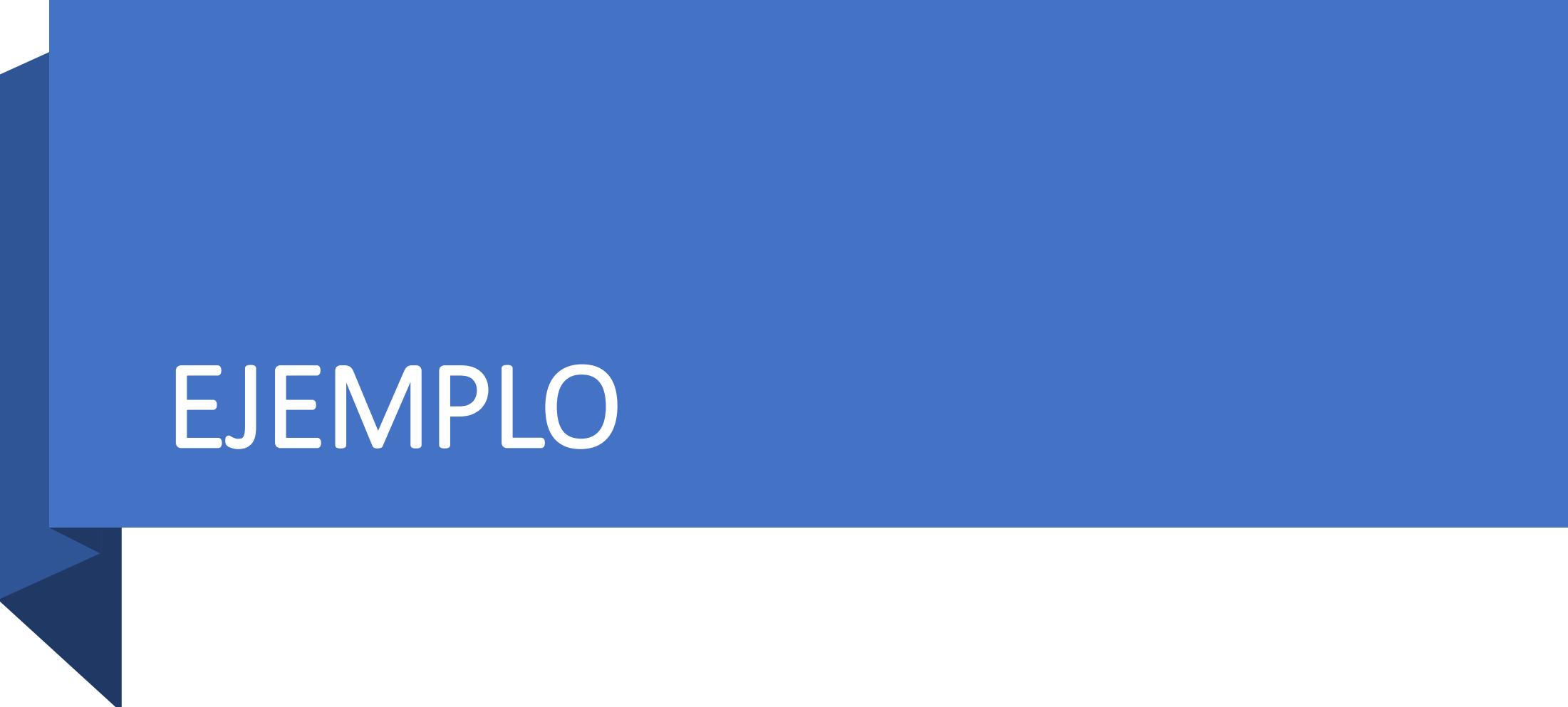
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun mostrarFragmento1(v : View){
        var manejador = supportFragmentManager
        var transicion = manejador.beginTransaction()
        var frag1 = Fragmento1()
        transicion.add(R.id.contenedor, frag1)
        transicion.commit()
    }

    fun mostrarFragmento2(v : View){
        var manejador = supportFragmentManager
        var transicion = manejador.beginTransaction()
        var frag2 = Fragmento2()
        transicion.add(R.id.contenedor, frag2)
        transicion.commit()
    }
}
```





EJEMPLO

Crear un Proyecto

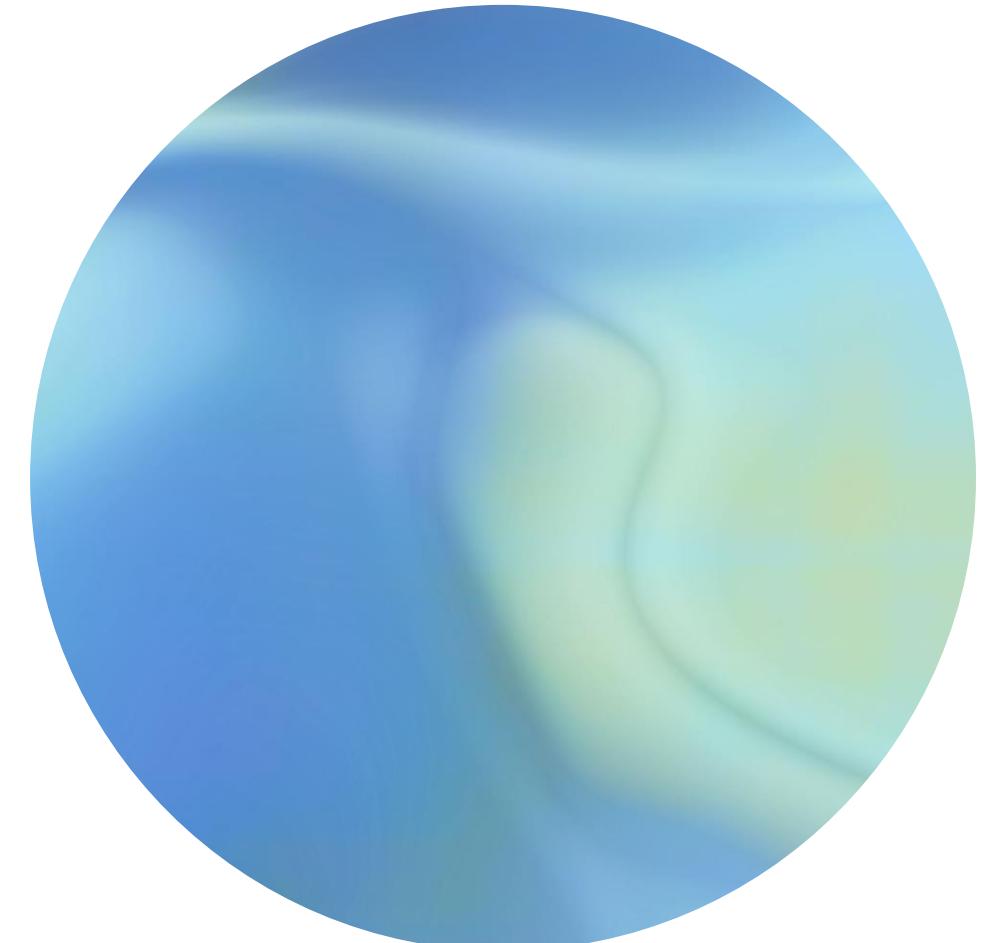
2 Actividades

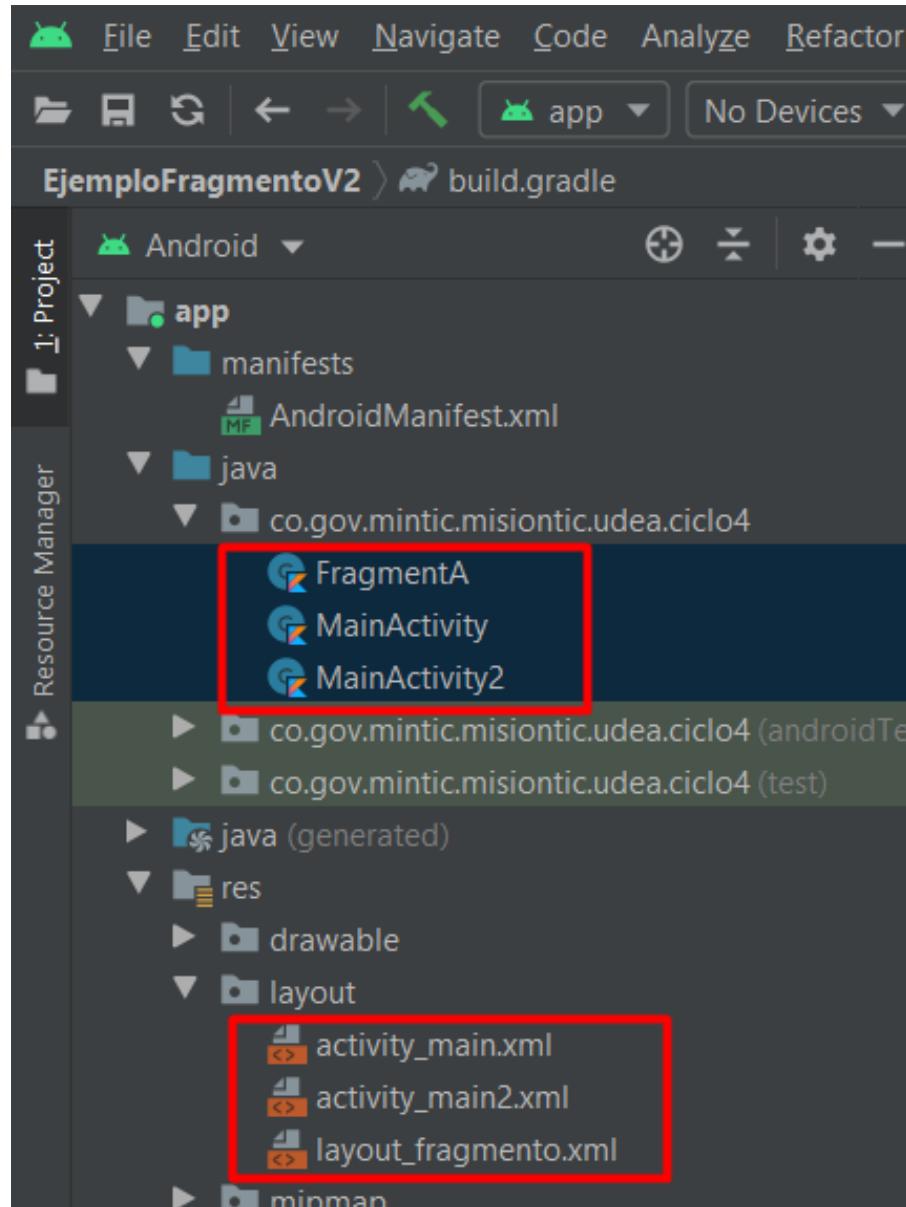
1 Fragmento

El Fragmento debe tener dos botones y una Etiqueta, la etiqueta debe mostrar el numero de clic que se dan en un boton y el otro boton reiniciara el Contador de clic

Las dos actividades deben agregar o vincular el mismo fragment

Una actividad debe poder mostrar a la otra y viceversa





```
<RelativeLayout 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:layout_width="match_parent"    android:layout_height="match_parent">

    <TextView
        android:id="@+id/txtMensaje"    android:layout_width="match_parent"
        android:layout_height="wrap_content"    android:layout_alignParentTop="true"
        android:layout_marginTop="85dp"    android:text=""
        android:textAlignment="center"    android:textSize="30sp" />

    <Button
        android:id="@+id(btnLimpilar"    android:layout_width="125dp"
        android:layout_height="wrap_content"    android:layout_below="@+id/txtMensaje"
        android:layout_alignParentStart="true"    android:layout_marginStart="43dp"
        android:layout_marginTop="93dp"    android:layout_marginEnd="57dp"
        android:layout_toStartOf="@+id	btnCuenta"
        android:drawableStart="@android:drawable/ic_menu_close_clear_cancel"
        android:text="Quitar" />

    <Button
        android:id="@+id	btnCuenta"    android:layout_width="wrap_content"
        android:layout_height="wrap_content"    android:layout_below="@+id/txtMensaje"
        android:layout_alignParentEnd="true"    android:layout_marginTop="89dp"
        android:layout_marginEnd="75dp"    android:drawableEnd="@android:drawable/ic_input_add"
        android:text="Contar"    tools:layout_editor_absoluteX="241dp"
        tools:layout_editor_absoluteY="307dp" />

</RelativeLayout>
```

```
package co.gov.mintic.misiontic.udea.ciclo4

import ...

class FragmentA : Fragment() {
    private lateinit var mensaje: TextView
    private var cuenta = 0

    override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.layout_fragmento, container, attachToRoot: false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        view.findViewById<Button>(R.id.btnLimpilar).setOnClickListener { limpiar() }
        view.findViewById<Button>(R.id.btnCuenta).setOnClickListener { contar() }
        mensaje = view.findViewById(R.id.txtMensaje)
        actualizarLaVista()
    }

    private fun contar() { cuenta++
        actualizarLaVista()
    }

    private fun limpiar() { cuenta = 0
        actualizarLaVista()
    }

    private fun actualizarLaVista() { mensaje.text = cuenta.toString() }
}
```

```
<?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" tools:context=".MainActivity"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="ACTIVIDAD PRINCIPAL"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintRight_toRightOf="parent" app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintVertical_bias="0.061" />

    <fragment
        android:id="@+id/fragment1" android:name="co.gov.mintic.misiontic.udea.ciclo4.FragmentA"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_marginStart="33dp" android:layout_marginTop="34dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/btnAbrirAct2" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_marginBottom="72dp"
        android:onClick="abrir" android:text="Abrir otra Actividad"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?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" tools:context=".MainActivity2"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"    android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="ACTIVIDAD SECUNDARIA"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintRight_toRightOf="parent"  app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintVertical_bias="0.061" />

    <fragment
        android:id="@+id/fragment2"    android:name="co.gov.mintic.misiontic.udea.ciclo4.FragmentA"
        android:layout_width="wrap_content"    android:layout_height="wrap_content"
        android:layout_marginStart="33dp"    android:layout_marginTop="34dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/btnAbrirAct1"    android:layout_width="wrap_content"
        android:layout_height="wrap_content"    android:layout_marginBottom="72dp"
        android:onClick="abrir"    android:text="Abrir otra Actividad"
        app:layout_constraintBottom_toBottomOf="parent"    app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
package co.gov.mintic.misiontic.udea.ciclo4

import ...

class MainActivity2 : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
    }

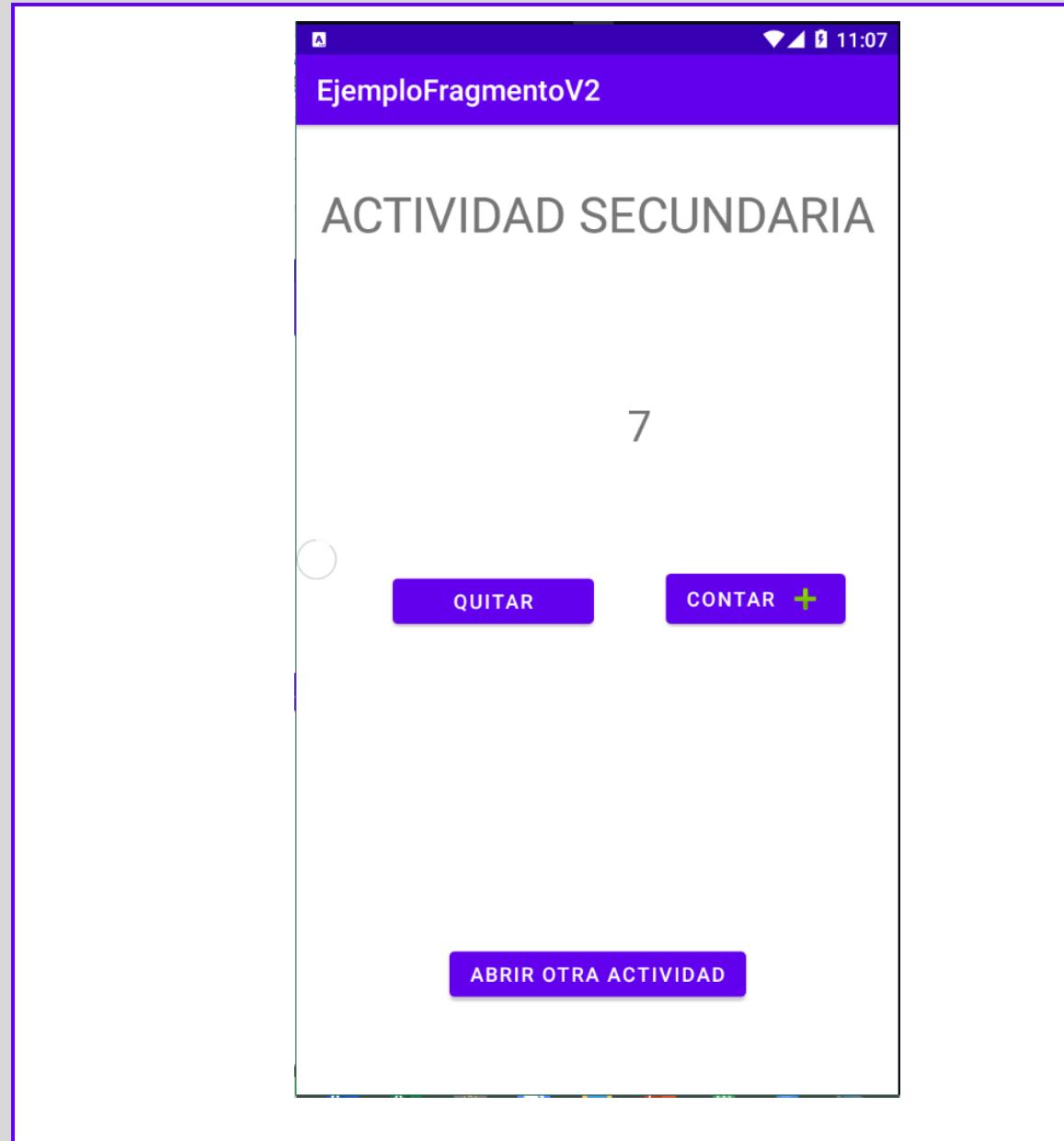
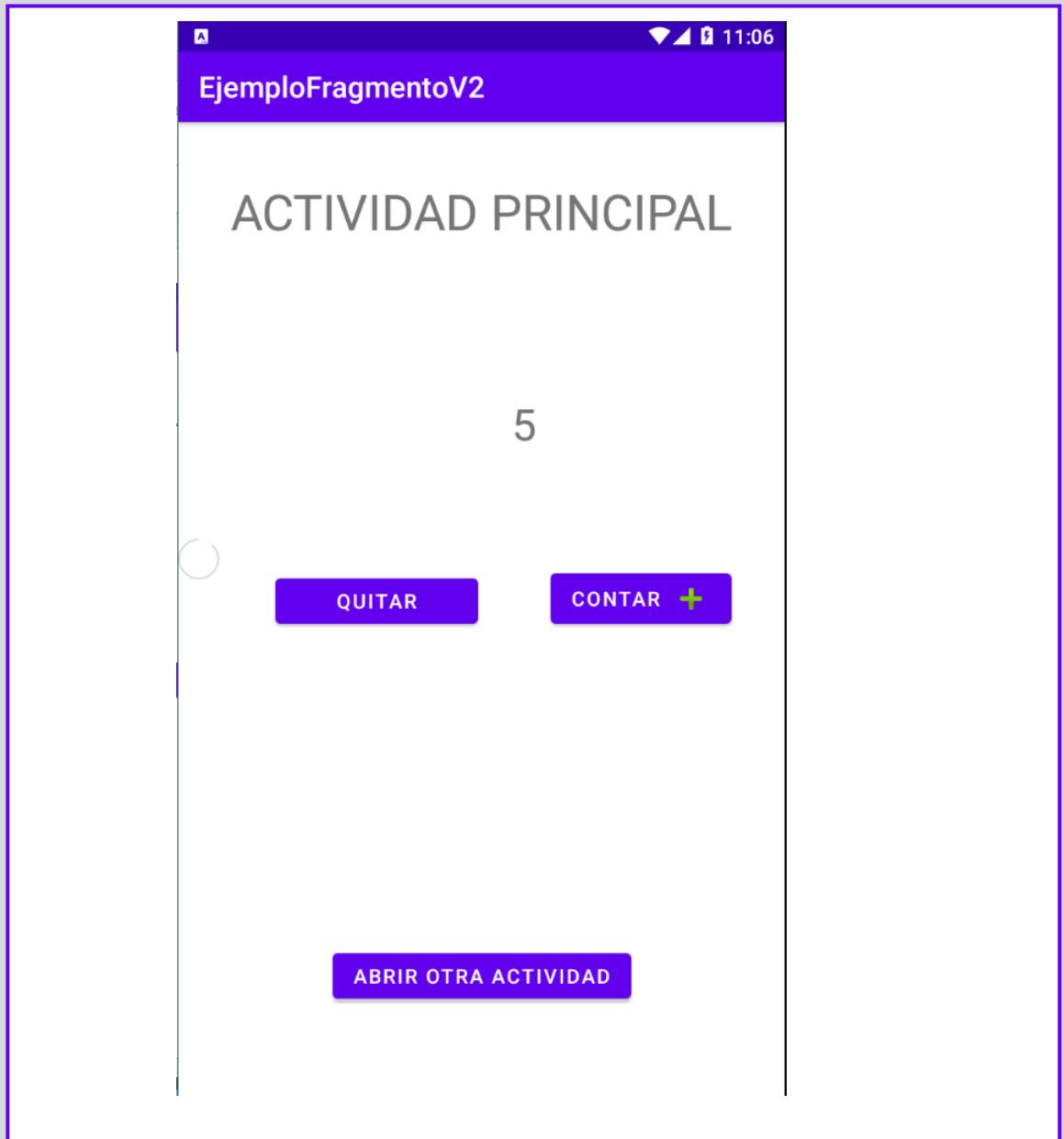
    fun abrir(v : View){
        val intento = Intent( packageContext: this, MainActivity::class.java)
        startActivity(intento)
    }
}
```

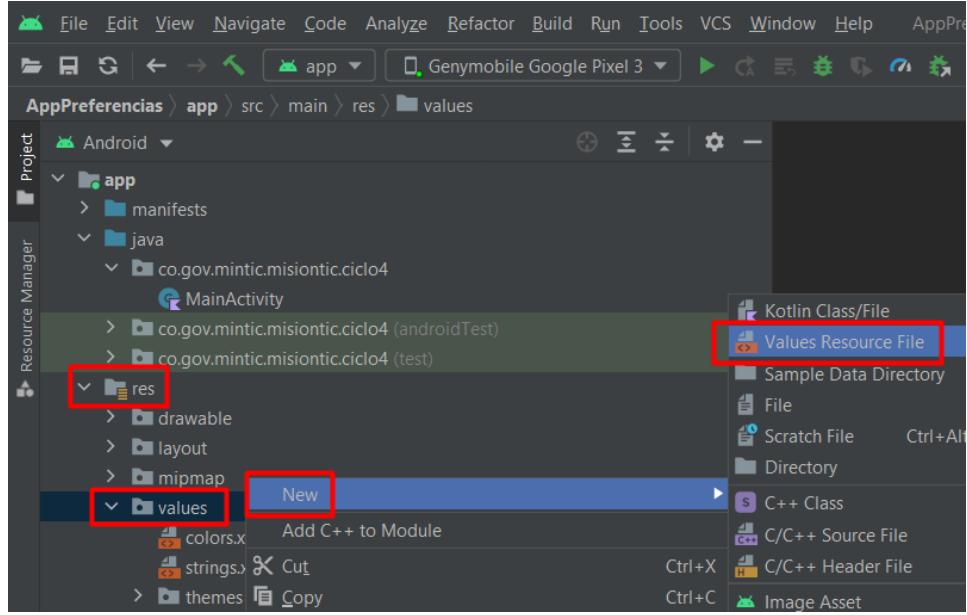
```
package co.gov.mintic.misiontic.udea.ciclo4

import ...

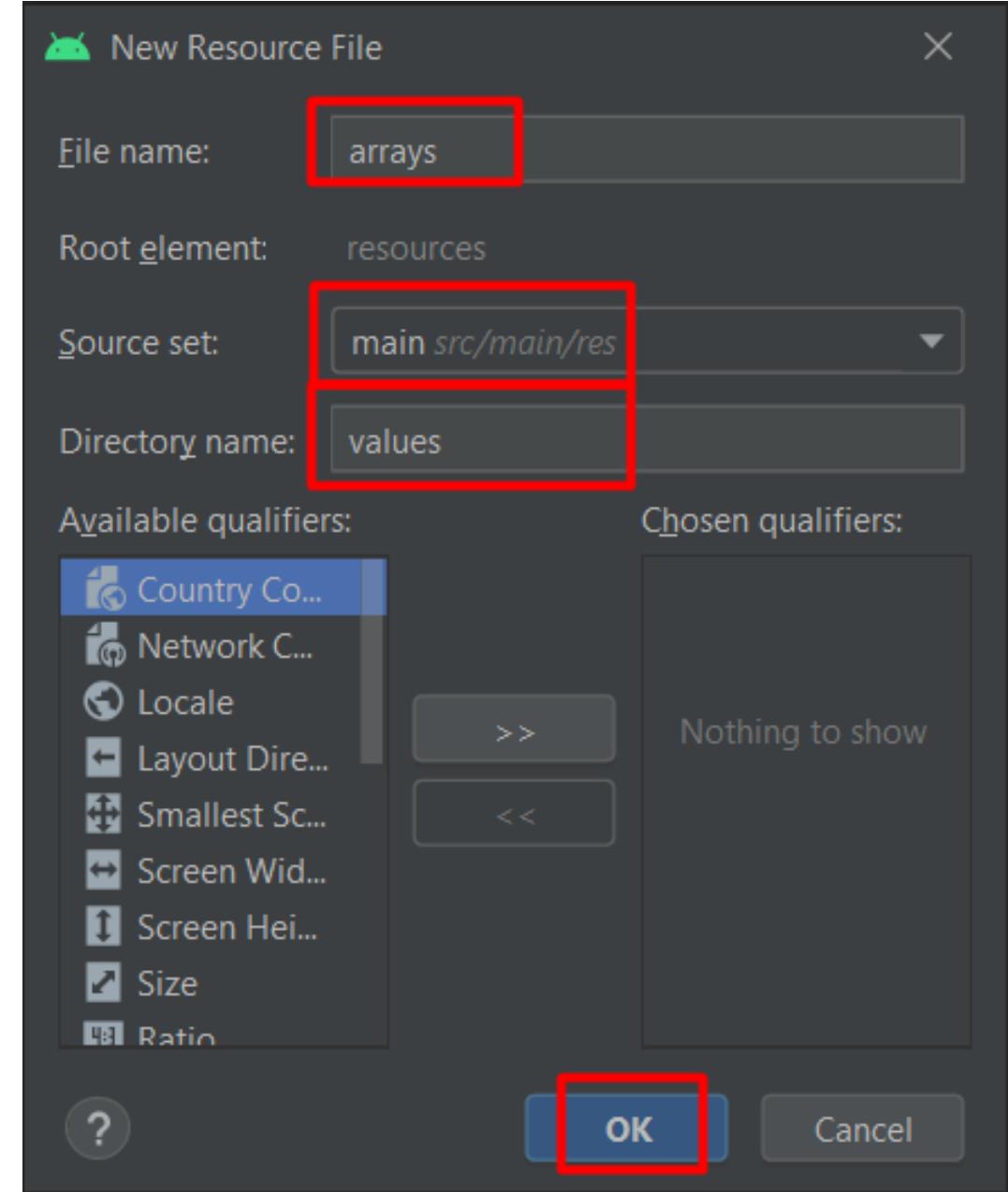
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun abrir(v : View){
        val intento = Intent( packageContext: this, MainActivity2::class.java)
        startActivity(intento)
    }
}
```

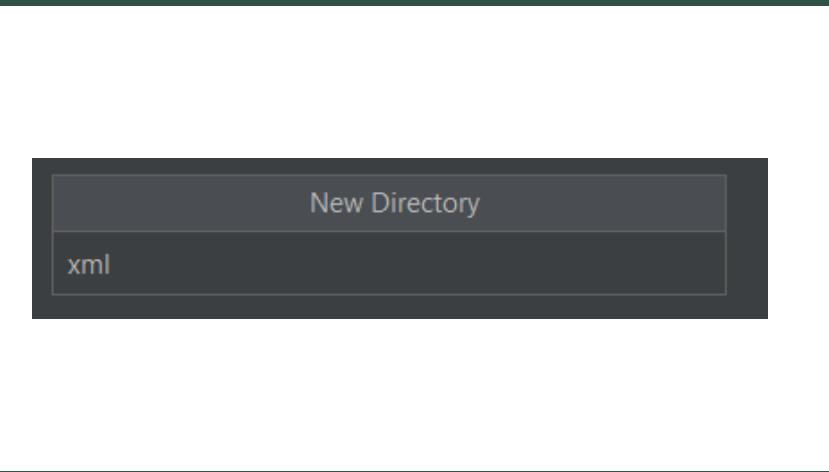
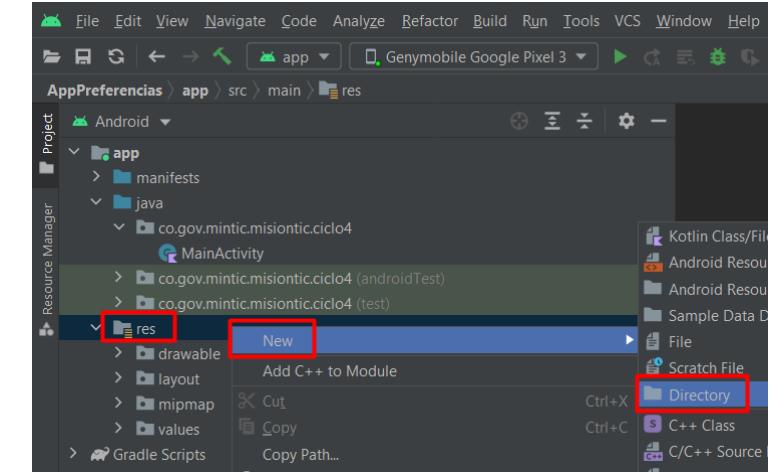


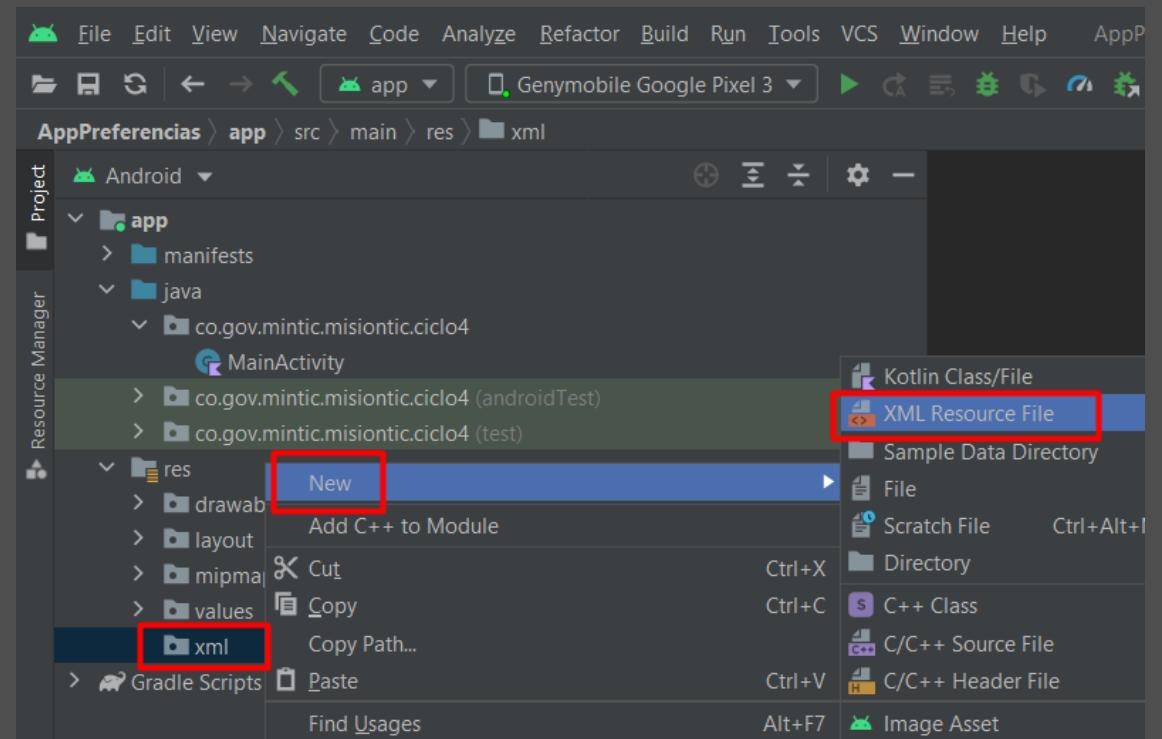
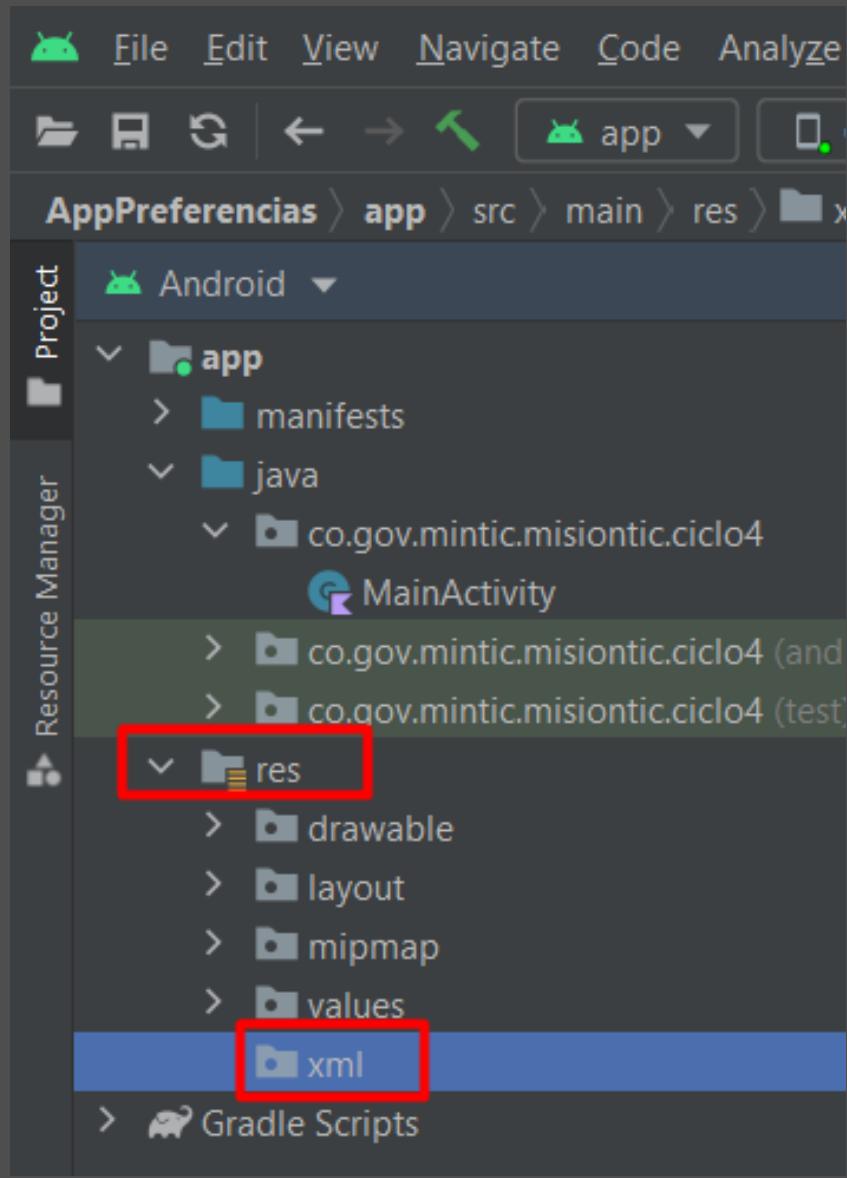


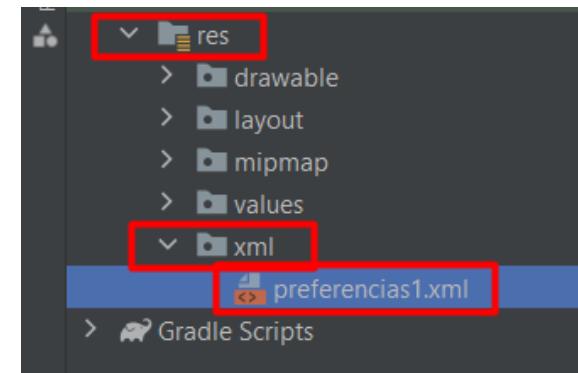
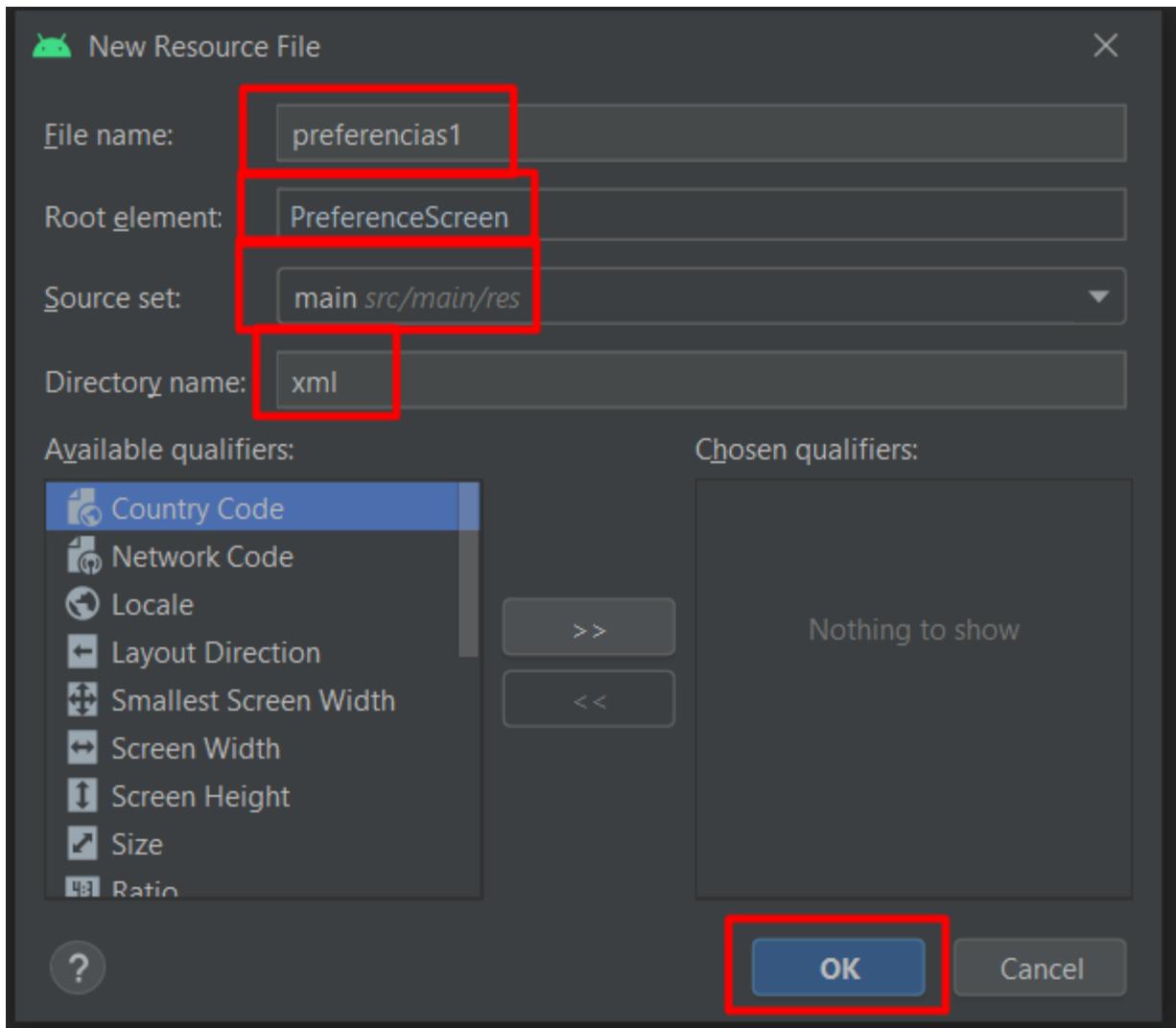
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
</resources>
```



```
arrays.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <array name="paises">
4          <item>Colombia</item>
5          <item>Venezuela</item>
6          <item>Mexico</item>
7          <item>Peru</item>
8          <item>Ecuador</item>
9          <item>Argentina</item>
10         <item>Chile</item>
11         <item>Brasil</item>
12         <item>Bolivia</item>
13         <item>Salvador</item>
14         <item>Panama</item>
15         <item>Cuba</item>
16     </array>
17 </resources>
```



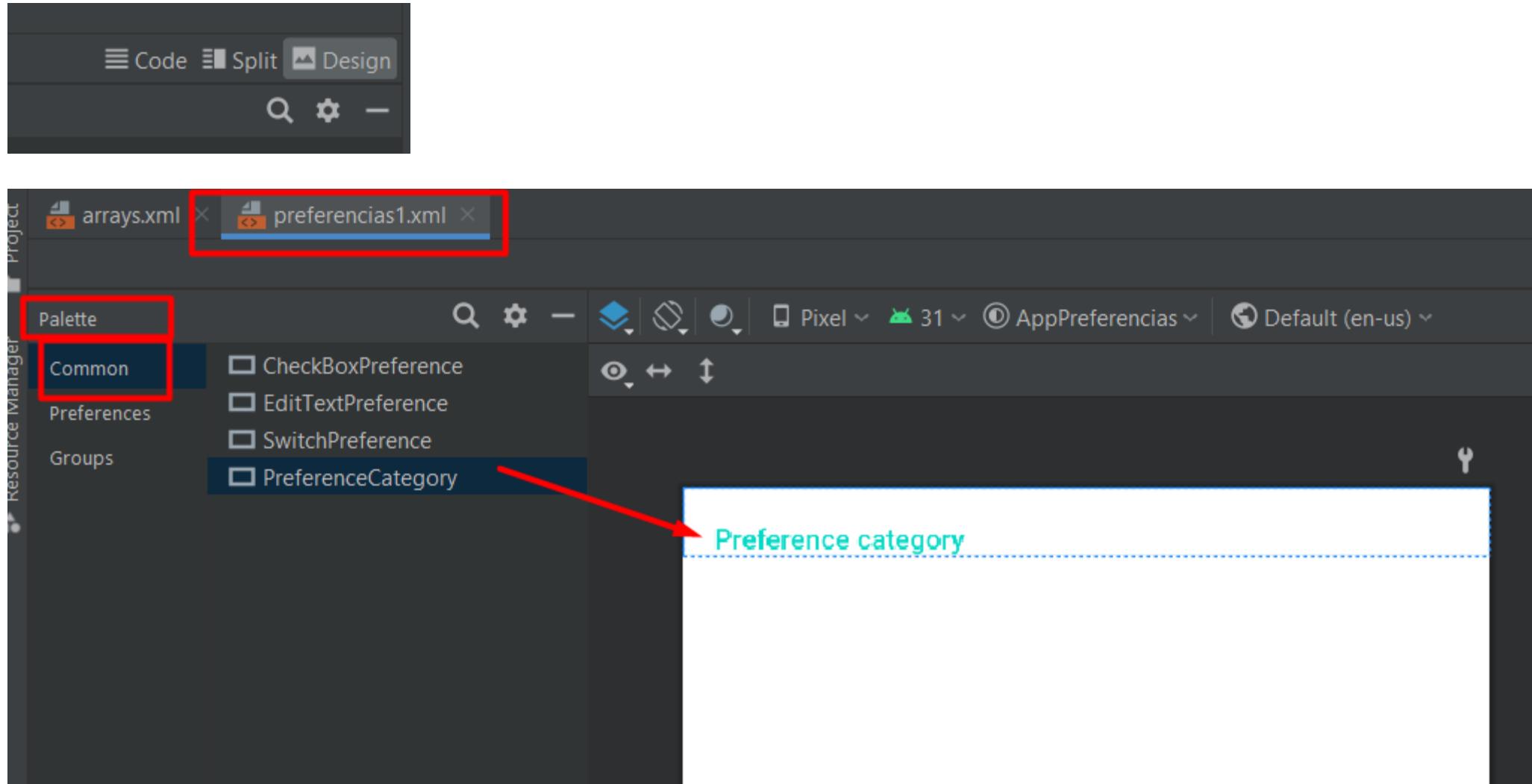


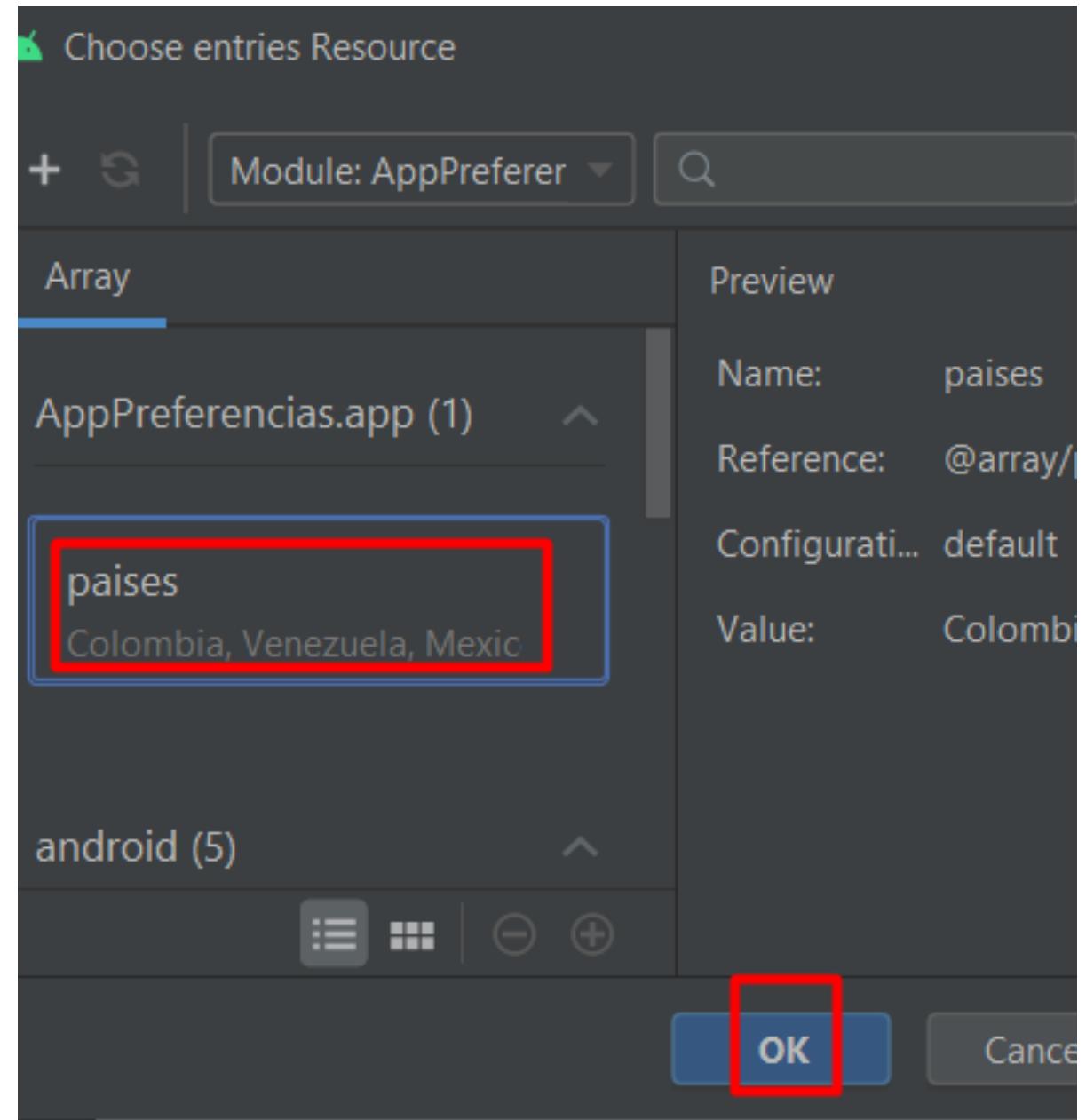
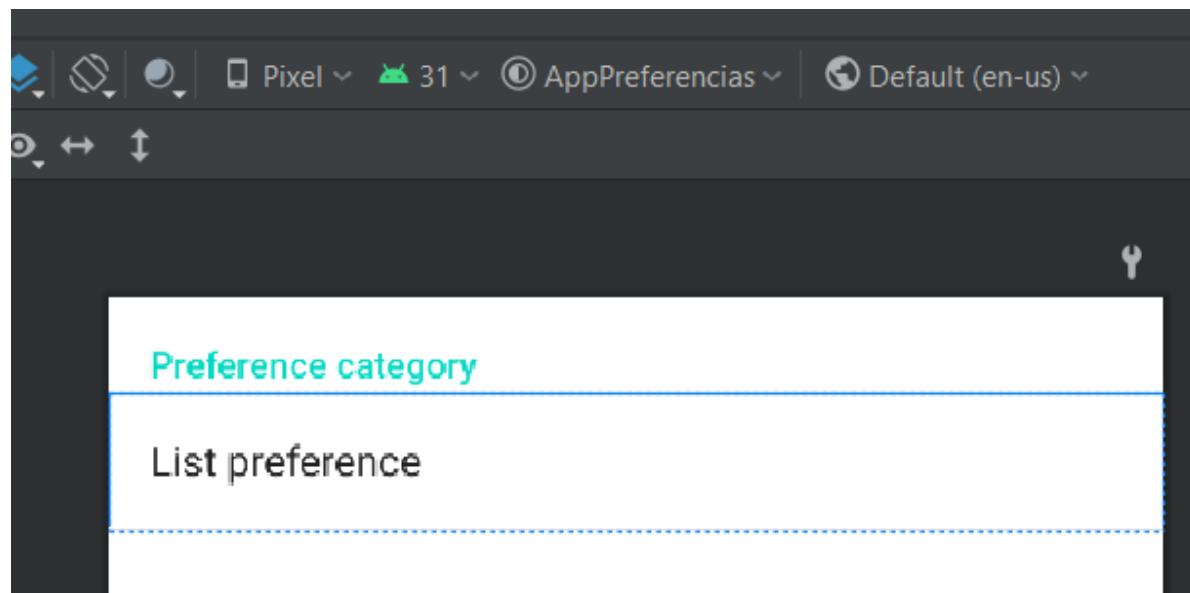
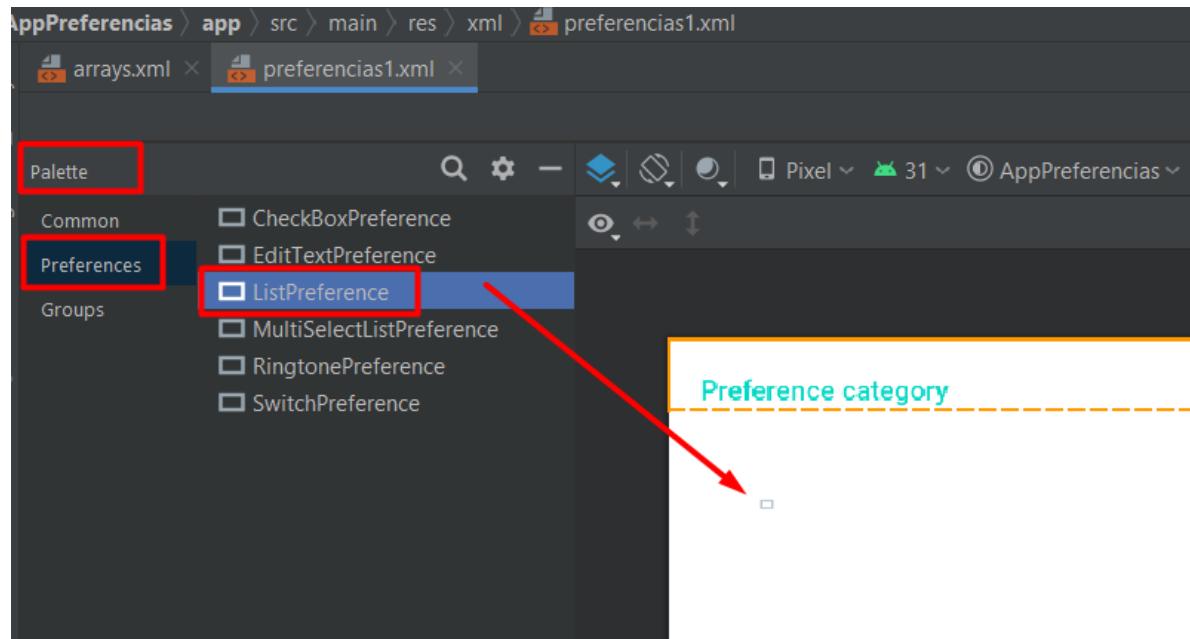


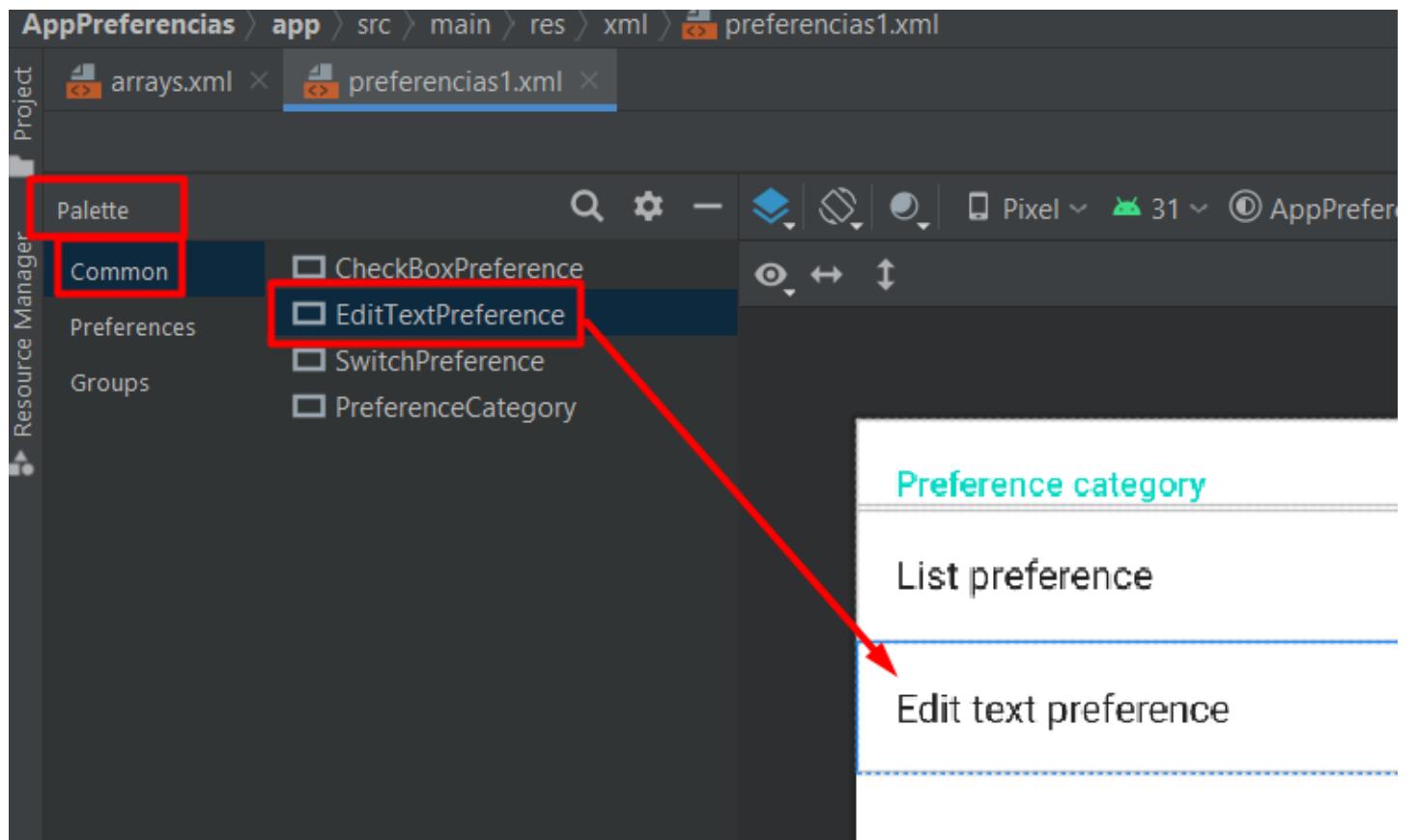
A screenshot of the XML editor showing the generated XML code for the PreferenceScreen.

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

The tab bar at the top shows 'arrays.xml' and 'preferencias1.xml', with 'preferencias1.xml' currently selected.

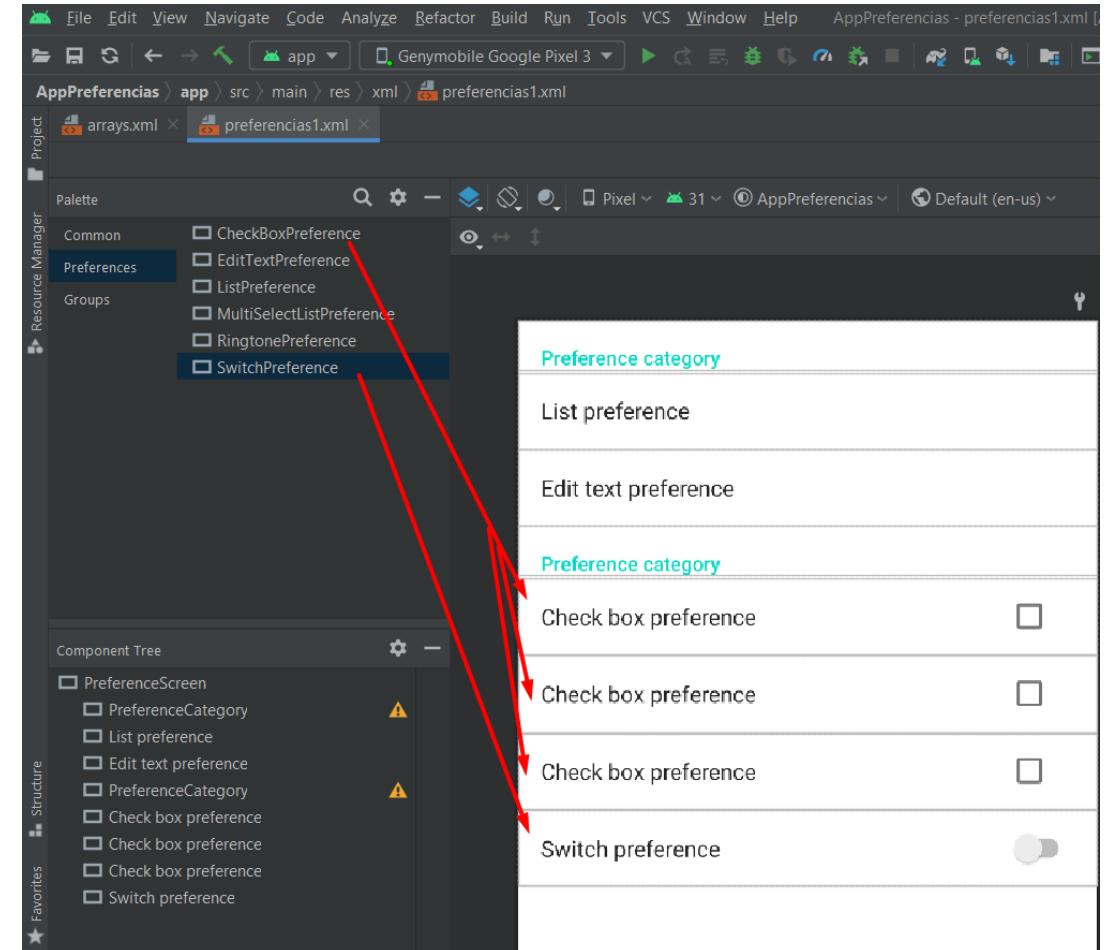


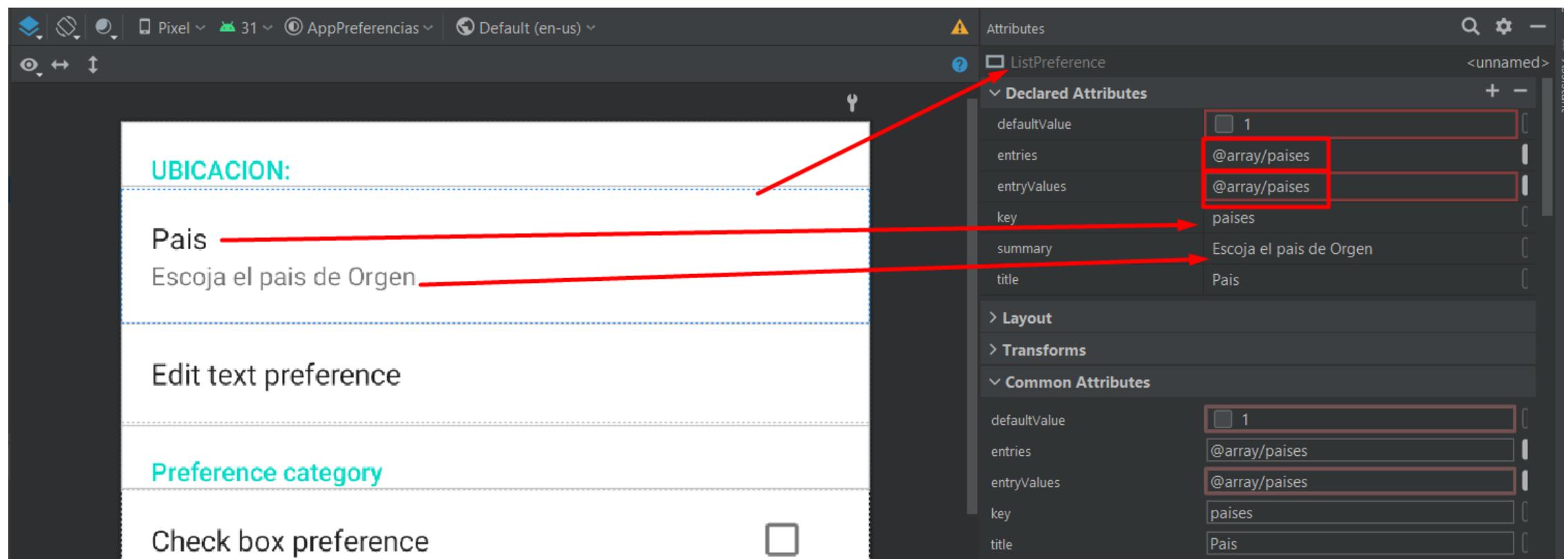
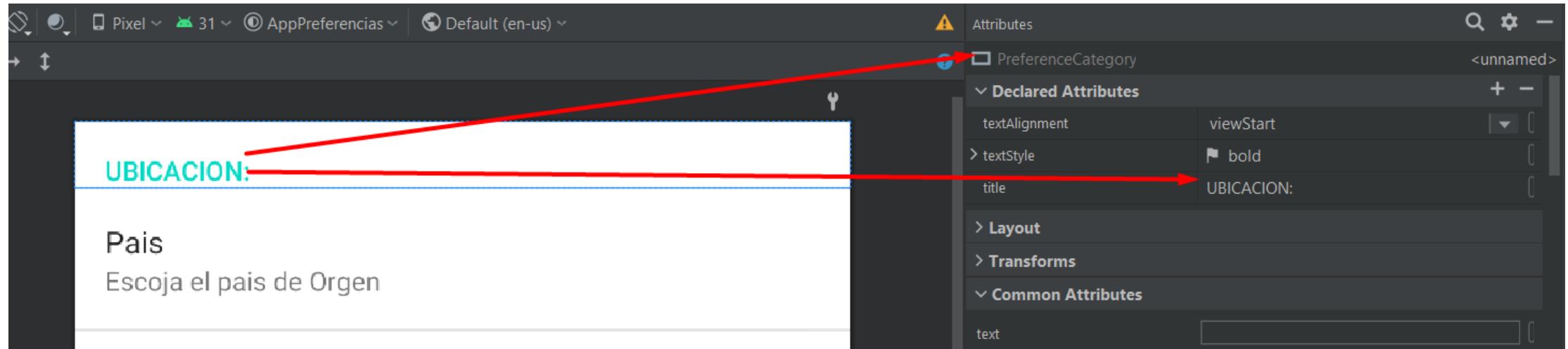


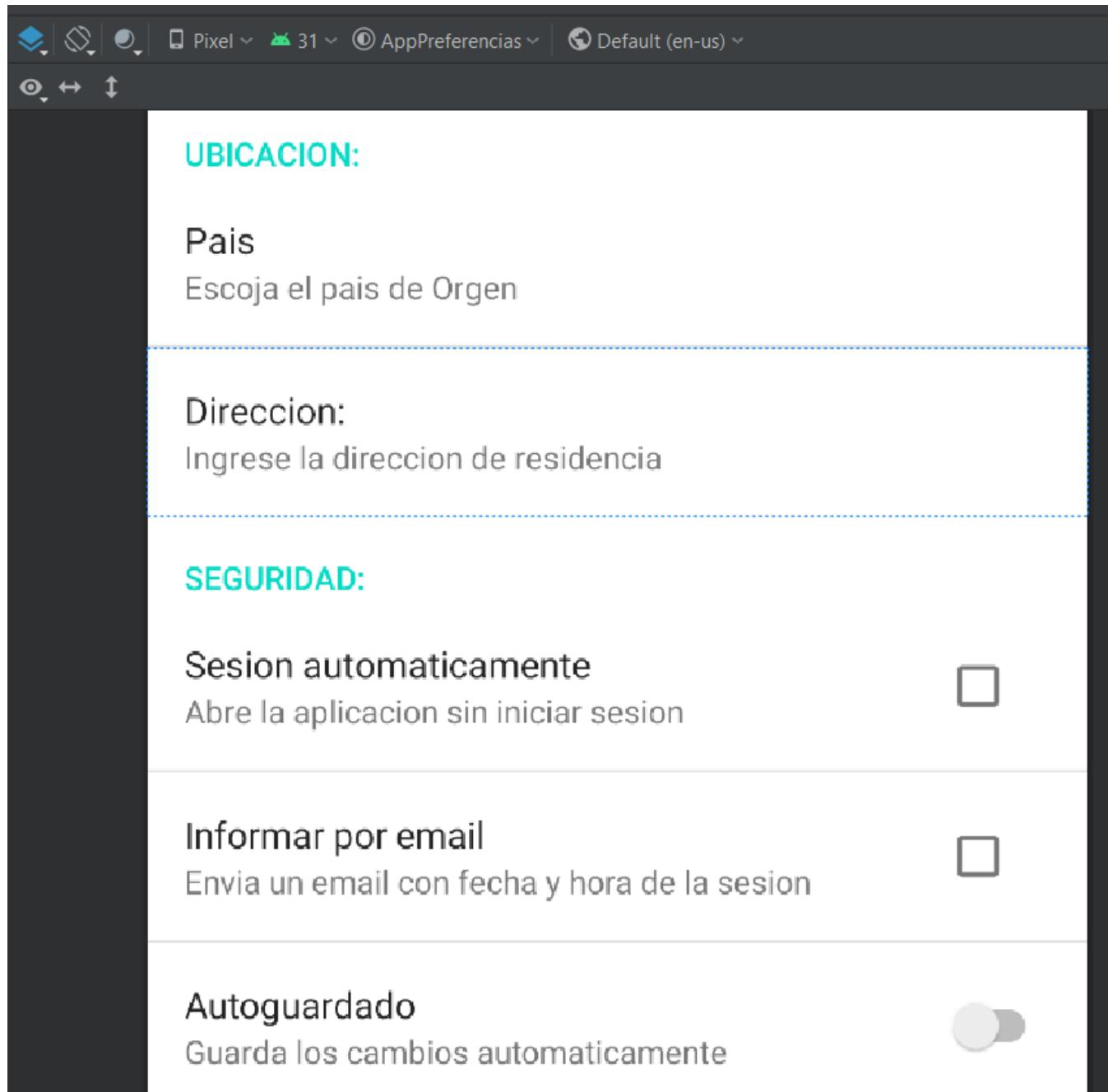




```
3     <PreferenceCategory android:title="Preference category">
4         <ListPreference
5             android:defaultValue="1"
6             android:entries="@array/paises"
7             android:entryValues="@array/paises"
8             android:key="list_preference_1"
9             android:title="List preference" />
10        <EditTextPreference
11            android:defaultValue="Default value"
12            android:key="edit_text_preference_1"
13            android:selectAllOnFocus="true"
14            android:singleLine="true"
15            android:title="Edit text preference" />
16    </PreferenceCategory>
17    <PreferenceCategory android:title="Preference category">
18        <CheckBoxPreference
19            android:defaultValue="false"
20            android:key="check_box_preference_1"
21            android:title="Check box preference" />
22        <CheckBoxPreference
23            android:defaultValue="false"
24            android:key="check_box_preference_2"
25            android:title="Check box preference" />
26        <CheckBoxPreference
27            android:defaultValue="false"
28            android:key="check_box_preference_3"
29            android:title="Check box preference" />
30        <SwitchPreference
31            android:defaultValue="false"
32            android:key="switch_preference_1"
33            android:title="Switch preference" />
34    </PreferenceCategory>
```

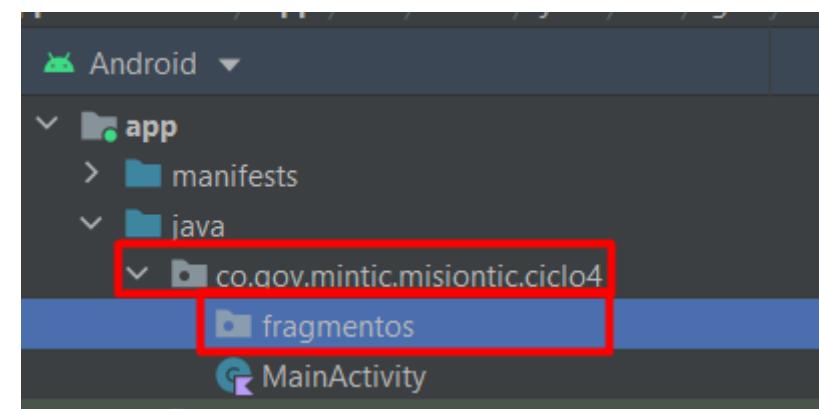
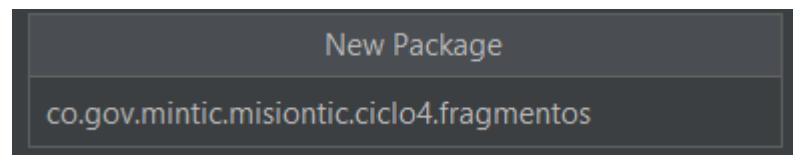
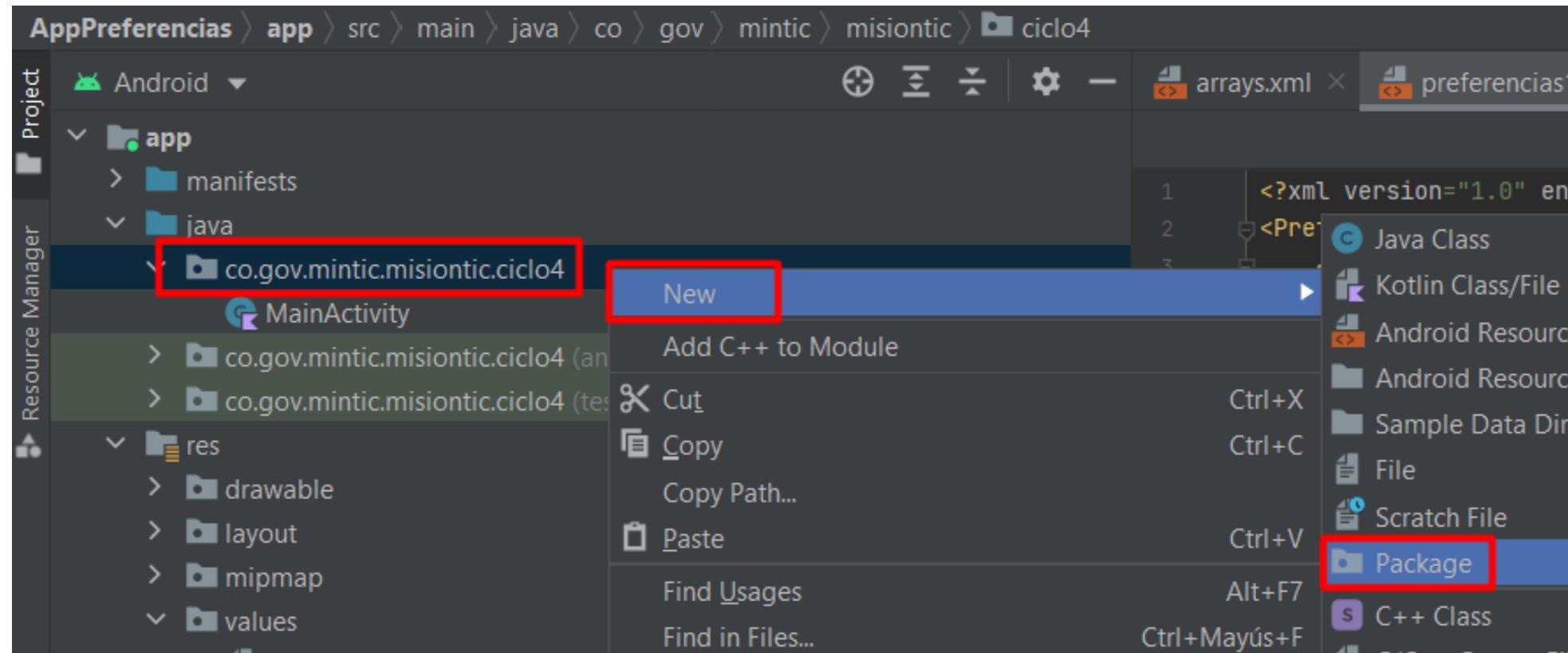


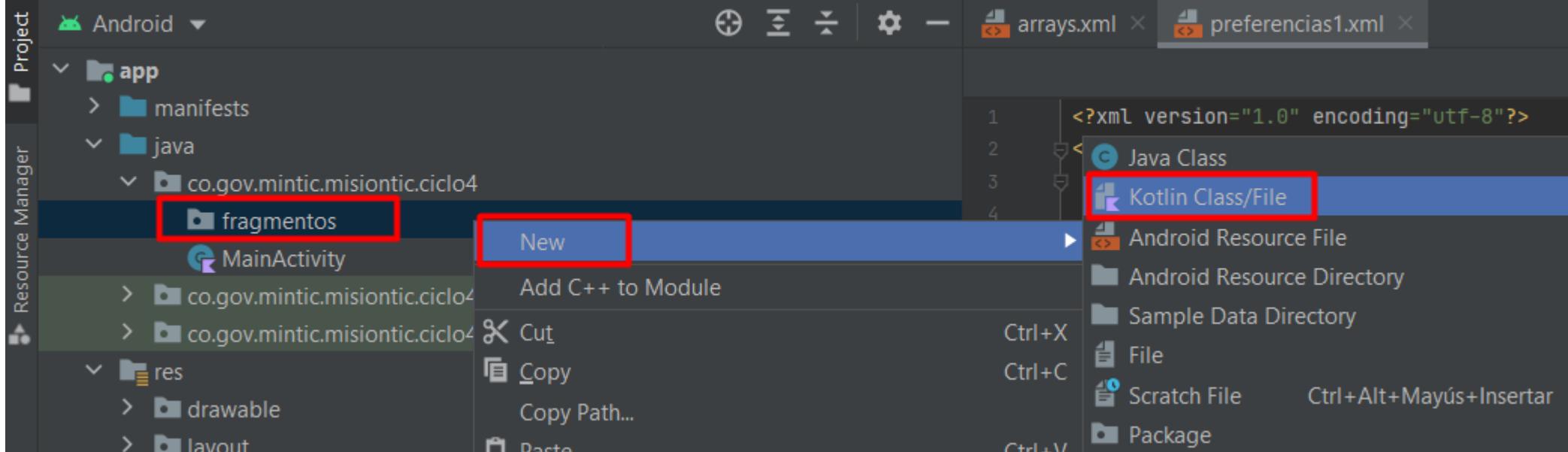




```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">

    <PreferenceCategory
        android:textAlignment="viewStart"
        android:textStyle="bold"
        android:title="UBICACION:">
        <ListPreference
            android:defaultValue="1"
            android:entries="@array/paises"
            android:entryValues="@array/paises"
            android:key="paises"
            android:summary="Escoga el pais de Origen"
            android:title="Pais" />
        <EditTextPreference
            android:defaultValue="Default value"
            android:key="direccion"
            android:selectAllOnFocus="true"
            android:singleLine="true"
            android:summary="Ingrrese la direccion de residencia"
            android:title="Direccion:" />
    </PreferenceCategory>
    <PreferenceCategory android:title="SEGURIDAD:">
        <CheckBoxPreference
            android:defaultValue="false"
            android:key="sesion_auto"
            android:summary="Abre la aplicacion sin iniciar sesion"
            android:title="Sesion automaticamente" />
        <CheckBoxPreference
            android:defaultValue="false"
            android:key="notificar_email"
            android:summary="Envia un email con fecha y hora de la sesion"
            android:title="Informar por email" />
        <SwitchPreference
            android:defaultValue="false"
            android:key="guardar_cambios"
            android:summary="Guarda los cambios automaticamente"
            android:title="Autoguardado" />
    </PreferenceCategory>
</PreferenceScreen>
```

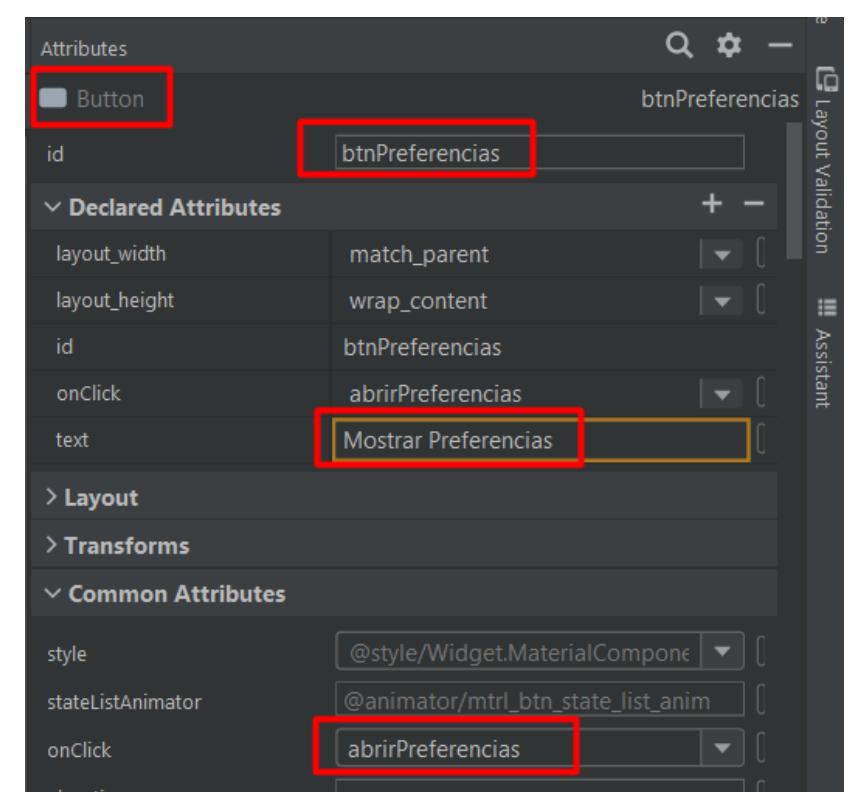
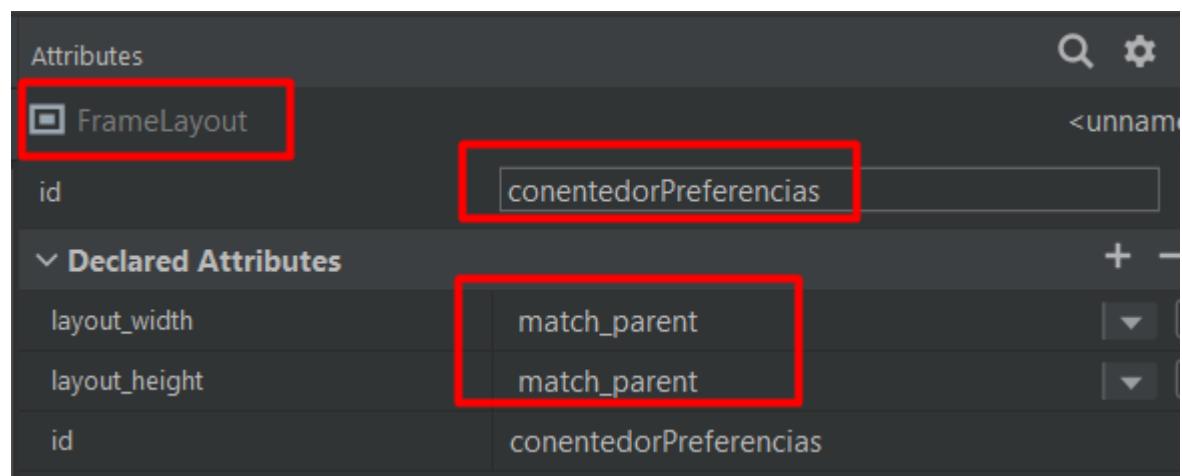
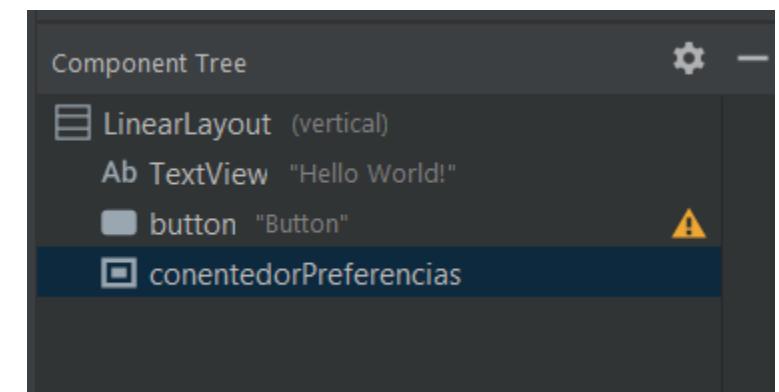
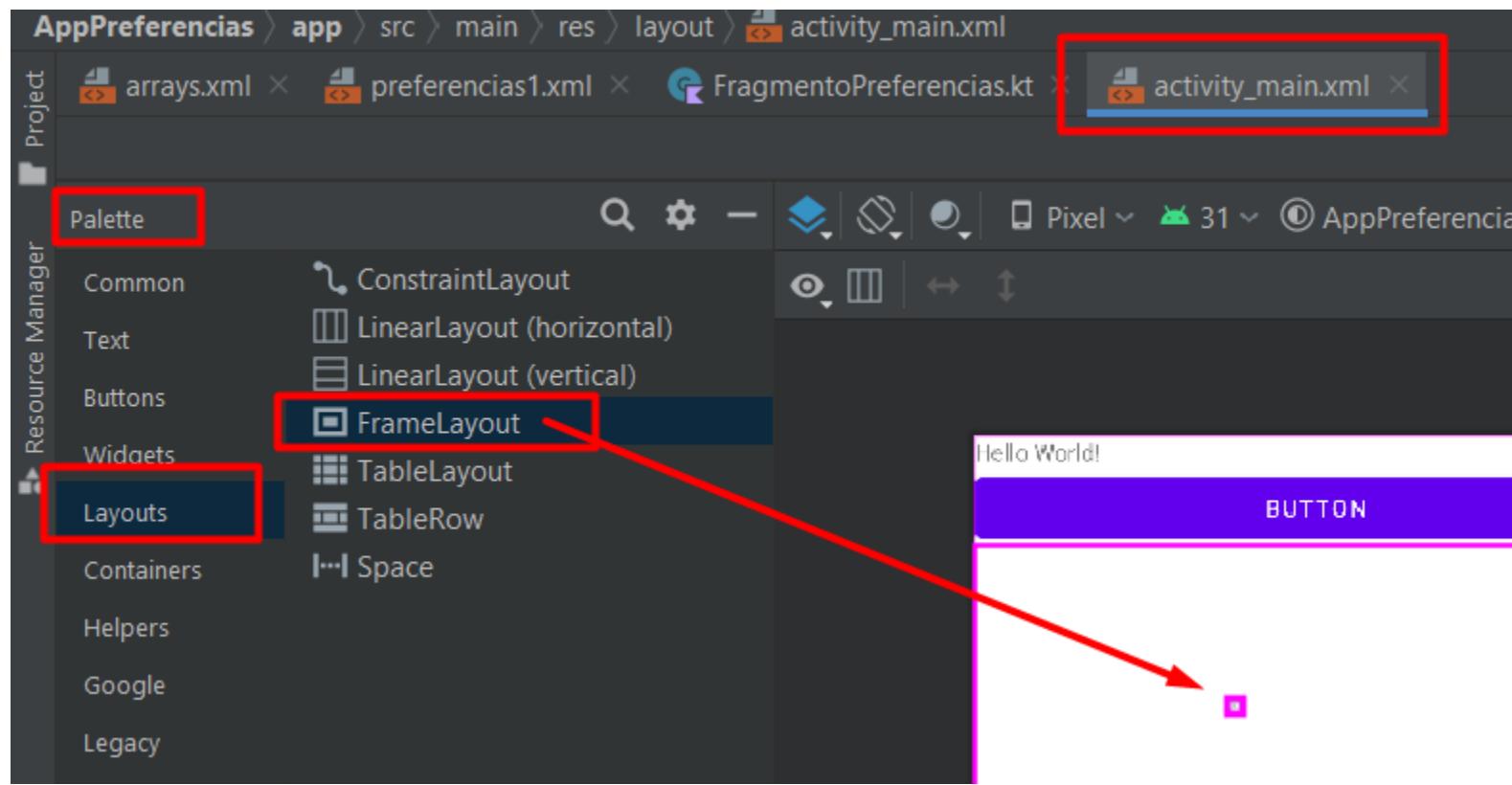




The screenshot shows the 'New Kotlin Class/File' dialog on the left, with 'Class' selected. On the right, the generated code for 'FragmentoPreferencias.kt' is displayed:

```
package co.gov.mintic.misiontic.ciclo4.fragmentos
import android.os.Bundle
import androidx.preference.PreferenceFragmentCompat
import co.gov.mintic.misiontic.ciclo4.R

class FragmentoPreferencias : PreferenceFragmentCompat() {
    override fun onCreatePreferences(savedInstanceState: Bundle?, clave: String?) {
        setPreferencesFromResource(R.xml.preferencias1, clave)
    }
}
```



The screenshot shows the Android Studio code editor with the file `MainActivity.java` open. A red box highlights the code block that handles button logic. The code uses a FragmentManager to add or remove a `FragmentoDePreferencias` fragment based on the button's hint value ('ver' or 'ocultar').

```
activity_main.xml x MainActivity.java x arrays.xml x preferencias1.xml x FragmentoDePreferencias.java  
15  
16     @Override  
17     protected void onCreate(Bundle savedInstanceState) {  
18         super.onCreate(savedInstanceState);  
19         setContentView(R.layout.activity_main);  
20     }  
21  
22     FragmentoDePreferencias fp = new FragmentoDePreferencias();  
23     public void mostarPreferencias(View v){  
24         Button b = (Button)v;  
25         if(b.getHint().toString().equals("ver")){  
26             b.setHint("ocultar");  
27             b.setText("OCULTAR PREFERENCIAS");  
28             FragmentManager mf = getSupportFragmentManager();  
29             FragmentTransaction ft = mf.beginTransaction();  
30             ft.add(R.id.contenedorDePreferencias,fp);  
31             ft.commit();  
32         }  
33         else{  
34             b.setHint("ver");  
35             b.setText("VER PREFERENCIAS");  
36             FragmentManager mf = getSupportFragmentManager();  
37             FragmentTransaction ft = mf.beginTransaction();  
38             ft.remove(fp);  
39             ft.commit();  
40         }  
41     }  
42 }  
43 }
```

## EjemploAppConPreferenciasG45Usa

### ACTIVIDAD CON PREFERENCIAS

VER PREFERENCIAS

## EjemploAppConPreferenciasG45Usa

### ACTIVIDAD CON PREFERENCIAS

OCULTAR PREFERENCIAS

#### UBICACIÓN:

Paises

Seleccione un pais

#### Direccion

Ingrese su direccion

#### SEGURIDAD

SESION AUTOMATICO

Iniciar sesion de forma automatica

#### EMAIL

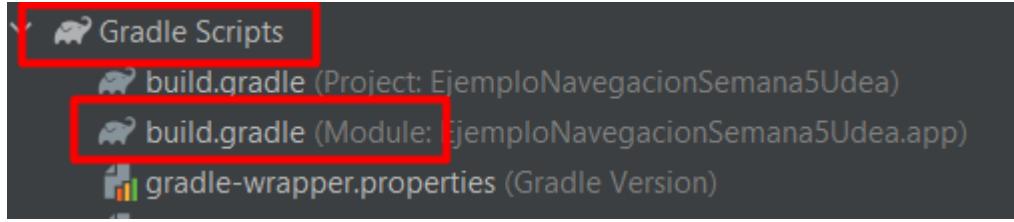
Ingrresa tu email

Informar por email

Enviar un email por cada sesion

# NAVIGATION

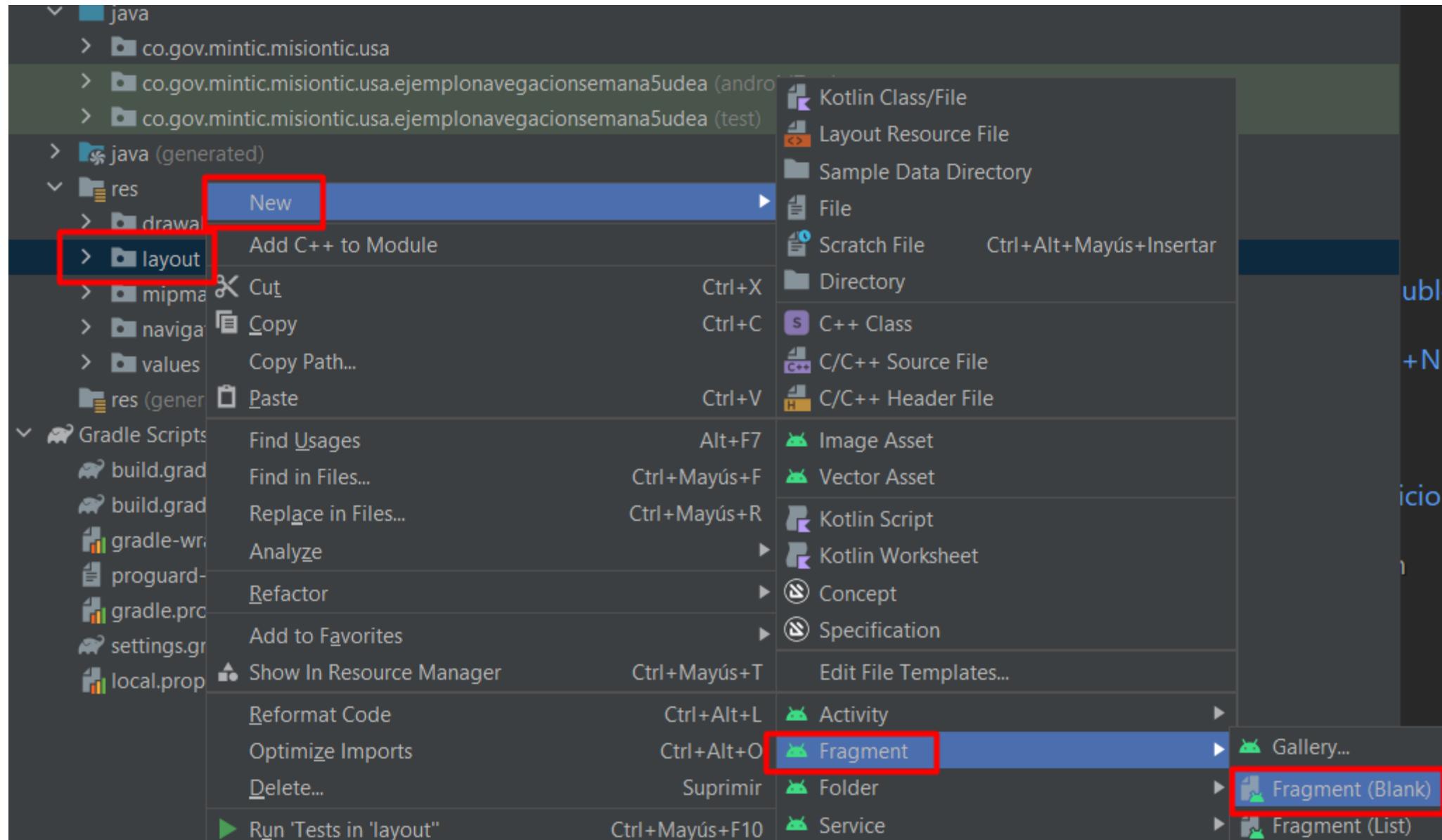
1. **Crear Un proyecto (AS 2x, SDK 19)**
2. **Agregar las librerías Gradle (Modulo)**
3. Sincronizare Gradle
4. Crear y diseñar una Actividad Principal
5. Crear y diseñar los Fragmentos
6. Crear un Grafico de Navegación
7. Agregar al grafico de navegación, ActividadPrincial
8. Agregar los fragmentos al grafico de navegación
9. Conectar los fragmentos desde el grafico según nuestro mapa de navegación.
10. Programar los evento

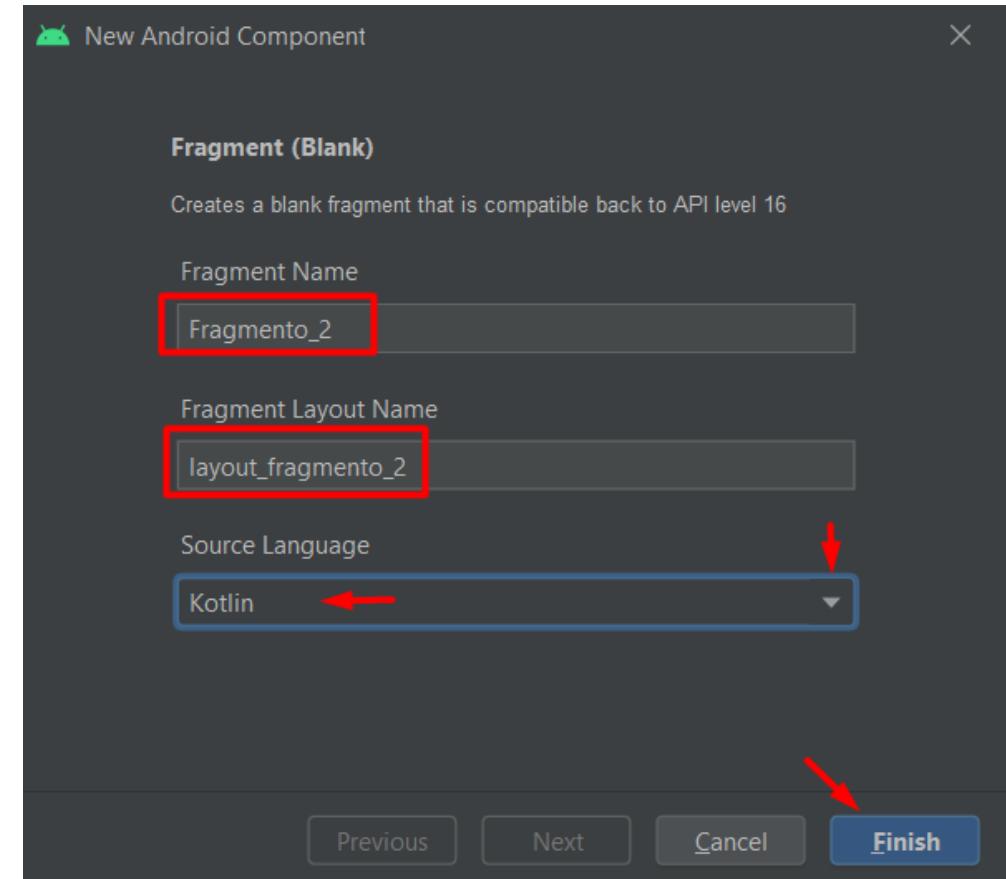
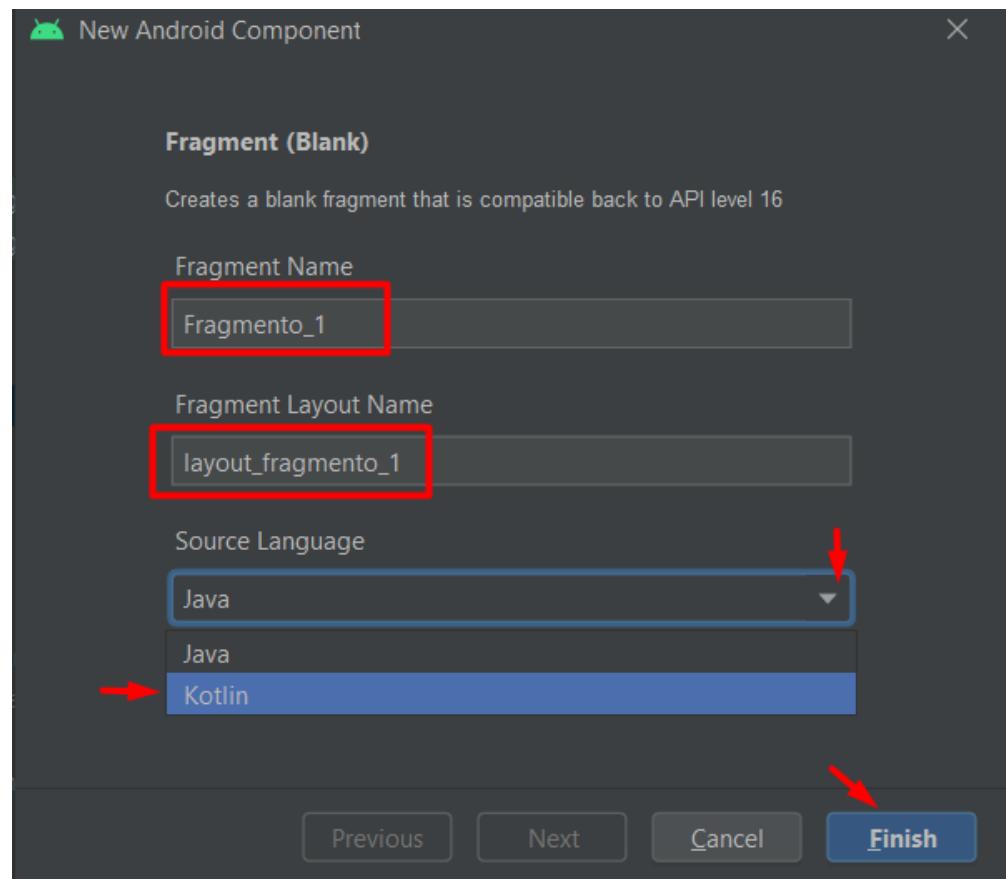


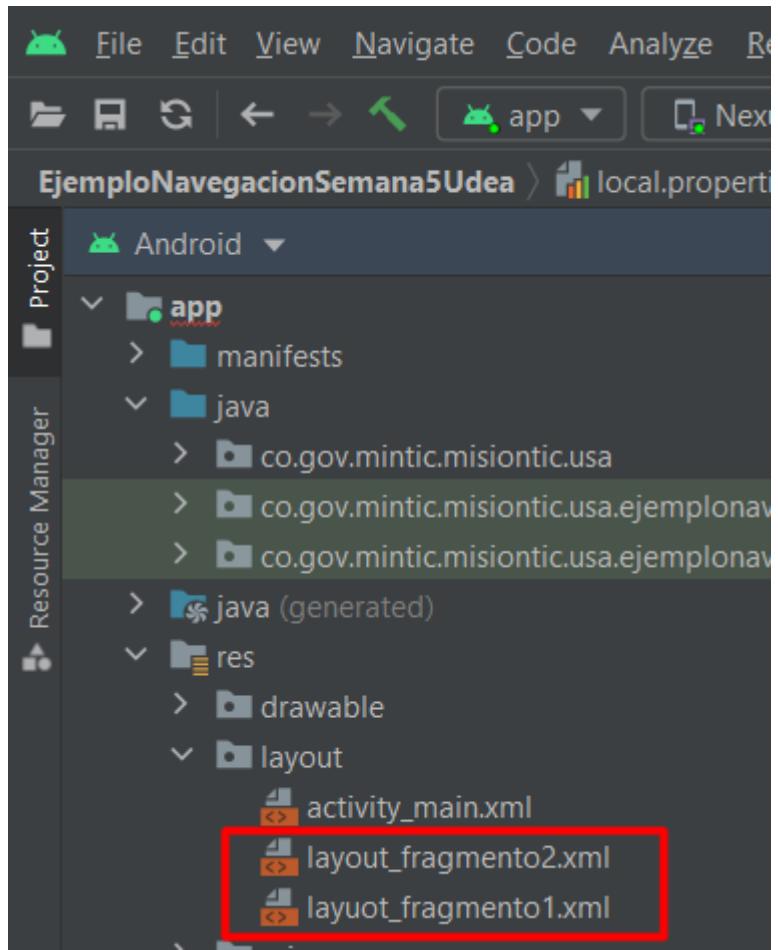
```
dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.4.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.2'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    /* para navegacion */
    def nav_version = "2.3.0-alpha01"
    implementation "androidx.navigation:navigation-fragment-ktx:$nav_version"
    implementation "androidx.navigation:navigation-ui-ktx:$nav_version"
    /* para navegacion */
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

}
```





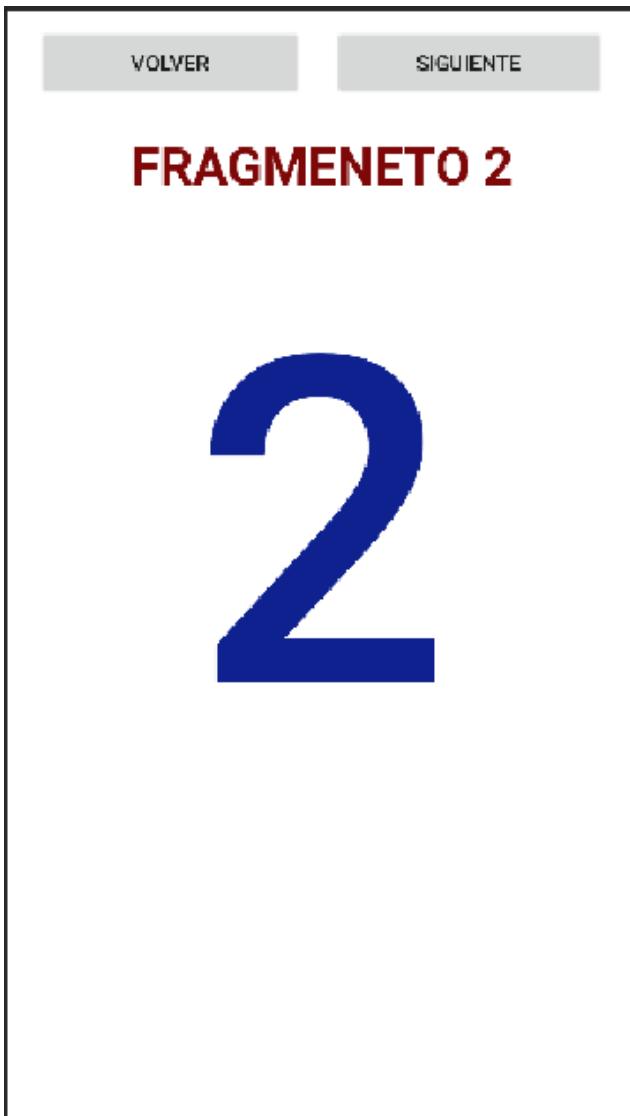
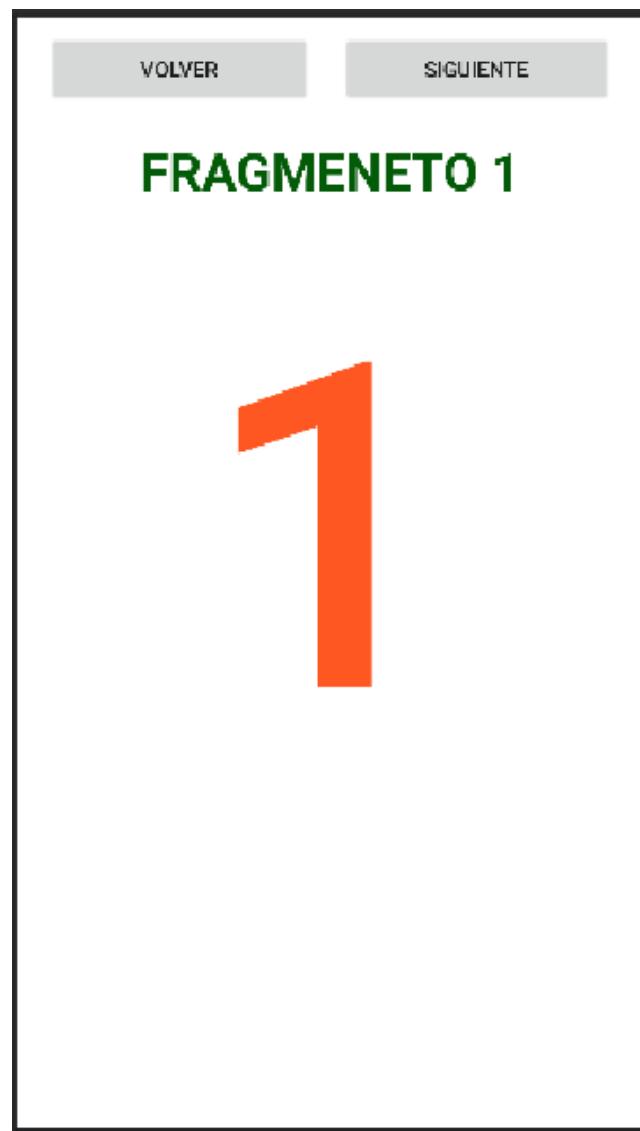


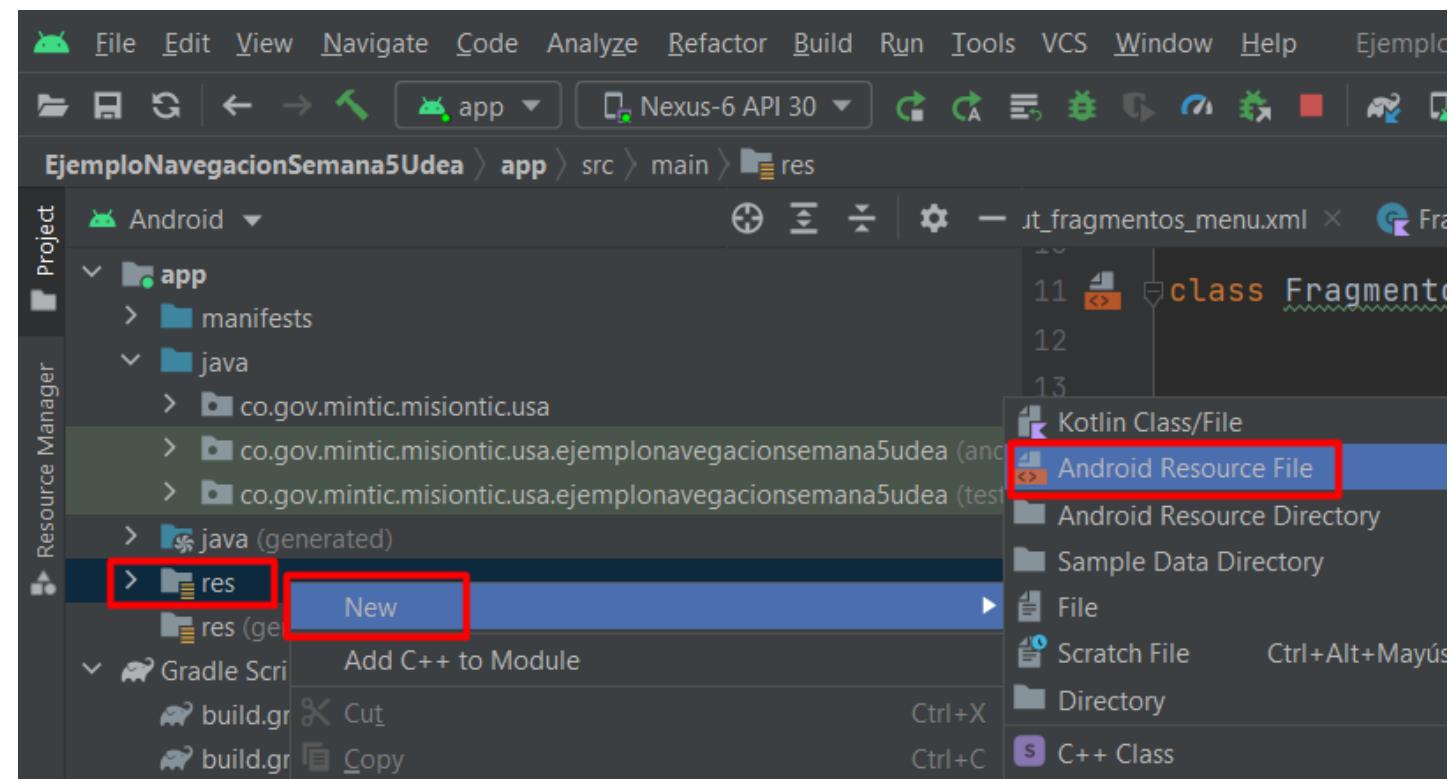
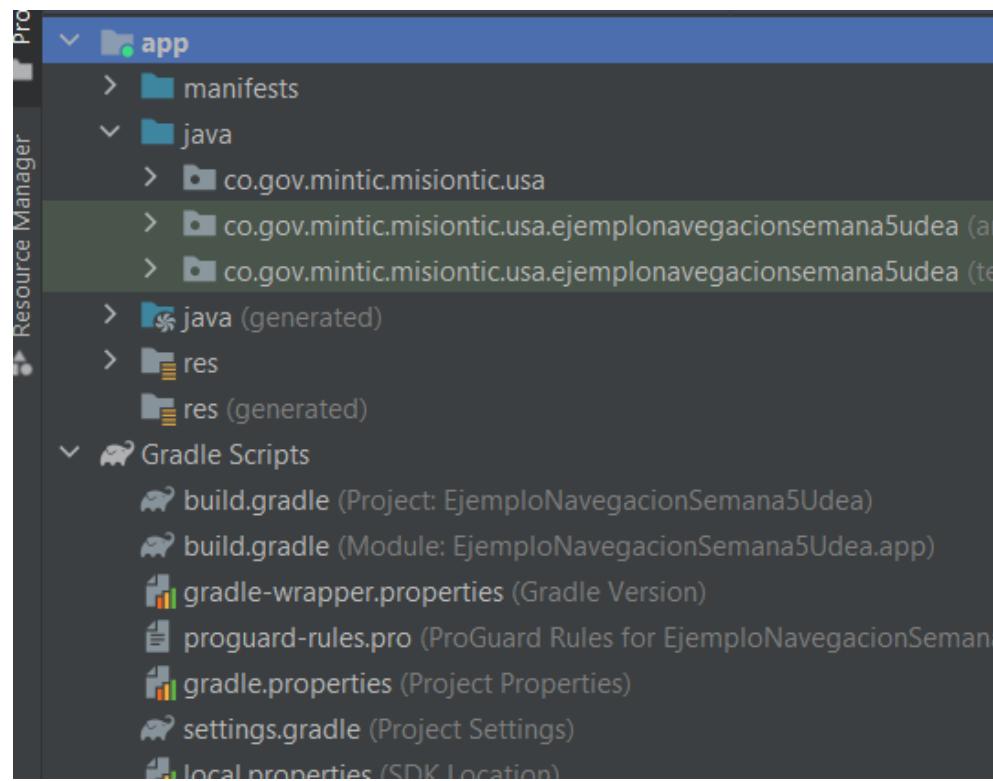
Component Tree

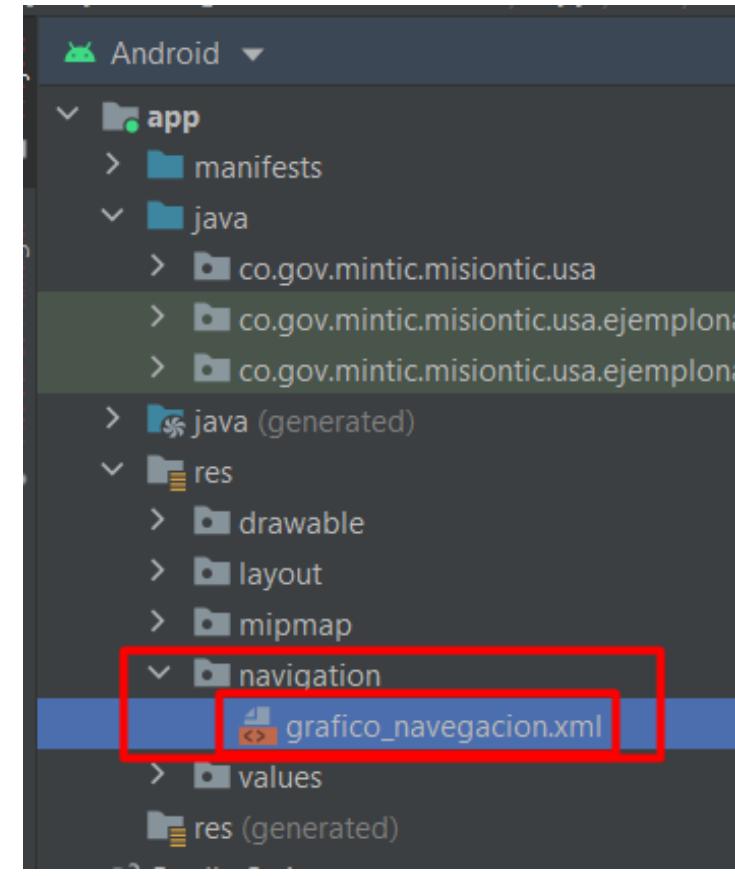
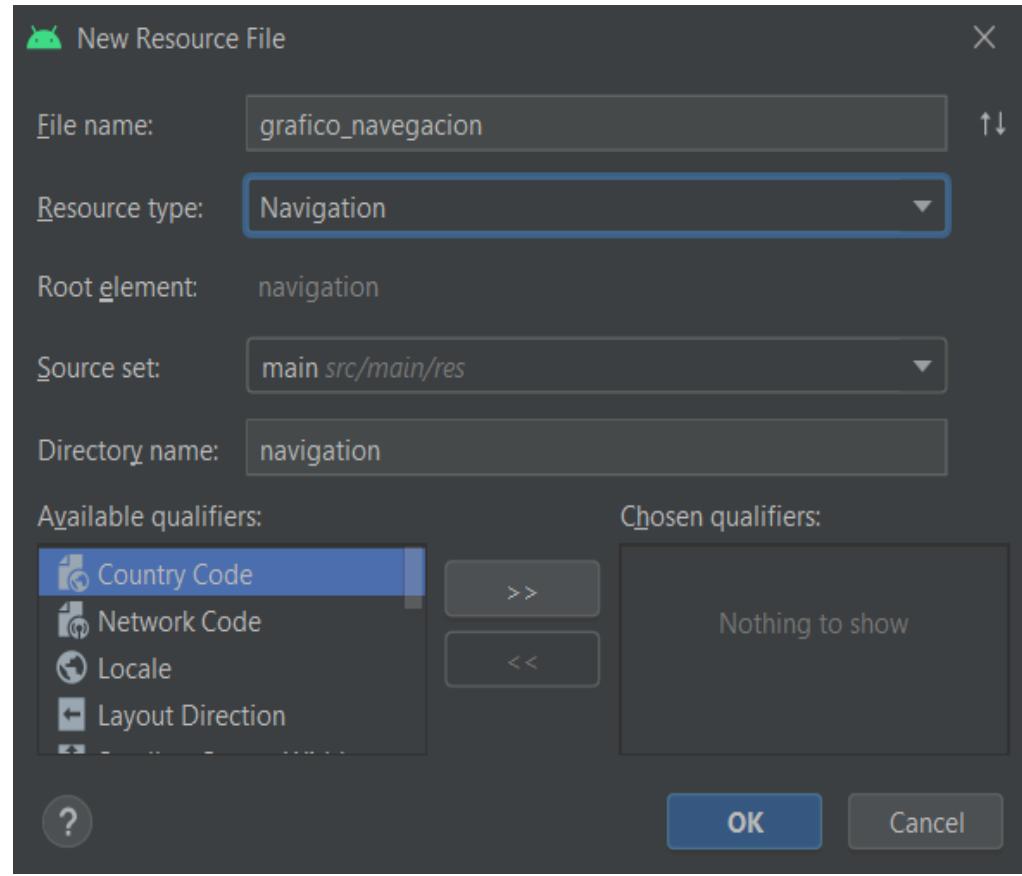
- LinearLayout (vertical)
  - LinearLayout (horizontal)
    - btnAtras1 "Volver"
    - btnSiguiente1 "Siguien..."
- Ab textView "@string/titulo\_fra..."
- Ab textView2 "1"

Component Tree

- LinearLayout (vertical)
  - LinearLayout (horizontal)
    - btnAtras2 "Volver"
    - btnSiguiente2 "Siguien..."
- Ab textView "@string/titulo\_fra..."
- Ab textView2 "2"









This operation requires the libraries androidx.navigation:navigation-fragment:+, androidx.navigation:navigation-ui:+.

Problem: Inconsistencies in the existing project dependencies found.

Version incompatibility between:

- com.google.android.material:material:1.4.0@aar
- and:
- com.google.android.material:material:1.4.0@aar

With the dependency:

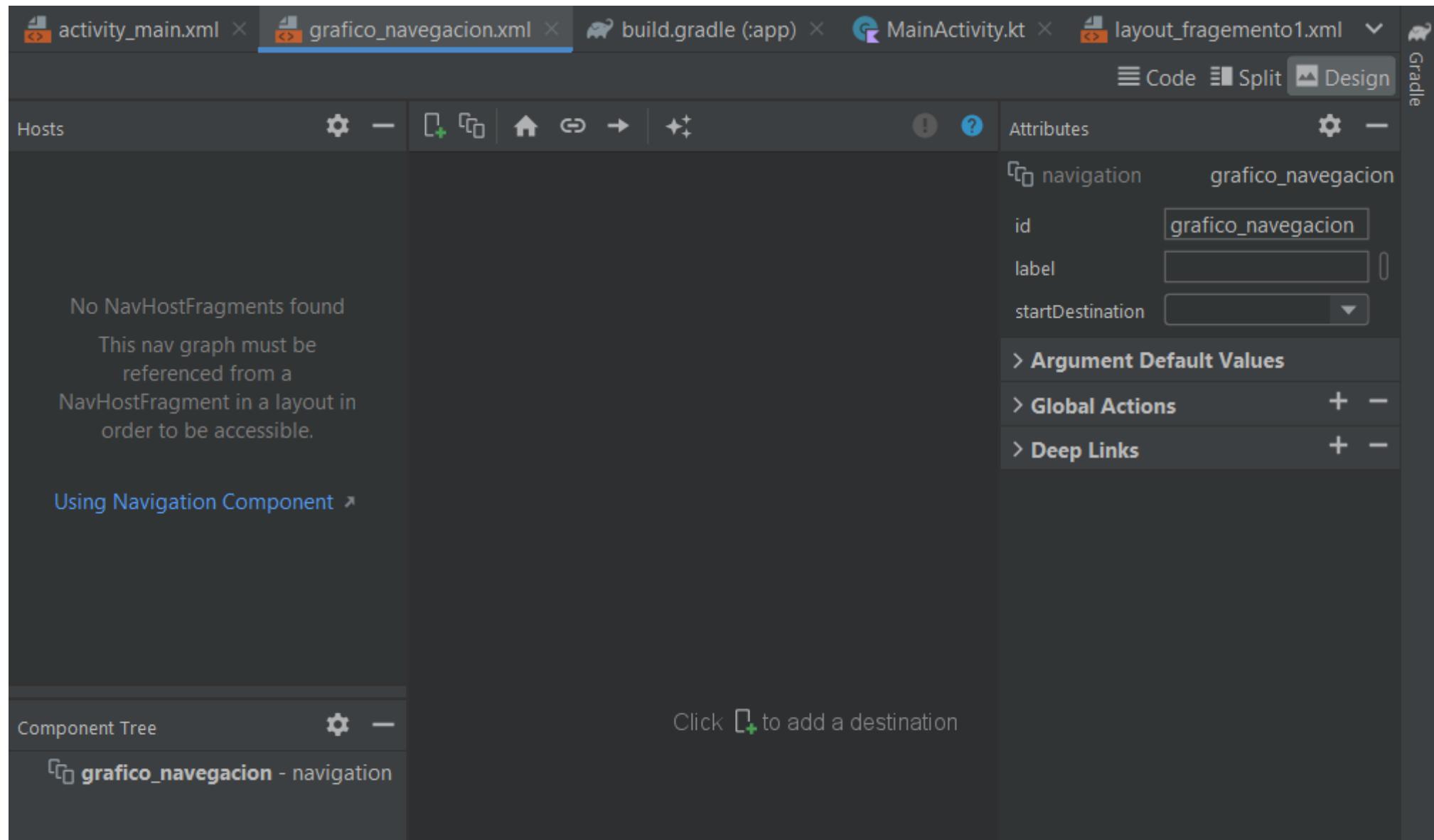
- androidx.annotation:~:1.1.0
- versus:
- androidx.annotation:~:2.0.0

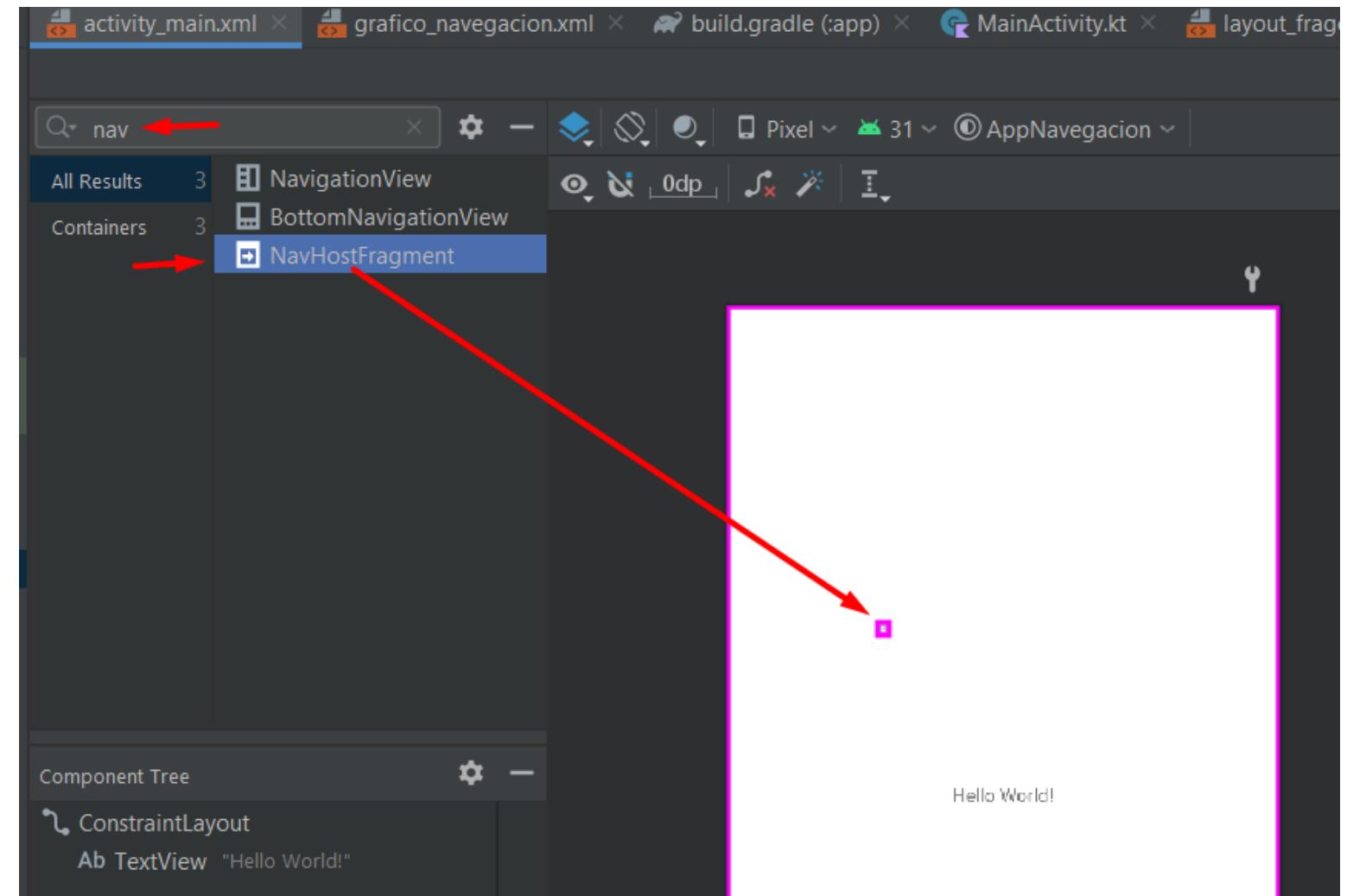
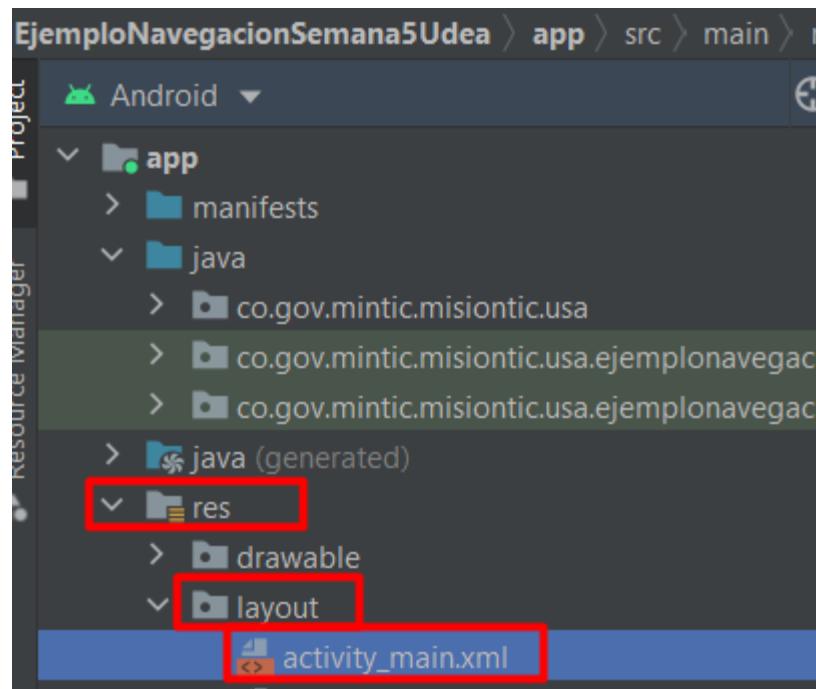
OK

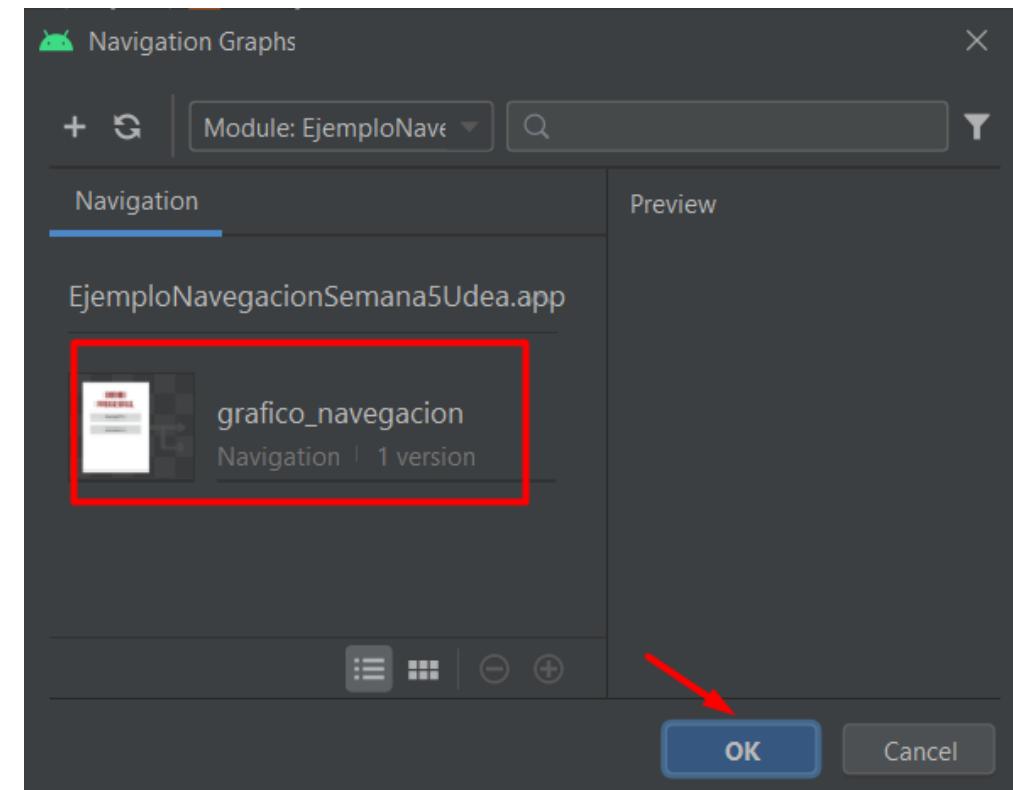
Cancel

```
implementation 'androidx.navigation:navigation-fragment:2.3.5'  
implementation 'androidx.navigation:navigation-ui:2.3.5'
```

```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.4.0'  
    implementation 'com.google.android.material:material:1.4.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.2'  
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'  
    implementation 'androidx.navigation:navigation-fragment:2.3.5'  
    implementation 'androidx.navigation:navigation-ui:2.3.5'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
}
```







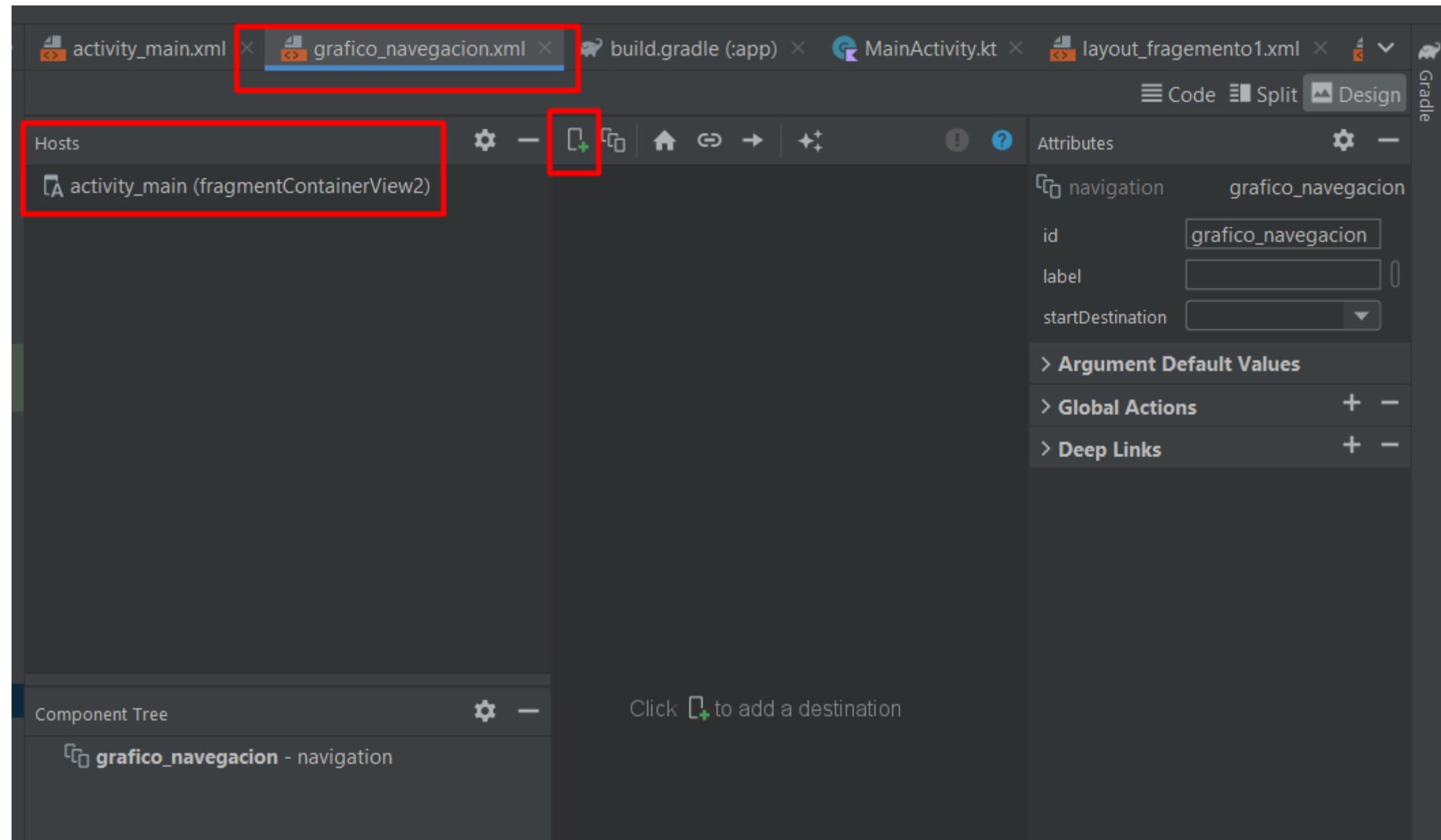
The screenshot shows the XML code for 'activity\_main.xml' in the Android Studio code editor. The code defines a linear layout with a vertical orientation, containing a text view with 'Hello World!' text and a fragment container view. The fragment container view is set as the default nav host and points to the 'grafico\_navegacion' navigation graph. The code is color-coded for syntax highlighting.

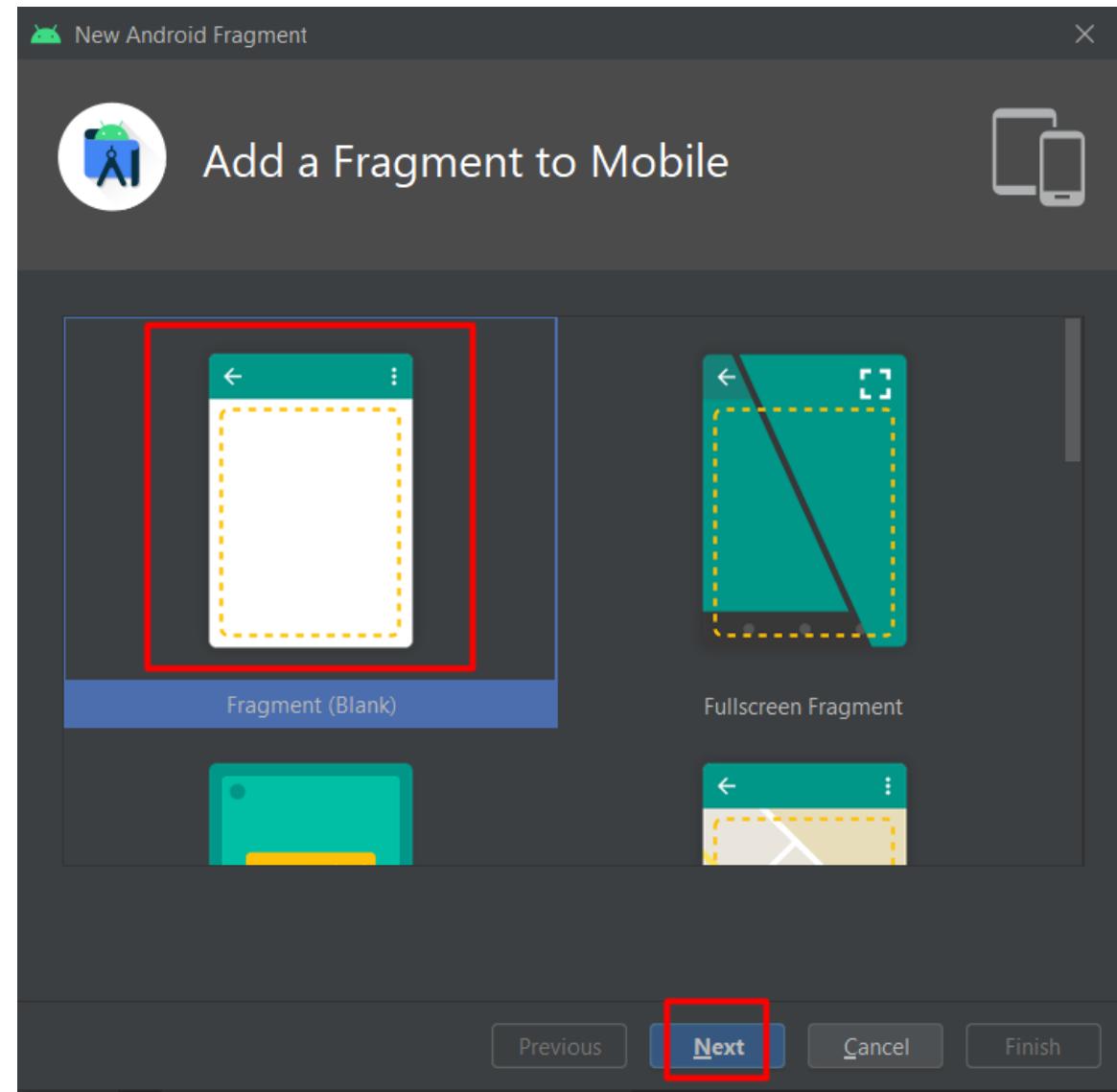
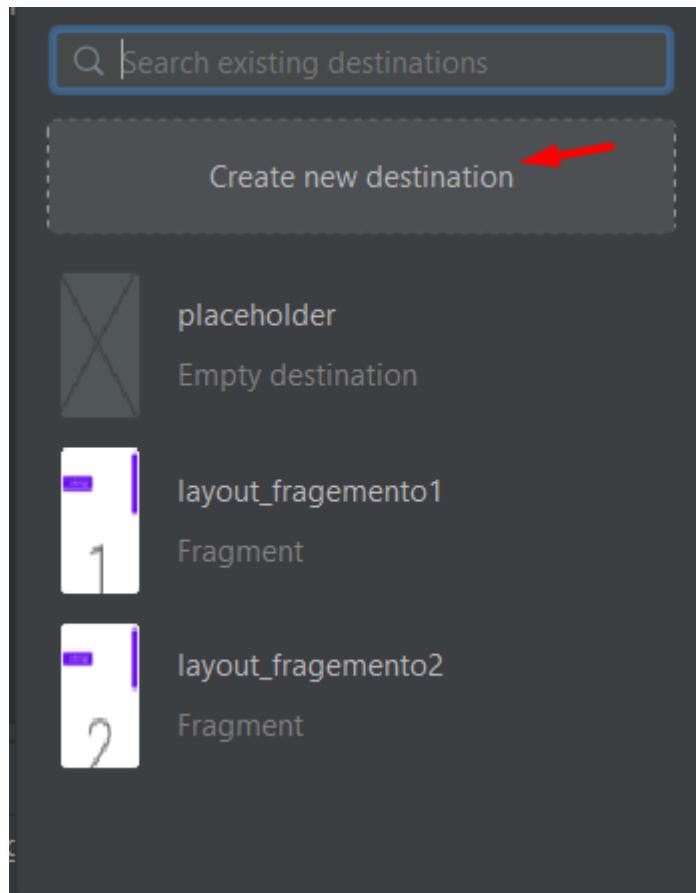
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

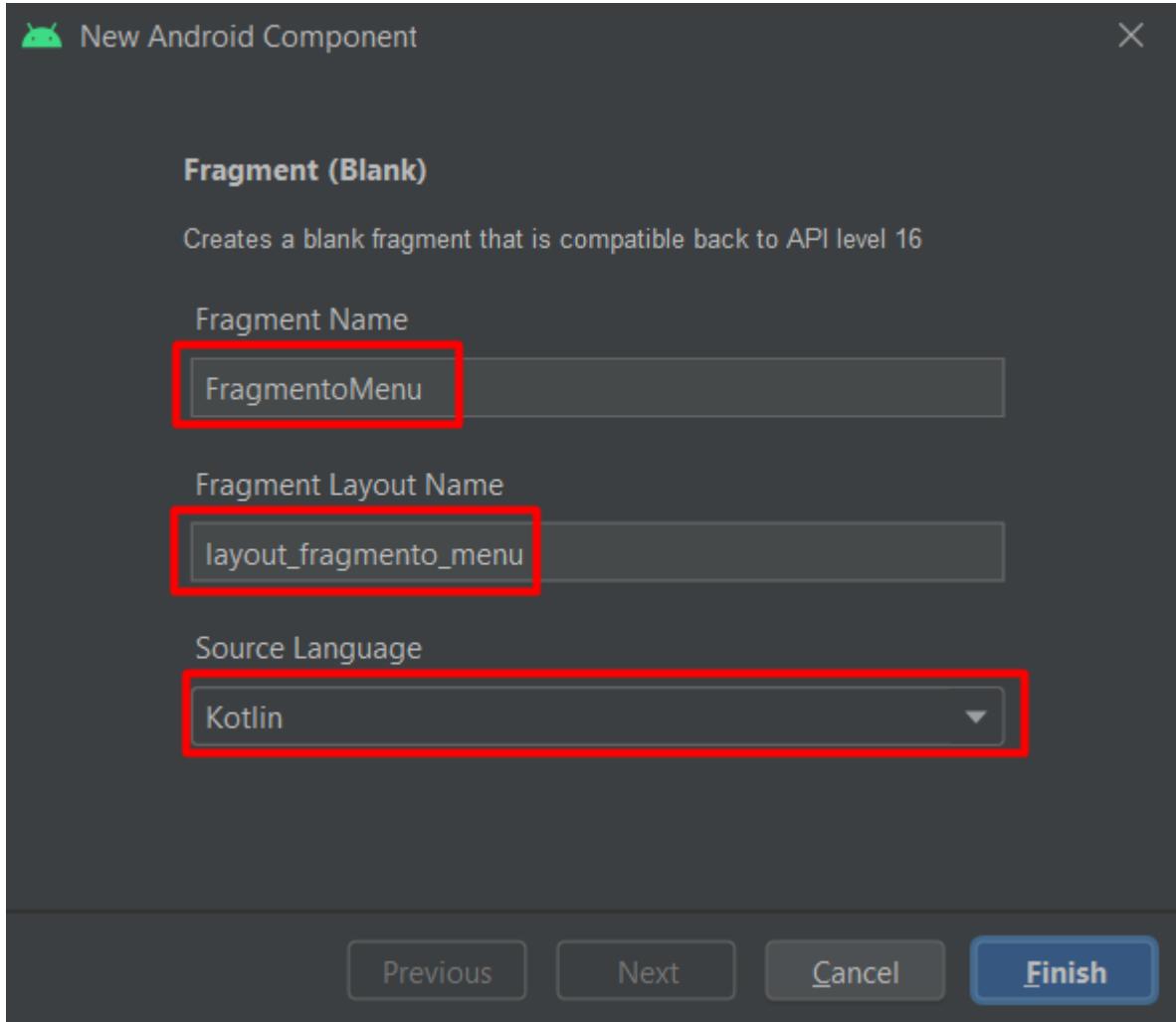
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

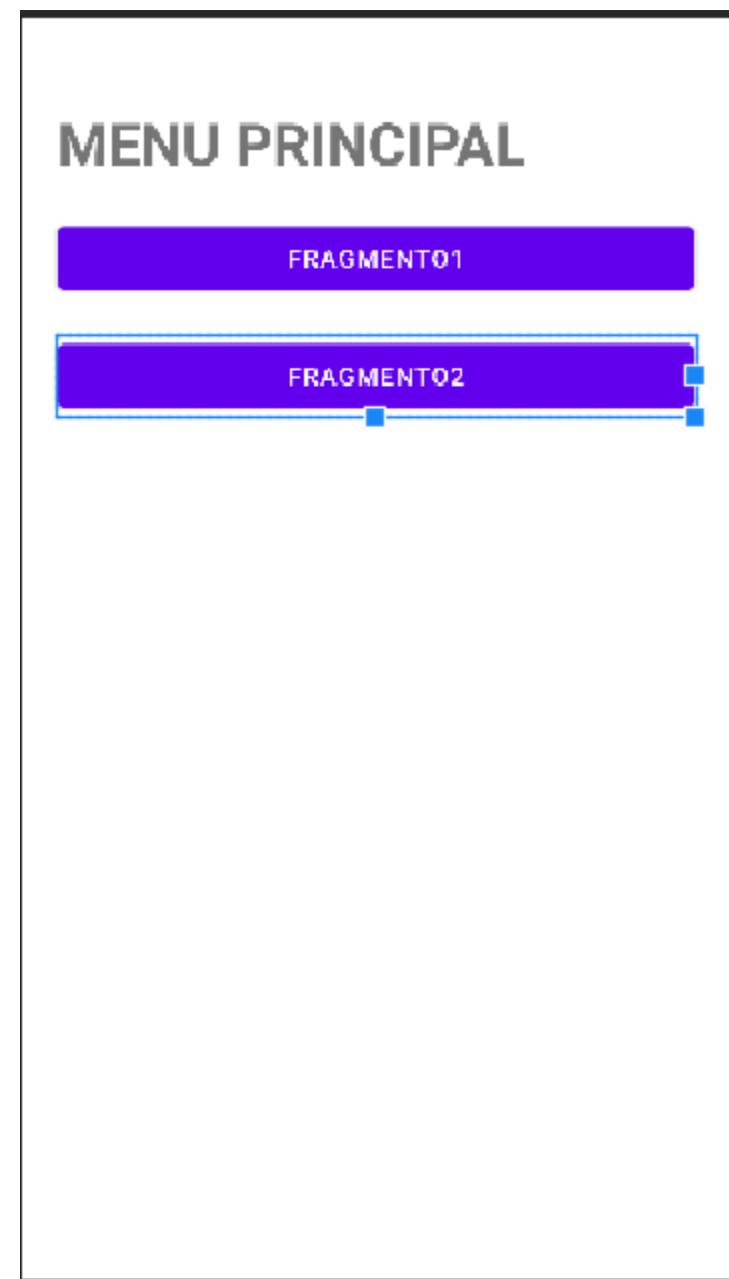
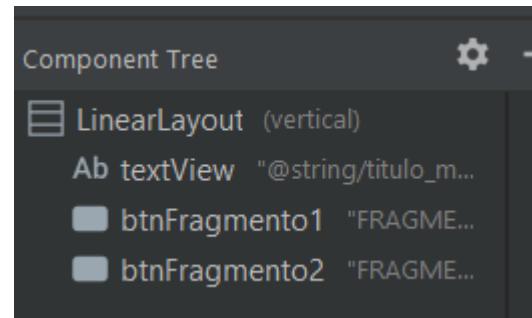
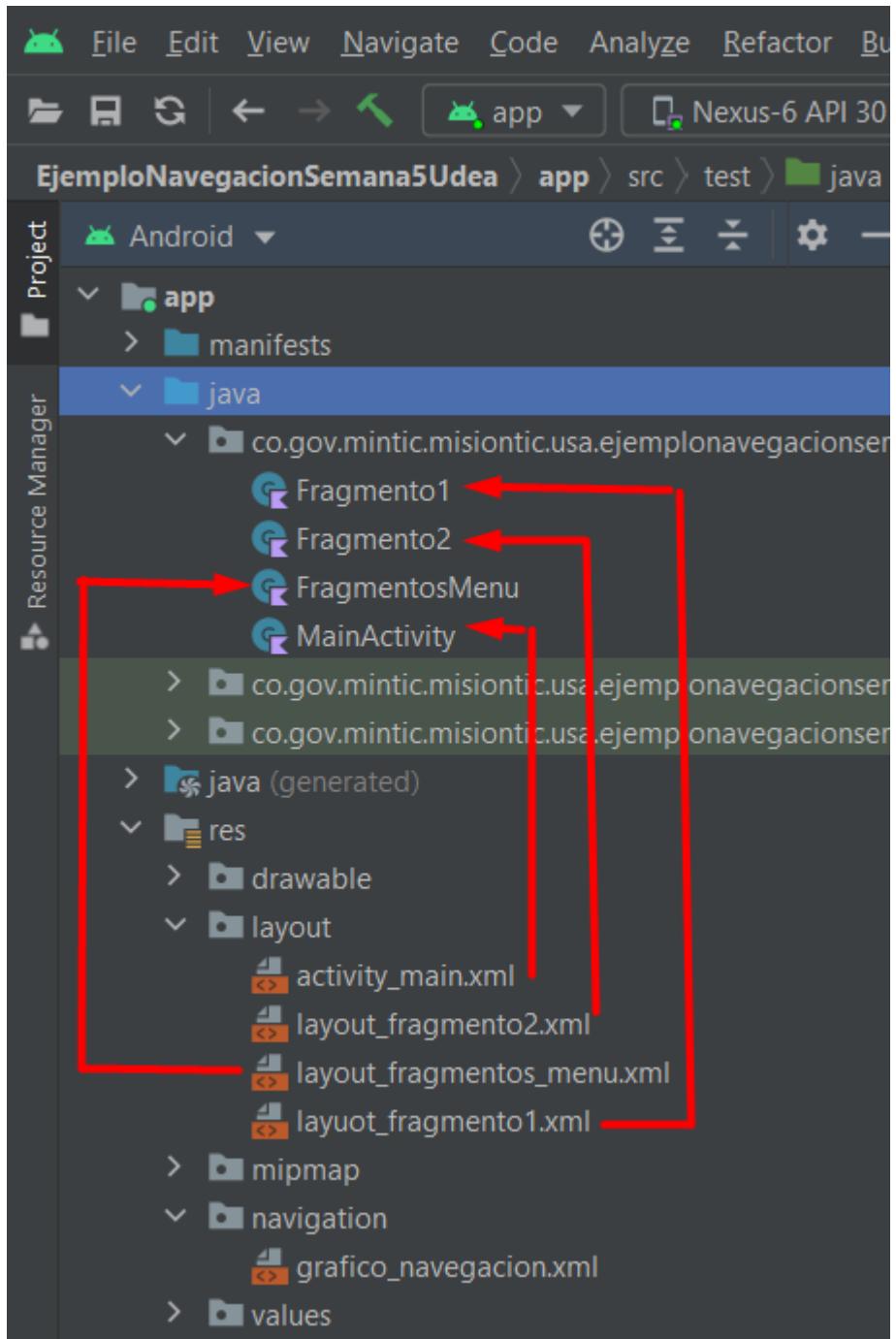
    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragmentContainerView2"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/grafico_navegacion" />

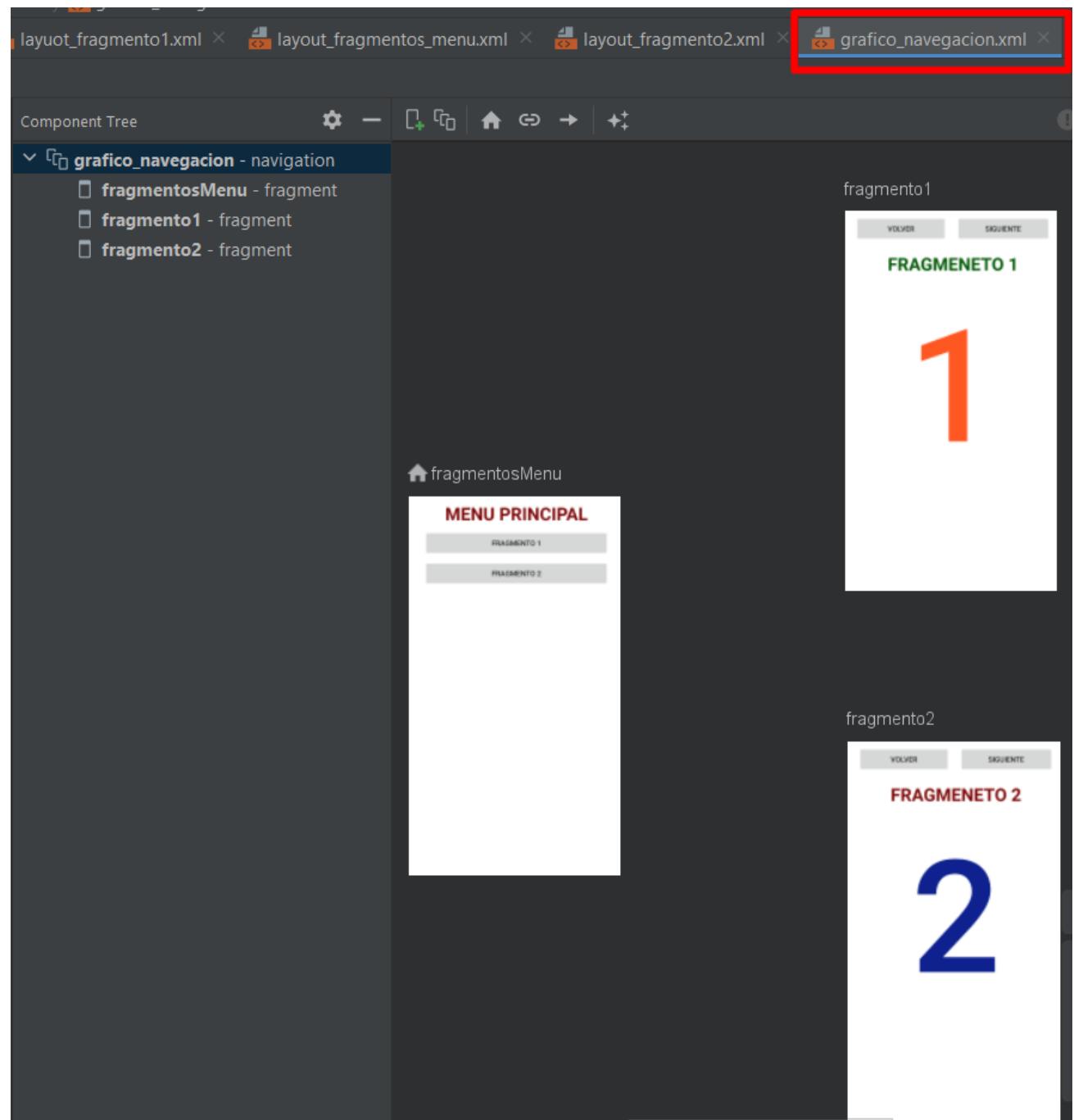
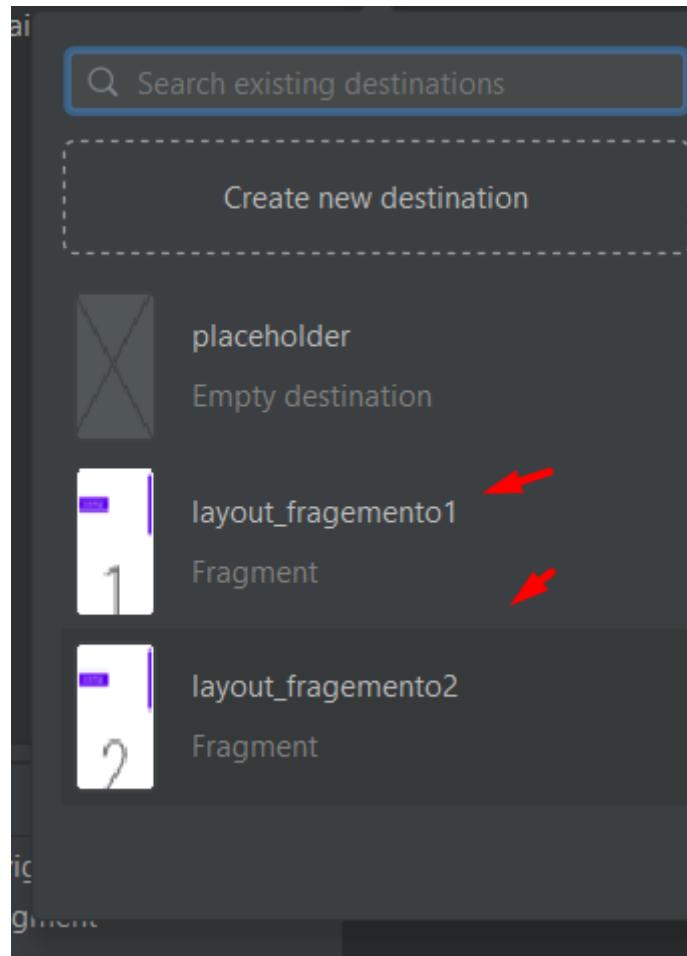
</LinearLayout>
```

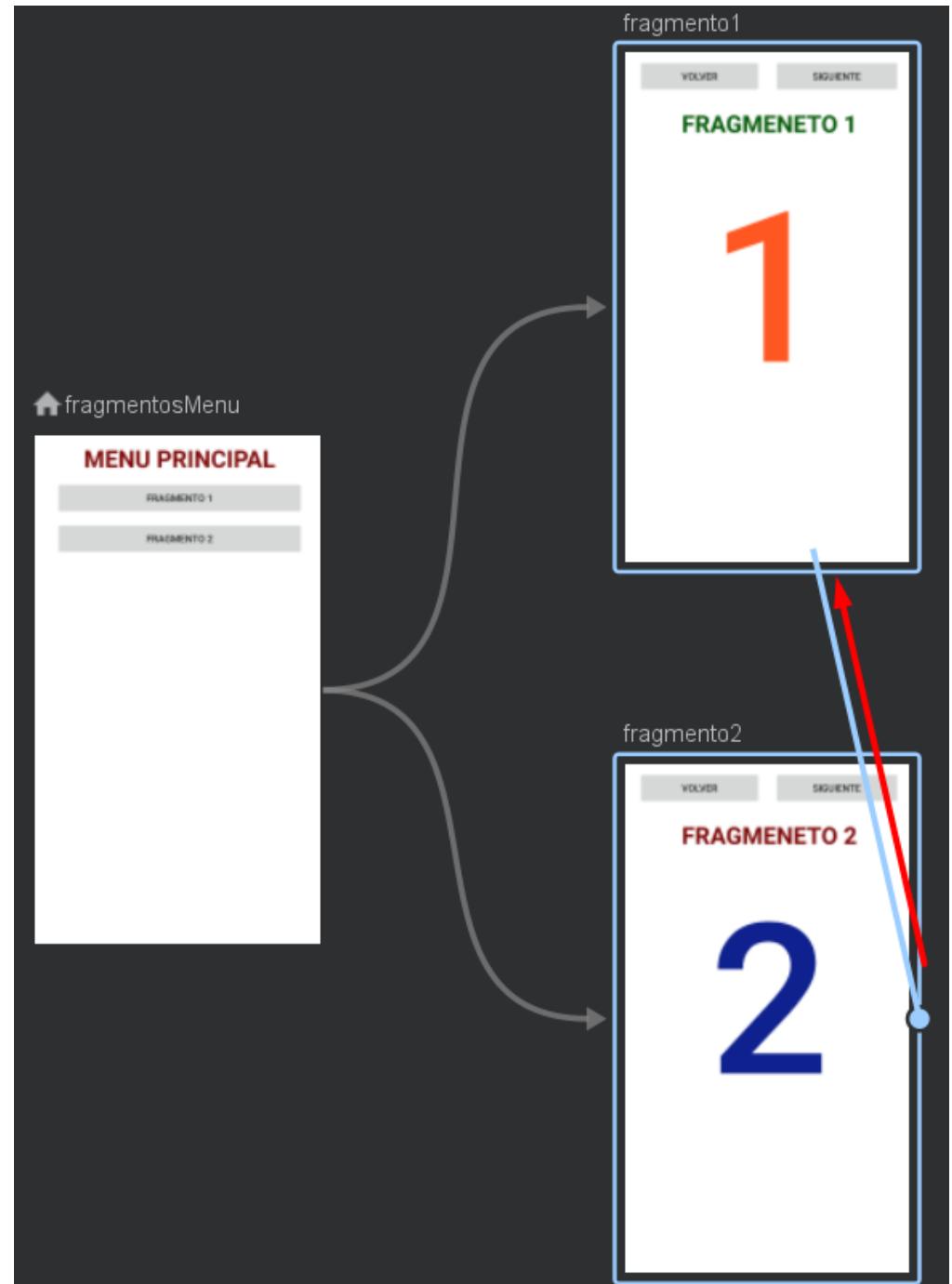
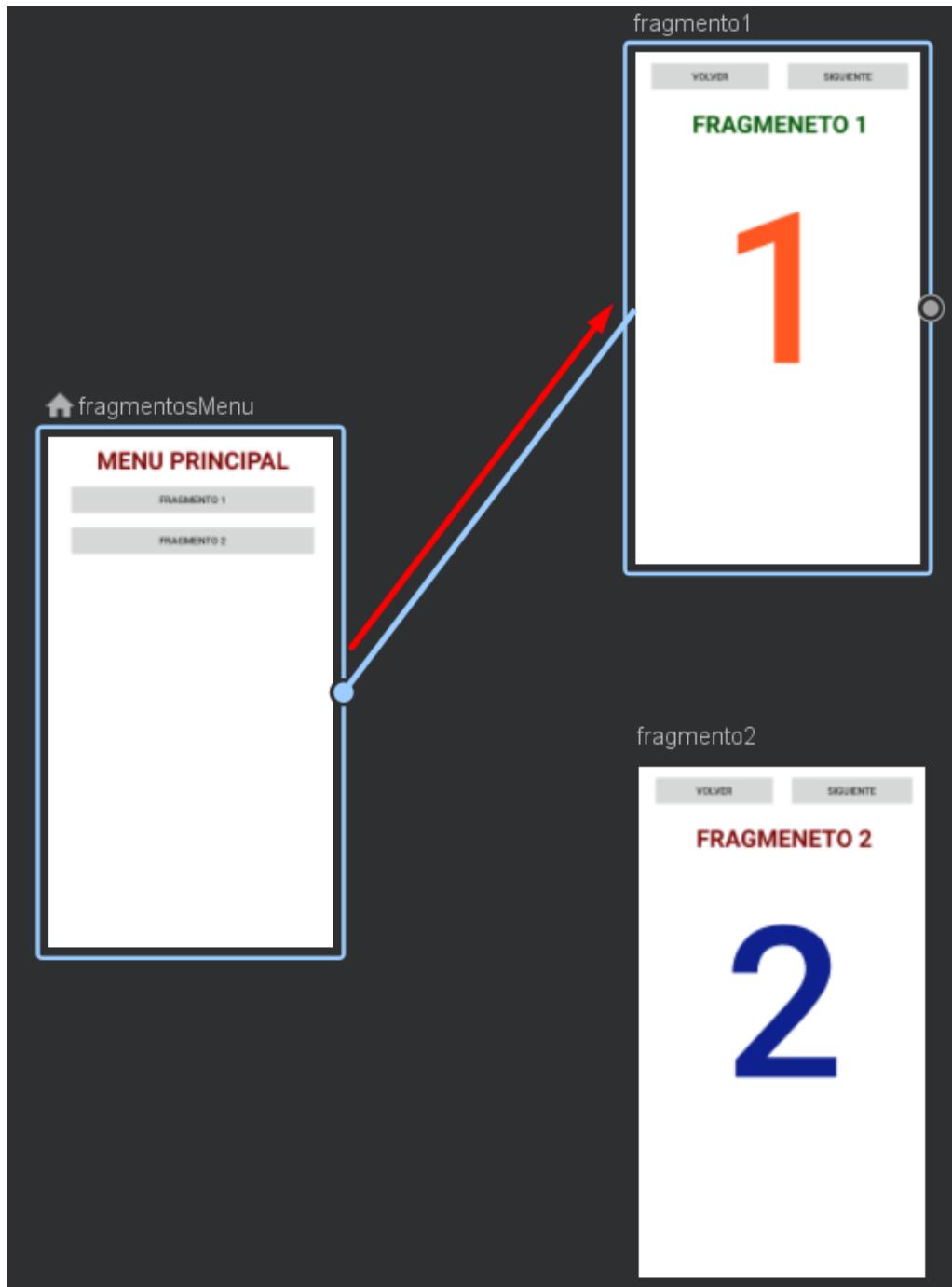


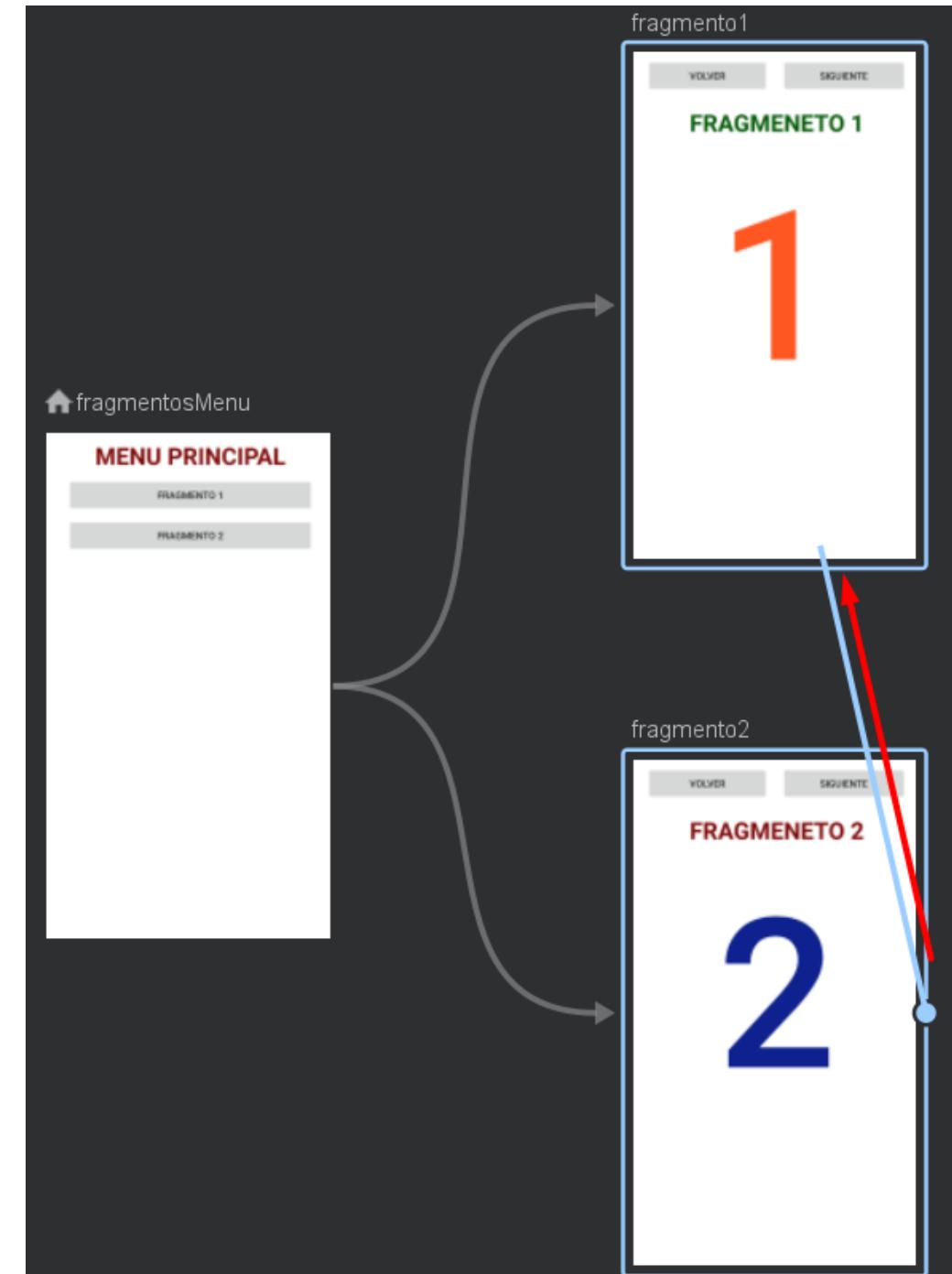
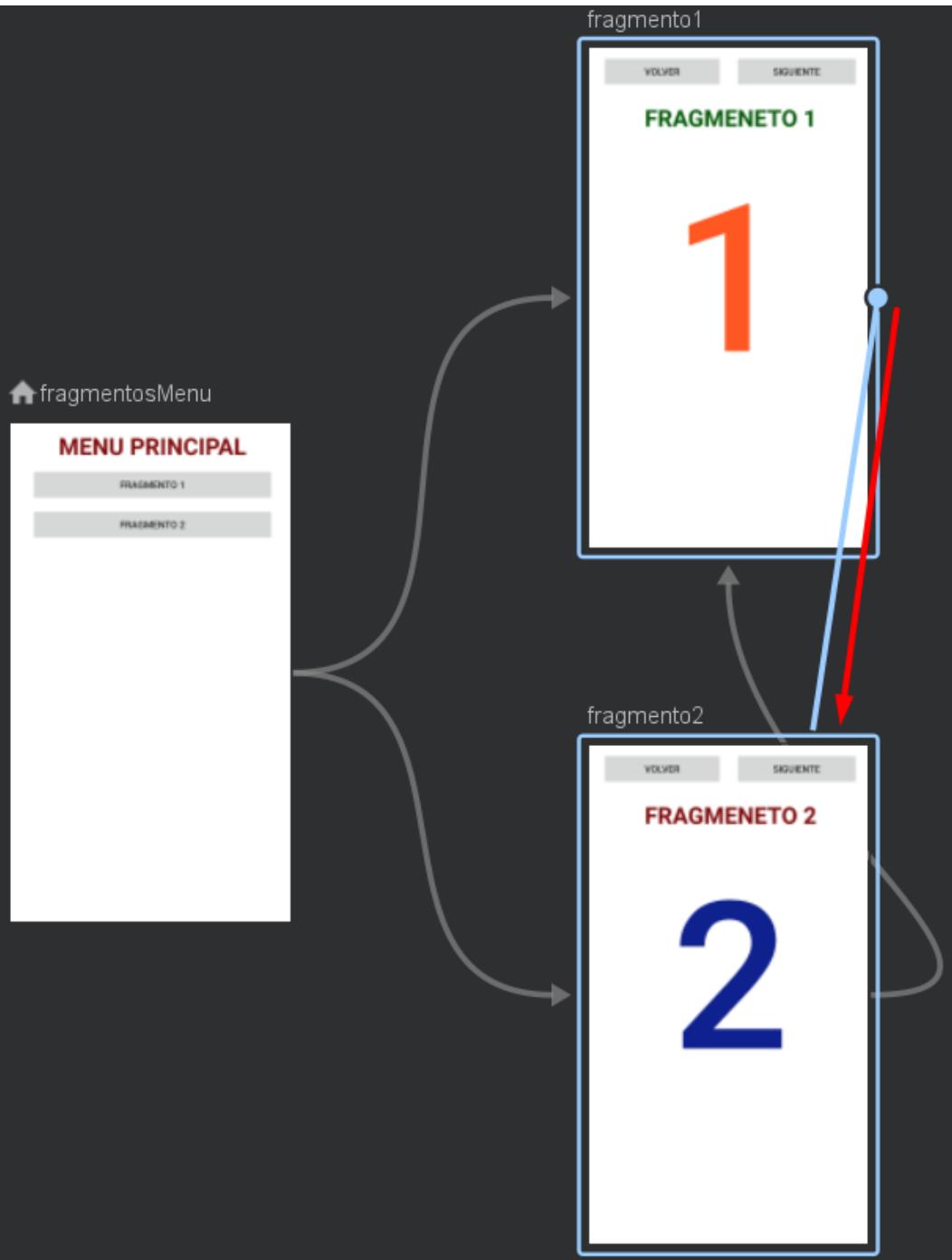


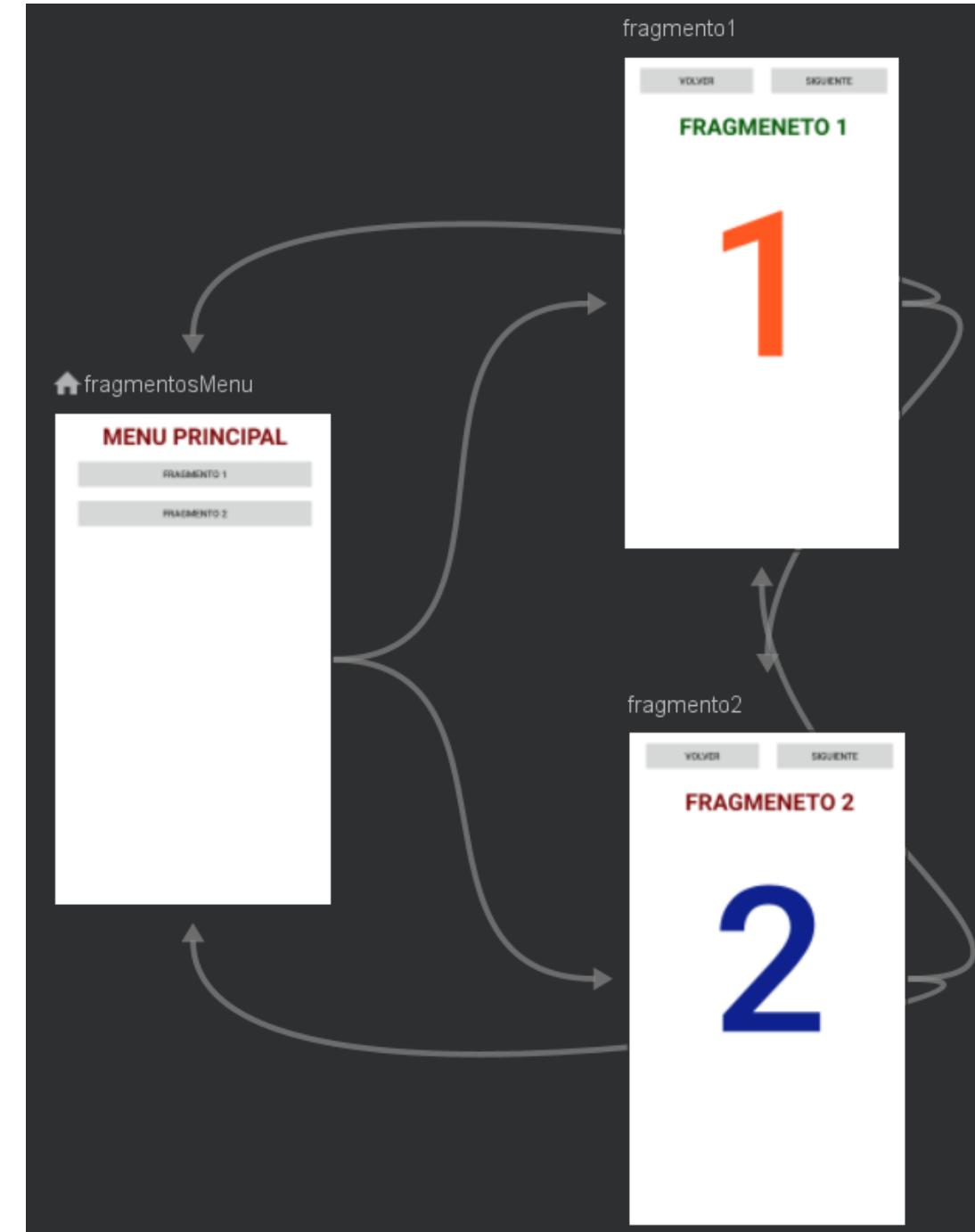
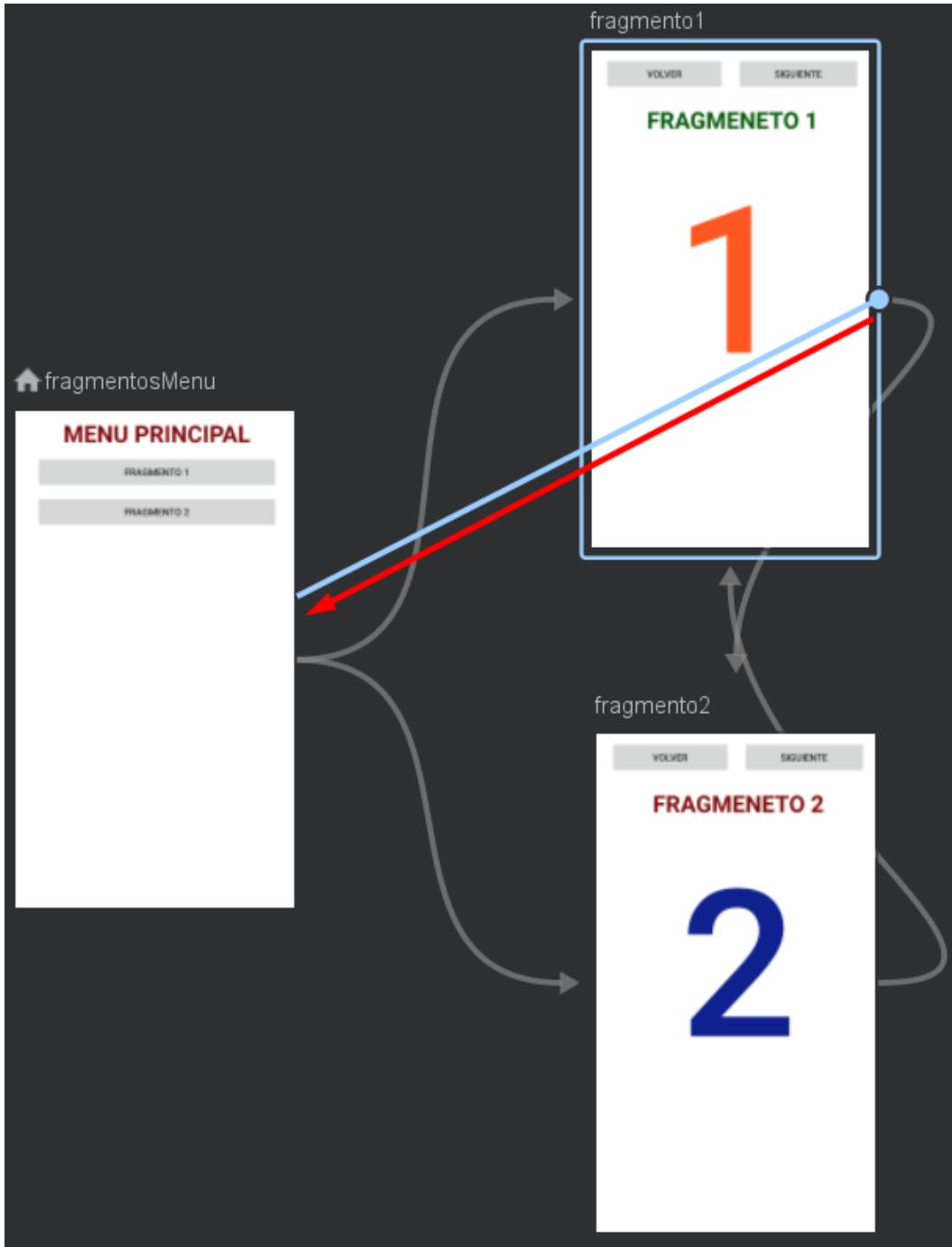


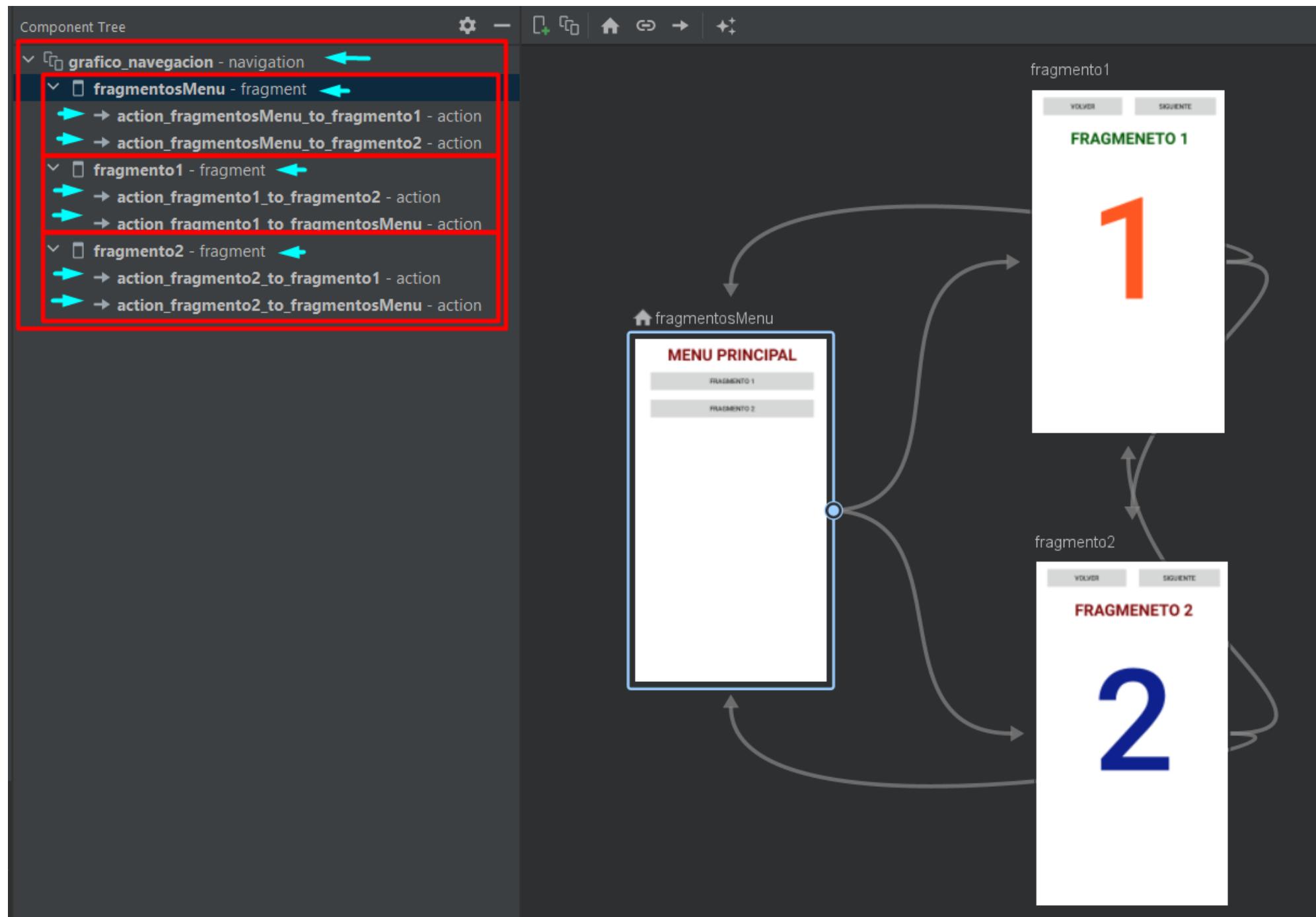












```
grafico_navegacion.xml X layout_fragmento1.xml X layout_fragmentos_menu.xml X layout_fragment
```

```
5     xmlns:android="http://schemas.android.com/apk/res/android"
6     android:id="@+id/grafico_navegacion"
7     app:startDestination="@+id/fragmentosMenu">
8
9     <fragment
10        android:id="@+id/fragmentosMenu"
11        android:name="co.gov.mintic.misiontic.usa.ejemplonavegacionsemana5udea.FragmentosMenu"
12        android:label="layout_fragmentos_menu"
13        tools:layout="@layout/layout_fragmentos_menu" >
14         <action
15            android:id="@+id/action_fragmentosMenu_to_fragmento1"
16            app:destination="@+id/fragmento1" />
17         <action
18            android:id="@+id/action_fragmentosMenu_to_fragmento2"
19            app:destination="@+id/fragmento2" />
20     </fragment>
21     <fragment
22        android:id="@+id/fragmento1"
23        android:name="co.gov.mintic.misiontic.usa.ejemplonavegacionsemana5udea.Fragmento1"
24        android:label="layout_fragmento1"
25        tools:layout="@layout/layout_fragmento1" >
26         <action
27            android:id="@+id/action_fragmento1_to_fragmento2"
28            app:destination="@+id/fragmento2" />
29         <action
30            android:id="@+id/action_fragmento1_to_fragmentosMenu"
31            app:destination="@+id/fragmentosMenu" />
32     </fragment>
33     <fragment
34        android:id="@+id/fragmento2"
35        android:name="co.gov.mintic.misiontic.usa.ejemplonavegacionsemana5udea.Fragmento2"
36        android:label="layout_fragmento2"
37        tools:layout="@layout/layout_fragmento2" >
38         <action
39            android:id="@+id/action_fragmento2_to_fragmento1"
40            app:destination="@+id/fragmento1" />
41         <action
42            android:id="@+id/action_fragmento2_to_fragmentosMenu"
43            app:destination="@+id/fragmentosMenu" />
44     </fragment>
45 </navigation>
```

```
grafico_navegacion.xml x FragmentosMenu.kt x layout_fragmento1.xml x layout_fragmentos_menu.xml x layout_fragmento2.xml x
1 package co.gov.mintic.misiontic.usa.ejemplonavegacionsemana5udea
2
3 import ...
4
5
6 class FragmentosMenu : Fragment() {
7
8     override fun onCreateView(
9             inflater: LayoutInflater, container: ViewGroup?,
10            savedInstanceState: Bundle?
11        ): View? {
12             // Inflate the layout for this fragment
13             return inflater.inflate(R.layout.layout_fragmentos_menu, container, attachToRoot: false)
14         }
15
16     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
17             super.onViewCreated(view, savedInstanceState)
18             var manejador = Navigation.findNavController(view)
19             var botonFrag1 = view.findViewById<Button>(R.id.btnFragmento1)
20             var botonFrag2 = view.findViewById<Button>(R.id.btnFragmento2)
21             botonFrag1.setOnClickListener { it: View!
22                 manejador.navigate(R.id.fragmento1)
23             }
24             botonFrag2.setOnClickListener { it: View!
25                 manejador.navigate(R.id.fragmento2)
26             }
27         }
28     }
29 }
```

```
1 package co.gov.mintic.misiontic.usa.ejemplonavegacionsemana5udea
2
3 import ...
10
11 class Fragmento1 : Fragment() {
12
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15     }
16
17     override fun onCreateView(
18         inflater: LayoutInflater, container: ViewGroup?,
19         savedInstanceState: Bundle?
20     ): View? {
21         // Inflate the layout for this fragment
22         return inflater.inflate(R.layout.layout_fragmento1, container, attachToRoot: false)
23     }
24
25     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
26
27         super.onViewCreated(view, savedInstanceState)
28         var btnAtras = view.findViewById<Button>(R.id.btnAtras1)
29         btnAtras.setOnClickListener { it: View! -->
30             Navigation.findNavController(view).navigate(R.id.fragmentosMenu)
31         }
32
33         var btnSig = view.findViewById<Button>(R.id.btnSigiente1)
34         btnSig.setOnClickListener { it: View! -->
35             Navigation.findNavController(view).navigate(R.id.fragmento2)
36         }
37     }
38 }
```

```
grafico_navegacion.xml x FragmentosMenu.kt x Fragmento1.kt x Fragmento2.kt x layout_fragmento1.xml x layout_frag
1 package co.gov.mintic.misiontic.usa.ejemplonavegacionsemana5udea
2
3 import ...
10
11 class Fragmento2 : Fragment() {
12
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15     }
16
17     override fun onCreateView(
18         inflater: LayoutInflater, container: ViewGroup?,
19         savedInstanceState: Bundle?
20     ): View? {
21         // Inflate the layout for this fragment
22         return inflater.inflate(R.layout.layout_fragmento2, container, attachToRoot: false)
23     }
24
25     override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
26
27         super.onViewCreated(view, savedInstanceState)
28         var btnAtras = view.findViewById<Button>(R.id.btnAtras2)
29         btnAtras.setOnClickListener { it: View!
30             Navigation.findNavController(view).navigate(R.id.fragmentosMenu)
31         }
32
33         var btnSig = view.findViewById<Button>(R.id.btnSiguiiente2)
34         btnSig.setOnClickListener { it: View!
35             Navigation.findNavController(view).navigate(R.id.fragmento1)
36         }
37     }
38 }
```

