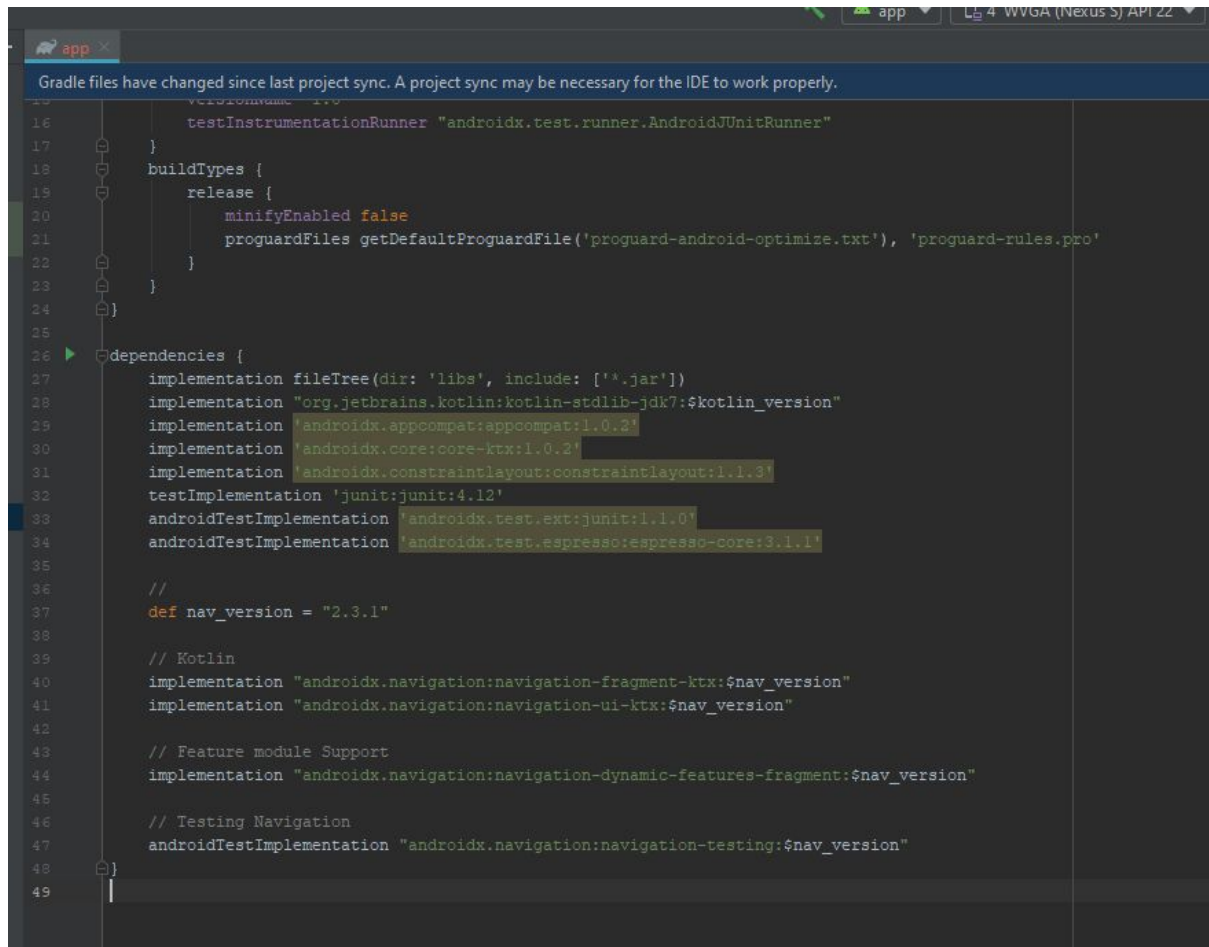
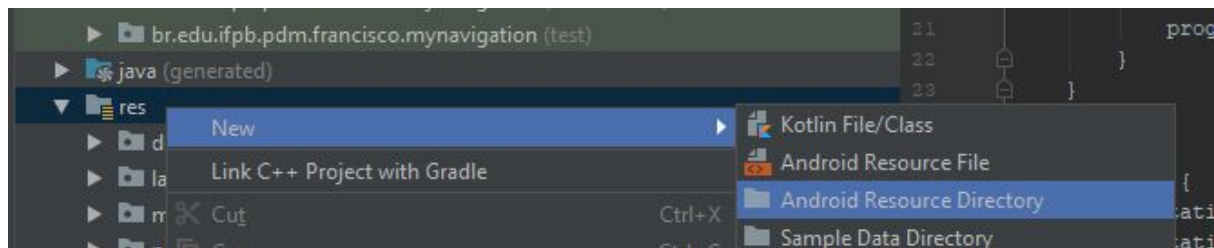


### ### Set up

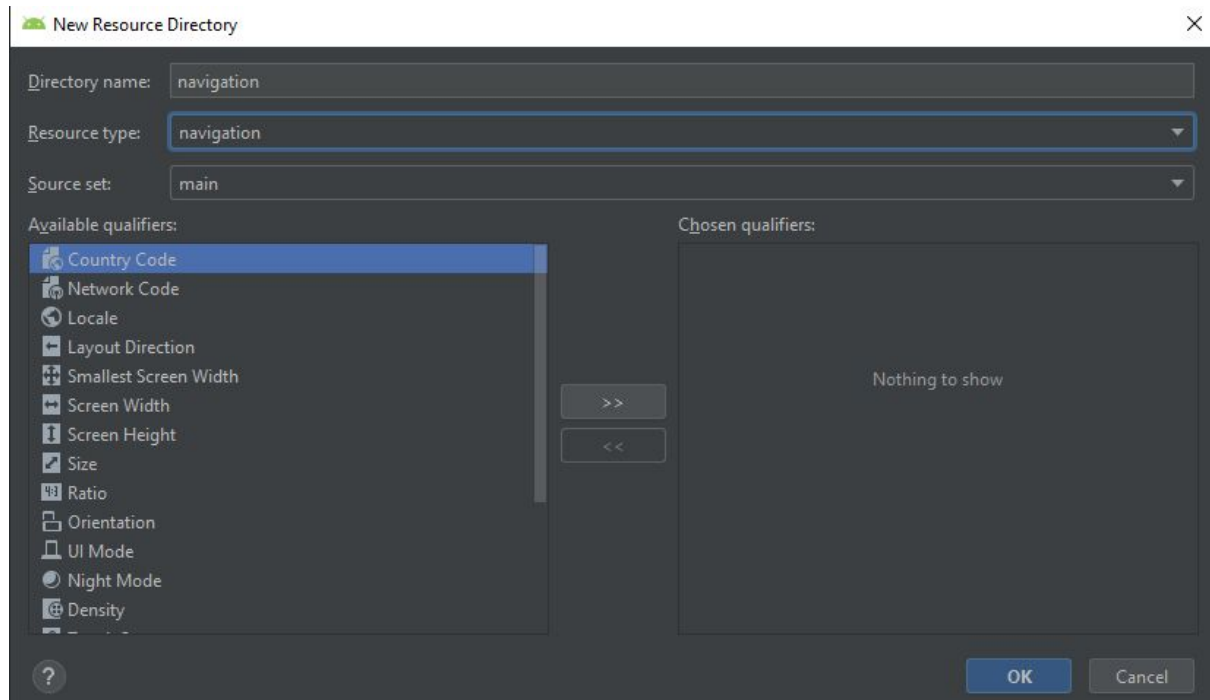
1) Você vai no gradle.build e adiciona o as dependências de navegação.



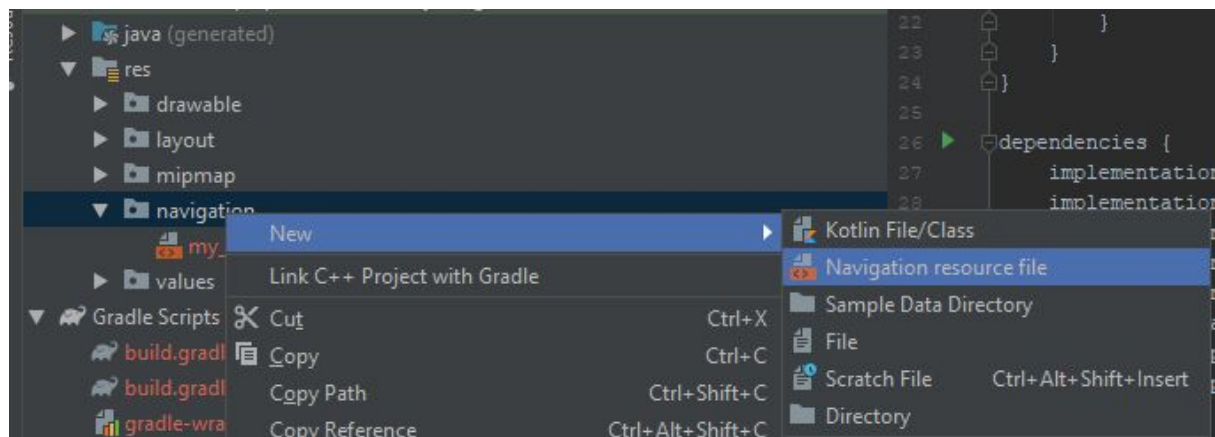
2) Você vai no res > new > android resource directory



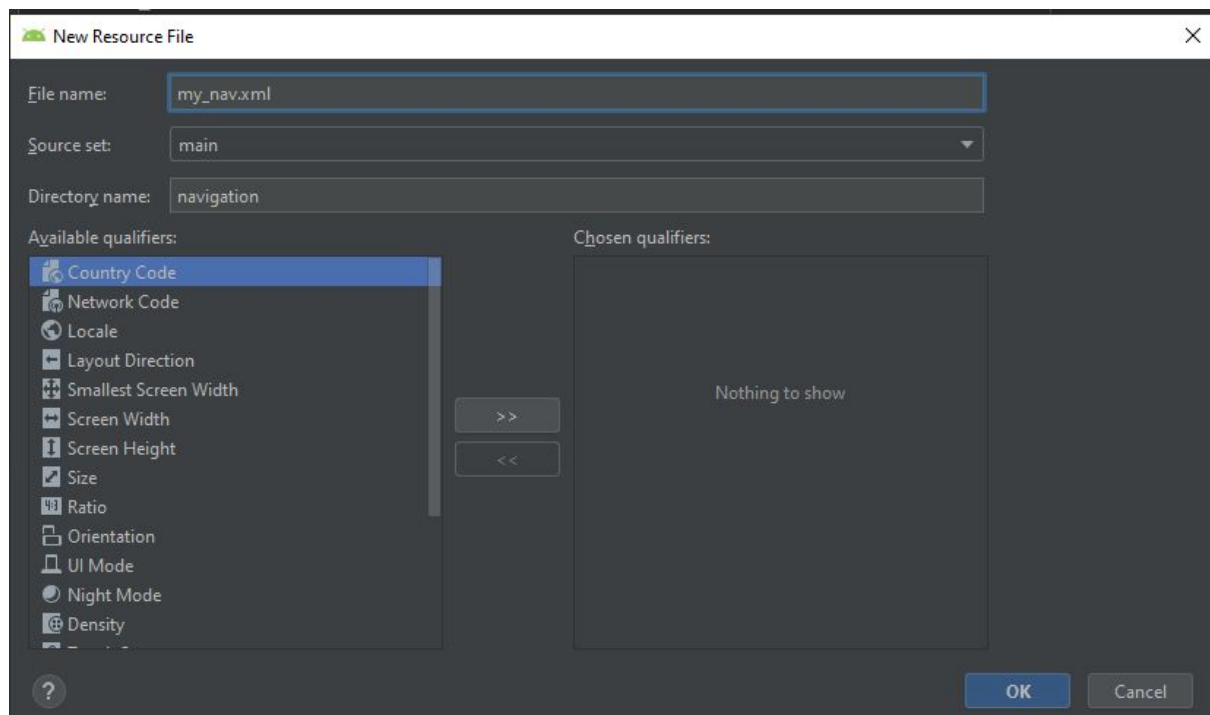
3) Resource type navigation, directory name também navigation



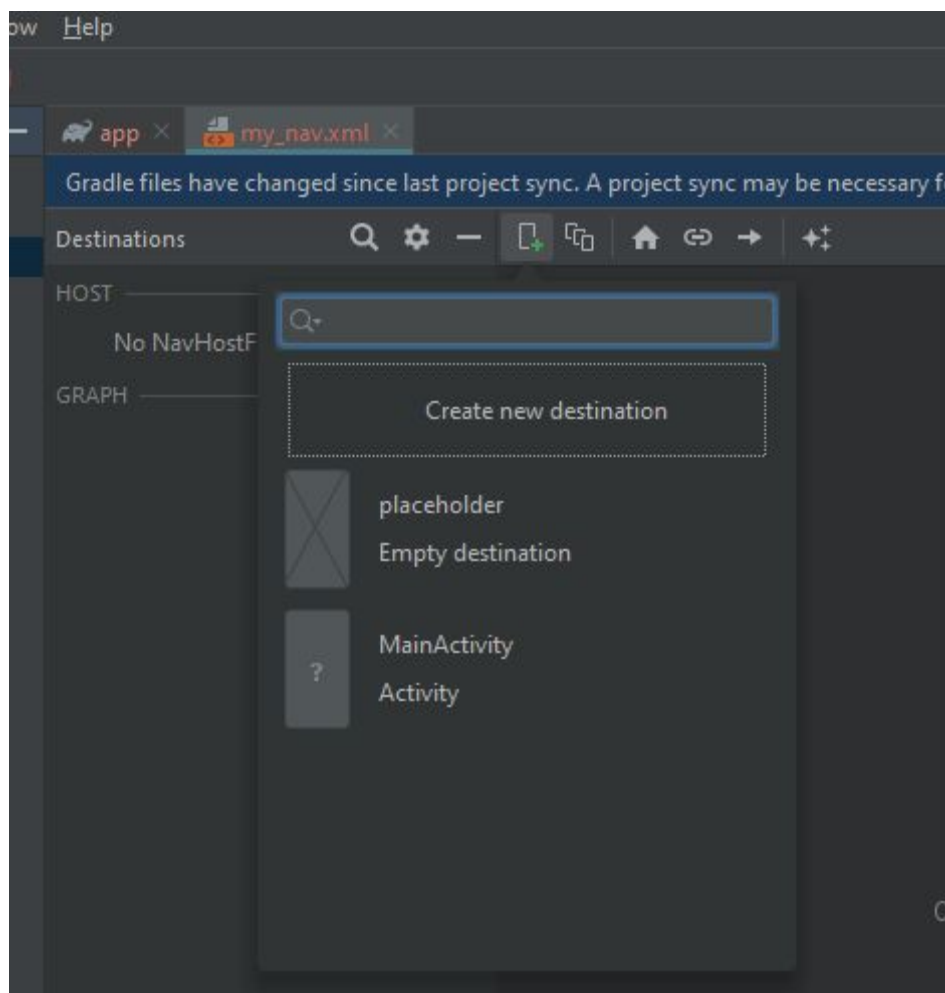
4) Clica com botão direito no directory navigation que foi criado em res > navigation resource file

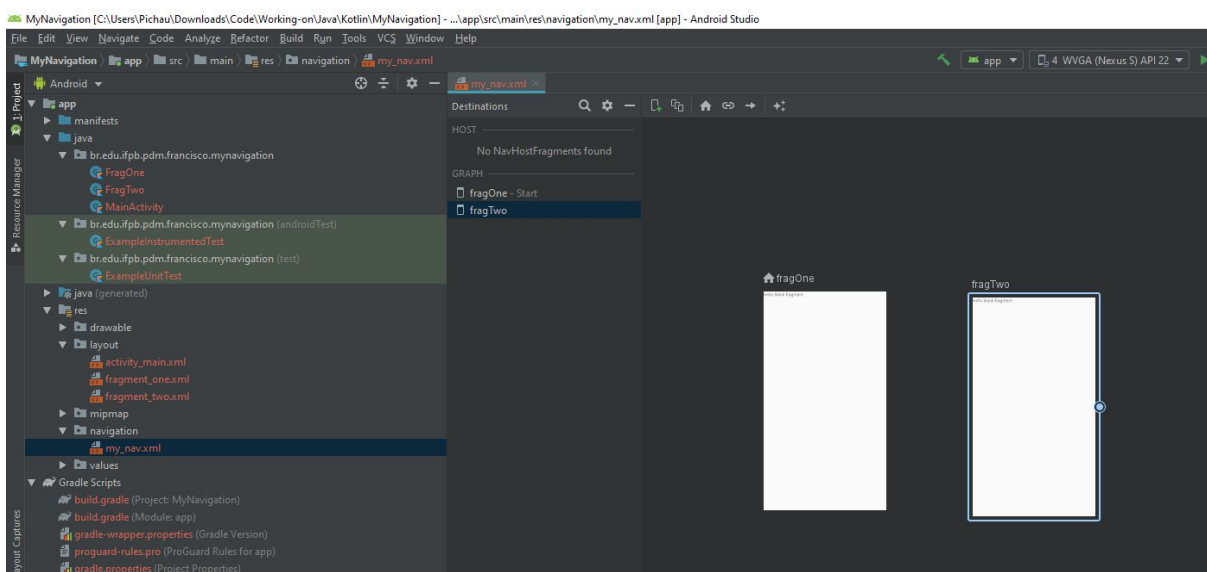
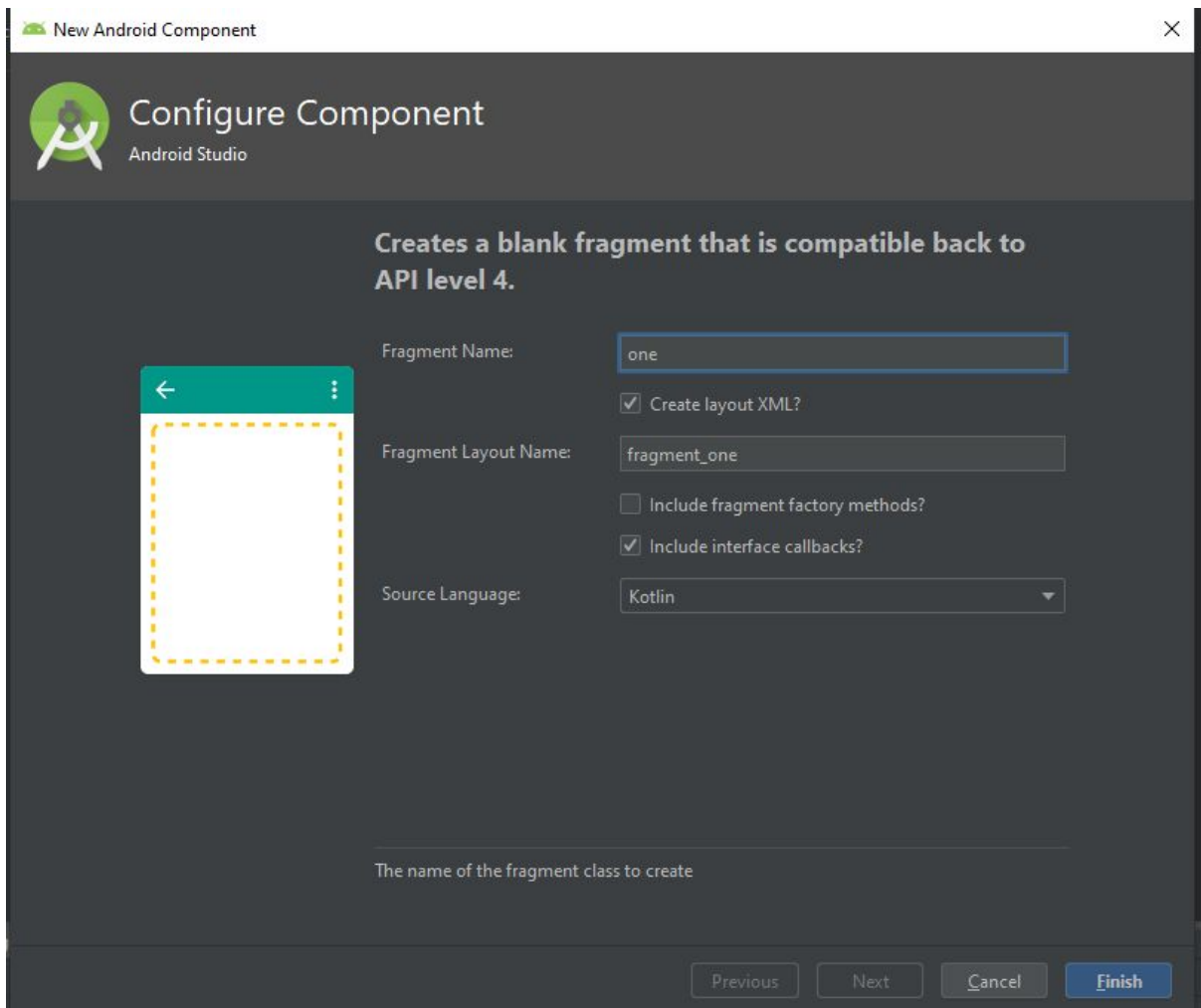


5) Dá um nome genérico, "my\_nav.xml"

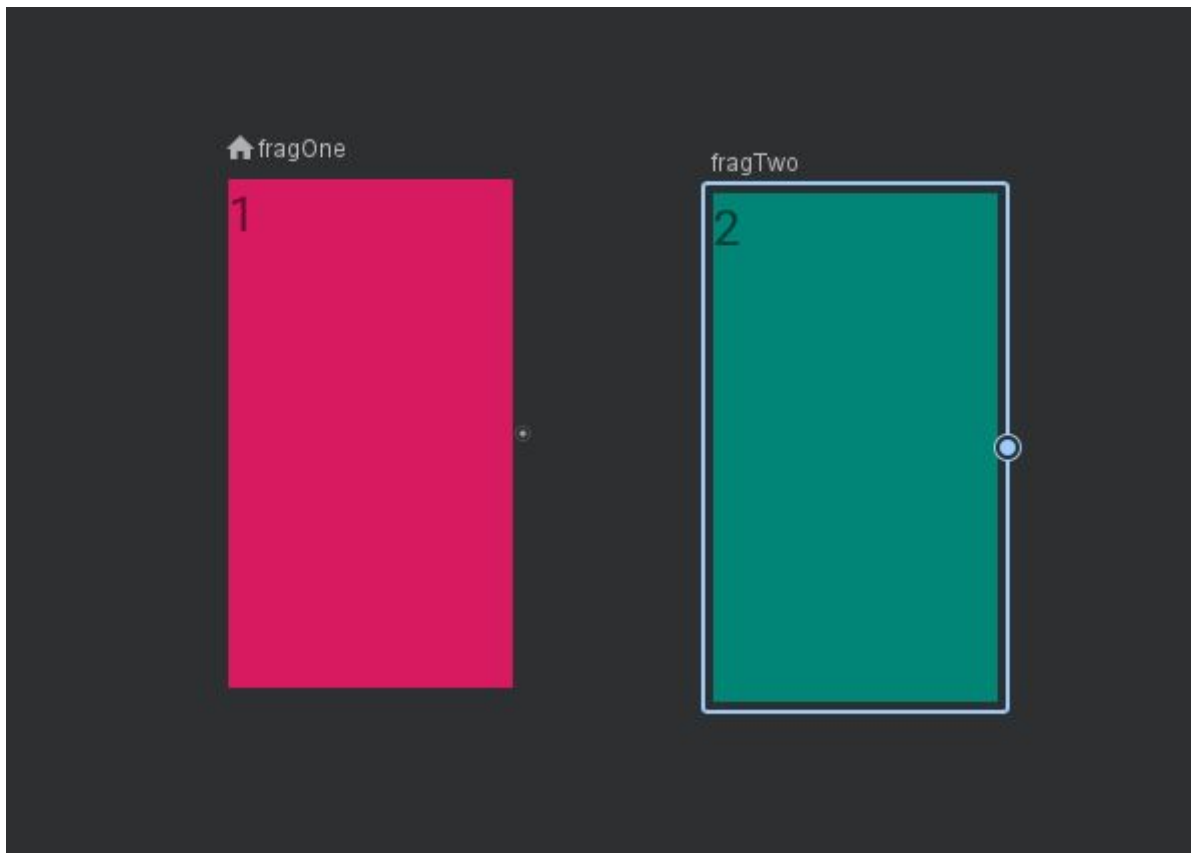


6) Nós precisamos de dois fragmentos agora.

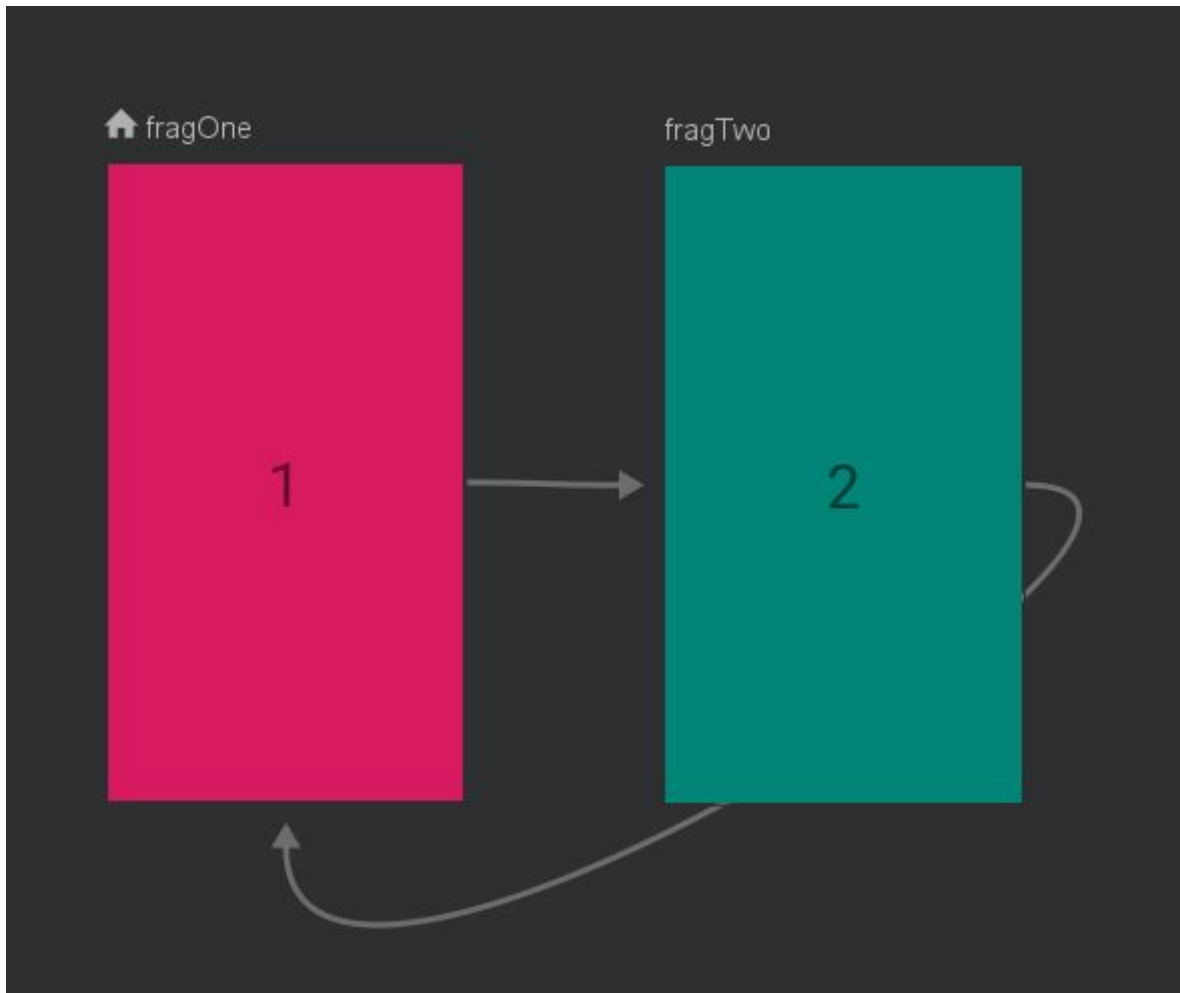




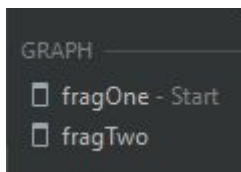
7) Cria um layout pros dois fragmentos



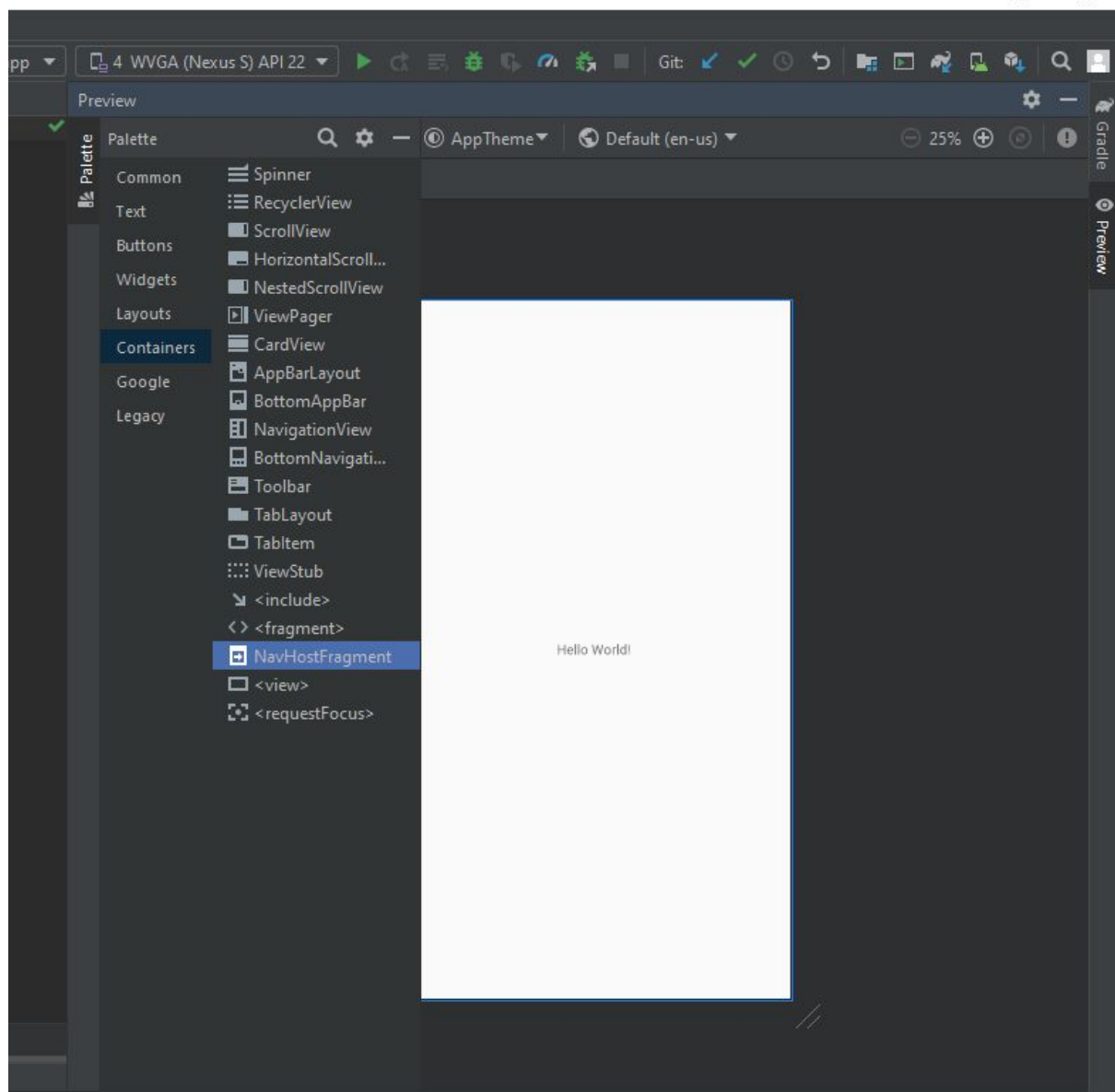
8) Clickando e arrastando de um fragmento pro outro, vc consegue conectar eles.  
Eu cliquei do fragOne -> fragTwo. Depois eu fiz o processo inverso.



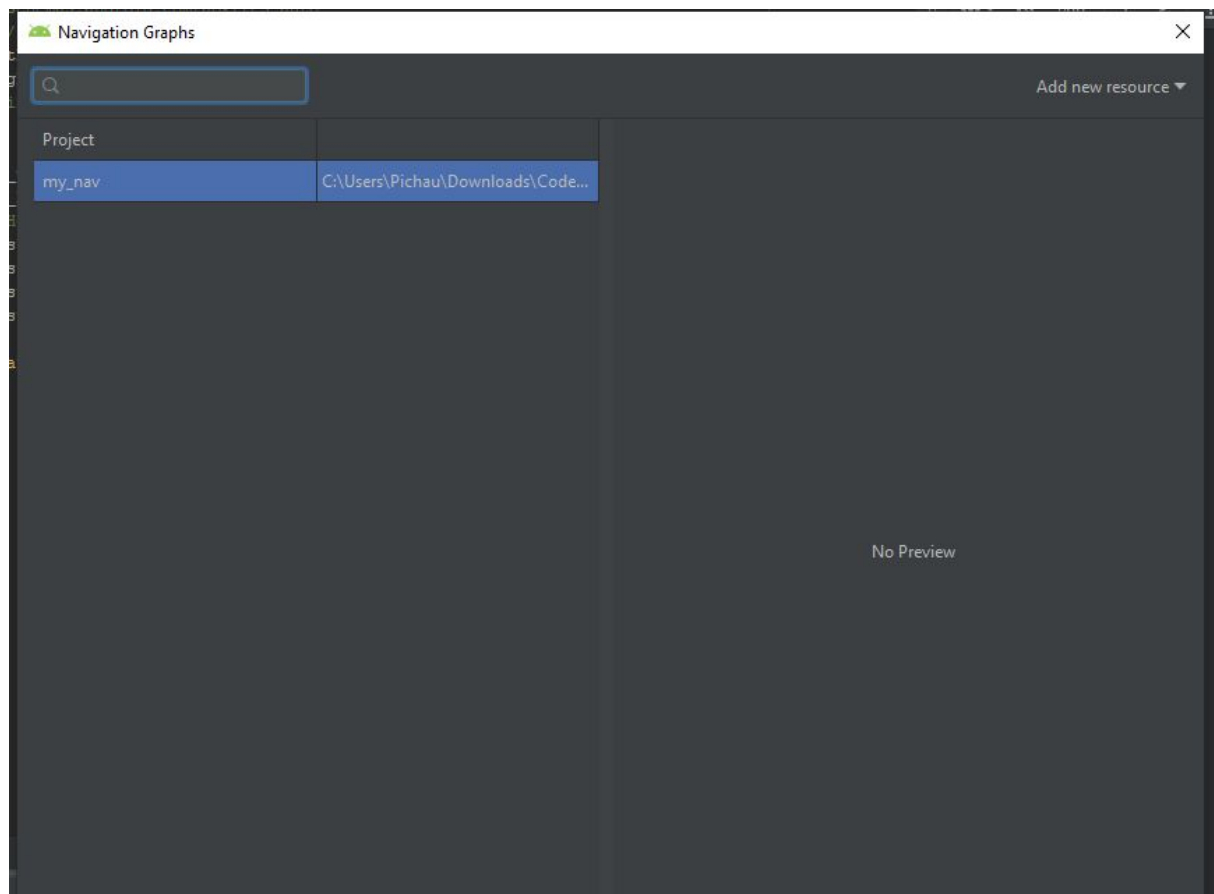
9) Importante ver que fragOne é o start



10) No layout da MainActivity, você coloca um NavHostFragment



11) Na janela que abrir, vc selecionar o seu navigation graph criado anteriormente

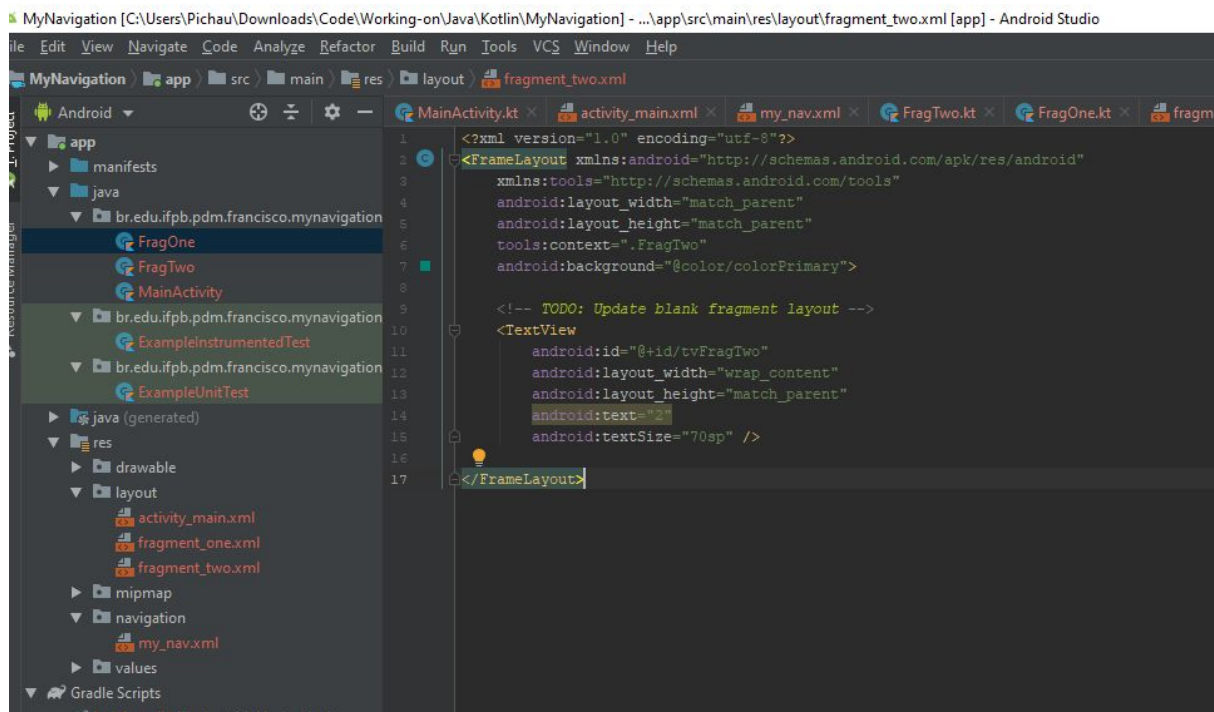


12) O fragmento criado, vc dá match no parent em width e height

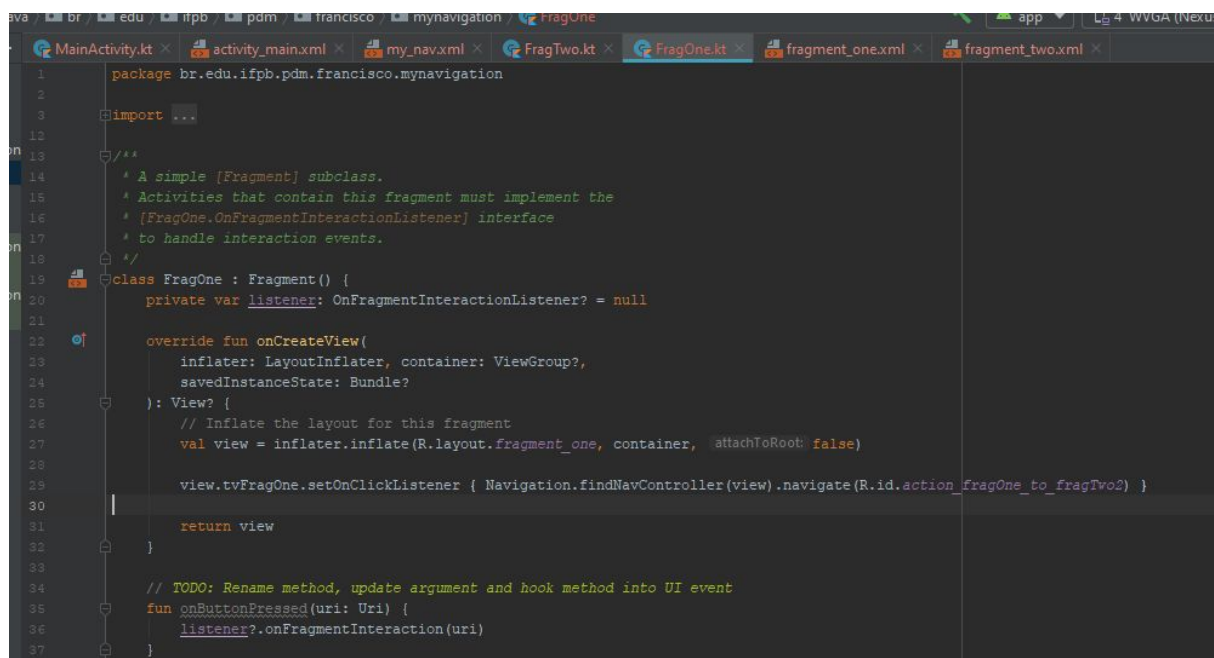


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintLeft_toLeftOf="parent"
15         app:layout_constraintRight_toRightOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18     <fragment
19         android:id="@+id/fragment"
20         android:name="androidx.navigation.fragment.NavHostFragment"
21         android:layout_width="match_parent"
22         android:layout_height="match_parent"
23         app:defaultNavHost="true"
24         app:navGraph="@navigation/my_nav"
25         tools:layout_editor_absoluteX="1dp"
26         tools:layout_editor_absoluteY="1dp" />
27
28 </androidx.constraintlayout.widget.ConstraintLayout>
```

13) Agora voltamos pro xml do fragmento um e dois e acrescentemos um id na text view que colocamos lá



14) Agora você entra na classe e diz acrescenta a seguinte linha:



Sobre essa linha: você está setando um onClickListener no fragmento, e utilizando o id da action presente dentro do fragmento (basta verificar o text do xml do gráfico de navegação). você está setando essa action para esse listener. "Toda vez que você clicar nesse fragmento, executa esta action." A action no caso é sair do frag um e ir pro frag dois. Repetimos a mesma coisa no fragmento dois, mas retornando pro fragmento um.

15) Agora você vai em MainActivity.kt e implementa ambos o OnFragmentInteractionListener de ambas as classes de fragmentos

```
class MainActivity : AppCompatActivity(), FragOne.OnFragmentInteractionListener, FragTwo.OnFragmentInteractionListener {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
  
    override fun onFragmentInteraction(uri: Uri) {}  
}
```