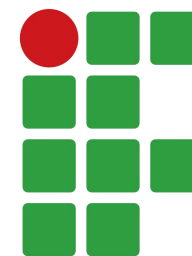


PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS

Curso de Engenharia de Software
Lucas Sampaio Leite



**INSTITUTO
FEDERAL**
Pernambuco

Fragments

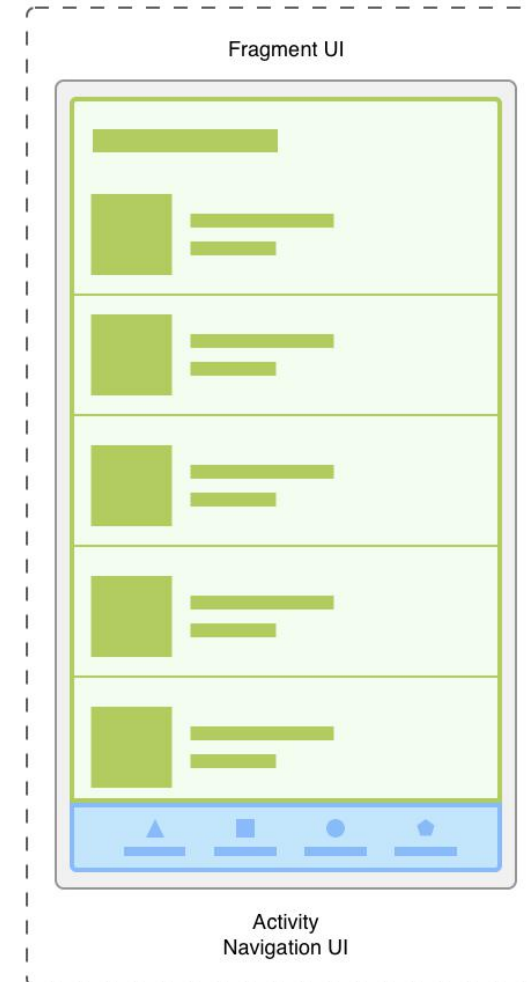
- Fragmentos (Fragments) são componentes modulares do Android que representam partes da interface e do comportamento de uma aplicação, permitindo criar estruturas mais flexíveis e reutilizáveis.
- Cada fragmento gerencia seu próprio layout, possui um ciclo de vida próprio e pode tratar eventos de interação do usuário.
- Os fragmentos não existem de forma independente: precisam ser hospedados por uma Activity ou por outro fragmento.

Fragments

Large Screen



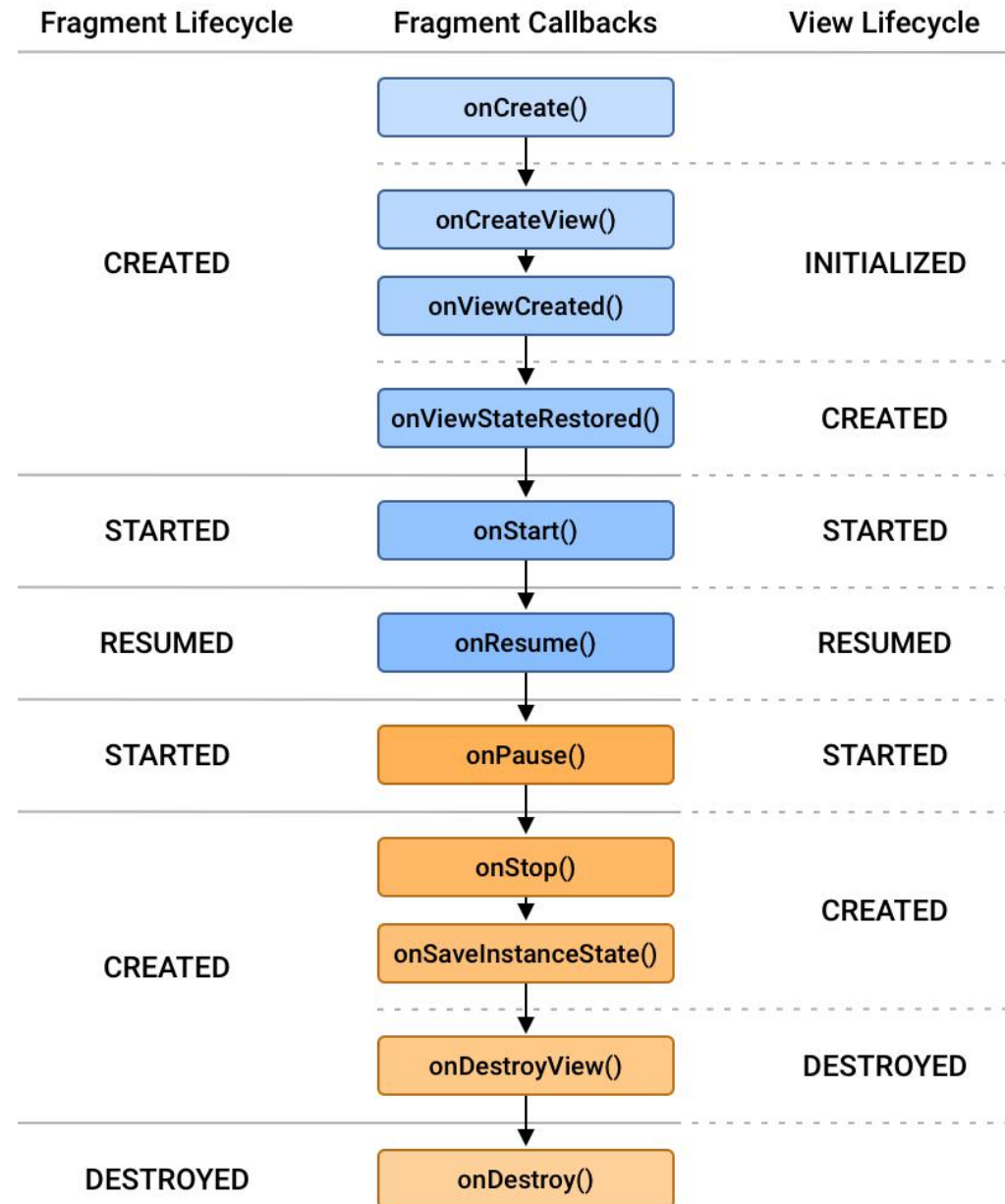
Small Screen



Fragments

- Fragmentos possuem um ciclo de vida próprio, relacionado ao da Activity que os hospeda, com métodos como `onCreate()`, `onCreateView()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroyView()` e `onDestroy()`.
- Fragmentos podem se comunicar entre si, diretamente ou por meio de componentes compartilhados, como a Activity permitindo a troca de dados e a coordenação de ações.

Fragments



Fragments

- Uma aplicação pode ser construída utilizando apenas Activities. No entanto, Fragments oferecem vantagens importantes em aplicações com interfaces mais complexas.
 - Reutilização de componentes: um Fragment pode ser utilizado em diferentes Activities sem precisar duplicar toda a implementação da interface.
 - Múltiplas áreas na mesma tela: é possível exibir uma lista de itens em uma região e os detalhes do item selecionado em outra, por exemplo, em tablets.
 - Navegação dentro da mesma Activity: o usuário pode alternar entre diferentes partes da interface sem que cada navegação precise abrir uma nova Activity.
 - Adaptação a diferentes telas: uma interface pode mostrar os Fragments separadamente em celulares e lado a lado em telas maiores.

Fragments

New Project

×

Empty Activity

Create a new empty activity with Jetpack Compose

Name

FragmentsApplication

Package name

leite.sampaio.lucas.fragmentsapplication

Save location

/home/lucas/AndroidStudioProjects/FragmentsApplication2

Minimum SDK

API 24 ("Nougat"; Android 7.0)

ⓘ Your app will run on approximately 99,2% of devices.

[Help me choose](#)

Build configuration language ⓘ

Kotlin DSL (build.gradle.kts) [Recommended]

Previous

Next

Cancel

Finish

Fragments

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import leite.sampaio.lucas.fragmentsapplication.databinding.ActivityMainBinding

1 Usage
class MainActivity : AppCompatActivity() {

    2 Usages
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}
```



Fragments

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import leite.sampaio.lucas.fragmentsapplication.databinding.ActivityMainBinding

1 Usage
class MainActivity : AppCompatActivity() {

    2 Usages
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}
```

A classe AppCompatActivity oferece compatibilidade com diferentes versões do Android e permite utilizar o supportFragmentManager para gerenciar fragments.

Fragments

- build.gradle.kts (Module)

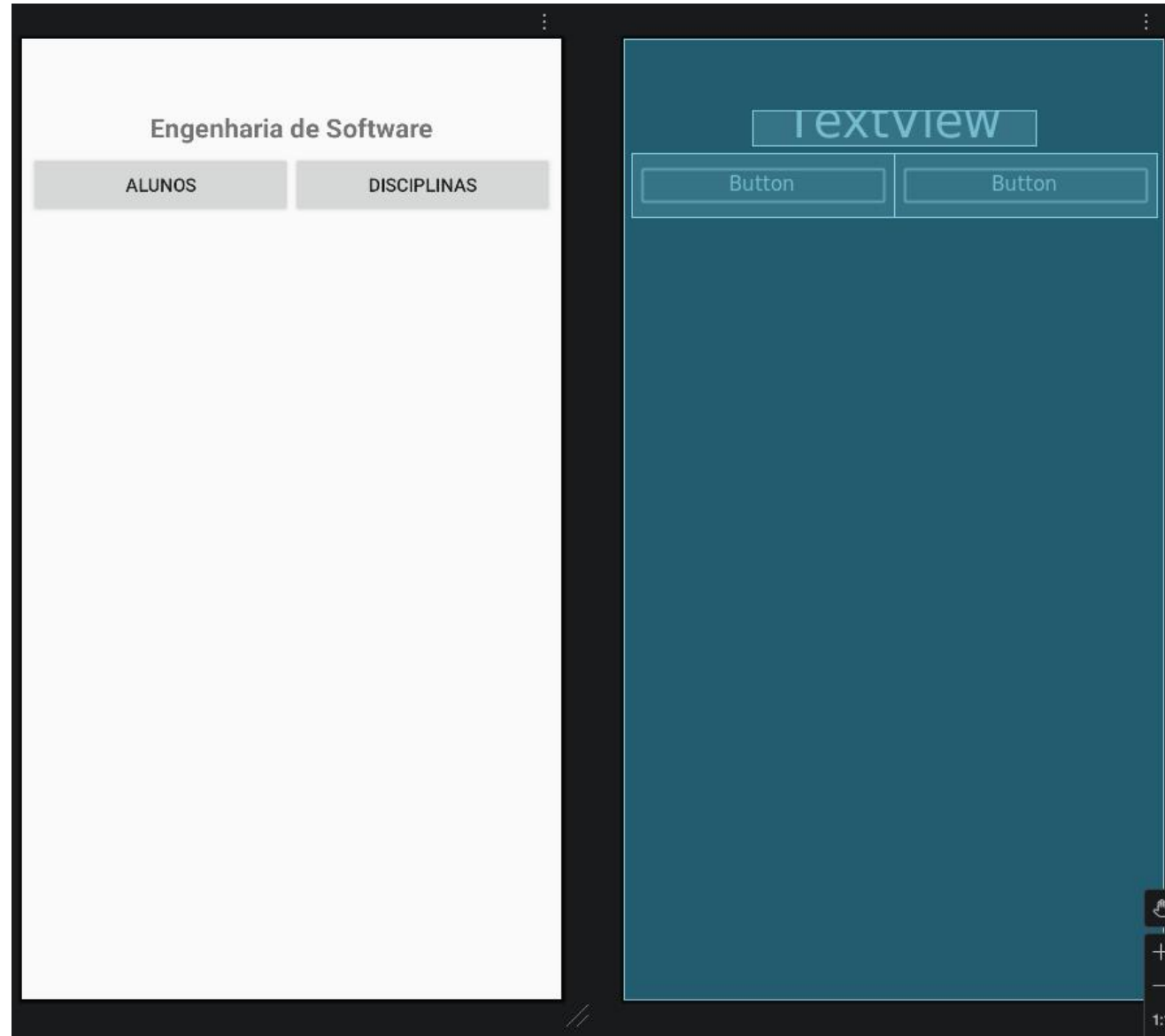
```
viewBinding{  
    enable = true  
}
```

```
dependencies {  
    implementation(libs.androidx.appcompat)
```

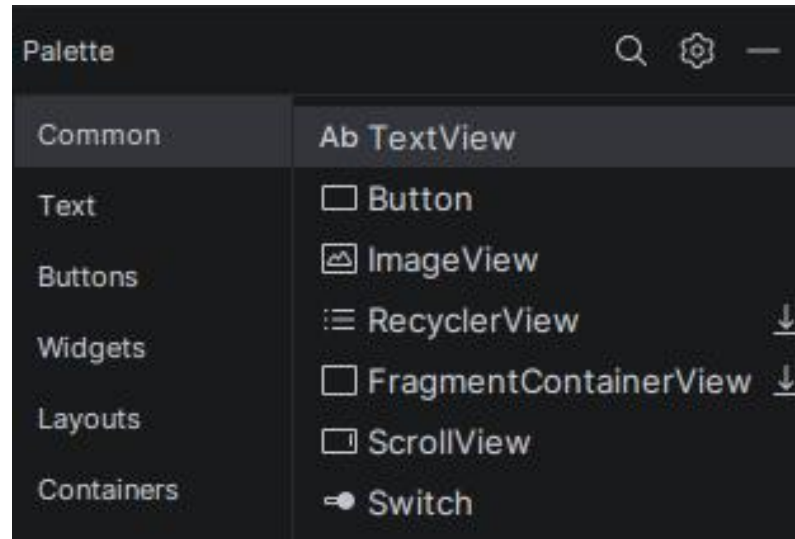
- res/values/themes.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <style name="Theme.FragmentsApplication" parent="Theme.AppCompat.DayNight.NoActionBar"/>  
</resources>
```

Fragments



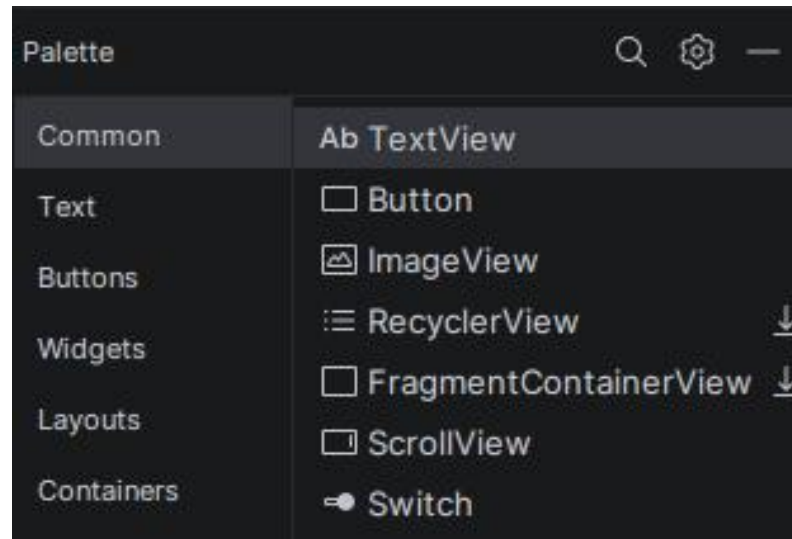
Fragments



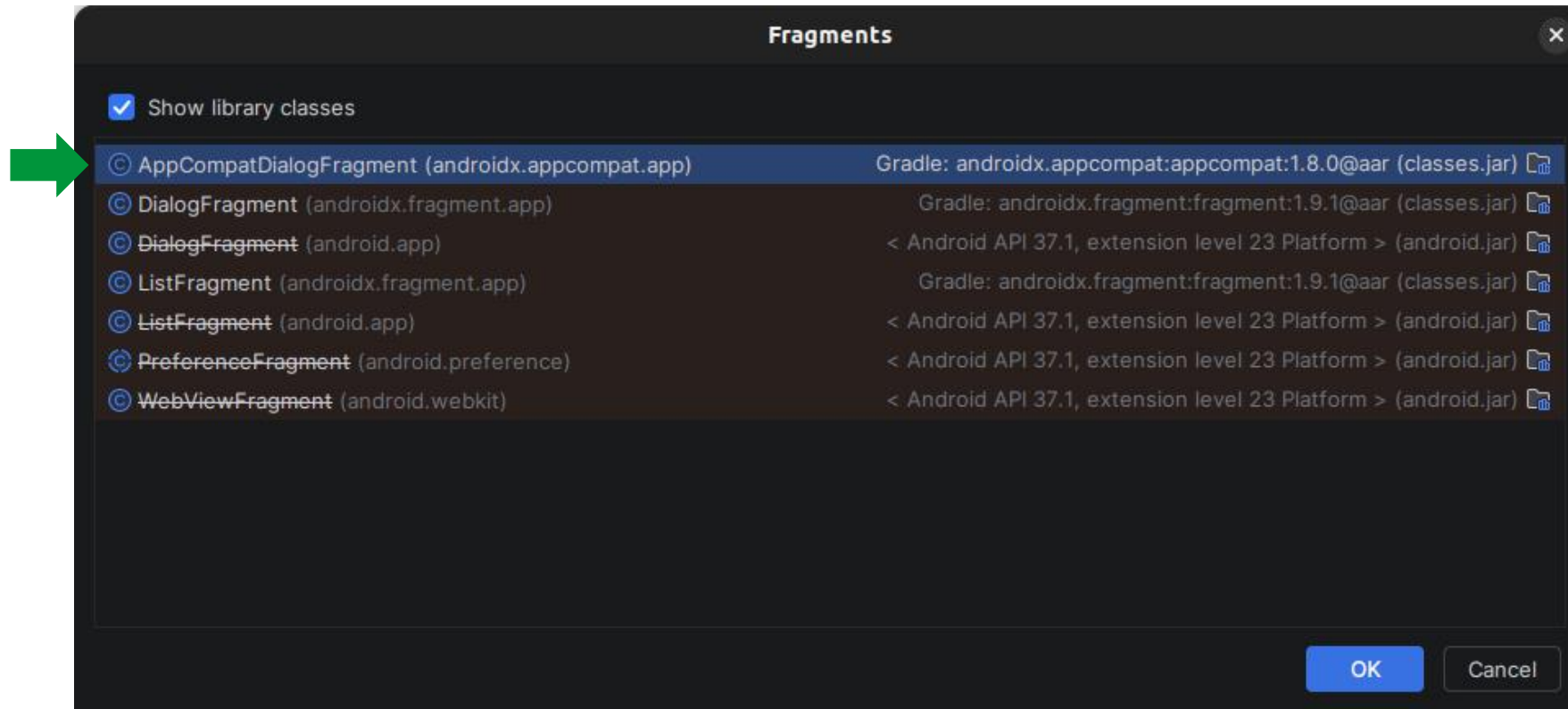
```
dependencies {  
    implementation(libs.androidx.fragment)
```



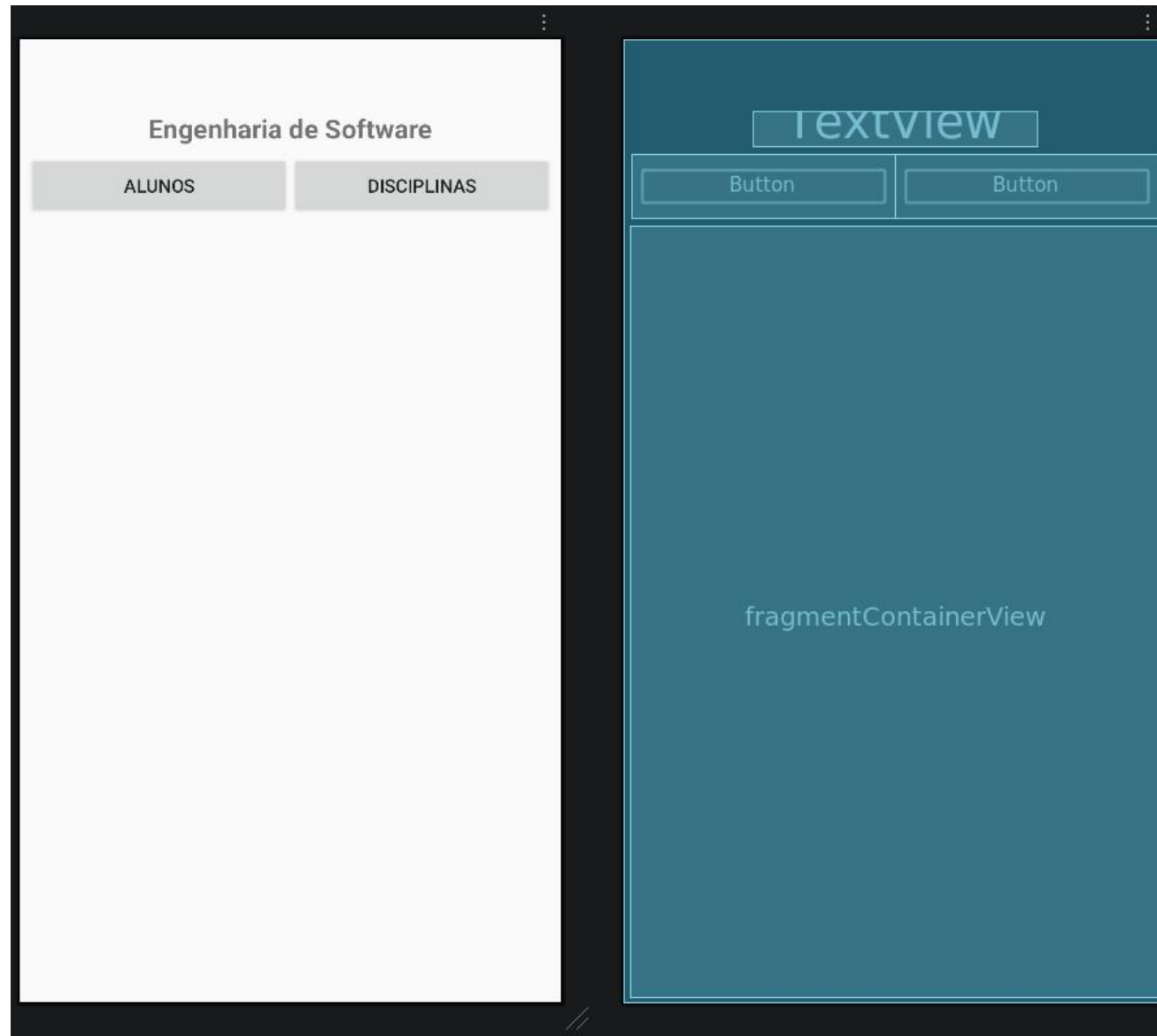
Fragments



```
dependencies {  
    implementation(libs.androidx.fragment)
```

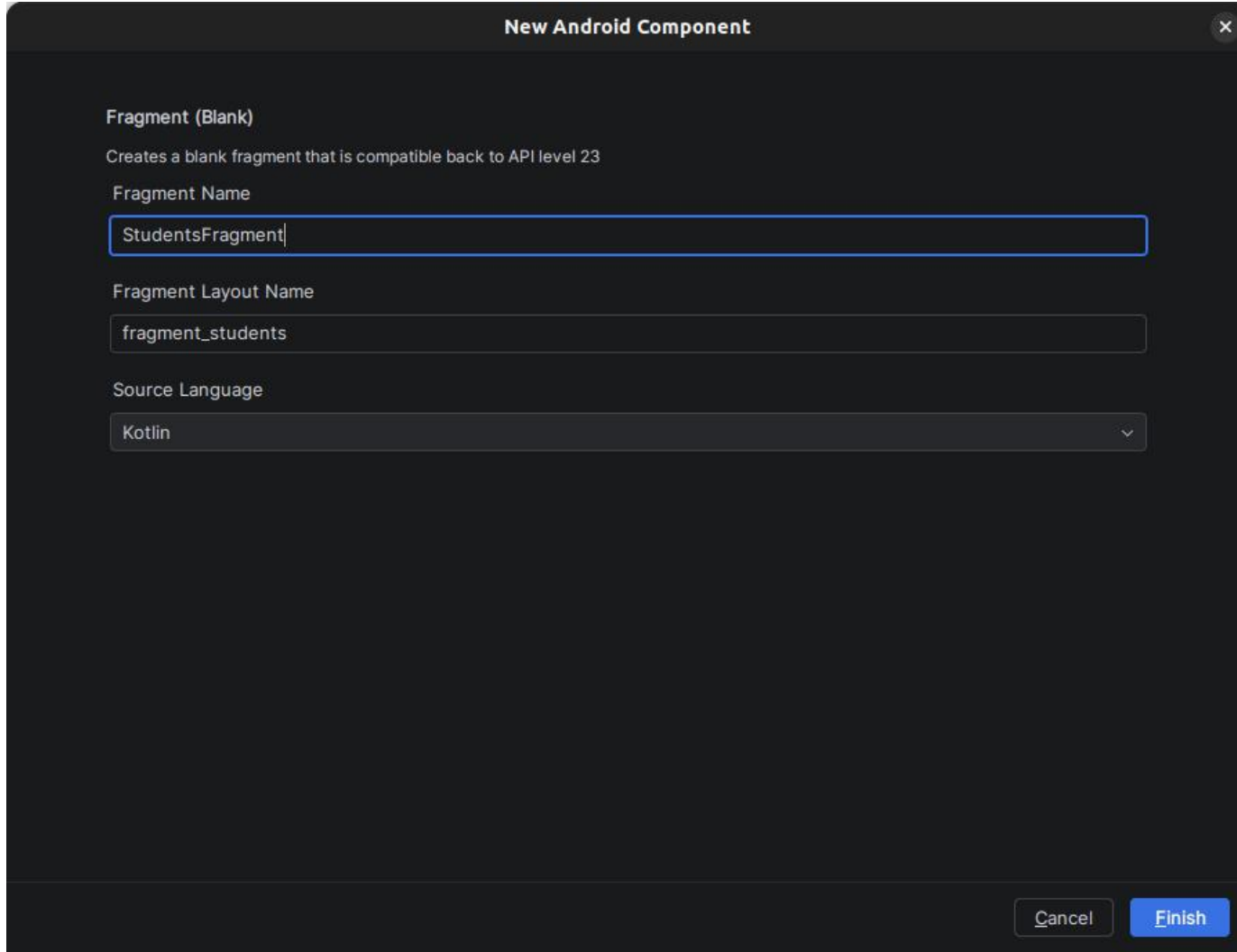


Fragments



Fragments

- packageName → new → Fragment → Fragment (Blank)



The screenshot shows the 'New Android Component' dialog box in Android Studio. The title bar reads 'New Android Component'. The main content area is titled 'Fragment (Blank)' and includes the description 'Creates a blank fragment that is compatible back to API level 23'. There are three input fields: 'Fragment Name' with the text 'StudentsFragment', 'Fragment Layout Name' with the text 'fragment_students', and 'Source Language' with a dropdown menu showing 'Kotlin'. At the bottom right, there are two buttons: 'Cancel' and 'Finish'.

New Android Component

Fragment (Blank)

Creates a blank fragment that is compatible back to API level 23

Fragment Name

StudentsFragment

Fragment Layout Name

fragment_students

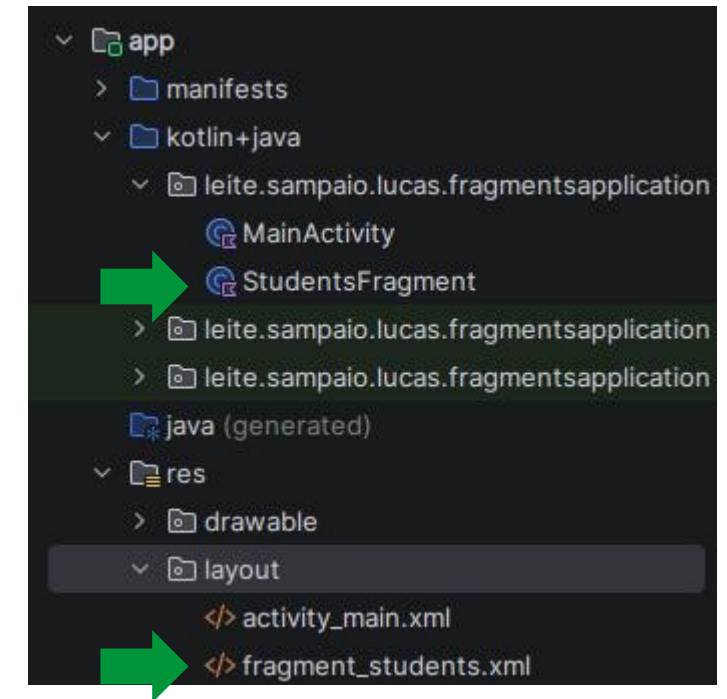
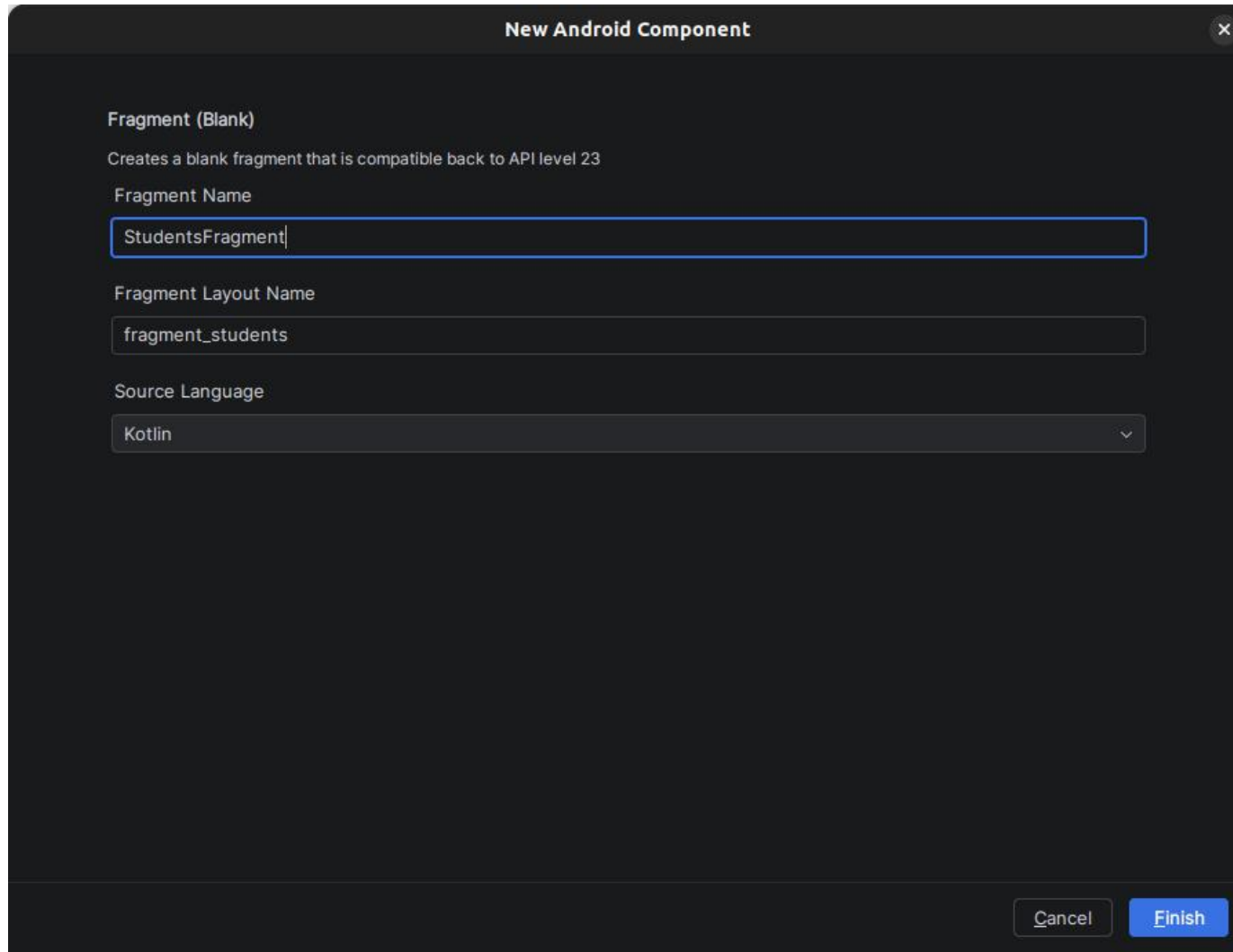
Source Language

Kotlin

Cancel Finish

Fragments

- packageName → new → Fragment → Fragment (Blank)



Fragments

- Refatorando o Fragmento:

```
class StudentsFragment : Fragment() {  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        return inflater.inflate(resource = R.layout.fragment_students, root = container,  
                                attachToRoot = false)  
    }  
  
}
```

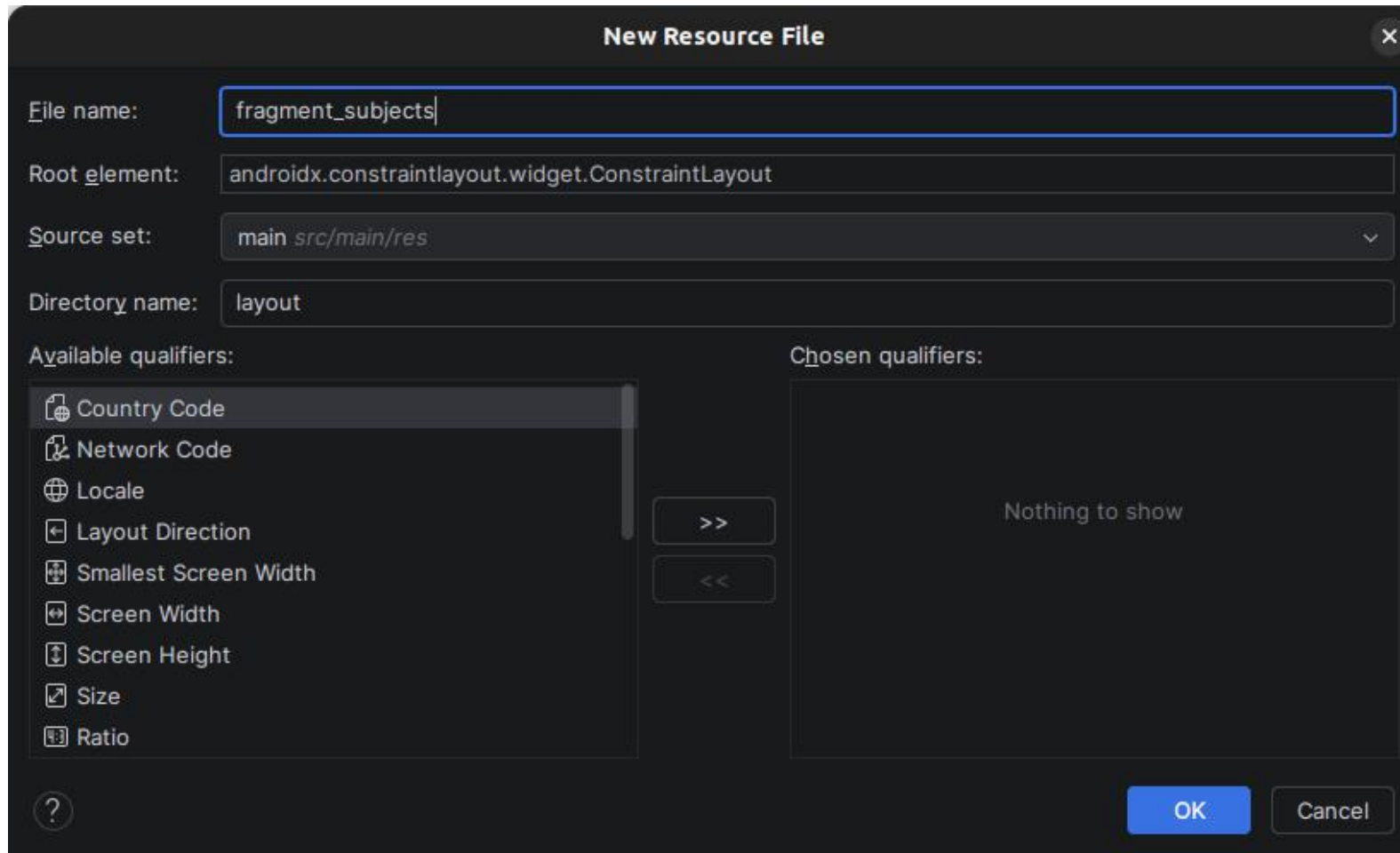
Fragments

- Habilitando o viewBinding:

```
class StudentsFragment : Fragment() {  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        ➡ val binding = FragmentStudentsBinding.inflate(  
            inflater, parent = container, attachToParent = false)  
        return binding.root  
    }  
  
}
```

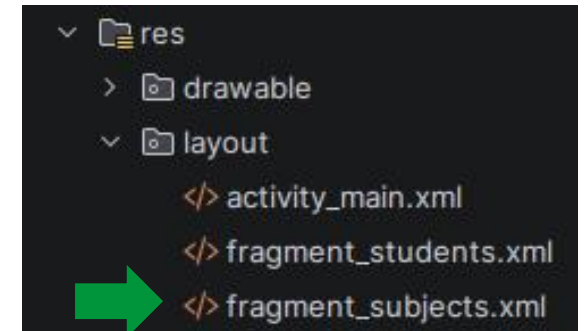
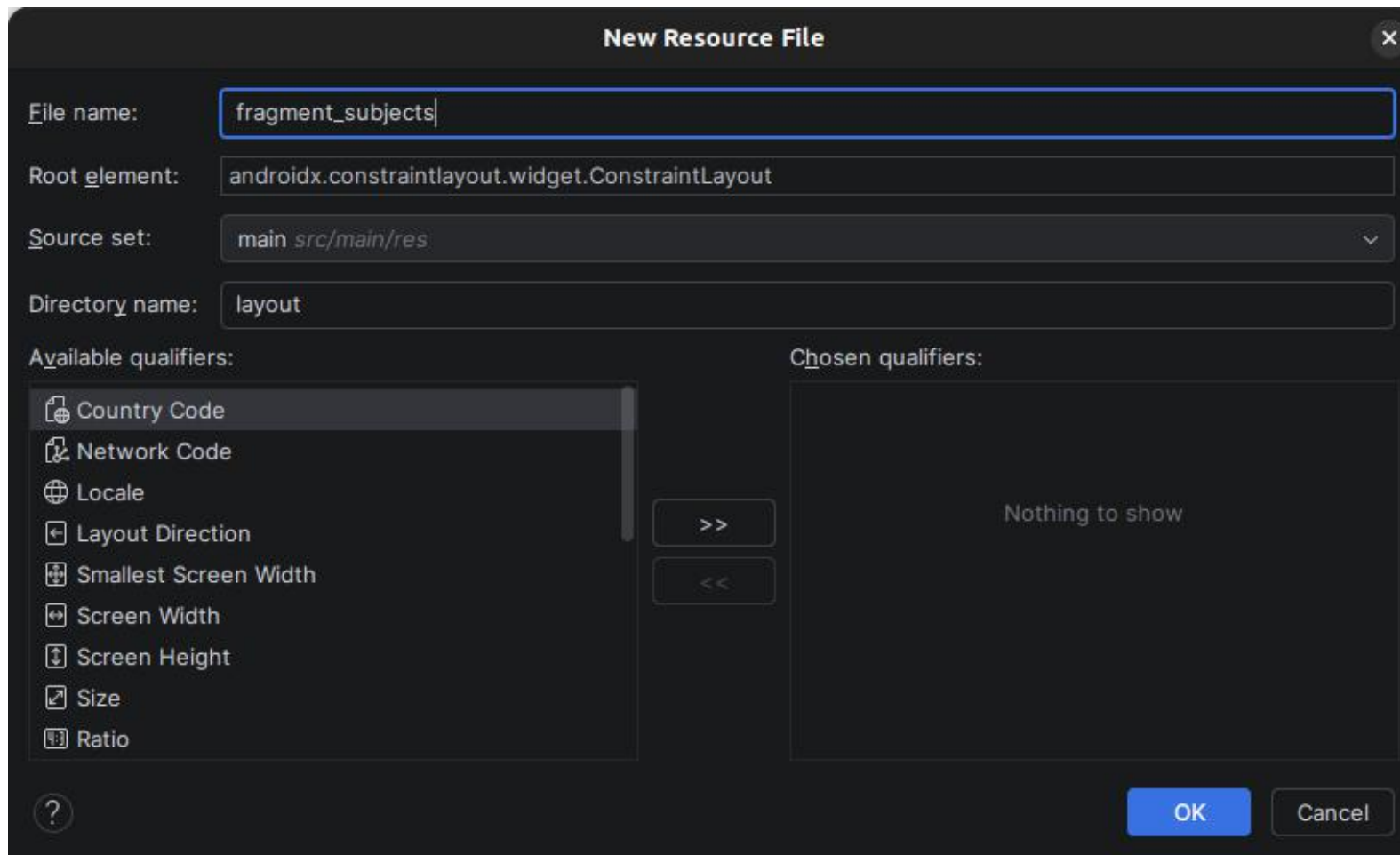
Fragments

- Criando o fragmento de disciplinas manualmente:
 - res/layout → new → Layout Resource File



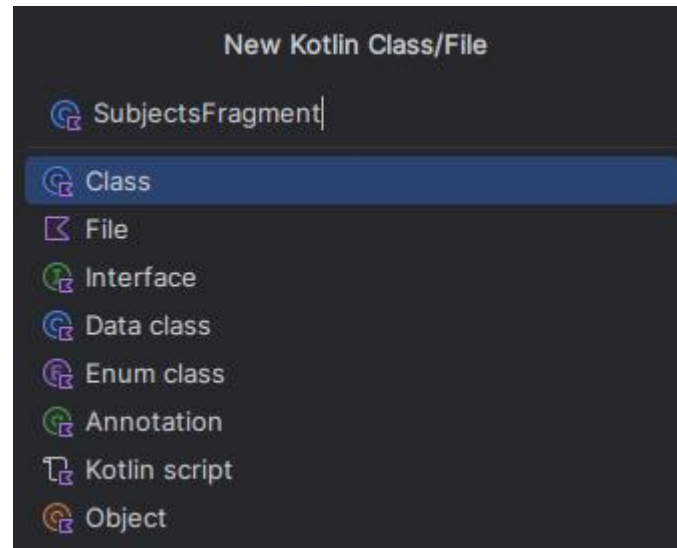
Fragments

- Criando o fragmento de disciplinas manualmente:
 - res/layout → new → Layout Resource File



Fragments

- Criando o fragmento de disciplinas manualmente:
 - packageName → new → Kotlin Class/File



```
class SubjectsFragment {  
}
```

Fragments

- Implementando SubjectsFragment:

```
class SubjectsFragment : Fragment() {  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        val binding = FragmentSubjectsBinding.inflate(  
            inflater, parent = container, attachToParent = false)  
        return binding.root  
    }  
  
}
```

Fragments

- Implementando MainActivity:

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    binding = ActivityMainBinding.inflate(layoutInflater)  
    setContentView(binding.root)  
  
    binding.btStudents.setOnClickListener {  
        ➡ showFragment(StudentsFragment())  
    }  
  
    binding.btSubjects.setOnClickListener {  
        ➡ showFragment(SubjectsFragment())  
    }  
}
```

Fragments

```
private fun showFragment(fragment: Fragment) {  
    supportFragmentManager  
        .beginTransaction()  
        .replace(containerViewId = binding.fragmentContainerView.id, fragment)  
        .commit()  
}
```

Fragments

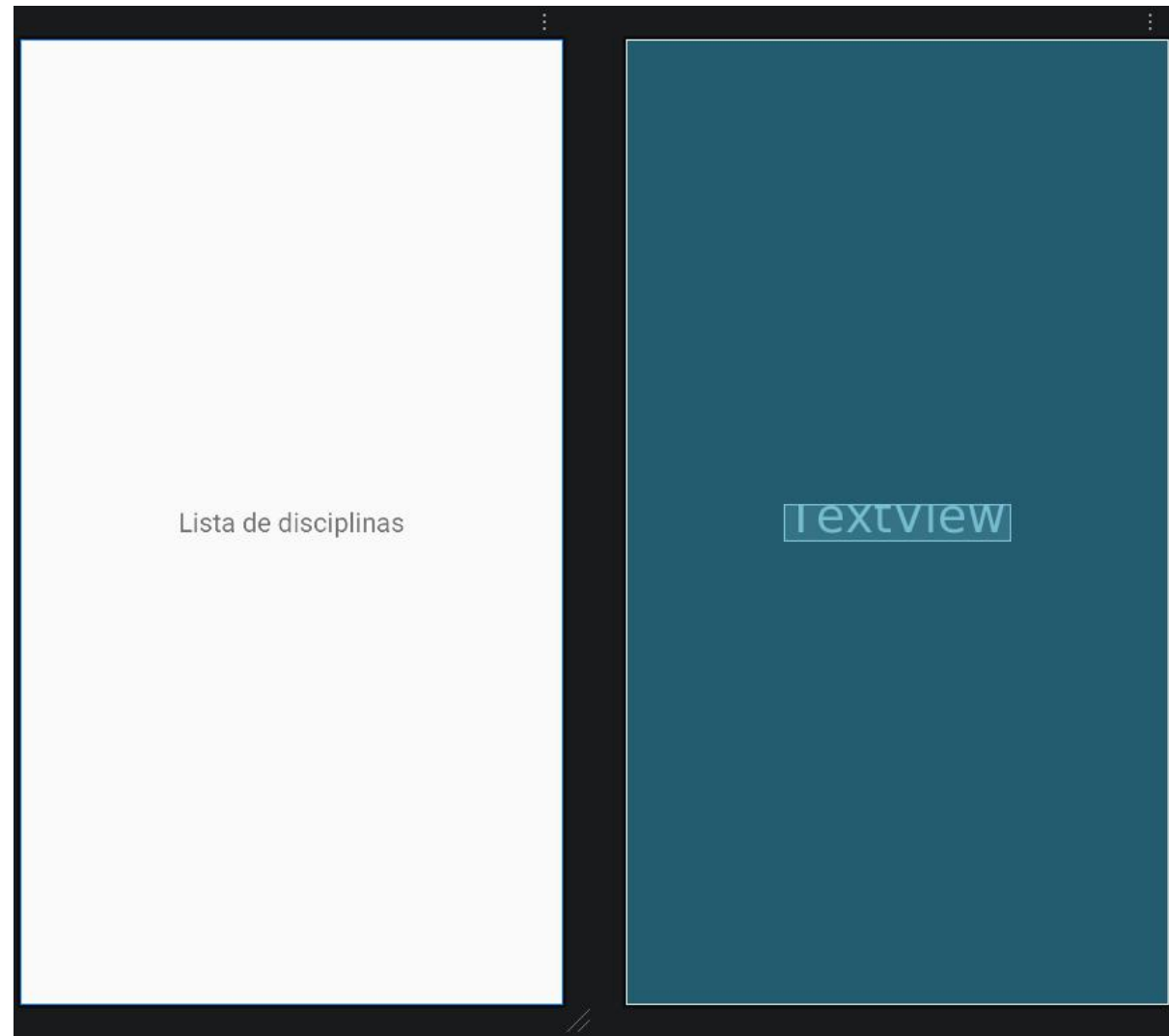
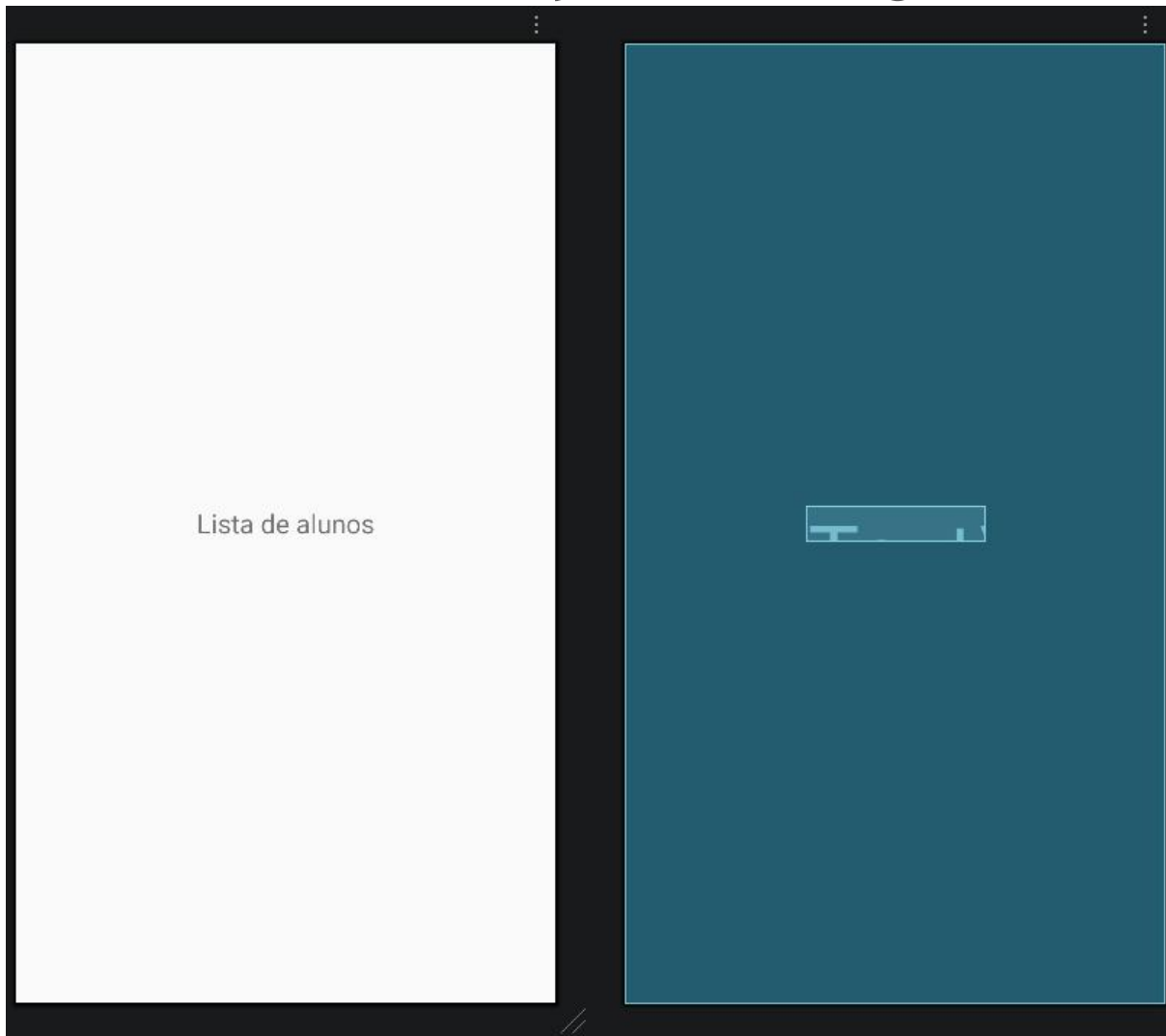
`supportFragmentManager` é um objeto disponibilizado pela classe `AppCompatActivity` que permite gerenciar os fragmentos associados a uma `Activity`, incluindo sua adição, substituição e remoção.

```
private fun showFragment(fragment: Fragment) {  
    supportFragmentManager  
        .beginTransaction()  
        .replace(containerViewId = binding.fragmentContainerView.id, fragment)  
        .commit()  
}
```

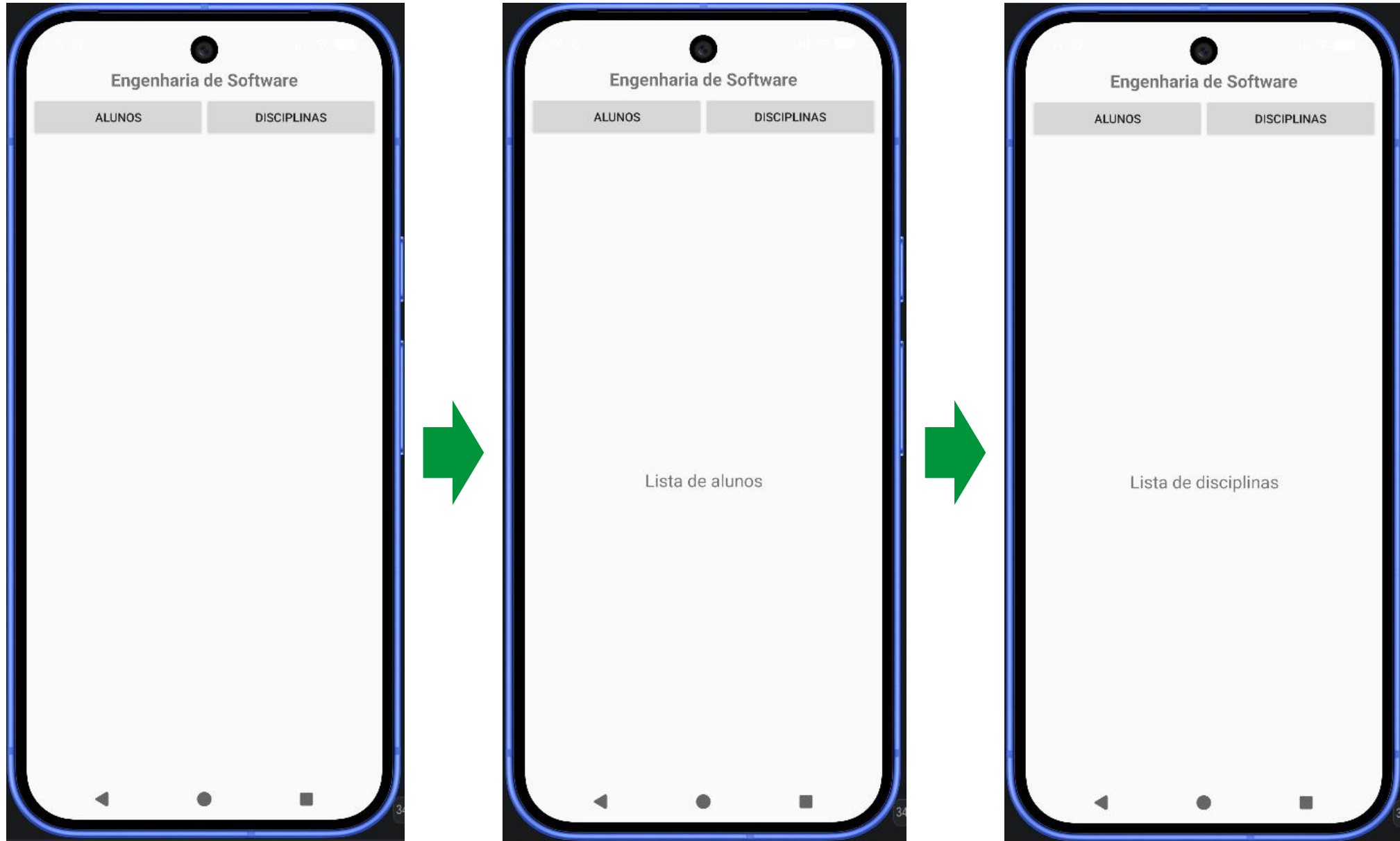


Fragments

- Alterando o layout dos fragmentos:

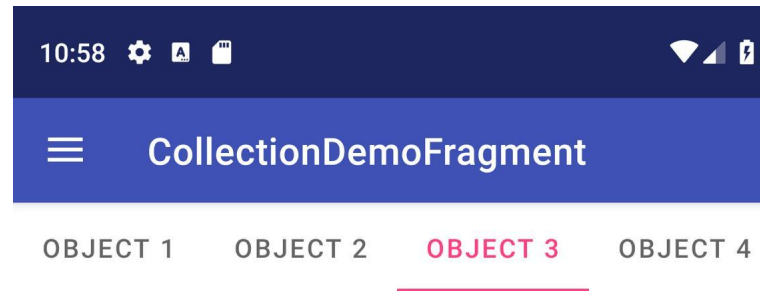


Fragments



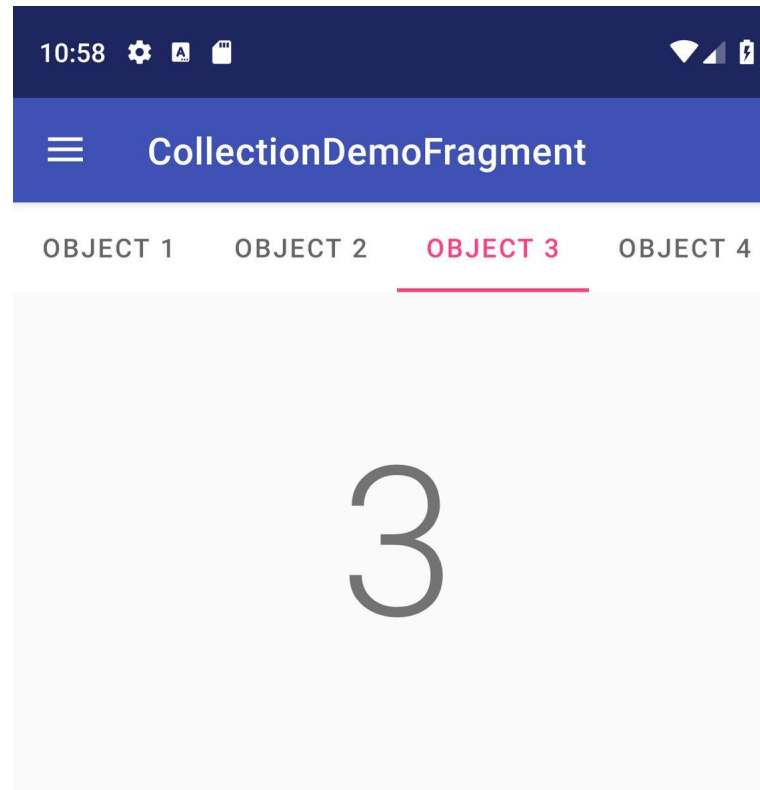
TabLayout

- O TabLayout é um componente visual do Android que organiza opções de navegação em abas, permitindo que o usuário selecione uma seção do aplicativo tocando na aba correspondente.



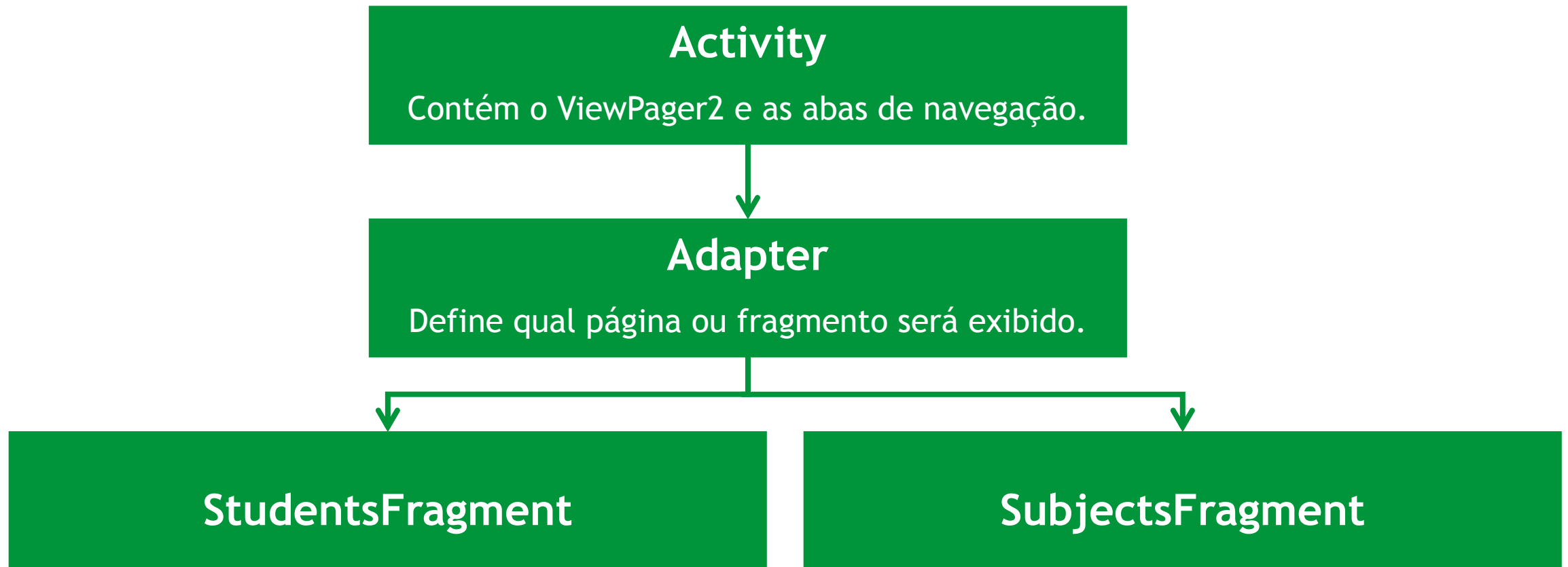
ViewPager2

- O ViewPager2 permite apresentar várias páginas em um mesmo espaço da interface, exibindo uma página por vez. O usuário pode alternar entre elas deslizando para a esquerda ou para a direita.

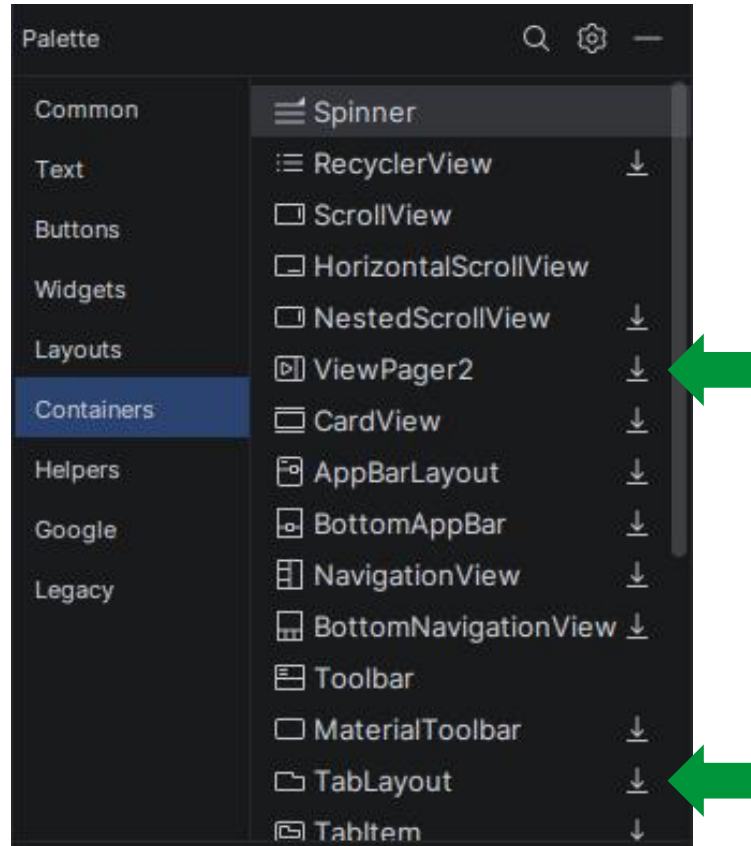


ViewPager2

- O ViewPager2 trabalha com um adaptador, responsável por fornecer as páginas que serão exibidas.



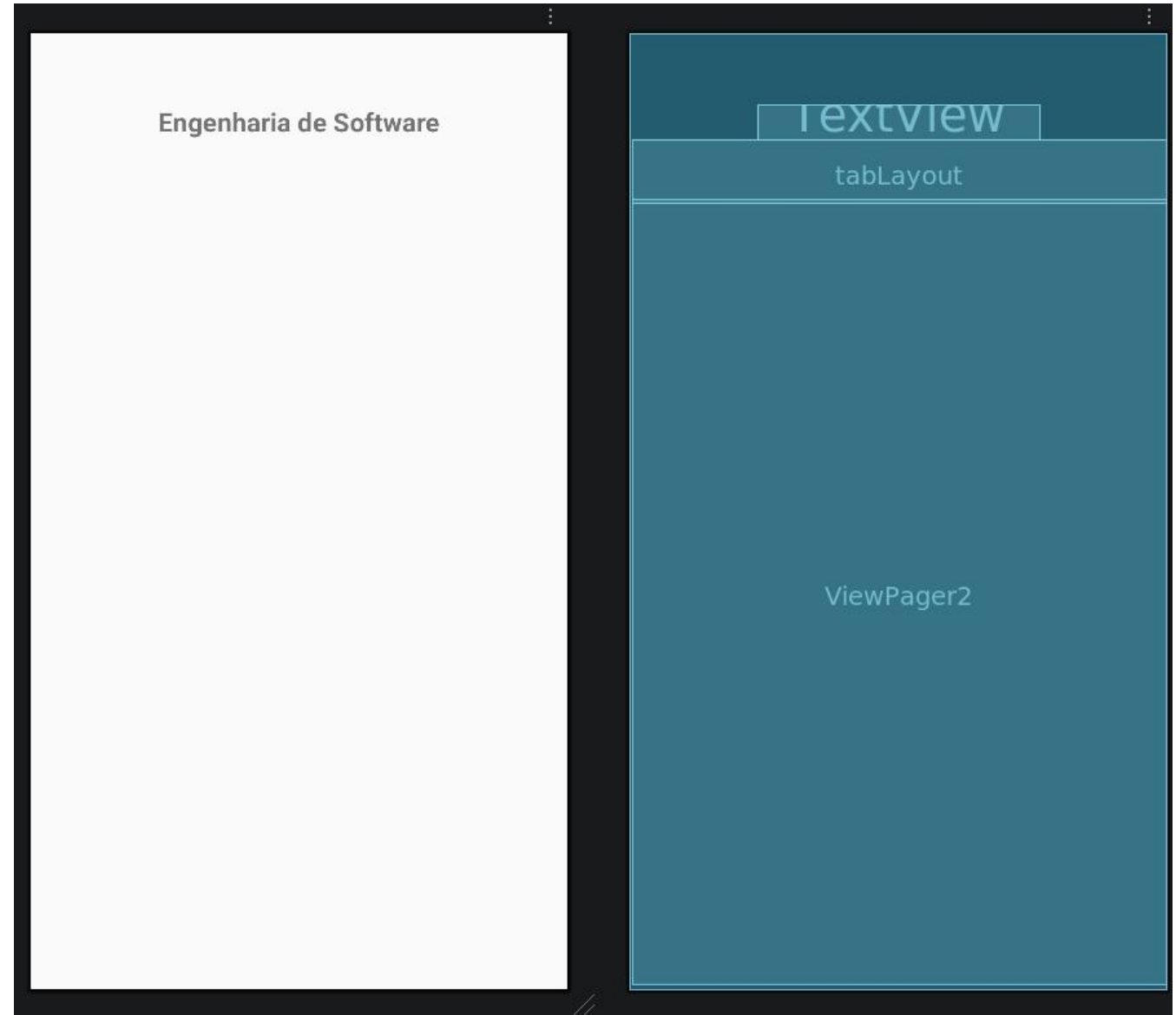
TabLayout + ViewPager2 + Fragments



```
dependencies {  
    implementation(libs.androidx.viewpager2)  
    implementation(libs.material)  
}
```

TabLayout + ViewPager2 + Fragments

- Modificando o layout da activity:



TabLayout + ViewPager2 + Fragments

- Implementando o Adapter:

```
class ScholarPagerAdapter(activity: MainActivity):  
    FragmentStateAdapter(fragmentActivity = activity) {  
  
    override fun getItemCount(): Int = 2  
  
    override fun createFragment(position: Int): Fragment {  
        return when (position) {  
            0 -> StudentsFragment()  
            1 -> SubjectsFragment()  
            else -> throw IllegalArgumentException(  
                "Posição inválida: $position"  
            )  
        }  
    }  
}
```

TabLayout + ViewPager2 + Fragments

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    binding = ActivityMainBinding.inflate(layoutInflater)  
    setContentView(binding.root)  
  
    binding.viewPager2.adapter = ScholarPagerAdapter(activity = this)  
  
    TabLayoutMediator(binding.tabLayout, viewPager = binding.viewPager2) { tab, position ->  
        tab.text = when (position) {  
            0 -> "Alunos"  
            1 -> "Disciplinas"  
            else -> throw IllegalArgumentException("Posição inválida: $position")  
        }  
    }.attach()  
}
```

TabLayout + ViewPager2 + Fragments

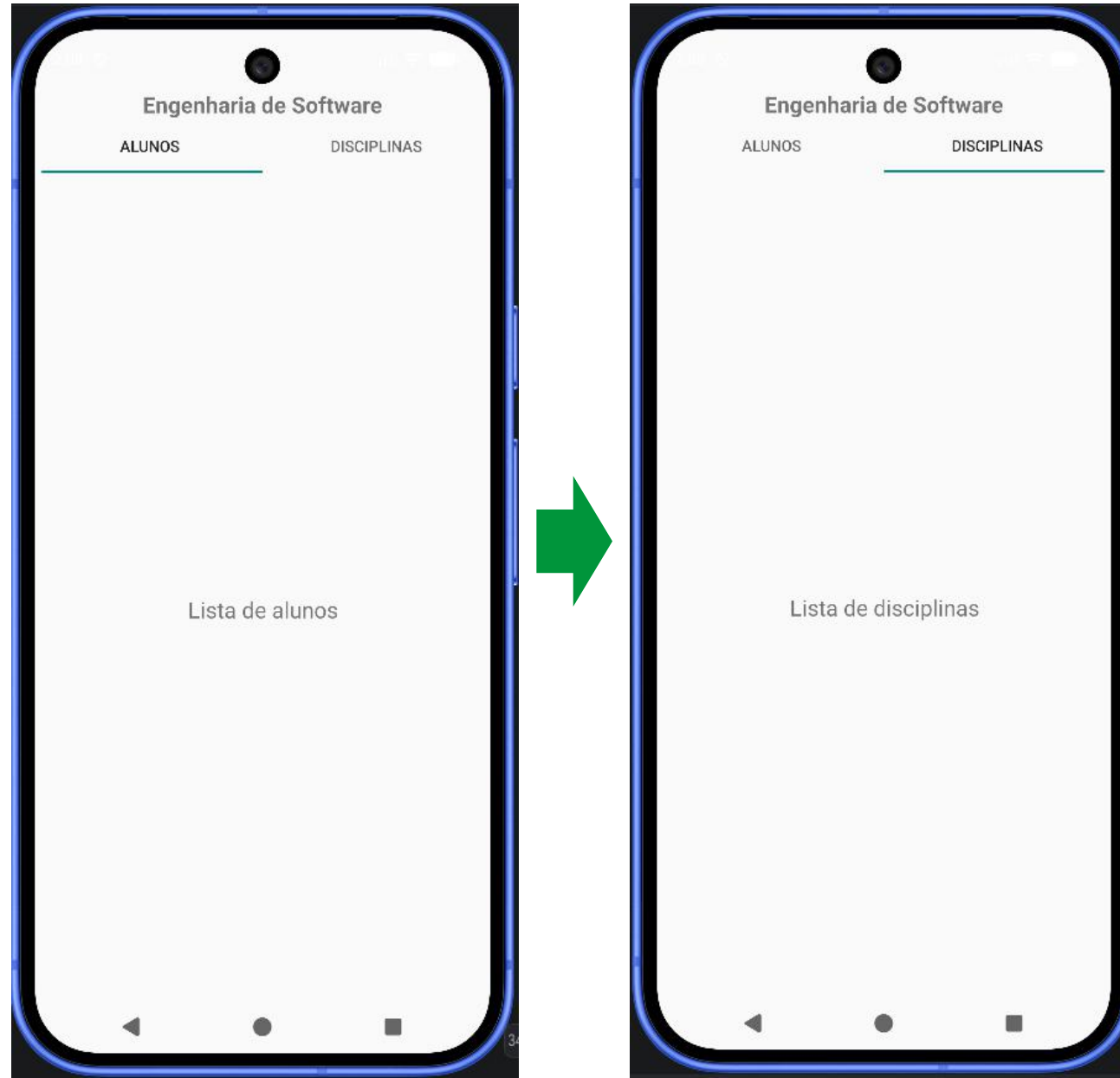
```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)
```

O TabLayoutMediator é uma classe da biblioteca Material Components do Android que conecta um TabLayout a um ViewPager2, mantendo os dois componentes sincronizados.

```
binding.viewPager2.adapter = ScholarPagerAdapter(activity = this)
```

```
TabLayoutMediator(binding.tabLayout, viewPager = binding.viewPager2) { tab, position ->  
    tab.text = when (position) {  
        0 -> "Alunos"  
        1 -> "Disciplinas"  
        else -> throw IllegalArgumentException("Posição inválida: $position")  
    }  
}.attach()  
}
```

Fragments



Exercícios

1. Na classe StudentsFragment, sobrescreva os seguintes métodos:
 - onCreate(), onCreateView(), onViewCreated(), onStart(), onResume(), onPause(), onStop(), onDestroyView(), onDestroy().
 - Em cada método, utilize Log.d() para registrar uma mensagem que identifique o fragmento e o método executado.
- Qual é a ordem dos métodos executados quando o fragmento é aberto pela primeira vez?
- O que acontece com o ciclo de vida quando o usuário alterna entre as abas Alunos e Disciplinas?
- Feche o aplicativo e abra-o novamente. Compare as mensagens registradas com as da primeira execução.

Exercícios

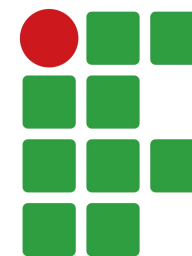
2. Pesquise e implemente a passagem de dados de uma Activity para um fragmento e a comunicação entre dois fragmentos, seguindo os requisitos:
- Passagem de dados da Activity para o fragmento: na MainActivity, declare uma variável contendo o nome do curso e passe esse valor para o StudentsFragment.
 - Exibição dos dados recebidos: no StudentsFragment, recupere o nome do curso recebido e exiba-o em um TextView.
 - Seleção de aluno: no StudentsFragment, disponibilize um botão que permita selecionar um aluno. Ao clicar nele, armazene o nome do aluno selecionado.
 - Compartilhamento de dados entre fragmentos: implemente uma forma de compartilhar o nome do aluno selecionado com o SubjectsFragment.
 - Exibição dos dados compartilhados: no SubjectsFragment, recupere o nome do aluno selecionado e exiba-o na interface.

Dúvidas



PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS

Curso de Engenharia de Software
Lucas Sampaio Leite



**INSTITUTO
FEDERAL**
Pernambuco