Aqui vamos criar uma interface a vamos navegar pelos fragmentos.

Já criamos o fragmente



**Agora para navegar sobre o FragmentContainerView vamos utilizar o**

fragmentManager e o .beginTrasaction(Inicar uma transação)

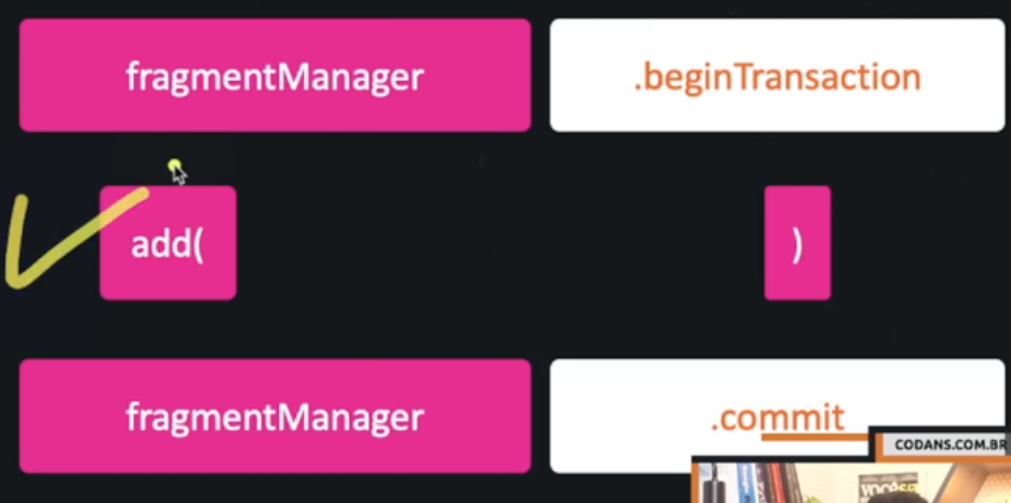
**Depois vamos utilizar o**

fragmentManager / .commit (Vai confirmar as alterações que fizemos)

**E aqui é onde fazemos realmente as alterações**

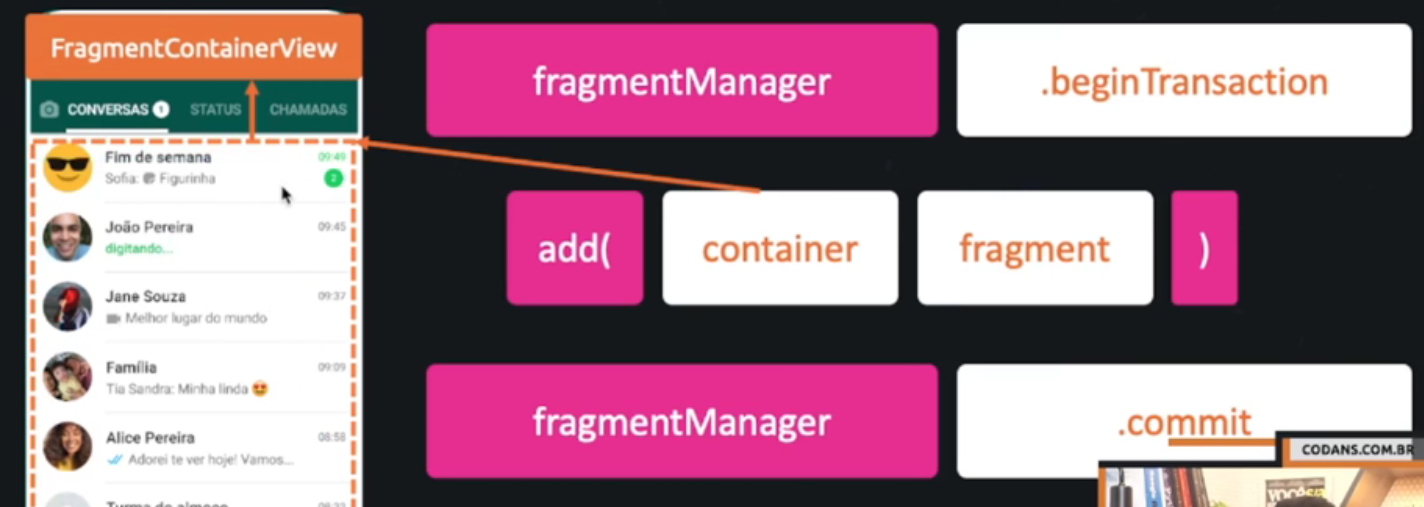
Add ( )

Ficando dessa forma



Ai dentro podemos fazer varias alterações, remover, adicionar...

Resumindo: .beginTransaction inicia a operação e o .commit ele confirma a alteração.



No add (vamos adicionar o container e o fragmente seria o fragmente\_conversas) que vamos colocar ali dentro.

# Mão na massa

Primeiro devemos deixar o nosso layout principal dessa forma

<androidx.fragment.app.FragmentContainerView  
 android:id="@+id/ *fragmentNaActivity* "  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/guideline"  
 tools:layout="@layout/fragment\_conversas"/>

Não podemos deixar o name ali porque vamos fazer de forma dinâmica, alterando os fragmente, conforme vamos clicando nos botões.

Agora quando o usuário clicar nos botões ele vai abrir um novo fragmente no local que indicamos

class MainActivity : AppCompatActivity() {  
  
 lateinit var btnConversas : Button  
 lateinit var btnStatus : Button  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 setContentView(R.layout.*activity\_main*)  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->** val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 **}** btnConversas = findViewById(R.id.*buttonConversas*)  
 btnStatus = findViewById(R.id.*buttonStatus*)  
  
  
 */\*\*  
 \* Quando clicar no botao quero abrir o fragment.  
 \*/* btnConversas.setOnClickListener **{** */\*\*  
 \* supportFragmentManager é um objeto que nos permite manipular os fragments  
 \* beginTransaction é ele que começa a transação ele inicializa as transações que voce pode fazer no fragment  
 \*  
 \*/* val fragmentManager = *supportFragmentManager*.beginTransaction()  
  
 */\*\*  
 \* Aqui podemos fazer diversas alterações no nosso fragment  
 \* A ALTERAÇÂO QUE ESTAMOS FAZENDO É: ADICIONAR ELE NA TELA.  
 \* Primeiro parametro é o container = objeto que vai ser exibido na tela  
 \* Segundo parametro é o objeto Fragment onde colocamos nossos codigos, reparar que precisamos instanciar ele()  
 \*/* fragmentManager.add( R.id.*fragmentNaActivity*, ConversasFragment() )  
  
 */\*\*  
 \* Aqui é onde vamos confirmar as alterações  
 \*/* fragmentManager.commit()  
 **}** */\*\*  
 \* Fazendo de outra forma para ser mais organizado e mais rapido  
 \*/* btnStatus.setOnClickListener **{** *supportFragmentManager*.beginTransaction().add(R.id.*fragmentNaActivity*, StatusFragment()).commit()  
 **}** }  
}

Atenção no método replace / add

Add = adicionar algo, ele vai adicionar por cima

Replace = ele vai substituir.

btnStatus.setOnClickListener **{** *supportFragmentManager*.beginTransaction().replace(R.id.*fragmentNaActivity*, StatusFragment()).commit()  
**}***/\*\*  
 \* Lembrar de uma coisa o metodo Add. ele vai adicionar um fragment, mas se ja existir um na tela, ele vai adicionar o outro por cima  
 \* Devemos sempre usar o replace para substituir o atual.  
 \*  
 \* ERRADO : supportFragmentManager.beginTransaction().add(R.id.fragmentNaActivity, StatusFragment()).commit()  
 \* CORRETO: supportFragmentManager.beginTransaction().\*replace\*(R.id.fragmentNaActivity, StatusFragment()).commit()  
 \*/*

# Removendo uma fragmente

*/\*\*  
 \* Agora imagina que queremos remover um fragment que não queremos mais utilizar  
 \* Para remover precisamos remover uma instancia especifica de um objeto  
 \* Porque precisamos instanciar um objeto dentro da variavel?  
 \*  
 \* Porque toda vez que instanciamos dentro do proprio codigo ele cria instancias diferentes  
 \* Imagina assim  
 \* jogador1 = ConversasFragment  
 \* jogador2 = ConversasFragment  
 \* Repara que ele vai criando varios objetos, então não vamos saber de fato qual o jogador que queremos remover  
 \* Pra isso vamos criar uma variavel e instanciar apenas 1 objeto dentro.  
 \* Criamos la em cima para abrir a conversa val conversasFragment = ConversasFragment()  
 \* Agora esse botao aqui vai ser responsavel por destruir esse objeto que esta dentro da variavel.  
 \*/*

val conversasFragment = ConversasFragment()  
btnConversas.setOnClickListener **{**val fragmentManager = *supportFragmentManager*.beginTransaction()

fragmentManager.replace( R.id.*fragmentNaActivity*, conversasFragment )  
fragmentManager.commit()  
**}**

btnRemoverFragment.setOnClickListener **{** *supportFragmentManager*.beginTransaction()  
 .remove(conversasFragment)  
 .commit()  
**}**