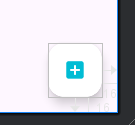
Float action button

Float = flutuar

Nessa aula vamos aprender como clicar em um button flutuante e abrir outros botões

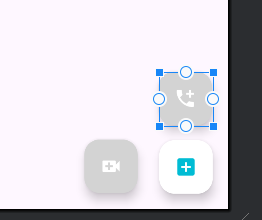


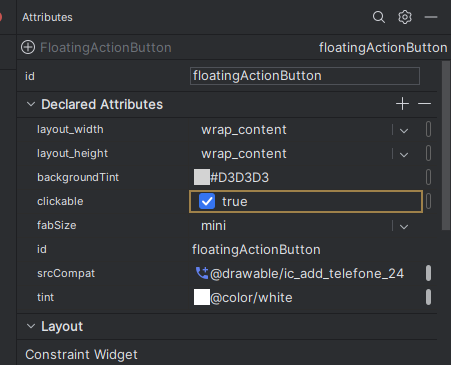
Vamos imaginar que quando a gente clicar nesse botão, vamos ter 2 novos.

1 para adicionar contado

2 para adicionar um novo vídeo

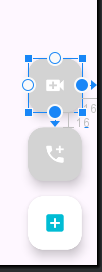
Tudo que foi modifcado no nosso floating



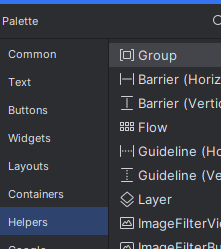


fabSize é o tamanho dele, como ele vai ser um sub botão, usamos o mini

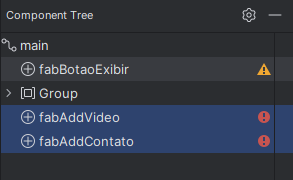
Ficou dessa maneira agora



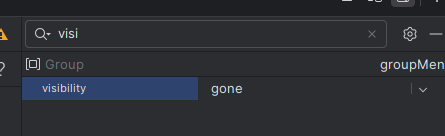
Agora vamos precisar usar o Group, ele vai agrupar os 2 itens, e vai ficar como se os 2 itens estivessem o mesmo conceito, assim quando a gente chamar no código, vamos por o botão para abrir e vai abrir os 2 subs botões



Agora vamos adicionar o group na tela e vamos adicionar nossos subs botões dentro dele



Agora damos um ID para o group e chamamos ele de group menu, agora podemos controlar a visibilidade dele, e quando o usuário clicar no botaoExibir vamos colocar ele visível de novo.



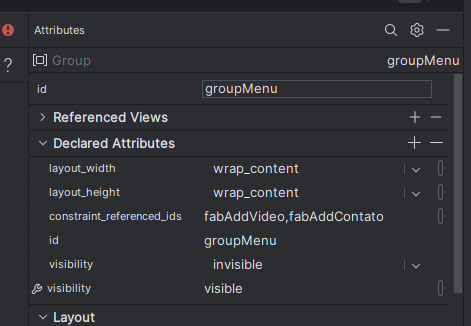
Temos o visible, invisible, gone

Visible = visível

Invisible = o usuário não vê, mas ele continua ali, ocupando espaço

Gone = ele deixa de existir, não ocupa espaço na tela.

Todas as configurações do meu group



Estamos associando esse group a 2 itens, quando a gente precisar exibir ou modificar algo, podemos fazer apenas no groupMenu e ele vai replicar todos os comandos para todos que estão dentro dele.

# Agora vamos configurar o ViewBinding para poder modificar esses dados

Primeiro vamos configurar no modulo do app

buildFeatures **{** viewBinding = true  
**}**

Agora vamos usar nosso XML no atributo

private val binding by *lazy* **{** ActivityFloatActionBottonBinding.inflate(*layoutInflater*)  
**}**

Terminando o resto da classe

class FloatActionBotton : AppCompatActivity() {  
  
 private val binding by *lazy* **{** // Lazy: carregamento preguiçoso ele so vai carregar quando for chamado  
 ActivityFloatActionBottonBinding.inflate(*layoutInflater*)  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 setContentView(binding.*root*)  
 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  
 **}** *with*(binding) **{** // Com whith ele ja vai saber que estamos falando do objeto binding, então nao precisamos repetir.  
 // Se nao tivesse o with, teriamos que ficar repetindo o objeto na frente de cada fabBotao por exemplo  
 // ficando assim ; binding.fabBotaoExibir  
 // binding.groupMenu  
 // o whith sabe que estamos tratando do binding   
  
 fabBotaoExibir.setOnClickListener **{** // Vamos verificar se o item esta visivel  
 // Se tiver visivel, vamos esconder  
 // Se tiver invisivel vamos tornar visivel  
  
 if (groupMenu.*visibility* == View.*VISIBLE*) {  
 groupMenu.*visibility* = View.*INVISIBLE* }else {  
 groupMenu.*visibility* = View.*VISIBLE* }  
 **}  
  
 }** }  
}