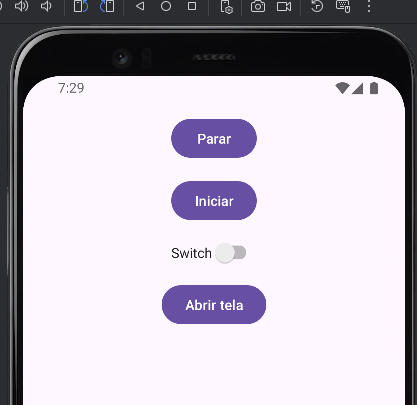
# 1 primeiro configuramos o viewBinding

# 2 configuramos o projeto e os objetos



# 3 criando um thread do 0

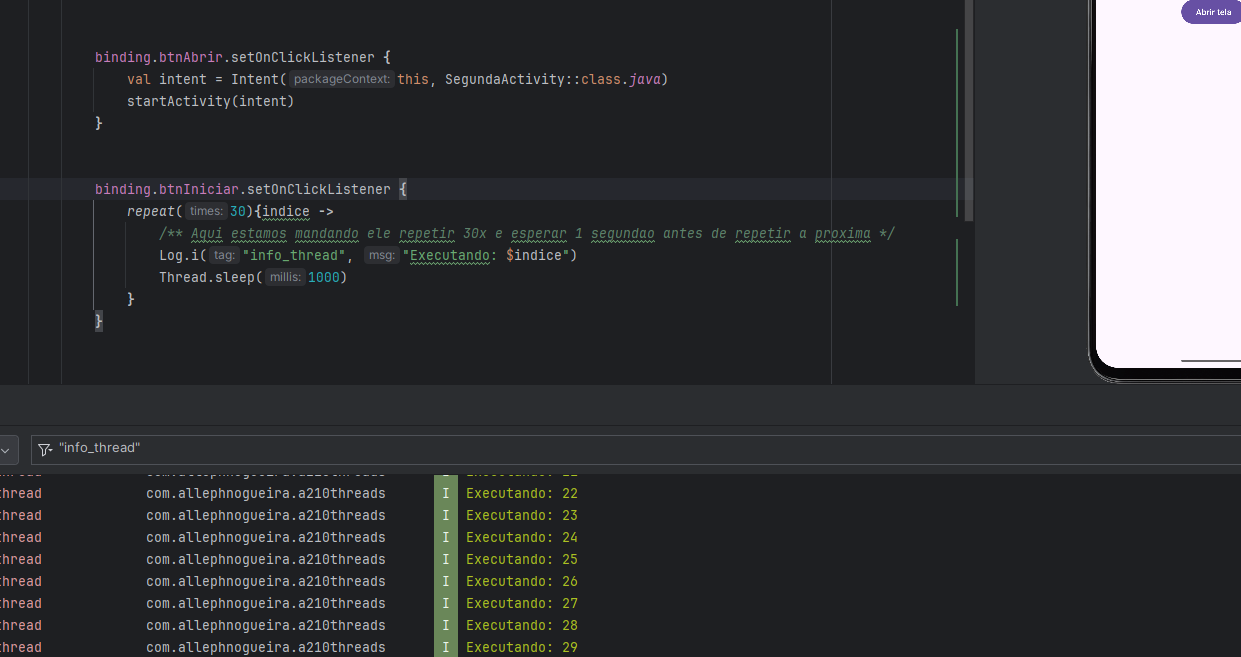
Run = executar é la que vamos por nosso código

Start = iniciar

## PRATICA:

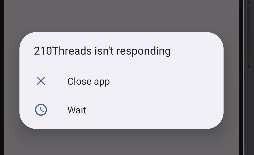
Vamos criar nossa classe.

Criamos nossa classe dentro do MainActivity mesmo, criamos uma inner class MinhaThread



Reparar que aqui estamos fazendo nossa thread principal executar um bloco de repetição, e quando tentamos clicar em outro botão, temos um problema, porque nossa thread esta ocupada tentando fazer sua tarefa primeiro.

E se você tentar executar outro botão você tem esse problema



Resumindo: Nossa thread principal ela só serve para atualizar a tela com os dados, os dados devem ser processados em thread que nos mesmo vamos criar.

Imagina os postos de GNV, imagina que estamos carregando os postos, estamos processando em um thread separado, então quando a gente terminar o processamento, vamos avisar para nossa thread principal que terminamos aí sim ela vai exibir os dados na tela.

Então agora vamos fazer diferente, vamos colocar para executar esse mesmo código na nossa thread e vamos testar para ver se vai travar também...

## Usando minha própria Thread

binding.btnIniciar.setOnClickListener **{** */\*\* Aqui estamos encadeando metodo, estamos instanciando e ja usando seu metodo  
 \* Reparar que o metodo start ele vem de herança da classe Thread()  
 \*/***MinhaThread().start()**  
**}**

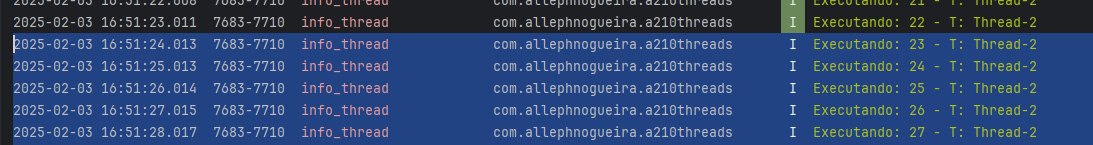
inner class MinhaThread : Thread() {  
 override fun run() {  
 super.run()  
  
 *repeat*(30)**{**indice **->** */\*\* Aqui estamos mandando ele repetir 30x e esperar 1 segundao antes de repetir a proxima \*/* Log.i("info\_thread", "Executando: $indice")  
 sleep(1000)  
 **}** }  
}

Agora sim quando clicamos no botão, não sentimos mais a tela travando e sim podemos acessar outros botões, porque a thread principal ficou apenas para exibir os dados e criamos uma thread por fora para processar os dados.

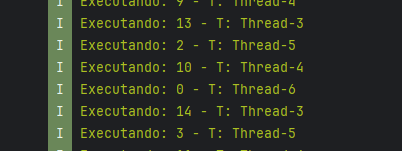
## Descobrindo qual thread esta sendo processada

inner class MinhaThread : Thread() {  
 override fun run() {  
 super.run()  
  
 *repeat*(30)**{**indice **->** */\*\* Aqui estamos mandando ele repetir 30x e esperar 1 segundao antes de repetir a proxima  
 \* Metodo Thread.CurrentThread().nome = Cada Thread tem um nome e ai vamos saber qual thread esta processando esse bloco de codigo \*/* Log.i("info\_thread", "Executando: $indice - T: ${Thread.currentThread().*name*}")  
 sleep(1000)  
 **}** }  
}

Nesse caso a thread processada é a 2



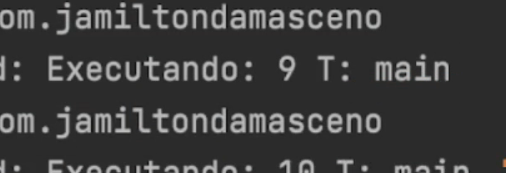
Outra coisa importante, você clica no botão de iniciar ele esta executando a thread 2, mas se você clicar novamente ele vai executar a thread 3, se você clicar novamente ele vai começar a thread4 e assim por diante.



Cada vez que clicamos em iniciar ele instancia e começa um nova thread.

## Usando a thread principal

Atenção pegamos apenas o código da nossa thread de processamento e jogamos para processar na principal.



Podemos utilizar o mesmo código para o teste e repara qual a thread esta sendo usada.

Main = thread principal.