Nesse exemplo vamos usar uma aula anterior.

Primeiro vamos em modulo do app e vamos adicionar o viewBinding la.

1 Ai vamos fazer o carregamento “preguiçoso”

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

Atençao: Repara que estamos usando a activity principal e não o layout que montamos o nosso modelo(recyclerView)

2 Vamos também definir nosso root

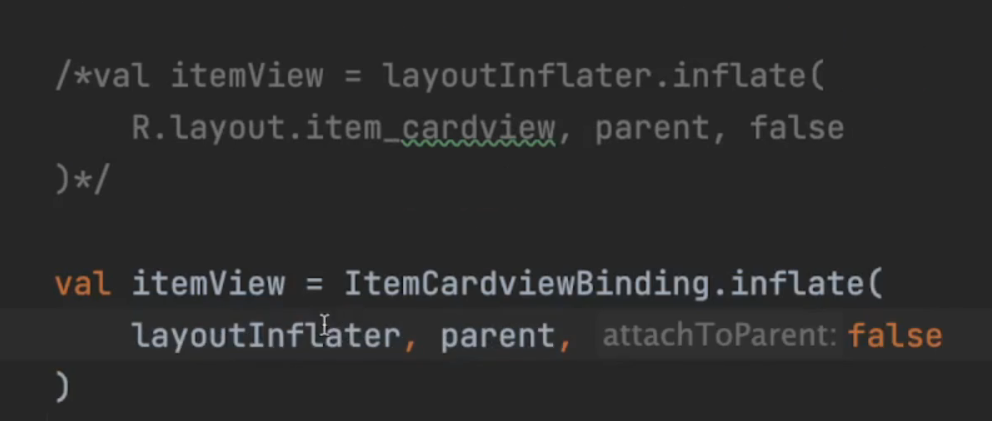
setContentView( binding.*root* )

Oque seria o root? Seria o layout que a variável binding esta tomando conta.

3 Atençao, agora não vamos mais usar o findByID e sim usar o binding.rvLista... por exemplo.

4 agora vamos para nosso adapter

O antes e depois da maneira certa



Repara que antes a gente usava o layout, igual la dentro do onCreate, agora a gente tem a classe de cada layout

ItemCardViewBinding... é um layout, mas o viewBinding transforma em uma classe.

Dentro do método vamos, inflar o layout, o parent

(Porque estamos fazendo assim? Porque fica diferente da classe original )

Aqui estamos pegando um layout que o viewBinding transformou em classe

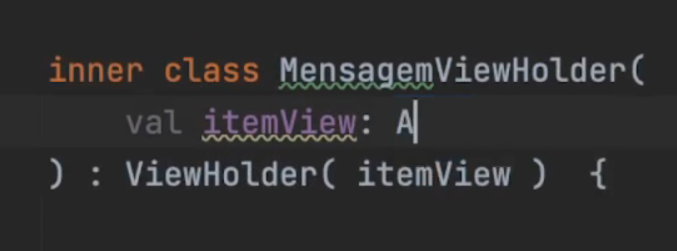
private val binding by *lazy* **{** ActivityRecyclerviewBinding.inflate( *layoutInflater* )  
**}**

Aqui estamos usando apenas um parâmetro porque vamos inflar o layout inteiro.

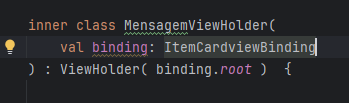
Já na parte de cima estamos usando o inflar, parent, false

Porque vamos inflar somente uma parte desse layout.

5 no viewHolder recebíamos um item do tipo View e agora vamos passar e receber uma classe do tipo binding.



Ficando dessa forma



De resto fica quase tudo igual, apenas onde estávamos procurando com o findById agora vamos utilizar o binding.

Veja o código e qualquer coisa retorne para a aula de RecyclerView