

## MensagemAdapter – ANDROID

```
class MensagemAdapter(private val lista: List<String>) :  
RecyclerView.Adapter<MensagemAdapter.MensagemViewHolder>() {  
  
    inner class MensagemViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
        // ... (código para acessar views dentro do item da lista)  
    }  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MensagemViewHolder {  
        val inflater = LayoutInflater.from(parent.context)  
        val itemView = inflater.inflate(R.layout.item_lista, parent, false)  
        return MensagemViewHolder(itemView)  
    }  
  
    override fun getItemCount(): Int {  
        return lista.size  
    }  
  
    override fun onBindViewHolder(holder: MensagemViewHolder, position: Int) {  
        TODO("Not yet implemented")  
    }  
}
```

### 1. class MensagemAdapter(...) : RecyclerView.Adapter<...>():

Define a classe MensagemAdapter, que herda de RecyclerView.Adapter. Isso significa que ela será responsável por fornecer dados e gerenciar as views para um RecyclerView.

O construtor recebe uma lista de strings (lista) que conterá as mensagens a serem exibidas.

<MensagemAdapter.MensagemViewHolder> especifica o tipo de ViewHolder que este adapter usará.

### 2. inner class MensagemViewHolder(...) : RecyclerView.ViewHolder(...):

Define a classe MensagemViewHolder, que herda de RecyclerView.ViewHolder.

O ViewHolder representa um único item na lista do RecyclerView.

Ele recebe uma itemView (a view que representa o layout do item) no construtor.

Você deve adicionar código aqui para acessar as views dentro do itemView usando findViewById ou View Binding, para que você possa atualizar os dados do item na tela.

### 3. override fun onCreateViewHolder(...):

Este método é chamado pelo RecyclerView para criar um novo ViewHolder quando necessário.

LayoutInflater é usado para inflar o layout do item (R.layout.item\_lista).

parent é o ViewGroup ao qual o novo item será anexado (o próprio RecyclerView).

false indica que o item inflado não deve ser anexado ao parent imediatamente, pois o RecyclerView fará isso mais tarde.

Um novo MensagemViewHolder é criado e retornado.

### 4. override fun getItemCount():

Este método retorna o número total de itens na lista de mensagens.

## 5. override fun onBindViewHolder(...):

Este método é chamado pelo RecyclerView para vincular os dados de um item a um ViewHolder. holder é o ViewHolder que precisa ser atualizado.

position é a posição do item na lista de mensagens.

Você deve implementar a lógica aqui para obter os dados da mensagem na posição position e atualizar as views dentro do holder com esses dados.

Em resumo:

O RecyclerView solicita um novo item chamando onCreateViewHolder.

onCreateViewHolder infla o layout do item e cria um ViewHolder.

O RecyclerView solicita a vinculação de dados chamando onBindViewHolder.

onBindViewHolder obtém os dados do item e os define nas views do ViewHolder.

O item é exibido na tela.

Próximos passos:

Implemente a lógica em onBindViewHolder para atualizar as views do item com os dados da mensagem.

Adicione código ao MensagemViewHolder para acessar as views dentro do layout do item.

Associe o MensagemAdapter ao seu RecyclerView na sua Activity ou Fragment.