

Aula 05 - 01/07/2021 - Adaptador e RecyclerView

INF4 - PDM

Resumo da aula: Esta aula aborda o sobre adaptadores e componentes de lista em Android.

Um resumo sobre esses componentes pode ser acessado em:

- [doc] Resumo sobre RecyclerView: [link](#).
- [slide] Slides sobre Adaptador e RecyclerView: [link](#).

Roteiro

Parte 1 (finalizar em até 60 min).

1. Ver dúvidas dos alunos sobre o exercício.
 - a. Falar sobre o gabarito em vídeo.
2. Abrir os slides e retomar o esquema
3. Criação de Projeto de Activity
4. **Atividade 01:** Copiar os códigos passo-a-passo:
 - a. Tela (Layout de modelo de linha).
 - b. Modelo de linha (holder)
 - c. Adaptador
 - d. Activity
5. Falar sobre evento, mostrar a diferença entre as duas formas
6. **Atividade 02:** Copiar eventos:
 - a. Eventos do mais fácil
7. Ver se todos os alunos conseguem executar a app.

Parte 2 (40 min):

1. Propor exercício em sala de aula: se der tempo
 - a. Pedir para alterar a tela, adicionar um botão e adicionar um EditText
2. Modificar para receber uma classe ao invés de Strings.
3. Explicar exercícios para fazer em casa.
 - a. Exercício resolvido: Lista de compras com alteração de valor
 - b. Exercício valendo nota: pegar um código e alterar ele para fazer uma calculadora.
 - i. EditText, operação, Botão e RecyclerView
 - ii. É possível trocar o valor na linha.
 - iii. Desafio, usar a Forma de evento 2.

https://drive.google.com/file/d/1uV7gcR7stJ3ziB3vv5NSaUNle_Fyokkh/view

Interface: MeuEventoDeClickDaLista

No adaptador Colar isso:

```
// declarar os eventos que vou usar na minha lista
public MeuEventoDeClickDaLista<String> eventoExcluirLinha = null;
public MeuEventoDeClickDaLista<String> eventoDuplicarLinha = null;
public MeuEventoDeClickDaLista<String> eventoClicarLinha = null;
```

colar no onBindView:

```
// adicionar eventos para interagir com a linha
if(eventoExcluirLinha != null){
    modeloDeLinha.iconeDelete.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // executar o evento
        eventoExcluirLinha.onItemClick(contenido);
    }
});
}
if(eventoDuplicarLinha != null){
    modeloDeLinha.iconeAdd.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // executar o evento
        eventoDuplicarLinha.onItemClick(contenido);
    }
});
}
if(eventoClicarLinha != null){
    modeloDeLinha.itemView.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // executar o evento
        eventoClicarLinha.onItemClick(contenido);
    }
});
}
```

Colar no onCreate do MainActivity

```
adaptador.eventoExcluirLinha = new
MeuEventoDeClickDaLista<String>() {
    public void onItemClick(String contenidoDaLinha) {
        // excluir item do adaptador
        adaptador.getDados().remove(contenidoDaLinha);
    }
}
```

```

        // excluir do banco de dados (veremos no futuro!) uuuuu
        // informar a tela sobre a alteração
        adaptador.notifyDataSetChanged();
    }
};
adaptador.eventoDuplicarLinha = new
MeuEventoDeClickDaLista<String>() {
    public void onItemClick(String conteudoDaLinha) {
        // duplica item do adaptador
        adaptador.getDados().add(conteudoDaLinha);
        // insere no banco de dados (veremos no futuro!) uuuuu
        // informar a tela sobre a alteração
        adaptador.notifyDataSetChanged();
    }
};
adaptador.eventoClicarLinha = new
MeuEventoDeClickDaLista<String>() {
    public void onItemClick(String conteudoDaLinha) {
        Toast.makeText(getBaseContext(),
            "Conteúdo: "+conteudoDaLinha,
            Toast.LENGTH_SHORT).show();
    }
};

```

No onCreateViewHolder do adaptador. Colar:

```

// adicionar eventos para interagir com a linha
if(eventoExcluirLinha != null){
    holder.iconeDelete.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String conteudo = getDados().get(holder.posicao);
            // executar o evento
            eventoExcluirLinha.onItemClick(conteudo);
        }
    });
}

```

Colocar no fim o onBindViewHolder

```

// atualizo a posição de quem está sendo exibido
modeloDeLinha.posicao = position;

```