

****Exercício: Aplicativo de Lista de Pedidos com Banco de Dados usando o padrão MVC****

Neste exercício, você será desafiado a desenvolver um aplicativo Android que permite aos clientes fazerem pedidos e armazena esses pedidos em um banco de dados SQLite local. O aplicativo utilizará o padrão de arquitetura MVC (Model-View-Controller) para organizar o código de forma mais eficiente.

****Requisitos:****

1. Android Studio instalado e configurado.
2. Conhecimentos básicos em Java e desenvolvimento de aplicativos Android.

****Descrição:****

Seu aplicativo deve conter as seguintes funcionalidades:

1. **Tela de Pedidos:**

- Na tela inicial, o cliente encontrará uma lista de itens disponíveis para pedir.
- Cada item deve ser exibido com um nome, preço e um botão "Adicionar ao Pedido" (Button) que permitirá ao cliente adicionar o item ao pedido.

2. **Lista de Pedidos:**

- Abaixo da lista de itens disponíveis, haverá um componente RecyclerView que exibirá todos os itens adicionados pelo cliente ao pedido.
- Cada item na lista de pedidos deve conter o nome do item, a quantidade e o preço total (quantidade x preço unitário).

3. **Armazenamento dos Pedidos:**

- Todos os pedidos feitos pelo cliente devem ser armazenados em um banco de dados SQLite local.
- O aplicativo deve ser capaz de criar, ler e excluir pedidos do banco de dados.

****Instruções:****

1. Crie um novo projeto no Android Studio com o nome "Aplicativo de Pedidos MVC".

2. Utilize o padrão MVC para organizar o código do aplicativo, criando classes separadas para Model, View e Controller.
3. Crie um layout XML para a tela principal (`activity_main.xml`) contendo os componentes necessários, como uma RecyclerView para a lista de itens disponíveis e outra RecyclerView para a lista de pedidos.
4. Crie um layout XML separado para os itens da lista de pedidos (`order_item.xml`) contendo os elementos visuais necessários para exibir os itens na RecyclerView.
5. Implemente a classe `Item` como o Modelo (Model) para representar os objetos de item com atributos como `id`, `name` e `price`.
6. Crie a classe `ItemAdapter` como a View para exibir os itens disponíveis na lista usando o layout padrão do Android (`simple_list_item_1`) ou um layout personalizado, se preferir.
7. Implemente a classe `OrderItem` como o Modelo (Model) para representar os objetos de item do pedido com atributos como `id`, `itemId`, `name`, `quantity` e `totalPrice`.
8. Crie a classe `OrderItemAdapter` como a View para exibir os itens do pedido na lista usando o layout `order_item.xml`.
9. Implemente a classe `OrderController` como o Controller para gerenciar as ações relacionadas aos pedidos, como adicionar, buscar e excluir pedidos do banco de dados.
10. Na classe `MainActivity`, inicialize os componentes de UI e crie instâncias de `ItemAdapter`, `OrderItemAdapter` e `OrderController`. Implemente a lógica para adicionar itens ao pedido chamando o método apropriado no `OrderController`. Além disso, exiba os itens do pedido do banco de dados na RecyclerView de pedidos através do `OrderItemAdapter`.
11. Implemente a funcionalidade para excluir itens do pedido ao clicar no botão de exclusão na lista, também através do `OrderController`.
12. Teste o aplicativo para garantir que as funcionalidades estejam funcionando conforme esperado.

****Observações:****

- Certifique-se de seguir boas práticas de programação, como evitar operações demoradas na thread principal (UI thread) e gerenciar corretamente o ciclo de vida do banco de dados.
- O objetivo deste exercício é criar uma versão básica do aplicativo de lista de pedidos usando o padrão MVC e armazenando os pedidos em um banco de dados local. Você pode expandir o aplicativo adicionando mais recursos e melhorias, conforme desejado.

Divirta-se codificando e aprendendo!