



DESENVOLVIMENTO JAVA WEB

Proposta de Resolução

Autoria: Ariel da Silva Dias

Leitura crítica: Rogério Colpani

Proposta de Resolução

Olá, estudante! Como dito no desafio, o MVC é um dos padrões de projetos mais empregados no mundo da programação web e, apesar de não ter sido criado especificamente para o Java, ele possui uma forte relação.

Então, vamos responder as perguntas que foram colocadas no desafio, lembrando que não há apenas uma resposta correta, logo, o que será apresentado é uma ideia geral de resolução.

Primeiramente, precisamos analisar o que no modelo MVC representaria cada parte de nossa aplicação.

O objetivo do MVC é separar a lógica do aplicativo da interface do usuário. Logo, o modelo (*model*) representa a camada de negócios do aplicativo, ou seja, toda a lógica de dar baixa no estoque durante uma venda, somar valores de crédito quando um cliente comprar um produto, entre outros. A camada de visualização (*view*) define a apresentação da aplicação, em outras palavras, a tela do PDV (ponto de venda ou frente de caixa) é a interface por onde o gerente monitora as vendas, por onde o vendedor interage e realiza uma venda. O controlador (*controller*) gerencia o fluxo do aplicativo, lidando com as solicitações do usuário. Se pensarmos em uma venda, quando o caixa registra os códigos de barras do produto, o papel do controlador é pegar os dados e enviar para a visão (caso seja só para apresentar os dados na tela) ou enviar para a camada de modelo (caso seja necessário armazenar no banco de dados).

Desse modo, observe que, na programação Java, o modelo (*model*) consiste em classes Java, a *view* exibe os dados e o controlador são servlets. Assim, podemos responder ao segundo questionamento, sobre a sequência de requisições.

O primeiro passo é o navegador, no caso, o caixa ou computador do gerente envia uma solicitação de página ao controlador que está no servidor.

O segundo passo é o controlador invocar o modelo, recuperando os dados que precisam, de acordo com a solicitação (por exemplo, se foi solicitado o número de vendas no dia de hoje, então o controlador solicita a camada *model* esta informação).

O terceiro passo é o controlador fornecer os dados recuperados para a camada de visualização.

Por fim, a visualização renderiza o conteúdo recebido pelo controlador e exibe no navegador.

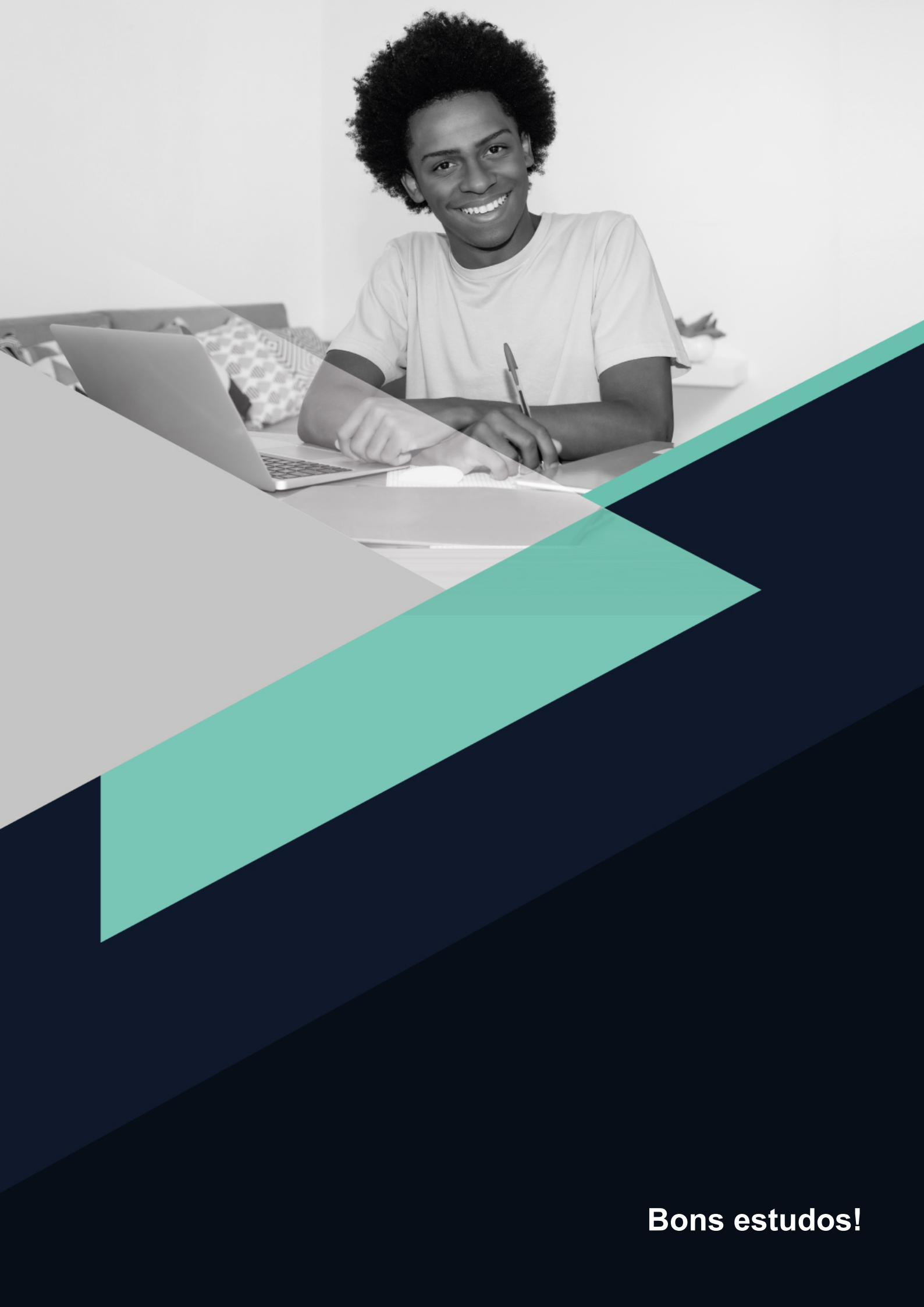
As vantagens com essa arquitetura MVC são:

- Vários desenvolvedores podem trabalhar simultaneamente (cada um/equipe em uma camada).
- Os componentes MVC são fáceis de serem mantidos uma vez que existe baixa dependência entre eles.



- Torna-se mais fácil de estender e testar um aplicativo MVC devido à baixa dependência das camadas.





Bons estudos!