
Fatec Guaratinguetá

Prova 1 – Programação Web (Parte1)

Data: 01/10/2025

Atividade: Sistema de Vendas em MVC

A equipe (até 3 alunos) irá desenvolver um sistema de gerenciamento de vendas seguindo o padrão MVC.

1. Estrutura do Banco de Dados (Model)

Um membro da equipe será responsável pela criação e organização do banco de dados, que servirá como a fonte de dados para o **Model**.

- **Banco de Dados:** Crie o banco de dados loja_virtual.
- **Tabela produtos:** Armazena as informações dos produtos (ID, nome, valor, estoque).
- **Tabela vendas:** Registra cada venda (ID, ID do produto, quantidade, valor total, data da venda).

Detalhe: Adicione alguns produtos de exemplo na tabela produtos para iniciar os testes.

2. Estrutura de Arquivos e Pastas (Organização do Projeto)

Outro membro da equipe pode criar a estrutura de pastas que reflete a arquitetura MVC.

- projeto_vendas/
 - /app/
 - /controllers/ (Lógica de controle)
 - /models/ (Lógica de dados)
 - /views/ (Interface do usuário)
 - /config/ (Configurações do banco de dados)
 - /public/ (Arquivos acessíveis pela web, como index.php)

A página index.php na pasta /public/ será a porta de entrada para todas as requisições.

3. Desenvolvimento das Camadas MVC

Model

Coloque aqui a lógica de negócio e interação com o banco de dados:

config/database.php: Arquivo para a conexão **PDO** com o banco de dados.

- **app/models/Produto.php:** Uma classe que representa um produto. Ela deve ter um método para **calcular o valor total** de uma venda e outros métodos para buscar produtos no banco de dados.
- **app/models/Venda.php:** Uma classe para gerenciar as operações de venda. Ela terá um método para **registrar uma nova venda**, que deve dar baixa no estoque (UPDATE) e inserir o registro da venda (INSERT).

Fatec Guaratinguetá

View

Apresentar a interface ao usuário.

- **app/views/produtos/listar.php**: Exibe a lista de produtos disponíveis. Deve conter a tabela de produtos e os formulários para a compra.
- **app/views/vendas/historico.php**: Apresenta a lista de vendas realizadas em uma tabela.
- **app/views/template/header.php** e **app/views/template/footer.php**: Componentes reutilizáveis para a estrutura do site.

Controller

Intermediário entre o **Model** e a **View**. Ele recebe as requisições, interage com o **Model** e decide qual **View** exibir.

- **app/controllers/ProdutoController.php**:
 - **Método listar()**: Busca os produtos no **Model** e passa esses dados para a **View produtos/listar.php**.
- **app/controllers/VendaController.php**:
 - **Método registrar()**: Recebe os dados do formulário (POST). Ele usa o **Model** para **realizar os cálculos** e salvar a venda no banco de dados.
 - **Método historico()**: Pede ao **Model** a lista de vendas e a exibe através da **View vendas/historico.php**.

4. Fluxo de Execução

- O arquivo public/index.php será a única porta de entrada. Ele deve identificar a URL e decidir qual **Controller** e **método** chamar.
- **Fluxo de Listagem**:
 1. O navegador acessa index.php (ex: index.php?controller=produto&action=listar).
 2. O index.php carrega o ProdutoController e executa o método listar().
 3. O ProdutoController pede os dados ao Produto.php (**Model**).
 4. O ProdutoController carrega a **View** listar.php e injeta os dados nela.
- **Fluxo de Venda**:
 1. O usuário preenche o formulário na **View** e envia a requisição para index.php (ex: index.php?controller=venda&action=registrar com dados via POST).
 2. O index.php chama o método registrar() do VendaController.
 3. O VendaController utiliza os **Models** (Produto e Venda) para processar a venda, realizar o **cálculo do valor total**, dar baixa no estoque e registrar a transação.
 4. O VendaController usa **sessões** para armazenar uma mensagem de sucesso ou erro e redireciona para a listagem de produtos.



SÃO PAULO
GOVERNO DO ESTADO

Fatec Guaratinguetá