Desenvolvimento de Aplicações Web Backend

Trabalho Final: Aplicação Simplificada de Comércio Eletrônico

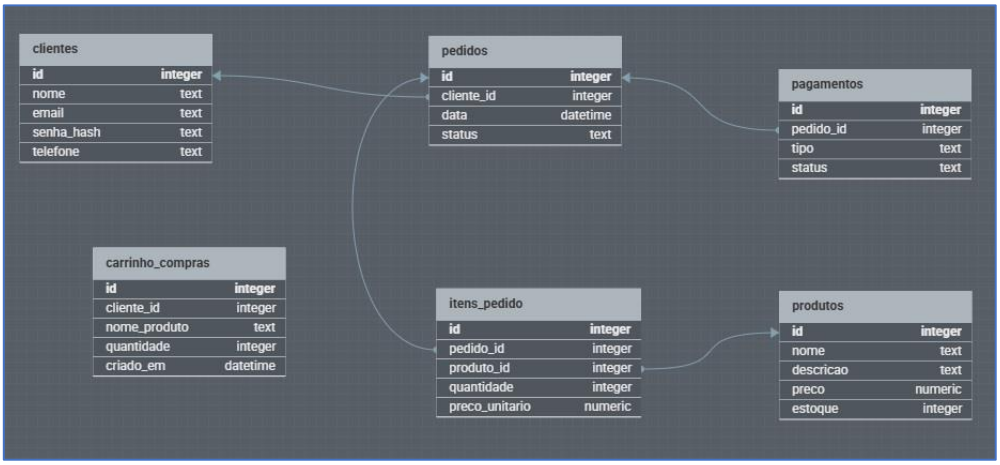
Objetivo: desenvolver uma aplicação web de comércio eletrônico simplificada utilizando Python e Flask, hospedada em uma instância AWS EC2, capaz de gerenciar clientes, produtos, carrinhos, pedidos e pagamentos. O objetivo é compreender o fluxo completo de um sistema de vendas online, desde a navegação por produtos até a finalização do pedido. Utilize os recursos abordados em aula, sempre que necessário.

Requisitos da Aplicação

Banco de dados

Utilize o script fornecido para criar as tabelas necessárias no SQLite (ou outro banco à escolha, mantendo a estrutura). As tabelas obrigatórias são:

- clientes
- produtos
- pedidos
- itens_pedido
- pagamentos
- carrinho_compras



O script disponibilizado é voltado para SQLite, se precisar mudar o banco, faça os ajustes necessários: ***script-ddl-loja_online.sql***

Rotas e telas básicas

Como sugestão de rotas e telas:

Rota	Método	Função / Tela	Observações
`/`	GET	Página inicial / catálogo de produtos	Lista todos os produtos disponíveis com nome, preço e link para detalhes
`/produto/<id>`	GET	Detalhes do produto	Mostra descrição, preço, estoque e opção de adicionar ao carrinho
`/carrinho`	GET / POST	Carrinho de compras	Exibe produtos adicionados, permite alterar quantidade e remover itens
`/checkout`	GET / POST	Finalização de pedido	Seleção de cliente, revisão de itens, escolha do método de pagamento (cartão, Pix, boleto)
`/pedido/<id>`	GET	Detalhes do pedido	Mostra itens, valores, status do pedido e pagamento
`/login`	GET / POST	Autenticação de clientes	Permite que clientes façam login para realizar pedidos
`/logout`	GET	Encerramento de sessão	Finaliza sessão do cliente
`/registrar`	GET / POST	Cadastro de novos clientes	Solicita nome, email, senha e telefone

Funcionalidades obrigatórias

- Cadastro e login de clientes com *hash* de senha**.**
- Carrinho de compras que permita adicionar produtos temporariamente, mesmo sem login.
- Atualização automática do estoque ao finalizar pedidos.
- Persistência dos dados no banco de dados e tratamento de erros básicos (ex.: produto fora de estoque, e-mail já cadastrado etc.).

Hospedagem

- A aplicação deve rodar em uma instância EC2 da AWS**.**
- Utilizar Gunicorn.
- Configuração básica de firewall/grupo de segurança para permitir acesso HTTP.

Critérios de Avaliação

- Domínio do código implementado pelos integrantes do grupo – mediante perguntas feitas pelo professor sobre o código apresentado.
- Funcionalidade completa das rotas e telas básicas.
- Persistência e integridade dos dados (cadastros, carrinho, pedidos, pagamentos).
- Qualidade do código Flask (uso de *templates*, rotas, funções e modularidade).
- Uso adequado de senhas seguras e tratamento de erros.
- Aplicação hospedada e acessível na EC2.

Regras do Trabalho

- **Valor:** 5 pontos
- **Individual ou grupo de até 4 alunos.**
- **Entregáveis:** códigos fontes (.py), arquivos html, css etc.
- **Data de entrega e apresentação dos trabalhos:** *24 de novembro.*
- **Forma de entrega:** *formulário a ser divulgado pelo professor na data de apresentação.*