



Universidade Federal de Uberlândia
Faculdade de Computação – Prof. Daniel A. Furtado
Aplicação Web com Java e Spring Boot
Atividade Individual

Este trabalho utiliza como base a API REST implementada no trabalho anterior, envolvendo o gerencialmente de clientes, produtos e fornecedores. Portanto, é obrigatório a utilização das classes implementadas no último trabalho como ponto de partida.

Registro de Vendas

Adicione à API a funcionalidade de gerencialmente de vendas de produtos. Toda venda deve estar associada ao cliente que a efetuou e aos produtos vendidos (uma venda pode conter vários produtos vendidos). Deve constar na venda a data e hora de sua realização, a quantidade vendida de cada produto, o valor unitário da venda do produto e eventual desconto aplicado (percentual) sobre o preço do produto. Para implementação das vendas, recomenda-se a criação de duas entidades: a entidade **Venda** e a entidade **ItemVenda** contendo o produto vendido, a quantidade, o preço unitário e o desconto aplicado. Dica: embora a relação indireta entre **Venda** e **Produto** seja do tipo **muitos-para-muitos**, a relação entre **Venda** e **ItemVenda** é similar à relação entre **Cliente** e **EnderecoCliente**, implementada anteriormente.

A API deve disponibilizar as seguintes funcionalidades no contexto das vendas:

- Criação de novas vendas
- Exclusão por completo de uma venda
- Listagem de todas as vendas realizadas
- Listagem de uma venda pelo ID

As classes de serviço, repositório e controlador devem ser implementadas adequadamente, conforme orientações dos trabalhos anteriores. Um objeto do tipo DTO pode ser necessário.

Criação de Aspecto para Logging Automático

Implemente um mecanismo de auditoria utilizando a Programação Orientada a Aspectos (AOP) para monitorar as operações da camada de serviço no contexto do gerencialmente de vendas. Deve ser aplicado o conceito de separação de interesses, onde a lógica de infraestrutura da aplicação fica separada do código relacionado à funcionalidade em si da API.

O componente de aspecto deve interceptar a execução dos métodos e apresentar na janela de console, de maneira automática, as seguintes informações:

- O nome do método sendo executado.
- Os valores dos parâmetros recebidos na chamada.
- O tempo total de execução do método, em milissegundos.
- As mensagens de eventuais exceções lançadas durante a execução.

A implementação deve ser feita sem alterar a classe de serviço, repositório ou controlador, ou seja, nenhuma linha de código relacionada a log deve ser inserida diretamente dentro dos métodos dessas classes.

Font-End - Cadastros

Deve haver páginas HTML para cadastro de **clientes**, **produtos** e **fornecedores**. O cadastro deve ser realizado utilizando requisições Ajax com a API Fetch e `asyn/await`. Scripts em JavaScript devem enviar os formulários, consumindo as respectivas funcionalidades da API. As páginas devem ser responsivas (o formulário deve se ajustar para uma boa exibição em smartphones). A estilização deve ser feita com CSS e flexbox. Não é necessário criar uma página para registro das vendas, mas a API deve suportar o registro de vendas utilizando um cliente HTTP (como a extensão Talend API Tester do Google Chrome).

Front-End – Listagem de Dados

Deve haver páginas HTML para listagem de **clientes**, **produtos**, **fornecedores** e **vendas**. As páginas devem se comunicar com a API utilizando requisições Ajax com Fetch e `async/await`. O conteúdo retornado pela API em JSON deve ser apresentado na página dinamicamente, por meio de atualização da árvore DOM com JavaScript (sem utilizar bibliotecas ou frameworks adicionais como jQuery e similares). Recomenda-se a utilização do exemplo **ex3-infinite-scroll**, disponibilizado como anexo do **Trabalho 6**, como base para a implementação. Não é necessário realizar o controle de paginação dinâmica (pode-se carregar e apresentar todo o conteúdo de uma vez). As páginas devem ser estilizadas com CSS e flexbox.

Front-End – Página Principal

Deve haver uma página HTML principal que forneça acesso às demais páginas, permitindo o cadastro e listagem dos itens solicitados.

Entrega e Avaliação

Este trabalho não será avaliado mediante a entrega dos arquivos. A avaliação ocorrerá por prova específica individual, em sala de aula, envolvendo a devida implementação das funcionalidades solicitadas neste documento e aspectos teóricos relacionados. A avaliação incluirá também aspectos teóricos e de implementação dos dois trabalhos anteriores que levaram à produção do estado atual da API.