

Trabalho Final – 2023/2

Sistema de Vendas – SOLID + Arquitetura CLEAN

Objetivo

O objetivo do trabalho final de 2023/2 é explorar os conceitos apresentados ao longo do semestre a saber: gerência de configuração e *deploy*, desenvolvimento em equipe, teste unitário, padrões de projeto e arquitetura de software (com uso de arquitetura *CLEAN* e aplicação dos princípios *SOLID*).

Execução

O TF deve ser desenvolvido em grupos de até 4 alunos, trabalhos individuais devem ser discutidos e aprovados pelo professor da turma (não é objetivo da disciplina a execução de trabalhos individuais).

O acompanhamento do trabalho se dará ao longo das aulas e os grupos devem entregar link do repositório (privativo, com o professor da turma como colaborador), relatório do projeto desenvolvido e preparar apresentação (10 minutos de apresentação seguido de 5 minutos de discussão) na data prevista no calendário da disciplina.

Enunciado

O trabalho proposto é a re-fatoração do sistema de vendas que iniciamos o desenvolvimento durante as aulas, com os requisitos (novos ou alterados) apresentados nesse documento.

Enunciado original:

Uma loja online oferece um conjunto de produtos em seu catálogo. Sobre cada produto armazena-se o código a descrição e o preço unitário atual. Os produtos disponíveis para venda ficam estocados em um galpão. Para cada produto é reservado um espaço onde cabe uma certa quantidade de produtos. Desta forma, para cada produto em estoque armazena-se a quantidade máxima que pode ser armazenada, a quantidade mínima que deve ser mantida em estoque e a quantidade atual disponível.

Quando um cliente quer comprar um ou mais itens ele faz um pedido. No pedido consta o identificador do pedido, o nome do cliente, e uma lista de itens. Cada item tem o código do produto, e a quantidade desejada. A partir do pedido é gerado um orçamento. Um orçamento possui um código, a data, o nome do cliente, o pedido, o somatório dos custos dos itens do pedido, o valor do imposto (10%), o valor do desconto (o padrão é 5% para pedidos com mais de 5 itens) e o valor final para o consumidor. Todos os orçamentos são armazenados para consulta futura. Se o cliente aprovar o orçamento basta indicar que deseja efetivar a venda informando o número do orçamento. Neste momento o sistema deve verificar a viabilidade da venda, isto é, se existem produtos suficientes no estoque para atender ao pedido. Em caso positivo, o sistema faz a baixa no estoque e marca o



estoque como “efetivado”. Por simplicidade vamos ignorar os aspectos relativos ao pagamento e entrega dos produtos.

Requisitos adicionais:

- Sistema deve prever que os produtos podem estar estocados em um ou mais centros de distribuição. Ao realizar a venda o sistema deve prever que a entrega pode utilizar estoques de diferentes centros (chamados pela empresa de galpões);
- A empresa está implantando uma política de descontos para seus clientes, inicialmente estão implantadas duas políticas – clientes com valor médio de suas últimas 3 compras superior a R\$ 10 mil recebem 10% de desconto e a cada R\$ 10 mil adicionais 5% adicionais até um limite de 30% de desconto. Clientes com mais de 10 compras nos últimos 6 meses recebem desconto de 25%, indiferente do valor. É aplicada a regra que dê o maior desconto. Novas políticas devem ser implantadas em breve;
- Os orçamentos devem ter validade de 21 dias a partir da sua solicitação, exceto para os períodos de baixa procura (meses de janeiro e fevereiro), quando são de 35 dias;
- O sistema deve possuir um módulo estatístico com pelo menos 3 modelos de análise definidos pelo grupo.

Previsão de “endpoints”

O sistema deve prever “endpoints” para:

- Listar os produtos disponíveis
- Solicitar orçamento para uma lista de itens informados
- Efetivar orçamento indicado se ainda for válido e houver produtos disponíveis
- Retornar o relatório de análise estatística indicado

Instruções para o desenvolvimento do trabalho

O futuro sistema será composto por dois módulos: a) um módulo *backend* (REST e monolítico) capaz de atender as requisições de futuros módulos de *frontend* dentro da lógica de negócios proposta. b) A construção de um módulo de *frontend* está fora do escopo deste trabalho, mas se o grupo conseguir fazer uma versão, mesmo que bastante inicial, será valorizada.

O objetivo do trabalho é evoluir o protótipo do módulo *backend* iniciada durante as aulas para uma versão capaz de atender todos os requisitos descritos acima atendendo os princípios SOLID e a arquitetura CLEAN. Padrões de projeto deverão ser explorados sempre que adequado. O módulo deve ser escrito em Java usando a tecnologia Spring. A persistência dos dados deverá ser feita usando JPA. A escolha do SGBD é livre (pode ser usado o H2 em memória). Um conjunto de dados de *seed* deve ser fornecido de forma a testar adequadamente o sistema.

O trabalho deverá ser desenvolvido em etapas:

1. Análise do problema utilizando DDD
2. Modelagem da solução segundo a arquitetura CLEAN
3. Implementação do *backend*



4. Implementação de drivers de teste para pelo menos duas das classes do sistema que contenham a lógica de negócio;
5. Elaboração de relatório contendo:
 - a. Diagrama de classes da solução
 - b. Detalhamento dos padrões de projeto utilizados e o objetivo de cada um
6. Apresentação do relatório e da respectiva solução funcionando com um *workspace* o *Postman* que permita testar todos os *endpoints* desenvolvidos.

Entregas

O trabalho pode ser desenvolvido individualmente ou em grupos de até 4 alunos. O projeto deve ser mantido, até a apresentação, em um repositório privativo e o professor convidado como colaborador. O uso adequado do *git* (*commits* e uso de *branches*) será considerada na avaliação do trabalho, bem como a efetiva participação de todos os integrantes do grupo.

Os entregáveis que não correspondem a código fonte (documentação) devem ser armazenados em uma pasta chamada “OutrosEntregaveis” a ser criada na raiz do projeto no “GitHub”.

Para a apresentação final todos os integrantes do grupo devem estar presentes e aptos a responder perguntas sobre o trabalho desenvolvido.

Trabalhos que configurem fraude acadêmica (mesmo entre turmas) não terão nota atribuída.