E-commerce

Trata-se de um sistema de cadastro de produtos e pedidos em um portal de e-commerce.

Requisitos Funcionais

* O portal possui vários produtos em estoque e com uma determinada quantidade em estoque de cada um desses produtos.
* Cada pedido possui um ou mais produtos e um cliente associado a esse pedido.
* Um produto pode aparecer em um ou mais pedidos.
* Cada produto possui um código, um nome, uma quantidade e um valor.
* Cada cliente possui seus dados pessoais e dados de entrega.

Requisitos Não Funcionais

* Aplicação, do tipo Java Project, onde todas as informações são fornecidas pelo usuário.
* Utilizar o maven para o gerenciamento de dependências.
* Esse sistema deve ser desenvolvido utilizando uma das três combinações de técnicas:

1. Sprng Data JPA + Cache Redis (os dois em um único projeto).
2. Spring Data JPA (em um projeto separado) e Neo4J (em outro projeto separado).
3. Spring Data JPA (em um projeto separado) e MongoDB (em outro projeto separado, sendo que neste projeto considere que pode existir um ou mais endereços de entrega cadastrados).

Banco de Dados

* MySql – Escolhemos o banco de dados MySql devido as seguintes características:
  + Escalabilidade e flexibilidade
  + Alto Desempenho
  + High Availability
  + Suporte transacional robusta
  + Proteção de Dados Strong
  + Open Source
  + Baixo Custo Total de Propriedade

Cache

* Redis - Escolhemos o Redis como sistema de cache devido as seguintes características:

Modelo de Banco de Dados

Entidades

Arquitetura

* Spring Data JPA + Cache Redis

Definimos a arquitetura Spring Data JPA + Cache Redis em um único projeto pois acreditamos ser a arquitetura mais adequada devido ao modelo de negócio exigir relacionamentos entre as entidades, portanto, um banco de dados relacional como o MySql mostra-se mais viável do que um banco de dados não relacional como mongoDB ou NEO4J neste último caso um banco de dados baseado no teorema de grafos não se aplica ao modelo de negócio que este projeto implementa.