

Universidade Estadual da Paraíba
Departamento de Computação – CCT
Disciplina – Métodos Avançados em Programação
Grupo – Arthur Lincoln, João V. Marinho, Philipe de O. e Luiz Felipe
Professor(a) – Sabrina de Figueirêdo Souto

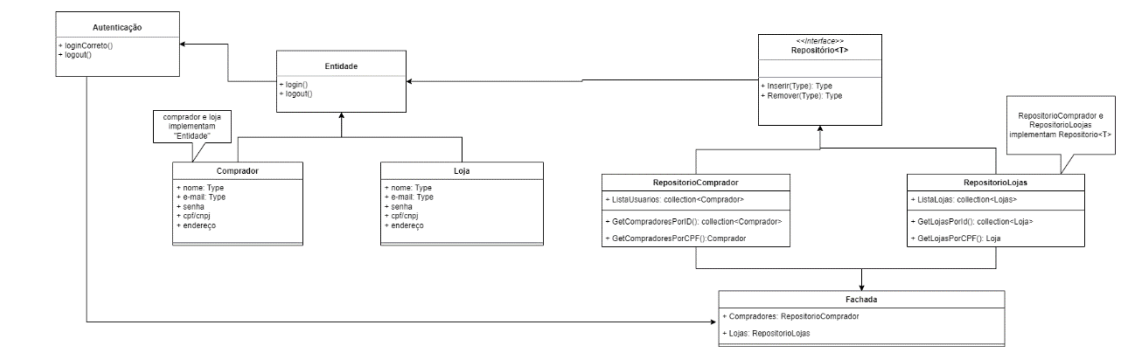
Relatório da release 1

Campina Grande, 01 de junho de 2023

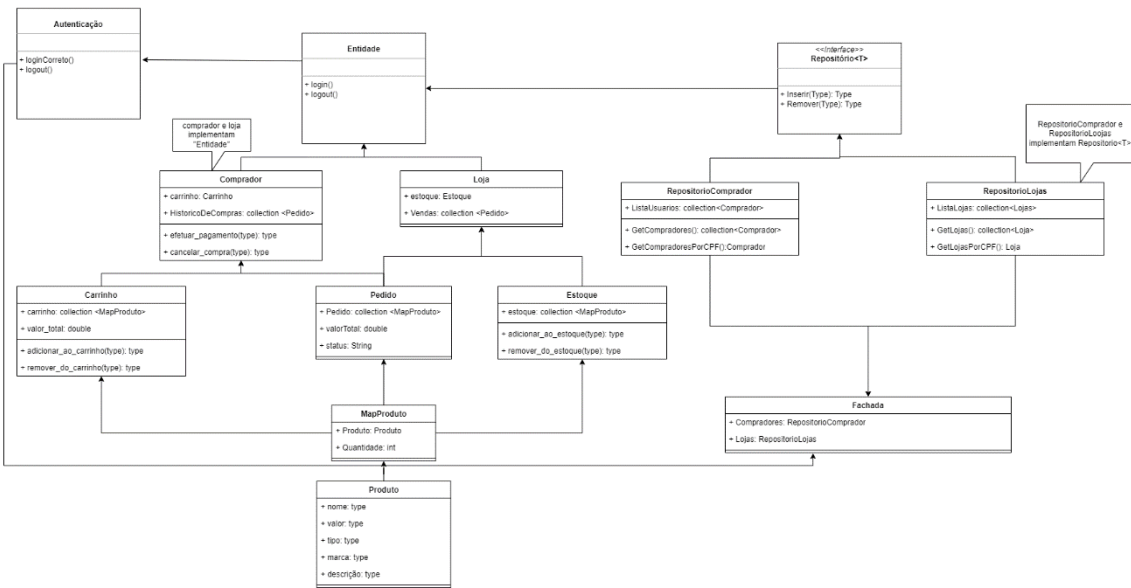
1. Diagrama de classes

Esse foi o diagrama de classes definido para o desenvolvimento do projeto:

CRUDS



SISTEMA COMPLETO



2. Patterns usados

Pattern Singleton: Esse padrão foi usado para dois requisitos, o “CRUD de lojas” e o “CRUD de Comparadores”. O Singleton foi usado nesses dois requisitos porque em ambos é importante que exista uma única instância da classe CRUD, evitando a criação de instâncias desnecessárias. Isso mantém a consistência dos dados e evita problemas de concorrência em um ambiente de acesso simultâneo, que é o que ocorre nesse projeto. Outro motivo para a utilização do Singleton nesses requisitos, é o compartilhamento de recursos importantes, evitando a necessidade de estabelecer uma nova conexão a cada operação. E pra finalizar outro motivo que fez com que o Singleton fosse usado foi a centralização da lógica de acesso aos dados em uma única classe, assim deixando as operações do CRUD encapsuladas em um único ponto.

Pattern Expert: Esse padrão foi usado para o requisito “CRUD de produtos a serem vendidos por cada loja”. O Expert foi usado nesse requisito porque a responsabilidade de executar as tarefas é atribuída a uma classe que possui maior quantidade de informações para realiza-la. No caso desse requisito a classe “Estoque” possui conhecimento necessário para realizar as operações específicas relacionadas ao estoque. Contendo uma lista “MapProduto” que é uma associação entre um objeto “Produto” e sua quantidade em estoque, assim a classe “Estoque” tem acesso direto aos objetos da lista.

Fachada (Facade): Esse padrão foi usado na “interface” do projeto, apesar do projeto não ter uma interface propriamente dita, esse padrão foi executado na visualização da execução do programa. A Fachada foi usado porque o sistema envolve várias entidades e componentes, ao utilizar desse padrão é criada uma interface única e simplificada que encapsula a lógica complexa e interações com essas entidades. Outro motivo é a definição dos métodos específicos que representam as principais funcionalidades do sistema, lidando com opções de segundo de plano.