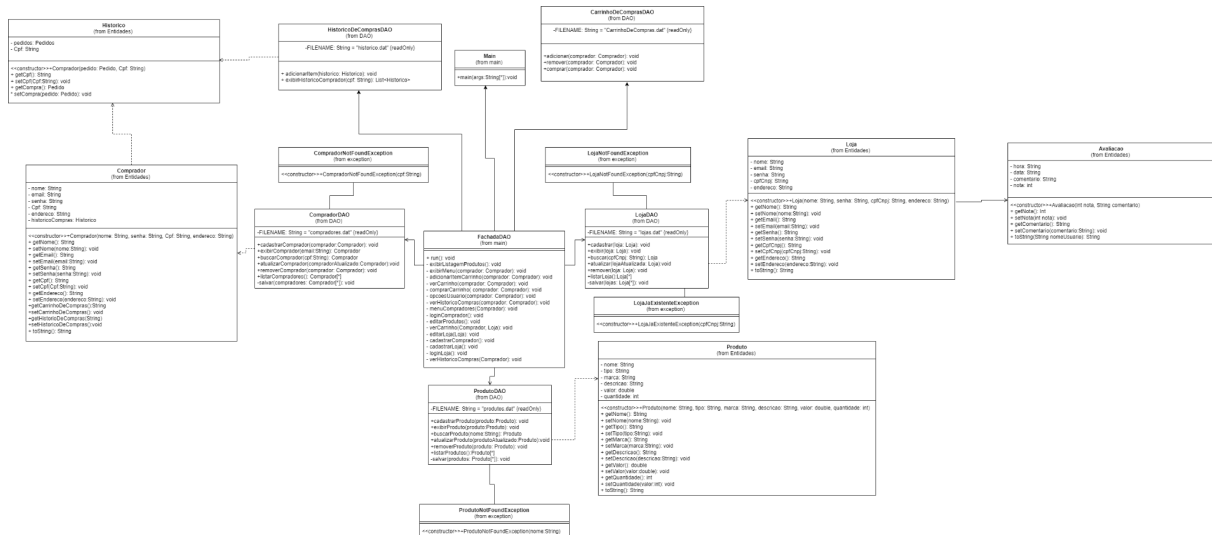




UNIVERSIDADE ESTADUAL DA PARAÍBA
Centro de Ciências e Tecnologia - CCT
Componente Curricular: **Metodos Avançados em
Programação**
Período: 2023.1
Professor(a): Dra Sabrina de Figueirêdo Souto
Turma: TARDE

Alunos: PEDRO HENRIQUE FRANÇA SALES
KAIQUE DA SILVA IVO
LUCAS FAUSTO MEDEIROS
MIKAELLE DOS SANTOS OLIVEIRA
NATÁLIA MARIA DE ARAÚJO LIMA
RAQUEL GOMES DE VASCONCELOS

02 de julho de 2023
Campina Grande, PB.



O Padrão Fachada: Todas as classes DAO do sistema se comunicam com a fachada e ela se comunica com main, simplificando assim, o acesso as classes DAO. Uma fachadaDAO foi implementada com o intuito de simplificar o acesso e a manipulação dos diferentes DAOs relacionados aos objects do sistema, como: LojaDAO, CompradorDAO, ProdutoDAO, CarrinhoDeComprasDAO, e HistoricoDeComprasDAO. A fachadaDAO simplifica o acesso aos dados no sistema , fornecendo uma interface abstrata e simplificada para interagir com os diferentes DAOs, tornando o código mais organizado, coeso e facilitando a manutenção e evolução do sistema.

Usamos a fachada ainda, para encapsular todas as classes do sistema, para não termos acesso direto a todas as classes que fazem parte da engrenagem de funcionamento da lógica de negócios do marketPlace, notamos a vantagem no desenvolvimento, pois ganhamos um pouco de tempo e compreensão melhor do código.

Padrão Creator: A classe fachada atua como um criador de objetos relacionados a persistência de dados, seguindo o padrão Creator para a maioria das classes do sistema, Produto, Loja, Comprador. Todas estas classes são construídas suas instâncias sob demanda, de acordo com as opções desejadas pelos usuários.

Padrão Composição: O sistema utiliza o conceito de composição em várias partes de sua estrutura, devido a sua natureza e relacionamento entre entidades. A composição é um padrão de design de orientação a objetos que permite construir objetos complexos combinando com outros objetos. No sistema usamos a composição diversas vezes, **Produto compõe Pedido, Produto compõe Loja, Pedidos compõe Histórico.**

Padrão Especialista: O padrão de projeto Expert é usado diversas vezes, em várias partes do sistema para garantir que a responsabilidade por determinadas operações seja atribuída ao objeto mais especializado em relação a essas operações. Se observarmos no diagrama de classe acima, **as classes Comprador, Produto e Loja**, se comunicam muito com outras classes do sistema, se não garantimos que o especialista fique com a informação, o sistema dificultaria a manutenção e perderia eficiência na comunicação entre as classes.

Alta Coesão: Tentamos aumentar a coesão das classes para facilitar o entendimento do código, dando tarefas simples para cada classe, e não alterando o valor dos parâmetros dentro dos métodos.

Baixo Acoplamento: Tentamos reduzir ao máximo o acoplamento entre as classes, apesar de não conseguirmos usar Interface e Herança no sistema, pois não conseguimos enxergar uma real necessidade de uso. As classes são independentes e possuem poucas dependências entre si.

OBS: Não escrevemos testes novos, porque nesta 3º release só temos alterações na classe

FachadaDao, ela é uma classe que executa um menu, sendo responsável pela apresentação da interface de usuário e interação, em vez de executar uma lógica complexa ou processamento. Não sabíamos como escrever testes para tais funcionalidades. E de novidade a classe Avaliacao, ambas as classes serão testadas manualmente no vídeo de apresentação do projeto.