



RELATÓRIO - RELEASE 3

Disciplina: Métodos Avançados em Programação

Discentes: Adriana Silva, João Morgan, José Antonio e Rayanne Vitoria
Docente: Sabrina Souto

Campina Grande, 2023

SUMÁRIO

1. Introdução	3
2. Design do projeto	4
2.1 Diagrama de Classes	6
3. Cobertura	7
4. Testes	8

1. INTRODUÇÃO

Neste relatório, apresentamos as atualizações e evoluções ocorridas no projeto, bem como a adição de dois novos requisitos: Requisito 07 e Requisito 08. O objetivo dessas melhorias é aprimorar a experiência do comprador e oferecer vantagens adicionais para os usuários, como a possibilidade de avaliar suas compras e obter benefícios.

Novos Requisitos Adicionados:

Requisito 07: Avaliação da Compra

Agora, os compradores podem avaliar suas compras, atribuindo notas e comentários aos produtos e vendedores. Essa funcionalidade promove transparência e confiança nas transações, além de ajudar outros compradores na tomada de decisões.

Requisito 08: Pontuação do Comprador

Implementamos um sistema de pontuação para os compradores, baseado em seu histórico de compras e avaliações. Essa pontuação proporciona benefícios exclusivos, como frete grátis, incentivando a fidelidade dos usuários.

2. DESIGN DO PROJETO

No decorrer do desenvolvimento do projeto, foram realizadas diversas atualizações e inclusões de novos recursos, proporcionando uma maior interação entre os compradores e as funcionalidades disponíveis. Nesta fase, destacam-se as seguintes alterações:

1. Criação da classe "Avaliacao":

Foi adicionada a classe "Avaliacao", contendo os atributos "comentario" (string), "produto" e "nota" (inteiro). Essa classe permite que os compradores avaliem suas compras, podendo atribuir uma nota de 1 a 5 e fornecer um comentário sobre o produto ou vendedor.

2. Introdução da interface "AvaliacaoDao":

Foi criada a interface "AvaliacaoDao" com os métodos "adicionar", "listar" e "remover", permitindo assim o gerenciamento das avaliações realizadas pelos compradores.

3. Implementação da classe "AvaliacaoDaoImpl":

A classe "AvaliacaoDaoImpl" foi desenvolvida para implementar a interface "AvaliacaoDao". Essa classe é responsável por adicionar, listar e remover as avaliações no sistema.

4. Adição de uma enumeração para representar o conceito:

Foi criada uma classe enum para representar o conceito atribuído às avaliações. Essa enumeração possibilita classificar as avaliações em categorias como "excelente", "bom", "regular", entre outras.

5. Atualização na classe "Loja":

Na classe "Loja", foi adicionado um método para calcular a média das notas recebidas nas avaliações. Essa média é utilizada para definir o conceito da loja, baseado na percepção dos compradores.

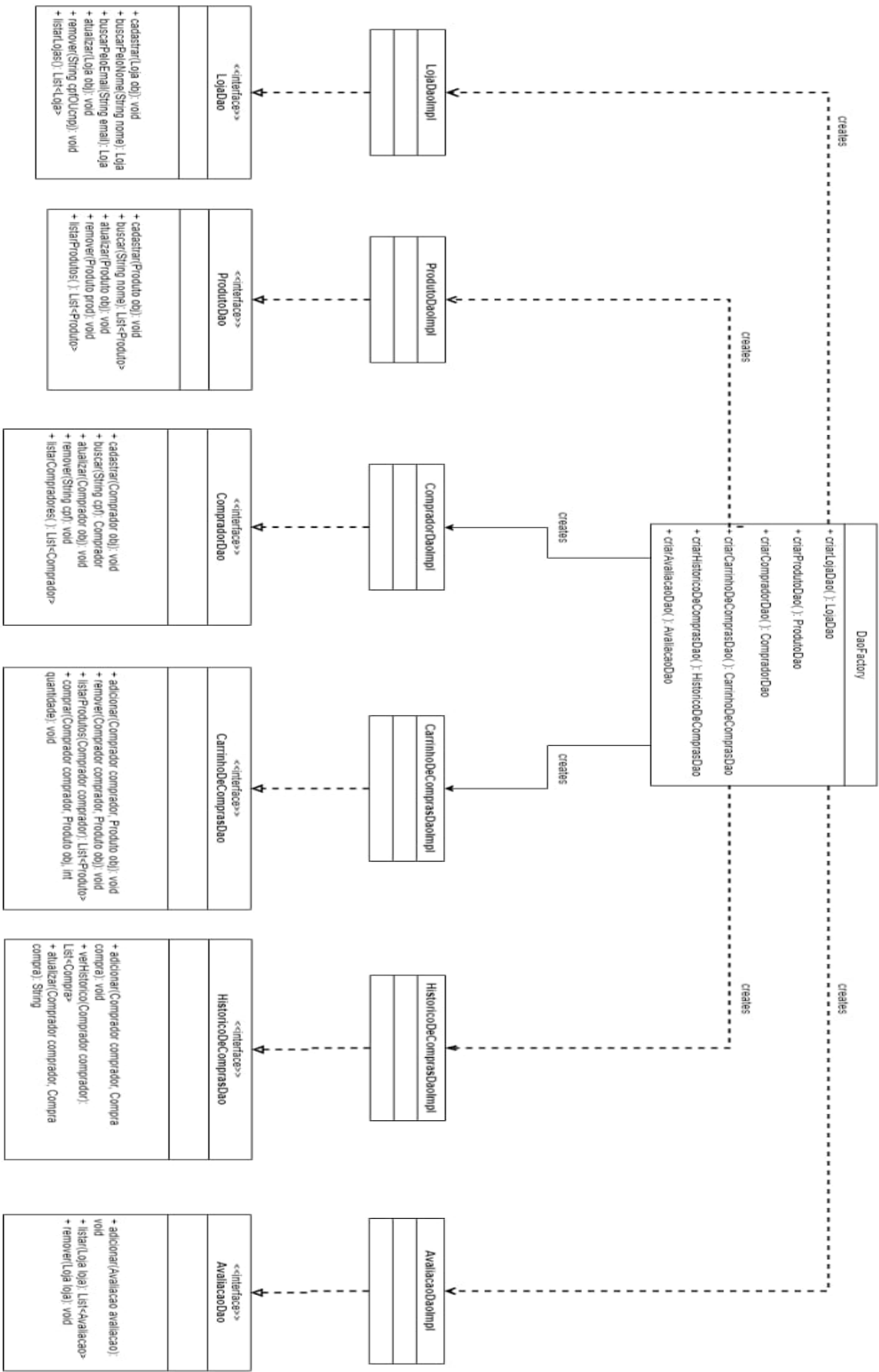
6. Atributo "pontuacao" na classe "Comprador":

Foi incluído o atributo "pontuacao" na classe "Comprador". Esse atributo é incrementado sempre que o comprador realiza uma compra e faz uma avaliação. Dessa forma, o comprador acumula pontos que podem lhe conceder benefícios, como frete grátis, quando atinge a pontuação mínima de 5 pontos.

7. Adição do método "atualizar" na classe "HistoricoDeComprasDaImpl":

Na classe "HistoricoDeComprasDaImpl", foi implementado o método "atualizar". Esse método permite a atualização das informações relacionadas ao histórico de compras, mantendo-o sempre atualizado.

2.1 DIAGRAMA DE CLASSES



3. COBERTURA

A cobertura de código no projeto foi avaliada para diferentes pacotes e classes. O pacote "Application" apresentou 70% de cobertura, enquanto o pacote "model.dao" alcançou 88,9% de cobertura, indicando uma boa abrangência dos testes. No pacote "model.dao.impl", a média de cobertura foi de 85,4%, destacando-se as classes "AvaliacaoDaoImpl" com 80,6%, "CarrinhoDeComprasDaoImpl" com 92,8%, "CompradorDaoImpl" com 84,8%, "HistoricoDeComprasDaoImpl" com 86,3%, "LojaDaoImpl" com 84,5% e "ProdutoDaoImpl" com 84,3%. O pacote "util" obteve uma média de 84% de cobertura, com destaque para as classes "CadastroMenu" com 86,2%, "CompradorMenu" com 90,4%, "LoginMenu" com 89,0%, "LojaMenu" com 89,6%, "MenuInterativo" com 80,0% e "Validar" com 51,3%.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
src	54,7 %	3,910	3,233	7,143
application	70,0 %	7	3	10
Program.java	70,0 %	7	3	10
exception	100,0 %	4	0	4
StoreNotFoundException.java	100,0 %	4	0	4
model.dao	88,9 %	24	3	27
DaoFactory.java	88,9 %	24	3	27
model.dao.impl	85,4 %	1,209	206	1,415
AvaliacaoDaoImpl.java	80,6 %	166	40	206
CarrinhoDeComprasDaoImpl.java	92,8 %	218	17	235
CompradorDaoImpl.java	84,8 %	257	46	303
HistoricoDeComprasDaoImpl.java	86,3 %	63	10	73
LojaDaoImpl.java	84,5 %	290	53	343
ProdutoDaoImpl.java	84,3 %	215	40	255
model.entities	42,1 %	626	862	1,488
Avaliacao.java	11,4 %	19	147	166
Compra.java	82,1 %	55	12	67
Comprador.java	25,1 %	92	275	367
Conceito.java	100,0 %	54	0	54
Loja.java	33,3 %	126	252	378
Pedido.java	62,5 %	45	27	72
Produto.java	54,8 %	181	149	330
ProdutoTipo.java	100,0 %	54	0	54
Testes	0,0 %	0	1,770	1,770
util	84,0 %	2,040	389	2,429
CadastroMenu.java	86,2 %	200	32	232
CompradorMenu.java	90,4 %	910	97	1,007
LoginMenu.java	89,0 %	195	24	219
LojaMenu.java	89,6 %	510	59	569
MenuInterativo.java	80,0 %	52	13	65
Validar.java	51,3 %	173	164	337

4. TESTES

Foram criados testes para validar a funcionalidade da classe "AvaliacaoDaoImpl" e da classe "Loja". Esses testes garantem a integridade e o correto funcionamento das implementações realizadas, incluindo um teste específico no "LojaTest" para verificar se o conceito está sendo atribuído de forma adequada. Esse teste verifica o conceito antes da loja receber alguma avaliação (nesse caso, o conceito é indefinido TBD), o que representa uma classe inválida, pois não está dentro do intervalo $1 \leq \text{conceito} \leq 5$. Em seguida, são realizadas avaliações, e o teste verifica se o conceito é atualizado corretamente.