



Universidade Estadual da Paraíba (UEPB)  
Centro de Ciências e Tecnologia (CCT)  
Curso Bacharelado em Ciências da Computação

Pedro Lucas Vaz Andrade

Alan Medeiros Alves Gonçalves

Zayon Tavares Oliveira

Relatório de Release 03 - Arquitetura

Campina Grande - 2025

## 1. Introdução

A Sprint 5 do projeto **Eng2Marketplace** foi direcionada à implementação de um dos aspectos mais importantes para a experiência do usuário dentro da plataforma: a **avaliação da compra**. Esta funcionalidade visa permitir que o comprador avalie tanto o produto adquirido quanto a loja vendedora após a finalização da compra. Além disso, também foram realizadas melhorias nos testes de regressão e uma rodada completa de atualização da documentação da release.

Esta sprint consolida o primeiro ciclo completo de compra + avaliação dentro do marketplace, e prepara o terreno para próximas funcionalidades como pedidos detalhados, relatórios de desempenho de lojas e recomendações baseadas em avaliações.

## 2. Arquitetura do Sistema na Sprint 5

A arquitetura do **Eng2Marketplace** seguiu a organização em camadas e foi expandida na Sprint 5 com a introdução de novos módulos relacionados à funcionalidade de avaliação. A estrutura geral segue os princípios de separação de responsabilidades (SRP), utilizando o padrão arquitetural **MVC** e o padrão de projeto **Facade** para simplificar o acesso às funcionalidades.

### 2.1 Camadas Existentes

- **Model:** Entidades do sistema (Loja, Produto, Comprador, Venda, Avaliacao)
- **Repository:** Persistência em arquivos `.txt`
- **Controller:** Regras de negócio e mediação entre View e Repository
- **View:** Interface textual (CLI)
- **Facade:** Ponto de entrada único, centralizando o uso dos controllers

### 2.2 Novas Classes Introduzidas

#### Avaliacao.java

Responsável por representar uma avaliação realizada por um comprador. Possui os atributos: `nota`, `comentario`, `produto`, `loja`, `comprador`, `data`.

## **AvaliacaoRepository.java**

Controla a leitura e escrita das avaliações no arquivo `avaliacoes.txt`. Mantém todas as operações de persistência da nova entidade.

## **Modificações em Loja.java**

Adicionada a função `getConceito()` que interpreta a média das notas recebidas e retorna um conceito textual (Ruim, Médio, Bom, Excelente).

## **2.3 Relacionamentos Adicionados**

- Uma **Venda** pode ser associada a uma ou mais **Avaliações**.
- Um **Produto** e uma **Loja** podem ser avaliados separadamente.
- O **Comprador** é o autor da avaliação, sempre vinculada a uma venda já finalizada.

## **2.4 Integração com a View**

A view CLI foi ajustada para:

- Solicitar avaliação após a compra
- Exibir conceito da loja ao visualizar o perfil
- Permitir inserção de nota e comentário via terminal

## **2.5 Fluxo de Avaliação**

Comprador realiza compra

↓

Sistema registra venda



Sistema oferece opção de avaliar produto/loja



Avaliacao é criada e salva



Loja atualiza conceito com base nas novas avaliações

## **2.6 Fluxo de Cálculo do Conceito da Loja**

Sistema recebe uma nova Avaliacao



Busca todas as avaliacoes associadas à loja



Calcula a média das notas



Verifica a faixa da média:

- 0-2 → Ruim

- >2-3 → Médio

- >3-4 → Bom

- >4-5 → Excelente



Atualiza conceito textual na exibição da loja

Esse fluxo é acionado automaticamente após o registro de uma nova avaliação e garante que o conceito exibido ao comprador esteja sempre atualizado com base na experiência dos clientes.

### 3. Considerações Arquiteturais

A introdução do módulo de avaliações ampliou a coesão do sistema sem comprometer sua manutenção. O uso do padrão **Facade** manteve a simplicidade no acesso a novas funcionalidades. A persistência em arquivo foi mantida para facilitar os testes manuais e reduzir a complexidade nesta fase do projeto.

As entidades e controladores adicionados seguem o mesmo padrão das funcionalidades anteriores, o que facilita a curva de aprendizado para novos desenvolvedores.

A arquitetura está preparada para futuras evoluções, como substituição da persistência por banco de dados, implementação de autenticação, e separação de interfaces com GUI ou web.



