MODELAGEM E VALIDAÇÃO DE SISTEMA

Felipe Eduardo Bohnen[1]
Miguel Schneiders Flach[2]
Murilo Morosini[3]
Roney Bieger Anschau[4]

Resumo

Este relatório detalha a modelagem e validação de um sistema CRUD para gerenciamento de produtos em um mercado. A modelagem inclui Diagramas de Casos de Uso, de Sequência, de Classes e o Modelo Entidade-Relacionamento, que definem a estrutura e o comportamento do software. Padrões de projeto, como Singleton para gerenciamento de conexão e Arquitetura em Camadas, foram aplicados para promover modularidade. A validação, realizada através de testes unitários e funcionais, revelou falhas em validações de dados e na interação da interface, necessitando de correções. Um teste de carga foi tentado, mas a ausência de resultados impede uma análise de desempenho conclusiva. O trabalho visa aprimorar o sistema para garantir funcionalidade e robustez.

[1] Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. felipebohnenf@gmail.com
[2] Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. miguelsflach65@gmail.com
[3] Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. murilomorosini@gmail.com
[4] Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. roneyronyy@hotmail.com

1. INTRODUÇÃO

O presente relatório tem como objetivo apresentar a modelagem, validação e aprimoramento de um sistema de gerenciamento de produtos para um mercado, conforme requisitos da disciplina de Engenharia de Software II. O sistema, que consiste em um CRUD (Create, Read, Update, Delete) completo, visa otimizar o controle de estoque e as operações relacionadas aos produtos. Este documento abordará as etapas de modelagem de software, utilizando diagramas UML (Unified Modeling Language) e o Modelo Entidade-Relacionamento (ER), além de identificar padrões de projeto aplicados. Posteriormente, serão detalhados os testes unitários, funcionais e de carga realizados para verificar a robustez e o desempenho do sistema.

2. DESENVOLVIMENTO

2.1 Modelagem de Software

A fase de modelagem é crucial para a compreensão da estrutura, comportamento e requisitos do sistema. Foram elaborados diagramas de Casos de Uso, de Sequência, de Classes e o Modelo Entidade-Relacionamento, que detalham diferentes perspectivas do software.

2.2 Diagrama de Casos de Uso

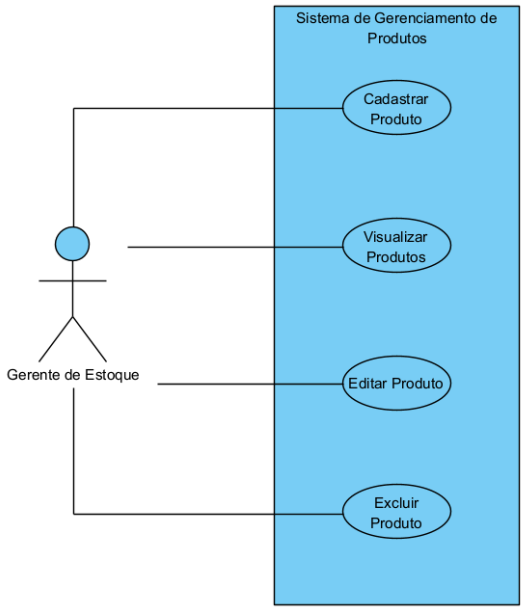
Este Diagrama de Casos de Uso modela as principais funcionalidades do Sistema de Gerenciamento de Produtos e o ator que interage com ele. O Gerente de Estoque é o ator central, representando o usuário que irá operar o sistema. Ele possui associações diretas com os seguintes casos de uso, que são as funcionalidades que o sistema oferece:

Cadastrar Produto: Permite ao Gerente de Estoque inserir novos produtos no sistema, adicionando suas informações essenciais.

Visualizar Produtos: Habilita o Gerente de Estoque a consultar a lista completa de produtos já cadastrados no sistema.

Editar Produto: Concede ao Gerente de Estoque a capacidade de modificar os detalhes de um produto existente, como preço ou quantidade em estoque.

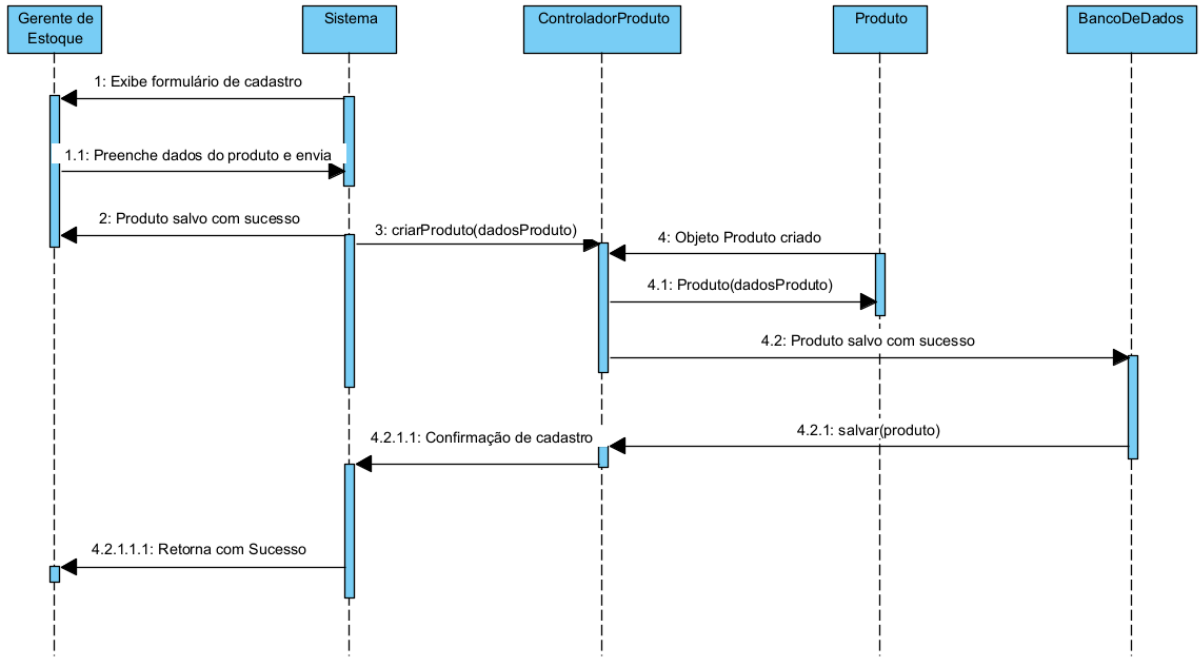
Excluir Produto: Permite que o Gerente de Estoque remova produtos do sistema quando não são mais necessários.



Fonte: Autoria própria (2025).

2.3 Diagrama de Sequência

Este Diagrama de Sequência detalha a ordem cronológica das interações entre os diferentes componentes do sistema quando um usuário (Gerente de Estoque) solicita o cadastro de um novo produto. O Gerente de Estoque inicia o fluxo ao solicitar a funcionalidade de cadastro. A Interface do Usuário (Sistema) responde exibindo um formulário para preenchimento dos dados do produto. Após o preenchimento, o Gerente de Estoque envia as informações. A Interface do Usuário passa esses dados para o ControladorProduto, que é responsável pela lógica de negócio. O ControladorProduto cria uma nova instância do objeto Produto com os dados fornecidos. Em seguida, o ControladorProduto interage com o Banco de Dados (ou a camada de persistência, como um DAO) para salvar as informações do novo produto. O Banco de Dados retorna uma confirmação de sucesso para o ControladorProduto. Finalmente, o ControladorProduto informa a Interface do Usuário sobre o sucesso da operação, que então exibe uma mensagem de confirmação ao Gerente de Estoque.



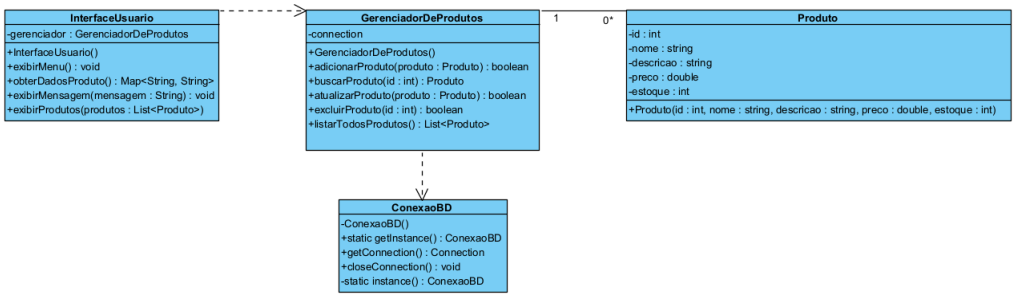
Fonte: Autoria própria (2025).

2.4 Diagrama de Classes

ConexaoBD (Singleton) garante uma única conexão com o banco de dados para todo o sistema. As classes se relacionam da seguinte forma: a InterfaceUsuario depende da GerenciadorDeProdutos para operar; a GerenciadorDeProdutos gerencia os objetos Produto; e a GerenciadorDeProdutos utiliza a ConexaoBD para acessar o banco de dados. O diagrama fornece uma visão clara da organização do código e da colaboração entre os componentes.

2.5 Modelo Entidade-Relacionamento (ER)

Este Modelo Entidade-Relacionamento (ER) ilustra a estrutura do banco de dados do sistema, com foco nas informações persistentes. O modelo é composto por uma única Entidade: PRODUTOS. Esta representa a tabela fundamental que armazena todos os dados dos produtos. Inclui atributos como id (chave primária e identificador único), nome, descricao, preco e estoque, cada um com seu tipo de dado e restrições de nulidade. Devido à simplicidade do escopo, este diagrama ER não apresenta relacionamentos entre múltiplas entidades, focado inteiramente na organização dos dados da tabela de produtos. Ele serve como blueprint para a criação da base de dados do sistema.



Fonte: Autoria própria (2025).

2.6 Uso de Padrões de Projeto

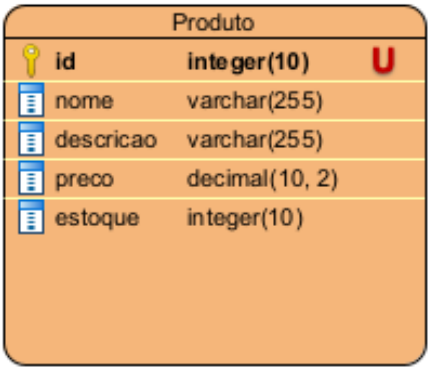
Os padrões de projeto a seguir foram aplicados e são visíveis na modelagem do sistema:

Padrão Singleton

Finalidade: Garante uma única instância para otimizar recursos.
Onde ver: Implementado na classe ConexaoBD (construtor privado e método getInstance()) no Diagrama de Classes.

Padrão de Camadas (Arquitetura)

Finalidade: Organizar o sistema em responsabilidades claras (separação de preocupações).
Onde ver: A estrutura em camadas (Apresentação, Lógica/Dados, Infraestrutura) é evidente nas relações entre InterfaceUsuario, GerenciadorDeProdutos e ConexaoBD no Diagrama de Classes.



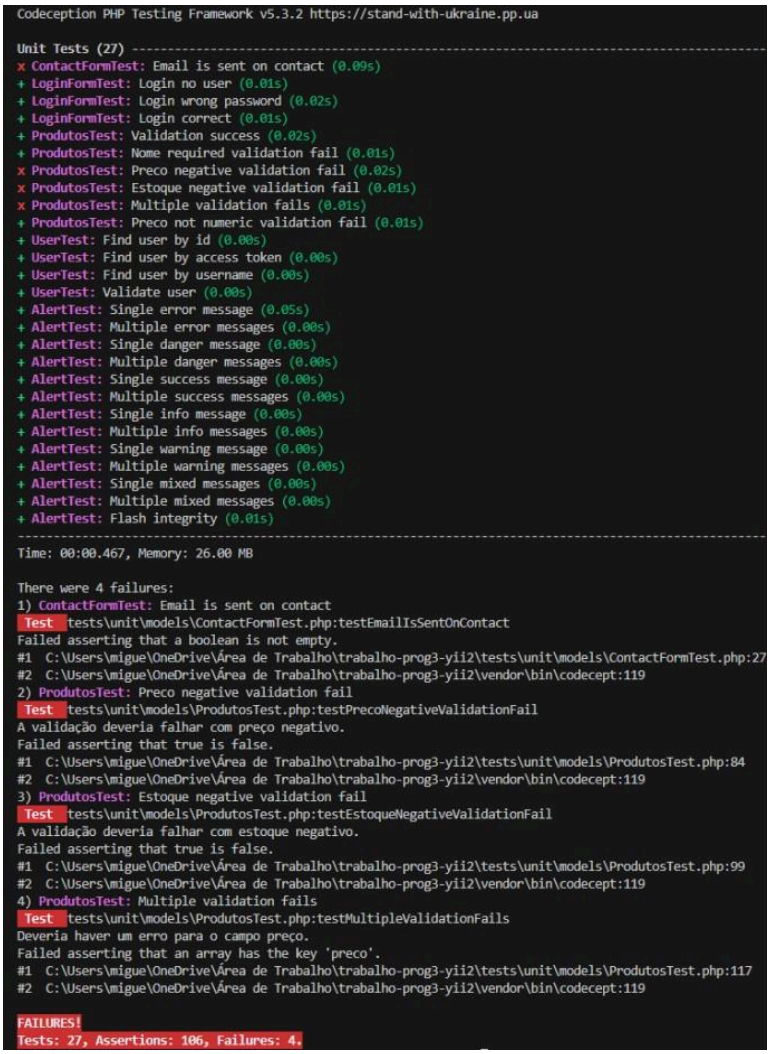
Fonte: Autoria própria (2025).

2.2 Validação e Testes

Esta seção apresenta os resultados obtidos nos testes unitários, de funcionalidade e de carga, fornecendo uma análise sobre o comportamento e desempenho do sistema.

2.2.1 Testes Unitários

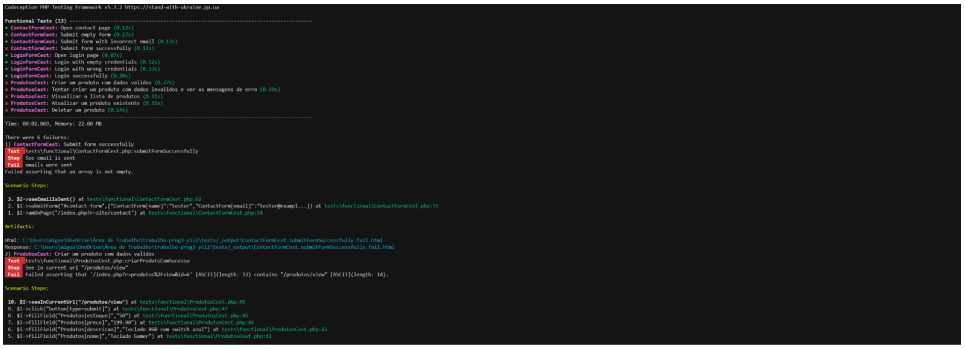
O Teste Unitário foca na verificação isolada de componentes. Nos 27 testes executados, 4 falharam. As falhas estão ligadas a validação do modelo Produto, especificamente permitindo preços e estoque negativos, e indicando problemas em validações múltiplas, o que exige revisão das regras de negócio.



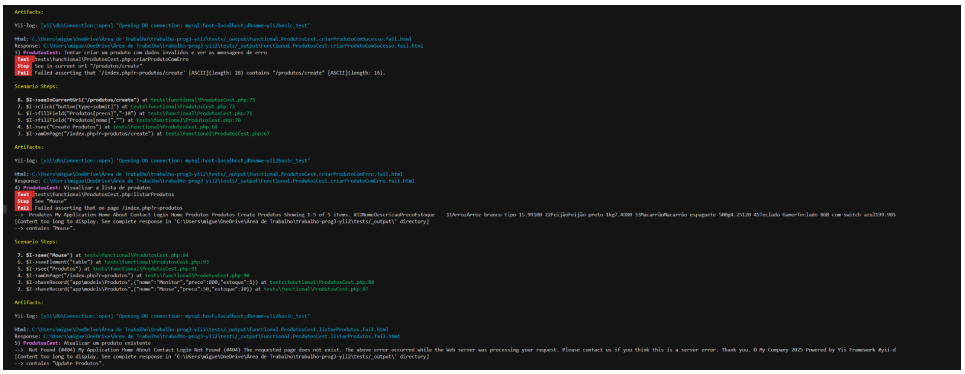
Fonte: Autoria própria (2025).

2.2.2 Testes Funcionais

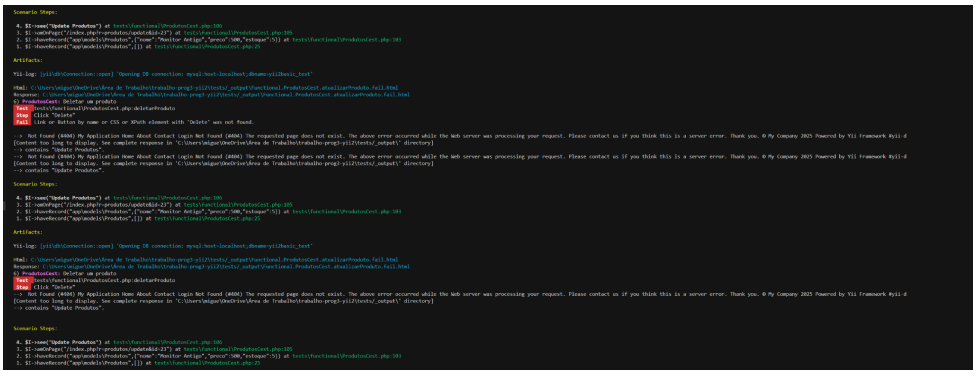
O Teste de Funcionalidade avalia se as funcionalidades correspondem aos requisitos. Das 13 funcionalidades testadas, 4 falharam, como mostram as figuras abaixo. As falhas incluem problemas no envio de formulário de contato, na exibição de mensagens de erro para dados inválidos, na atualização (erro 404) e na exclusão de produtos (botão 'Delete' não encontrado).



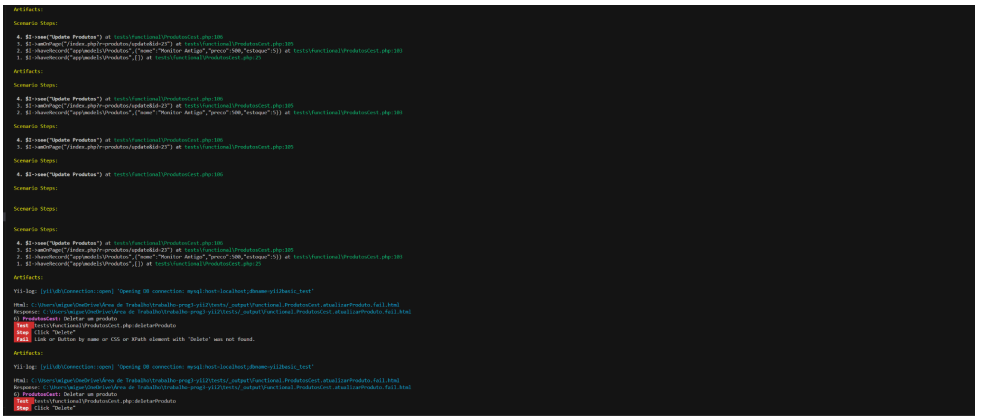
Fonte: Autoria própria (2025).



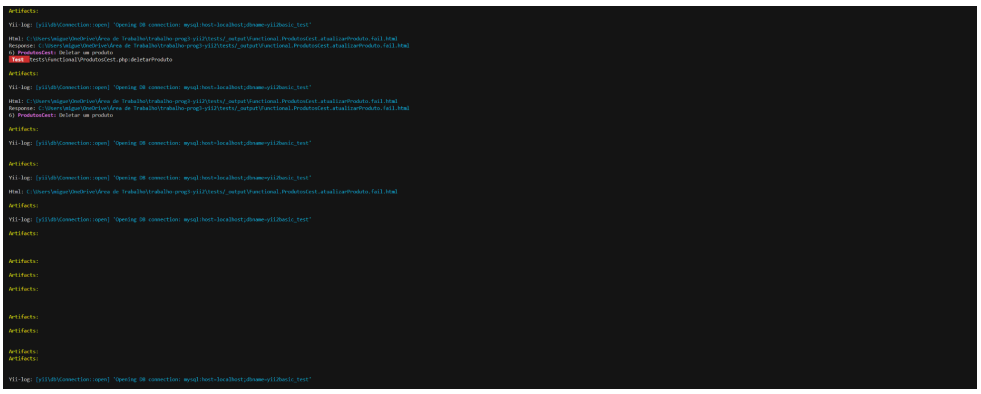
Fonte: Autoria própria (2025).



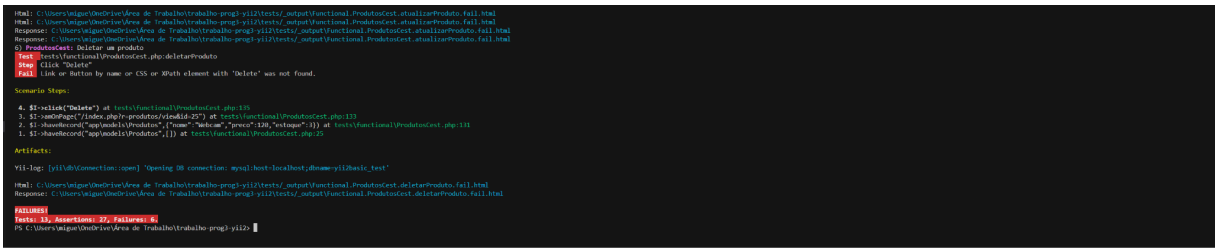
Fonte: Autoria própria (2025).



Fonte: Autoria própria (2025).



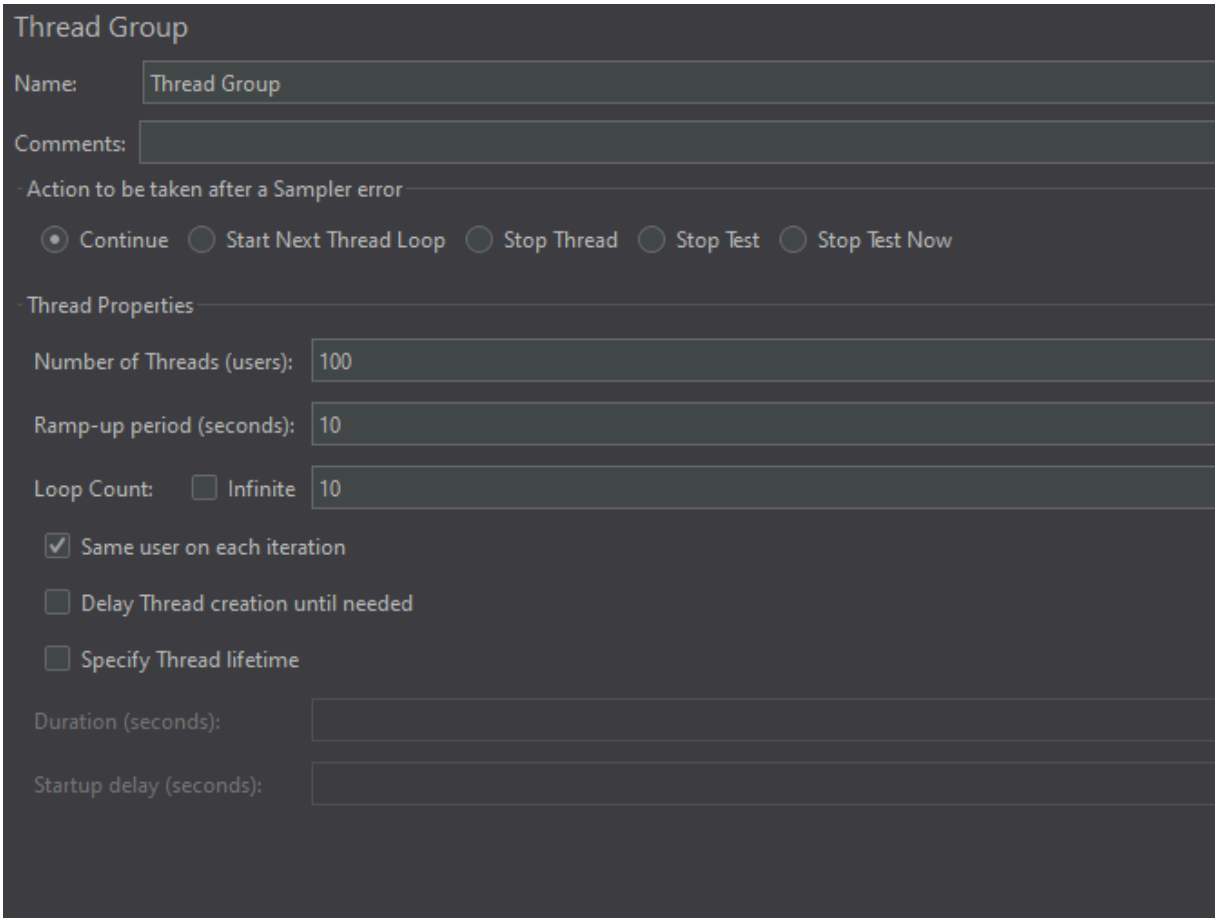
Fonte: Autoria própria (2025).



Fonte: Autoria própria (2025).

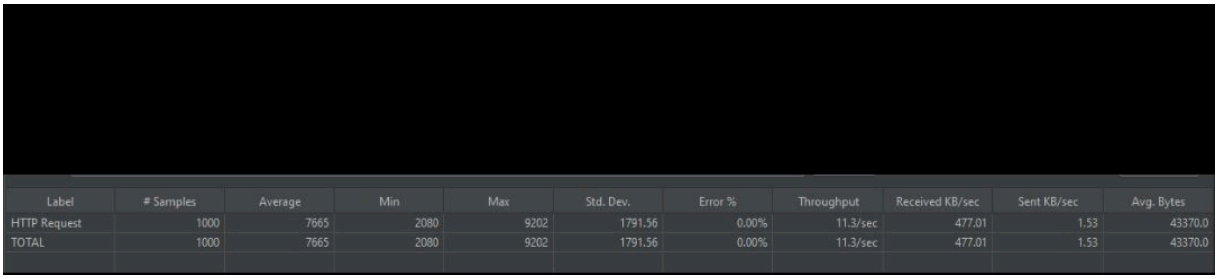
2.2.3 Testes de Carga

O Teste de Carga tem como objetivo avaliar o desempenho do sistema sob diferentes níveis de utilização e identificar possíveis gargalos ou limitações. Para este trabalho, o teste foi planejado com as configurações apresentadas abaixo, utilizando 100 usuários virtuais simulando requisições por um período de 10 segundos para rampa de carga, com 10 iterações de loop.



Fonte: Autoria própria (2025).

Contudo, os resultados apontam para desafios significativos de desempenho: o tempo médio de resposta de 7.665 milissegundos é consideravelmente elevado, e o desvio padrão de 1.791,56 milissegundos indica uma inconsistência notável na experiência do usuário. A vazão observada foi de 11.3 requisições por segundo. Tais métricas sugerem a necessidade de otimização para melhorar a agilidade e a previsibilidade do sistema sob carga.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	1000	7665	2080	9202	1791.56	0.00%	11.3/sec	477.01	1.53	43370.0
TOTAL	1000	7665	2080	9202	1791.56	0.00%	11.3/sec	477.01	1.53	43370.0

Fonte: Autoria própria (2025).

3. CONCLUSÃO

Este relatório apresentou a fase de modelagem de software para o sistema de gerenciamento de produtos, detalhando seus requisitos funcionais e estruturais através de diagramas UML e ER. A aplicação de padrões de projeto demonstrou a preocupação com a organização e manutenibilidade do código. A fase de validação e testes, embora tenha revelado algumas falhas importantes nos testes unitários e funcionais (especialmente relacionadas a validações e interação com a interface), forneceu insights valiosos sobre o comportamento do sistema. Adicionalmente, os testes de carga evidenciaram a necessidade de otimização de desempenho, com tempos de resposta elevados e inconsistentes sob simulação de uso.

https://www.youtube.com/watch?v=I8mFWieYke8