

CEFET-MARKET

1st Álvaro Augusto José Silva
Dept. de Engenharia de Computação
CEFET-MG
Divinópolis, Brasil
alvaro.ajsilva@gmail.com

2nd Humberto Henrique Lima Cunha
Dept. de Engenharia de Computação
CEFET-MG
Divinópolis, Brasil
humberto17henrique@gmail.com

3rd João Paula da Cunha Faria
Dept. de Engenharia de Computação
CEFET-MG
Divinópolis, Brasil
joaopaulofaria98@gmail.com

4th Kauã Lucas de Jesus Silva
Dept. de Engenharia de Computação
CEFET-MG
Divinópolis, Brasil
kaualucas396@gmail.com

5th Marcus Vinicius Nogueira Santos
Dept. de Engenharia de Computação
CEFET-MG
Divinópolis, Brasil
marcuscefet@gmail.com

6th Otávio Hiratsuka Camilo
Dept. de Engenharia de Computação
CEFET-MG
Divinópolis, Brasil
otaviohiratsukac@gmail.com

Abstract—Este projeto apresenta o desenvolvimento do Cefet-Market, um marketplace acadêmico baseado na web, projetado para facilitar a troca de livros, componentes eletrônicos e materiais de estudo entre alunos e professores. Implementado com tecnologias modernas como Angular no front-end e Spring Boot no back-end, a plataforma oferece funcionalidades específicas de acordo com três tipos de usuários: cliente, vendedor e administrador. Os principais recursos incluem gerenciamento de produtos, rastreamento de pedidos, controle de permissões e uma interface responsiva. A arquitetura do sistema foi guiada por princípios da engenharia de software, com requisitos bem definidos, análise de riscos e metodologia estruturada. O resultado é um sistema escalável e extensível que atende a desafios comuns no acesso a recursos educacionais em instituições acadêmicas.

I. INTRODUÇÃO

O desenvolvimento de sistemas de e-commerce tem se tornado cada vez mais relevante no cenário tecnológico, impulsionado pela crescente digitalização do comércio e a necessidade de plataformas robustas, escaláveis e seguras. Este projeto, desenvolvido como parte da disciplina de Engenharia de Software do curso de Engenharia da Computação no CEFET-MG, visa aplicar os conceitos teóricos aprendidos em sala de aula na construção de uma aplicação real, simulando um ambiente de desenvolvimento profissional. A proposta consiste na criação de um sistema de e-commerce completo, abrangendo desde a engenharia de requisitos até a implementação de funcionalidades críticas, como cadastro de produtos, carrinho de compras, processamento de pagamentos e gestão de usuários. Além disso, o projeto servirá como um estudo de caso para a padrões de projeto e boas práticas de arquitetura de software, utilizando tecnologias modernas como Spring Boot (back-end) e Angular (front-end).

II. PROPOSTA

A proposta do Cefet-Market é a facilitação para alunos e/ou entusiastas da área de conseguirem livros e materiais de estudo de determinada área ou matéria, promovendo um ambiente virtual colaborativo que conecta vendedores e

compradores. O sistema funcionará como um marketplace acadêmico, permitindo que usuários publiquem e adquiram materiais relevantes, com foco especial em atender às necessidades da comunidade acadêmica do CEFET-MG. Os produtos disponíveis incluem livros usados, apostilas, kits eletrônicos e componentes diversos voltados para cursos técnicos e superiores.

III. ESPECIFICAÇÃO DO PROBLEMA

Muitos alunos sentem a necessidade de estudar além do horário da aula, porém, por não terem em mãos o material que é utilizado pelo professor em aula — como livros específicos, apostilas ou componentes eletrônicos — ou por não terem acesso fácil a esses materiais online, acabam ficando um pouco perdidos em relação ao conteúdo. Isso gera lacunas no aprendizado e diminui o aproveitamento das disciplinas. A ausência de uma plataforma confiável e especializada para troca e venda desses recursos dentro do ambiente acadêmico agrava ainda mais o problema.

IV. MISSÃO DO PRODUTO

Facilitar a busca e aquisição de materiais de estudo por parte dos alunos, promovendo uma plataforma acessível e organizada para a compra e venda de livros, apostilas, kits eletrônicos e outros recursos utilizados em sala de aula. A missão inclui também o incentivo ao reaproveitamento de materiais, à economia colaborativa e ao fortalecimento da rede de apoio entre estudantes e professores.

V. JUSTIFICATIVA PARA O PRODUTO

A justificativa do produto vem para solucionar e facilitar que alunos possam adquirir os materiais e conteúdos que professores utilizam em sala de aulas, livros, documentações, materiais e peças eletrônicas.

VI. TEMPO GASTO E PREÇO PARA DESENVOLVIMENTO DO SOFTWARE

O desenvolvimento do projeto teve início no dia **10/04/2025** e tem como data final estipulada o **15/07/2025**, totalizando aproximadamente 14 semanas. O cronograma foi organizado em sprints semanais, conforme a Tabela I.

TABLE I
CRONOGRAMA DE DESENVOLVIMENTO

Período	Atividades
10/04 a 17/04	Planejamento, definição de requisitos e divisão de tarefas
18/04 a 02/05	Estruturação do projeto, configuração do ambiente e repositório
03/05 a 17/05	Cadastro de usuários, login, autenticação e autorização
18/05 a 31/05	Módulo de produtos e perfil do vendedor
01/06 a 14/06	Carrinho de compras, pedidos e filtragem de produtos
15/06 a 28/06	Funcionalidades administrativas e controle de permissões
29/06 a 06/07	Integração front-end/back-end e ajustes de layout
07/07 a 13/07	Testes finais e correções de bugs
14/07 a 15/07	Entrega final, documentação e apresentação

A. Estimativa de Custo

A estimativa de custo considera o tempo de trabalho de cada profissional envolvido e um valor médio de mercado por hora de serviço. A Tabela II resume essa estimativa:

TABLE II
ESTIMATIVA DE CUSTOS POR PROFISSIONAL

Função	Horas	R\$ /hora	Subtotal (R\$)
Back-End	140	60	8.400,00
Front-End	120	60	7.200,00
Documentação	60	40	2.400,00
Design UI	30	50	1.500,00
Total			19.500,00

B. Custos Operacionais

- Hospedagem do servidor (3 meses): R\$ 150,00
- Banco de dados (plano básico): R\$ 90,00
- Domínio (proporcional): R\$ 30,00
- Ferramentas auxiliares (GitHub, Trello): Gratuitas

Total de custos operacionais: R\$ 270,00

Custo total estimado do projeto: R\$ 19.770,00

VII. ORGANIZAÇÃO DA EQUIPE

A organização da equipe foi estruturada de forma a refletir a divisão de responsabilidades típicas de um projeto de desenvolvimento profissional. Com base nos conhecimentos e afinidades de cada membro, o grupo foi dividido em três frentes principais: desenvolvimento de back-end, desenvolvimento de front-end e documentação.

A equipe de **Front-End**, composta por Otávio, Humberto, Álvaro e Kauã, foi responsável pela implementação da interface gráfica da aplicação. Suas tarefas incluíram a criação das telas responsivas, o consumo das APIs REST fornecidas pelo back-end, a navegação entre componentes, e a experiência do usuário como um todo. Utilizando o framework Angular, essa equipe também cuidou da integração com bibliotecas de componentes visuais, filtros de produtos e simulação do fluxo de compras.

A parte de **Back-End** ficou sob responsabilidade de João Paulo, que desenvolveu os serviços e controladores usando o framework Spring Boot. Ele foi responsável por criar a estrutura de entidades e repositórios, realizar o mapeamento entre os objetos e o banco de dados (JPA/Hibernate), além de implementar a lógica de autenticação, controle de usuários e rotas seguras. João Paulo também trabalhou na integração com o banco de dados relacional e no desenvolvimento das APIs RESTful.

A **Documentação** do projeto ficou a cargo de Marcus Vinícius, que foi responsável pela redação técnica de todos os tópicos abordados, incluindo a especificação de requisitos, cronograma de execução, modelagem UML (casos de uso, classes, sequência), análise de riscos, metodologia, instruções de uso e elaboração do relatório final no modelo IEEE. Além disso, Marcus fez a curadoria das imagens utilizadas na apresentação e contribuiu para garantir a consistência entre a documentação e a implementação do sistema.

Essa divisão permitiu que as tarefas fossem executadas de forma paralela e eficiente, com reuniões semanais para sincronização do progresso, identificação de impedimentos e alinhamento entre os grupos. O trabalho colaborativo foi essencial para o cumprimento dos prazos e para o desenvolvimento de um produto funcional e bem documentado.

VIII. CRONOGRAMA

Para o cronograma do projeto trabalhamos de maneira onde semanalmente houvessem reuniões para ver o andamento do projeto, juntamente com as funcionalidades aplicadas na semana, erros a serem corrigidos, e funcionalidades a serem adicionadas.

IX. ESCOPO DE PRODUTO

O escopo do *Cefet-Market* abrange o desenvolvimento de uma plataforma web acessível, segura e intuitiva, voltada para a compra e venda de livros, materiais de estudo e componentes eletrônicos. O sistema será implementado como um marketplace acadêmico, possibilitando que alunos e professores da instituição publiquem, busquem e adquiram recursos úteis ao processo de ensino-aprendizagem.

O sistema oferecerá funcionalidades específicas para três tipos distintos de usuários:

- **Cliente:** poderá se cadastrar, realizar login, pesquisar e filtrar produtos, adicionar itens ao carrinho, finalizar pedidos e acompanhar seu histórico de compras.
- **Vendedor:** terá todas as permissões do cliente, além da capacidade de cadastrar, editar e remover seus produtos,

visualizar pedidos de seus itens e acompanhar suas vendas.

- **Administrador:** além das funcionalidades dos demais, poderá acessar um painel administrativo com controle total sobre os usuários, incluindo a possibilidade de promover ou rebaixar perfis, monitorar produtos e acompanhar estatísticas de uso da plataforma.

As funcionalidades previstas para o sistema incluem:

- Cadastro e autenticação de usuários com diferentes perfis;
- Publicação e gerenciamento de produtos por parte dos vendedores;
- Pesquisa e filtragem de itens por nome, categoria ou preço;
- Adição de itens ao carrinho e finalização de pedidos;
- Histórico de compras para clientes e histórico de vendas para vendedores;
- Painel administrativo para gestão dos usuários e permissões;
- Interface responsiva, compatível com dispositivos móveis e navegadores modernos.

O front-end será desenvolvido em **Angular**, enquanto o back-end utilizará **Spring Boot**, com integração via API RESTful. O sistema utilizará banco de dados relacional para armazenamento das informações e será projetado como um produto mínimo viável (MVP), com estrutura modular para futuras expansões, como integração com meios de pagamento, sistema de avaliação de produtos e notificações automatizadas.

X. RESTRIÇÕES DE ESCOPO

Algumas restrições para o projeto são a não realização de pagamentos, o usuário "cliente" só poderá se tornar um vendedor se entrar em contato pessoalmente com algum administrador

XI. INFRA-ESTRUTURA

Para a infraestrutura do projeto será necessário, para um bom funcionamento, um servidor dedicado para hospedagem do site e do banco de dados, garantindo alta disponibilidade e desempenho. O banco de dados armazenará informações de usuários, produtos, categorias, pedidos e permissões. A aplicação será dividida entre front-end (Angular) e back-end (Spring Boot), conectados por uma API RESTful. Além disso, medidas de segurança como autenticação com tokens (JWT), controle de acesso baseado em níveis de usuário (cliente, vendedor e administrador) e criptografia de dados sensíveis serão implementadas para garantir a integridade e a privacidade dos dados.

XII. ANÁLISE E DOCUMENTAÇÃO DE RISCOS

Durante o desenvolvimento do *Cefet-Market*, foram identificados diversos riscos que poderiam comprometer o sucesso do projeto. Esses riscos foram classificados em quatro categorias principais: técnicos, de cronograma, organizacionais e de segurança. A seguir, apresentamos a análise e documentação desses riscos, acompanhada de suas respectivas estratégias de mitigação.

A. Riscos Técnicos

- **Integração entre front-end e back-end:** dificuldades na comunicação entre Angular (front-end) e Spring Boot (back-end) podem gerar mal funcionamento da aplicação.
 - *Mitigação:* Definir claramente os contratos da API REST logo nas primeiras sprints e utilizar ferramentas como Postman para testes de endpoints.
- **Desempenho em ambiente de produção:** o uso de servidores gratuitos ou limitados pode afetar a escalabilidade da plataforma.
 - *Mitigação:* Realizar testes de carga e considerar futura migração para servidores dedicados ou escaláveis em nuvem.

B. Riscos de Cronograma

- **Atrasos no desenvolvimento de funcionalidades críticas:** como login, carrinho, pedidos e painel de administração.
 - *Mitigação:* Estabelecer prioridades e entregar funcionalidades mínimas primeiro. Revisões semanais com acompanhamento de progresso.
- **Acúmulo de tarefas próximas ao prazo final:** risco de atraso na entrega completa do projeto.
 - *Mitigação:* Divisão equilibrada de tarefas e controle de produtividade via reuniões semanais e ferramentas como Trello.

C. Riscos Organizacionais

- **Desalinhamento da equipe:** falhas na comunicação ou execução das tarefas designadas.
 - *Mitigação:* Definir claramente responsabilidades, realizar reuniões frequentes e promover uso de versionamento (GitHub) e planejamento (Trello).
- **Indisponibilidade de membros:** ausência de integrantes pode afetar partes específicas do sistema.
 - *Mitigação:* Garantir documentação interna do código e backup de tarefas entre os membros.

D. Riscos de Segurança

- **Falhas na autenticação ou controle de acesso:** usuários com permissões incorretas podem acessar funcionalidades indevidas.
 - *Mitigação:* Implementação de autenticação JWT e checagem rigorosa de permissões por tipo de usuário (cliente, vendedor, administrador).
- **Exposição de dados sensíveis:** como senhas ou informações pessoais de usuários.
 - *Mitigação:* Utilização de criptografia (bcrypt) para senhas e boas práticas de segurança como HTTPS e validação de entrada.

TABLE III
MATRIZ DE RISCOS DO PROJETO

Risco	Probabilidade	Impacto
Integração Angular/Spring	Média	Alta
Atraso nas funcionalidades críticas	Alta	Alta
Desalinhamento da equipe	Média	Média
Exposição de dados sensíveis	Baixa	Alta
Indisponibilidade de membros	Baixa	Média
Limitações do servidor em produção	Média	Média

E. Resumo dos Riscos

A identificação e análise desses riscos permitiram a definição de estratégias preventivas que foram integradas ao planejamento do projeto. Com isso, o desenvolvimento ocorreu de forma mais segura, colaborativa e organizada, aumentando a confiabilidade e a qualidade do sistema final.

XIII. EXPLICAÇÃO DA METODOLOGIA

A metodologia adotada para o desenvolvimento do projeto foi baseada em práticas ágeis, com enfoque no framework Scrum. O Scrum é uma metodologia incremental e iterativa que visa a entrega contínua de valor por meio de ciclos curtos denominados *sprints*. Essa abordagem foi escolhida por permitir flexibilidade, adaptação rápida a mudanças e maior colaboração entre os membros da equipe.

No contexto do projeto *Cefet-Market*, o trabalho foi dividido em sprints semanais, com reuniões realizadas de forma remota pela plataforma **Discord**. Nessas reuniões, a equipe discutia o progresso alcançado, avaliava possíveis impedimentos e definia novas funcionalidades ou correções a serem implementadas. Embora o Trello não tenha sido utilizado, a comunicação e divisão de tarefas foram organizadas verbalmente e de maneira colaborativa, respeitando as responsabilidades de cada integrante (front-end, back-end e documentação).

A metodologia foi complementada pelo uso do **GitHub** para controle de versão e integração do código entre os participantes. Esse controle foi essencial para evitar conflitos durante o desenvolvimento e possibilitou que todos os membros tivessem acesso à versão mais atual do sistema.

A. Sprints realizados

O desenvolvimento do sistema foi organizado em quatro sprints principais, cada uma com objetivos bem definidos:

- **Sprint 1 – Cadastro Inicial:** Estruturação das funcionalidades básicas de autenticação, como login e cadastro de usuários, além da criação da página inicial (*home*). Essa sprint foi fundamental para estabelecer o ponto de entrada no sistema.
- **Sprint 2 – Gestão de Produtos:** Implementação das funcionalidades de cadastro, listagem e visualização de produtos. Os vendedores puderam, a partir dessa sprint, começar a inserir seus itens no marketplace.
- **Sprint 3 – Funcionalidades Avançadas:** Criação das páginas “Minha Conta”, histórico de pedidos, edição de

dados pessoais e listagem/edição de usuários. Também foram aplicadas melhorias na navegação e validações nos formulários.

- **Sprint 4 – Correções Finais e Integração:** Padronização visual, integração completa entre o front-end (Angular) e o back-end (Spring Boot), testes finais, correções de bugs e refinamento geral da experiência do usuário.

Essa organização permitiu que o projeto evoluísse de maneira progressiva, com entregas parciais e funcionais a cada sprint. A aplicação do Scrum se mostrou eficaz para gerenciar o tempo, priorizar tarefas e garantir o alinhamento entre os objetivos do sistema e as ações da equipe.

XIV. REQUISITOS

O projeto se baseia em um site web de venda (marketplace), que para a utilização do mesmo o usuário necessitaria, apenas de um dispositivo com acesso a internet, e um navegador para acessar a página.

XV. DIAGRAMAS DE CASOS DE USO E DOCUMENTAÇÃO DOS CASOS DE USO

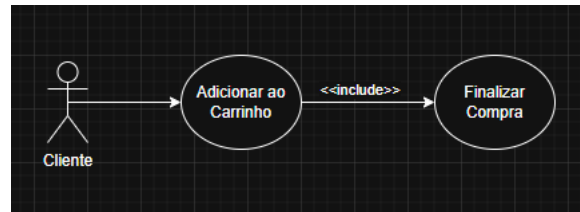


Fig. 1. Cliente adicionar ao carrinho para Finalizar a compra

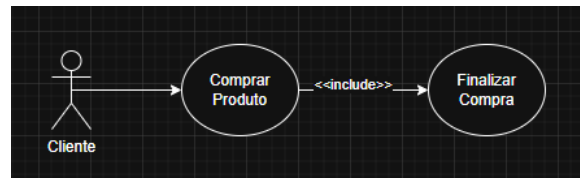


Fig. 2. Cliente comprar Produto para Finalizar a compra

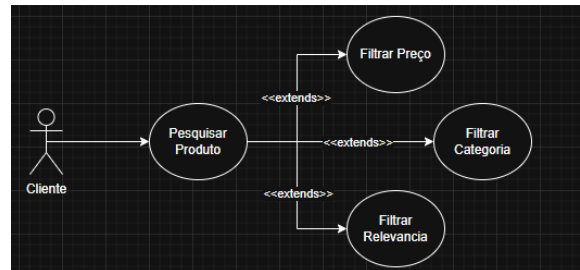


Fig. 3. Cliente Pesquisar produtos e filtrar

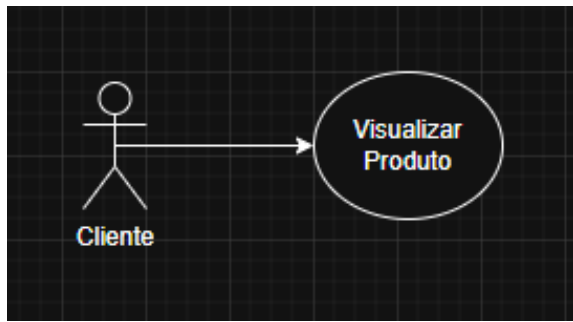


Fig. 4. Cliente visualizar produto

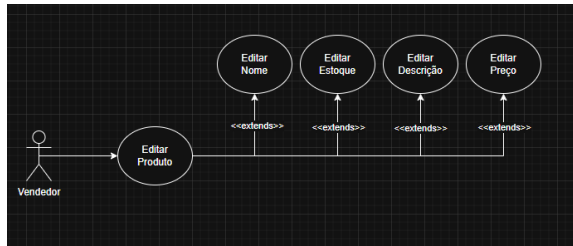


Fig. 5. Vendedor Editar Produtos

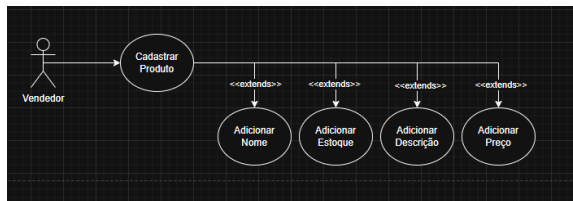


Fig. 6. Vendedor Cadastrar Produtos

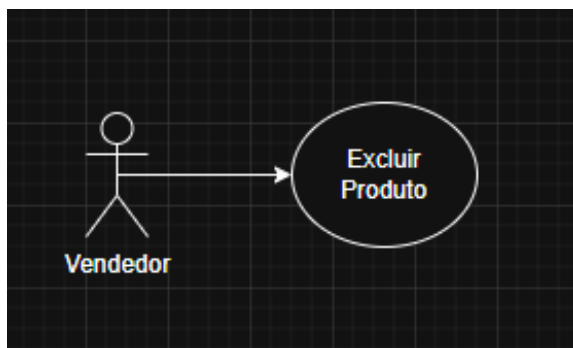


Fig. 7. Vendedor Excluir Produtos

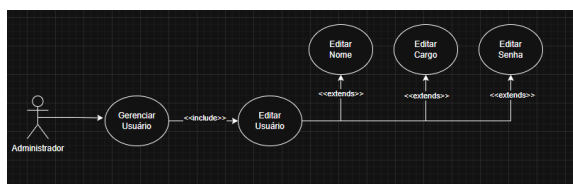


Fig. 8. Administrador -> Gerenciar Usuarios -> Editar Usuários

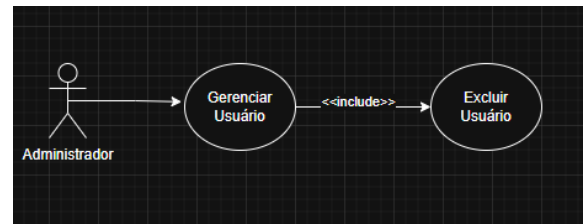


Fig. 9. Administrador -> Gerenciar Usuarios -> Excluir Usuários

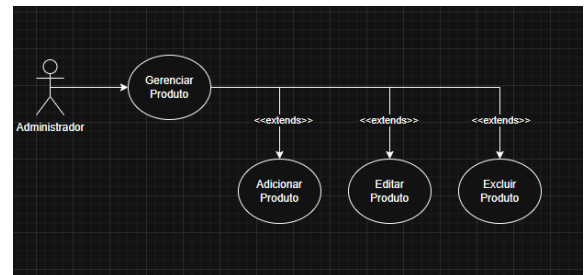


Fig. 10. Administrador -> Gerenciar Produtos

XVI. DIAGRAMA CLASSE

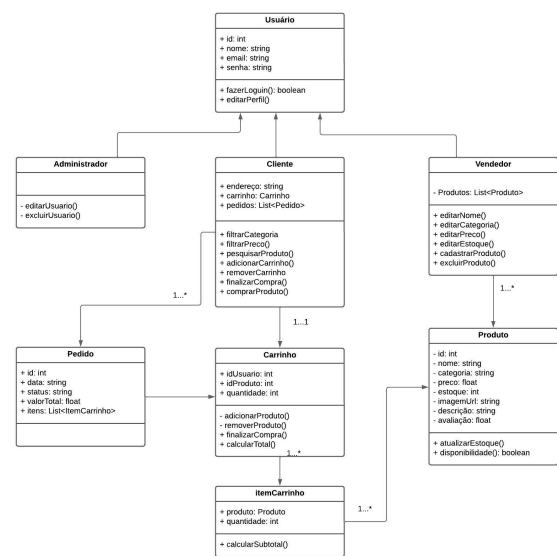


Fig. 11. Diagrama de Classes

XVII. BANCO DE DADOS

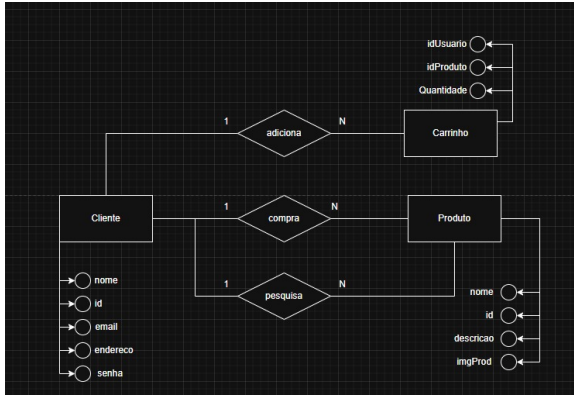


Fig. 12. Diagrama de Entidade e Relacionamento - Cliente

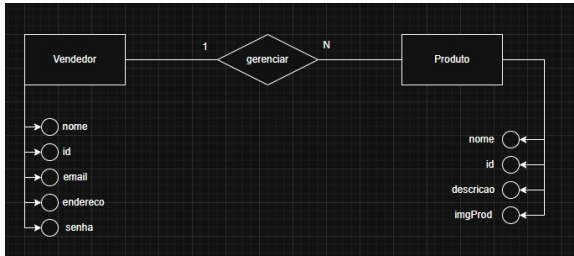


Fig. 13. Diagrama de Entidade e Relacionamento - Vendedor

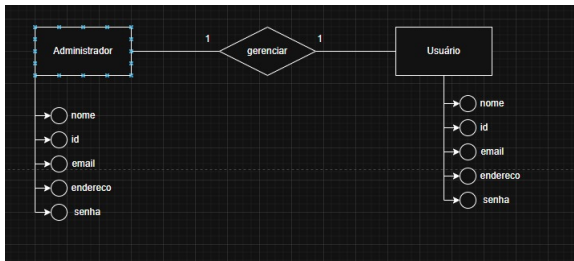


Fig. 14. Diagrama de Entidade e Relacionamento - Administrador

XVIII. DIAGRAMA DE SEQUÊNCIA

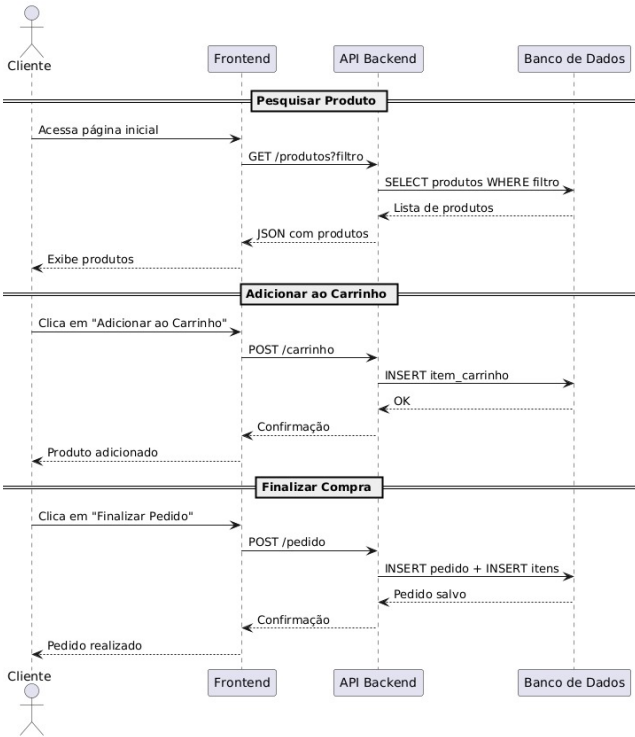


Fig. 15. Diagrama de Sequência - Cliente

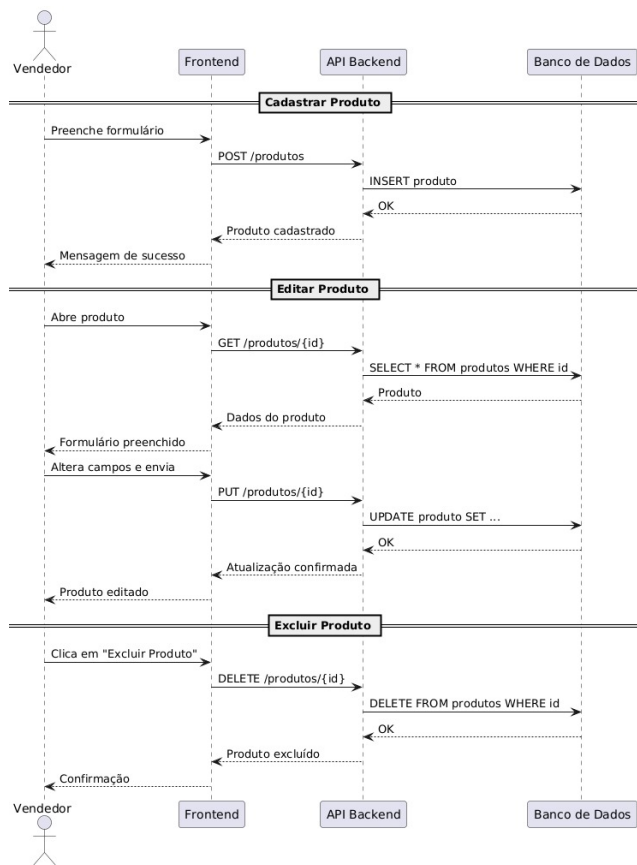


Fig. 16. Diagrama de Sequência - Vendedor

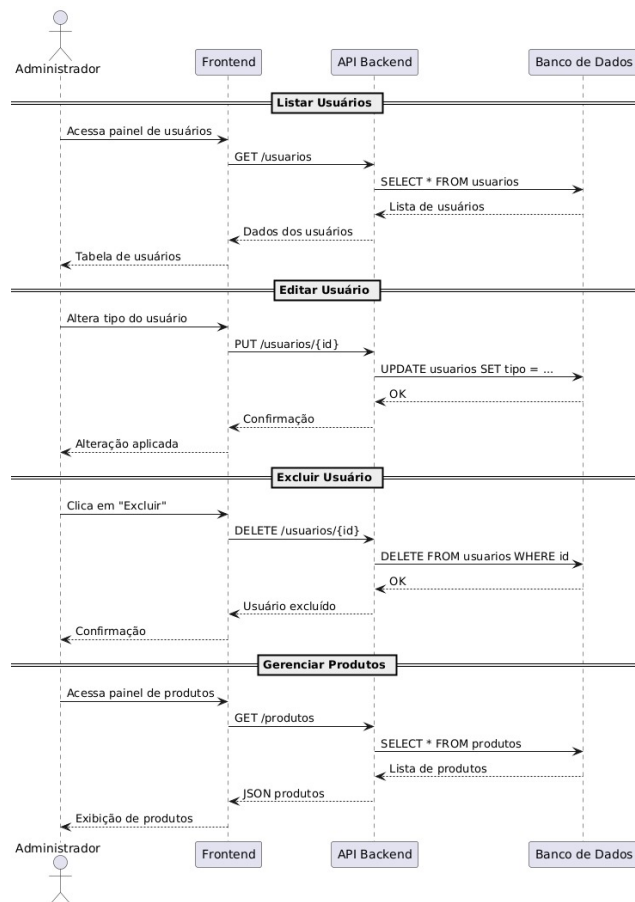


Fig. 17. Diagrama de Sequência - Administrador

XIX. TESTES

A seguir são apresentados testes simulando a utilização do sistema *Cefet-Market* por diferentes tipos de usuários. Os testes foram realizados manualmente após a integração das funcionalidades no ambiente de desenvolvimento local, com o objetivo de validar o comportamento esperado da aplicação.

A. Teste 1 – Cliente realiza compra

Objetivo: Verificar se um cliente consegue navegar pelo sistema, buscar produtos, adicionar ao carrinho e finalizar uma compra com sucesso.

- **Cenário:** Usuário autenticado com perfil de cliente.
- **Ações:**
 - 1) Acessar o sistema e realizar login.
 - 2) Navegar até a página de produtos.
 - 3) Utilizar filtros para buscar "Arduino".
 - 4) Adicionar o produto "Arduino UNO R3" ao carrinho.
 - 5) Acessar o carrinho e clicar em "Finalizar Compra".
- **Resultado Esperado:** Pedido é criado com status "pendente", os itens são registrados corretamente e o carrinho é esvaziado.
- **Resultado Obtido:** Comportamento conforme esperado. Pedido registrado com ID #0021.

B. Teste 2 – Vendedor cadastra novo produto

Objetivo: Verificar se o usuário com perfil de vendedor consegue cadastrar um novo produto e visualizá-lo na listagem.

- **Cenário:** Usuário autenticado com perfil de vendedor.
- **Ações:**
 - 1) Realizar login como vendedor.
 - 2) Acessar a área "Meus Produtos".
 - 3) Clicar em "Adicionar Produto".
 - 4) Preencher dados do produto: nome, descrição, preço, categoria e quantidade.
 - 5) Confirmar o cadastro.
- **Resultado Esperado:** Produto é exibido na listagem com todos os dados preenchidos.
- **Resultado Obtido:** Produto "Multímetro Digital" cadastrado com sucesso e listado na página do vendedor.

C. Teste 3 – Administrador altera perfil de um usuário

Objetivo: Verificar se o administrador consegue alterar o perfil de um usuário comum para vendedor.

- **Cenário:** Administrador autenticado acessa o painel de usuários.
- **Ações:**
 - 1) Realizar login como administrador.
 - 2) Acessar a aba "Gerenciar Usuários".
 - 3) Localizar o usuário "Lucas Silva" com perfil atual "Cliente".
 - 4) Selecionar a opção "Alterar Cargo" e modificar para "Vendedor".
 - 5) Confirmar a alteração.
- **Resultado Esperado:** O perfil do usuário é alterado e ele passa a ter acesso à área de cadastro de produtos.
- **Resultado Obtido:** Perfil de "Lucas Silva" atualizado com sucesso. Novo menu "Meus Produtos" disponível após o próximo login.

XX. INSTRUÇÕES DE COMO RODAR O SOFTWARE

Para executar o sistema *Cefet-Market*, é necessário possuir:

- Java 17 ou superior instalado;
- Node.js e Angular CLI instalados;
- Um SGBD como MySQL ou PostgreSQL configurado (de acordo com o arquivo de `application.properties` no back-end);
- IDEs recomendadas: IntelliJ ou VSCode;
- Navegador web moderno (Google Chrome, Firefox).

A. Execução do Back-End (Spring Boot)

- 1) Navegue até a pasta do projeto back-end.
- 2) Configure o banco de dados no arquivo `application.properties`.
- 3) Execute o projeto via terminal com o comando `./mvnw spring-boot:run` ou via IDE.

B. Execução do Front-End (Angular)

- 1) Acesse a pasta do front-end.
- 2) Execute `npm install` para instalar as dependências.
- 3) Execute `ng serve` para iniciar o servidor de desenvolvimento.
- 4) Acesse a aplicação no navegador por `http://localhost:4200`.

XXI. INSTRUÇÕES DE COMO UTILIZAR O SISTEMA

O sistema simula um ambiente de marketplace voltado à comunidade acadêmica, permitindo a compra e venda de materiais de estudo, livros e itens eletrônicos. A aplicação está dividida em três perfis de usuário: Cliente, Vendedor e Administrador.

A. Cliente

- Pode se cadastrar e realizar login.
- Pode navegar por produtos, visualizar detalhes e aplicar filtros por nome, categoria ou preço.
- Pode adicionar produtos ao carrinho e finalizar a compra.
- Visualiza seus pedidos anteriores.

B. Vendedor

- Possui todas as funcionalidades do cliente.
- Pode cadastrar novos produtos, incluindo nome, descrição, preço, quantidade e categoria.
- Pode editar e excluir produtos que cadastrou.
- A solicitação para se tornar vendedor é feita diretamente com o administrador.

C. Administrador

- Possui controle total sobre o sistema.
- Pode realizar todas as ações dos demais usuários.
- Pode promover ou rebaixar usuários entre os papéis de cliente e vendedor.
- Pode gerenciar (editar/excluir) qualquer produto.
- Acompanha as estatísticas gerais da aplicação.

O sistema foi projetado para ser intuitivo, funcional e responsivo. Todas as funcionalidades principais estão acessíveis através de menus na barra de navegação, com feedback visual para ações realizadas (cadastro, edição, compra, etc.).

XXII. RESULTADOS

Os resultados obtidos com o desenvolvimento do *Cefet-Market* demonstram que todas as expectativas inicialmente propostas foram devidamente cumpridas e alcançadas. A equipe conseguiu entregar um sistema funcional, com as principais funcionalidades implementadas e operacionais, contemplando as necessidades dos três tipos de usuários: cliente, vendedor e administrador.

Durante a fase de testes, os fluxos principais — como navegação entre páginas, autenticação, filtragem de produtos, adição ao carrinho, finalização de pedidos, cadastro de produtos e controle administrativo — foram validados com sucesso, apresentando comportamento estável e de acordo com os requisitos definidos na etapa de especificação. A

plataforma mostrou-se responsiva, com bom desempenho no carregamento de dados e usabilidade satisfatória.

Além disso, o projeto também demonstrou que a aplicação dos conceitos de engenharia de software — como análise de requisitos, modelagem UML, planejamento de cronograma e divisão de tarefas — contribuiu significativamente para a organização, qualidade e conclusão eficiente do sistema.

A integração entre o front-end (Angular) e o back-end (Spring Boot) foi realizada de maneira consistente, e o armazenamento de dados em banco relacional permitiu fácil manutenção e escalabilidade futura.

Conclui-se, portanto, que o *Cefet-Market* atende plenamente aos objetivos pedagógicos e técnicos do projeto, representando uma solução viável e funcional para a comercialização de materiais entre membros da comunidade acadêmica.

XXIII. CONCLUSÃO

O desenvolvimento do *Cefet-Market* representou uma aplicação prática dos conhecimentos adquiridos ao longo da disciplina de Engenharia de Software, proporcionando à equipe uma vivência próxima à realidade de projetos profissionais de desenvolvimento web. A proposta de criar um marketplace acadêmico voltado à comunidade do CEFET-MG permitiu abordar desafios reais enfrentados por alunos e professores na obtenção de materiais de estudo.

Durante o projeto, foi possível aplicar conceitos essenciais da engenharia de software, como levantamento de requisitos, análise e documentação de riscos, definição de escopo, cronograma de sprints, além da adoção de uma metodologia ágil baseada em Scrum. As tecnologias escolhidas — Angular no front-end e Spring Boot no back-end — foram fundamentais para estruturar uma solução moderna, escalável e segura.

A divisão clara de papéis entre os membros da equipe e o uso de ferramentas colaborativas contribuíram para o bom andamento do projeto, que resultou em um produto mínimo viável funcional. O sistema implementa os principais recursos esperados de um marketplace, incluindo gerenciamento de produtos, carrinho de compras, autenticação de usuários e controle de permissões por tipo de perfil (cliente, vendedor e administrador).

Concluímos que o *Cefet-Market* não só atende ao objetivo proposto, como também abre espaço para futuras melhorias, como integração com gateways de pagamento, sistema de avaliação e comunicação entre usuários. A experiência reforçou a importância do planejamento, da organização e do trabalho em equipe para o sucesso de projetos de software.

REFERENCES

- [1] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. McGraw-Hill Education, 2019.
- [2] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.
- [3] Spring Boot Documentation. [Online]. Available: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- [4] Angular Documentation. [Online]. Available: <https://angular.io/docs>
- [5] K. Schwaber and J. Sutherland, *The Scrum Guide*. [Online]. Available: <https://scrumguides.org>
- [6] Grady Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 2nd ed. Addison-Wesley, 2005.
- [7] Silberschatz, A., Korth, H. F., and Sudarshan, S., *Database System Concepts*, 7th ed. McGraw-Hill, 2020.