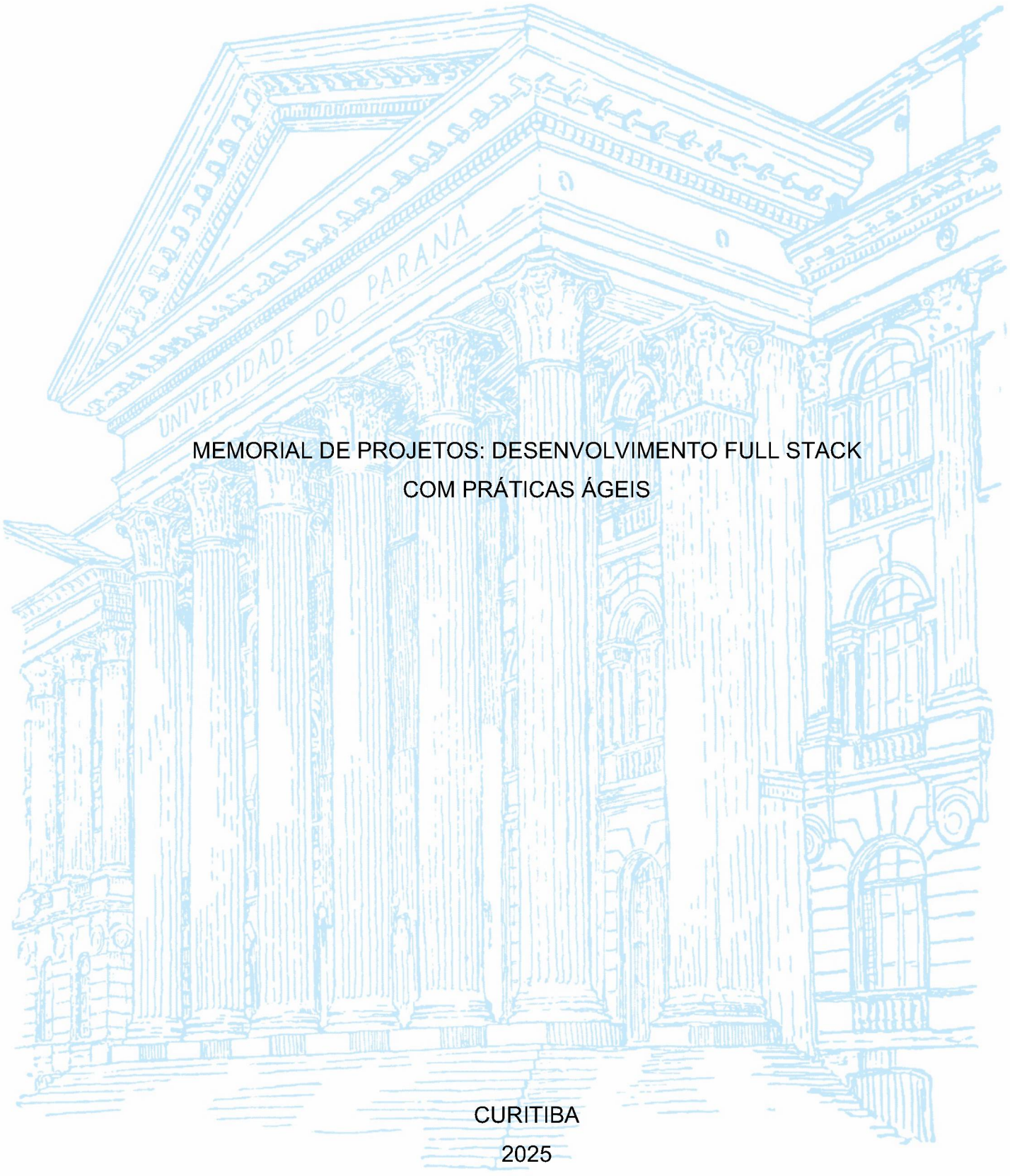
UNIVERSIDADE FEDERAL DO PARANÁ

ENZO GABRIEL BELLOTO FURLAN




# MEMORIAL DE PROJETOS: DESENVOLVIMENTO FULL STACK COM PRÁTICAS ÁGEIS

CURITIBA

2025

ENZO GABRIEL BELLOTO FURLAN

# MEMORIAL DE PROJETOS: DESENVOLVIMENTO FULL STACK COM PRÁTICAS ÁGEIS

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Desenvolvimento Ágil de Software, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Desenvolvimento Ágil de Software.

Orientador: Prof. Dr. Jaime Wojciechowski

CURITIBA

2025

MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO DESENVOLVIMENTO ÁGIL DE SOFTWARE - 40001016398E1

# TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação Desenvolvimento Ágil de Software da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **ENZO GABRIEL BELLOTO FURLAN**, intitulada: **MEMORIAL DE PROJETOS: DESENVOLVIMENTO FULL STACK COM PRÁTICAS ÁGEIS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua **aprovação** no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 10 de Junho de 2025.

JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora

RAFAELA MANTOVANI FONTANA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

## RESUMO

Este memorial de projetos consolida as principais atividades desenvolvidas ao longo da especialização em Desenvolvimento Ágil de *Software*, compondo um portfólio técnico focado na construção de soluções completas com uso de práticas ágeis. Os projetos abordam desde a modelagem inicial até a implementação de sistemas *web* e *mobile*, persistência de dados, automação de testes e entrega contínua com ferramentas de *DevOps*. A abordagem adotada reflete o perfil do desenvolvedor *full stack* moderno, com domínio de múltiplas tecnologias e atuação colaborativa em equipes ágeis. A organização dos artefatos evidencia a aplicação prática de princípios como iteração contínua, adaptação a mudanças e foco no valor entregue ao usuário. Além das competências técnicas, a vivência em projetos colaborativos fortaleceu habilidades interpessoais e de trabalho em equipe, fundamentais para a atuação em ambientes reais de desenvolvimento ágil.



Palavras-chave: Desenvolvimento Ágil; Desenvolvimento *Full Stack*; Aplicações *Web* e *Mobile*; Modelagem de *Software*;

## ABSTRACT

This project portfolio consolidates the main activities developed throughout the postgraduate program in Agile Software Development, presenting a technical body of work focused on building complete solutions using agile practices. The projects cover system modeling, implementation of web and mobile applications, data persistence, test automation, and continuous delivery using DevOps tools. The adopted approach reflects the profile of the modern full stack developer, combining technical versatility with collaborative work in agile teams. The organization of the deliverables demonstrates the practical application of principles such as continuous iteration, adaptation to change, and value-driven development. In addition to technical skills, collaborative project experience also fostered interpersonal abilities and teamwork, key aspects for effective performance in real-world agile environments.



Keywords: Agile Development; Full-Stack Development; Web and Mobile Applications; Software Modeling;

**SUMÁRIO**

# 1 PARECER TÉCNICO

O presente memorial, intitulado "Desenvolvimento *Full Stack* com Práticas Ágeis" consolida a trajetória prática do aluno ao longo da especialização em Desenvolvimento Ágil de *Software*. Este documento reúne grande parte dos projetos desenvolvidos nas disciplinas do curso, organizando-os sob uma perspectiva técnica e integrada, com foco na aplicação de princípios ágeis em todas as etapas do ciclo de vida do *software*.

Durante a especialização, foram desenvolvidos projetos que abrangem desde a modelagem de sistemas, passando pela implementação em diferentes plataformas (*Web* e *Mobile*), gerenciamento de ambientes com infraestrutura como código, persistência de dados, até a automação de testes. A prática foi orientada pelos valores do Manifesto Ágil (BECK et al., 2001), priorizando entrega contínua, colaboração e adaptação a mudanças.

Em disciplinas como MAG1 e MAG2, os fundamentos de modelagem foram aplicados na construção de diagramas de caso de uso, classes e sequências, refletindo uma estrutura sólida para os projetos implementados posteriormente em WEB2, MOB2 e BD. O gerenciamento ágil de projetos foi explorado em GAP1 e GAP2, com a elaboração de planos de release e simulações de fluxo de trabalho com *Kanban*, reforçando a importância do planejamento adaptativo e da gestão visual.

Na parte de desenvolvimento, os projetos de WEB1 e WEB2 envolveram a criação de aplicações *full stack* utilizando tecnologias modernas como *Angular*, *Bootstrap* e *Spring Boot*, com persistência em banco de dados *PostgreSQL*. Já nas disciplinas MOB1 e MOB2, foram desenvolvidos aplicativos Android, com foco em controle de dados locais e integração com APIs externas, utilizando práticas como corrotinas e chamadas assíncronas.

A disciplina de UX complementou os projetos com foco no design centrado no usuário, por meio da prototipação no *Figma* e da validação visual da experiência. Em TEST, a automação de testes foi aplicada com o uso do *framework Playwright*, evidenciando a importância da validação contínua no ciclo de desenvolvimento. Já em INFRA, o uso de *Docker* e *GitLab* permitiu simular um ambiente de versionamento real, fortalecendo o domínio de práticas relacionadas à integração e entrega contínua.

O título deste memorial, "Desenvolvimento *Full Stack* com Práticas Ágeis", reflete com precisão o escopo dos trabalhos realizados ao longo da especialização: a

construção de soluções completas, do *frontend* ao *backend*, com atenção à modularidade, testes, usabilidade e entrega de valor contínua. Essa abordagem condiz com o perfil de um desenvolvedor *full stack*, que precisa dominar múltiplas camadas tecnológicas e ser capaz de colaborar em times ágeis e multifuncionais (Bahrehvar & Moshirpour, 2022). A formação prática baseada em projetos, conforme proposta na especialização, permitiu aos alunos vivenciar o desenvolvimento de ponta a ponta de *software* seguindo um modelo de aprendizado ativo e orientado à prática real.

Entre os desafios enfrentados ao longo do curso, destacam-se a adaptação rápida a novas ferramentas, o equilíbrio entre entrega e qualidade, e a integração entre diferentes disciplinas e tecnologias. A maioria dos projetos foi desenvolvida em grupo, o que reforçou a importância de habilidades como comunicação contínua, divisão de responsabilidades e colaboração ativa. Esses elementos são apontados por Strode et al. (2022) como fundamentais para a eficácia de equipes ágeis, que devem apresentar confiança mútua e orientação ao time. A vivência em equipe proporcionou uma simulação realista do ambiente profissional, permitindo o desenvolvimento de competências interpessoais e de autogestão, essenciais para equipes ágeis e autônomas.

Assim, este memorial não apenas documenta a execução técnica dos projetos, mas também representa a evolução profissional do aluno, demonstrando sua capacidade de aplicar os princípios ágeis na prática, com entregas iterativas, foco em valor e aprendizado contínuo. Além das competências técnicas, a experiência colaborativa reforçou o desenvolvimento de habilidades como resolução de problemas em grupo, empatia e adaptabilidade. Esses elementos, como reforçado por Bahrehvar e Moshirpour (2022), são imprescindíveis na formação de profissionais preparados para lidar com os desafios de projetos reais e complexos, sendo valorizados tanto na indústria quanto na academia.

# 2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

A disciplina MADS teve como principal objetivo introduzir os fundamentos teóricos dos métodos ágeis de desenvolvimento de *software*, "preparando o terreno" para uma atuação mais consciente e fundamentada em projetos ágeis ao longo do curso. O projeto final consistiu na elaboração de um mapa mental, utilizando a ferramenta sugerida *MindX*, que organizasse e conectasse os principais conceitos relativos aos processos de *software* e suas abordagens tradicionais e ágeis.

O mapa mental desenvolvido abordou de forma estruturada os seguintes tópicos: Processo de *Software*, Modelos Tradicionais de Processo, Manifesto Ágil, Princípios Ágeis, *Lean* Software *Development, Scrum, Extreme Programming* (XP), *Kanban* e Entrega Contínua (Figuras 1, 2, 3 e 4). Cada tema foi detalhado com subtópicos que incluíam definições, características, práticas, valores e etapas típicas, promovendo uma visão abrangente e comparativa entre os modelos clássicos e as abordagens ágeis.

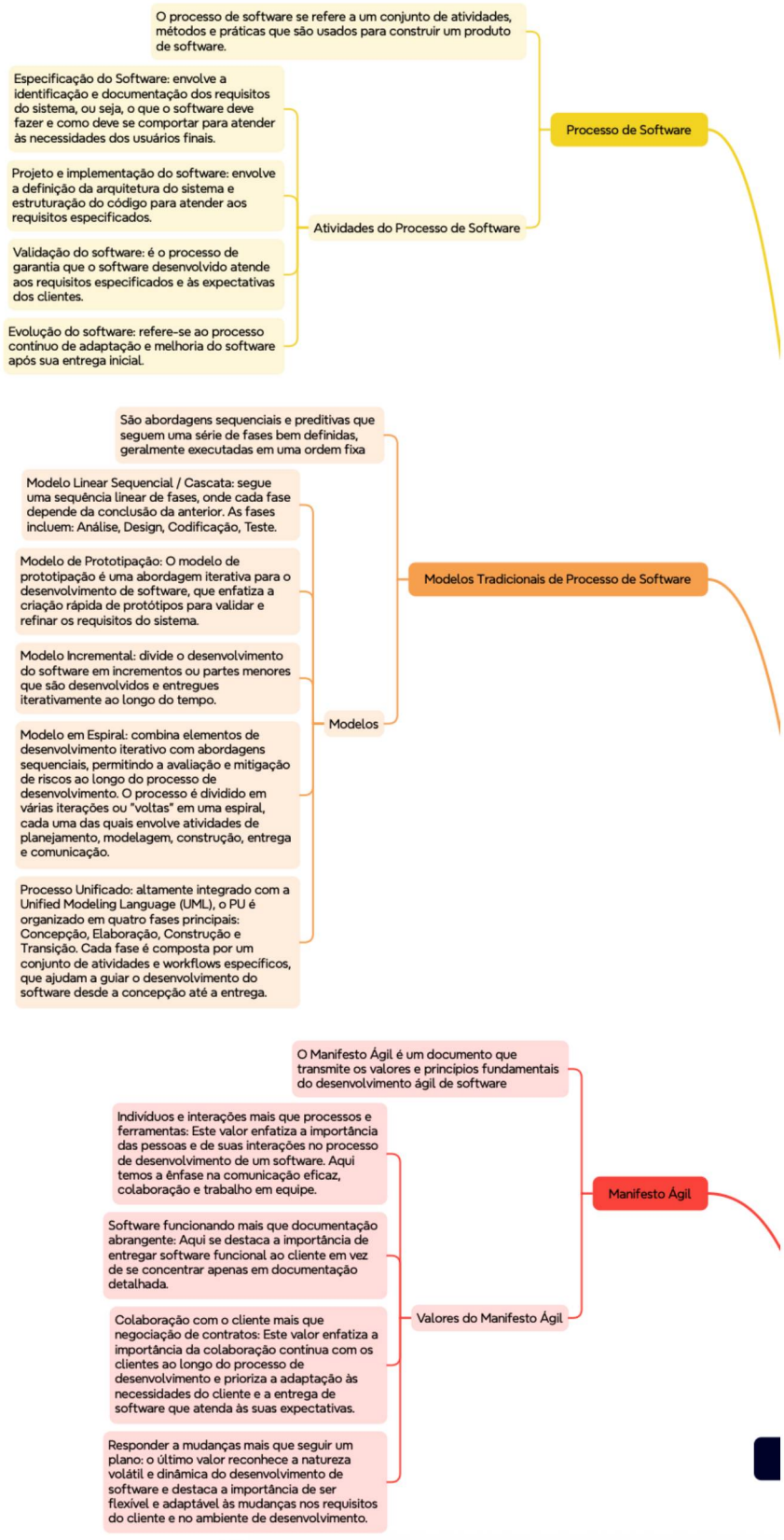
A construção do mapa mental proporcionou uma compreensão sistêmica das abordagens ágeis, destacando seus pilares — como colaboração, entrega contínua e adaptação — e demonstrando como esses princípios impactam diretamente todas as fases do desenvolvimento de *software*.

Essa atividade foi especialmente relevante para integrar conceitos que se manifestaram na execução dos projetos das seguintes disciplinas, em que práticas como iterações curtas, feedback contínuo, testes automatizados e foco no valor entregue ao cliente estiveram presentes.

A atividade também reforçou o papel da agilidade como mentalidade e não apenas como conjunto de métodos, evidenciando como a teoria bem assimilada contribui diretamente para decisões práticas mais eficazes ao longo do ciclo de vida do *software*.

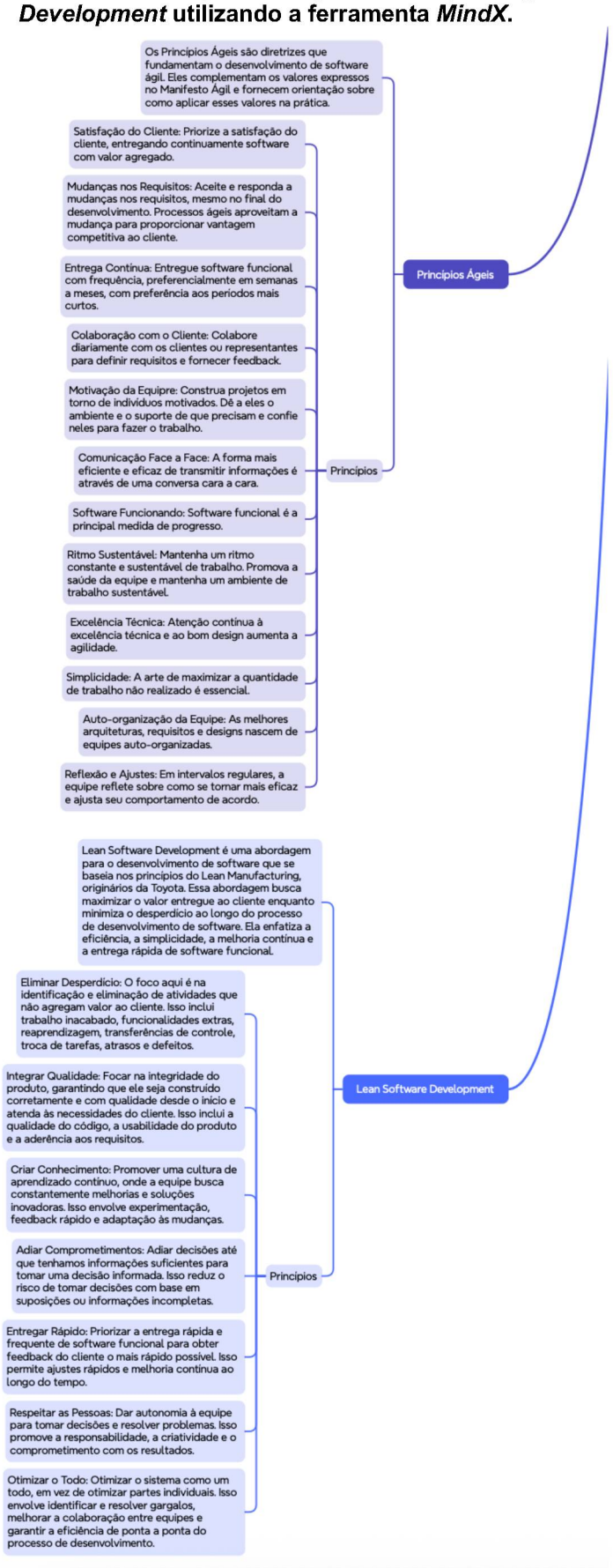
## 2.1 ARTEFATOS DO PROJETO

**Figura 1: Mapa mental desenvolvido para os tópicos Processo de *Software*, Modelos Tradicionais de Processo e Manifesto Ágil utilizando a ferramenta *MindX*.**



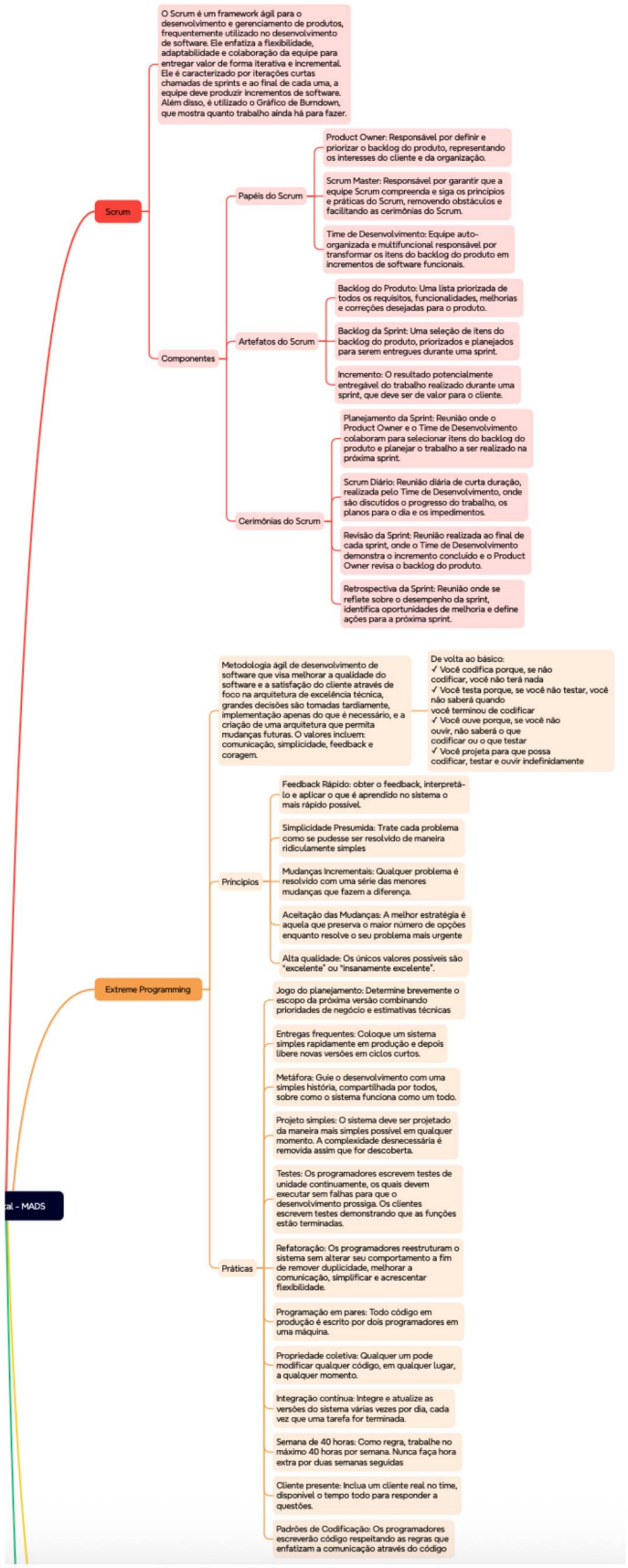

Fonte: Elaborado pelo autor.

**Figura 2: Mapa mental desenvolvido para os tópicos Princípios Ágeis, *Lean Software Development* utilizando a ferramenta *MindX*.**



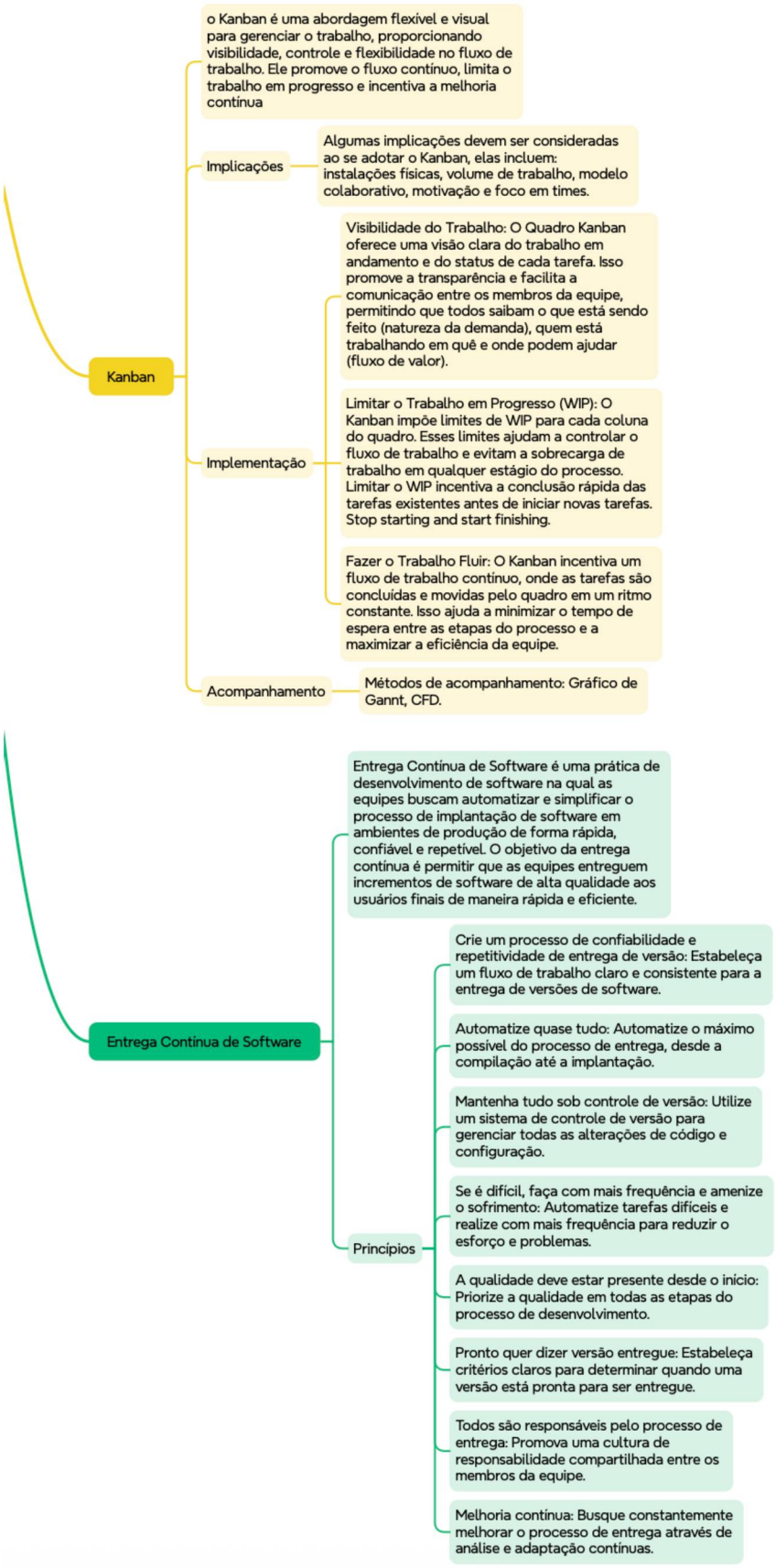

Fonte: Elaborado pelo autor.

**Figura 3: Mapa mental desenvolvido para os tópicos *Scrum* e *Extreme Programming* utilizando a ferramenta *MindX*.**




Fonte: Elaborado pelo autor.

**Figura 4: Mapa mental desenvolvido para os tópicos *Kanban* e Entrega Contínua de *Software* utilizando a ferramenta *MindX*.**




Fonte: Elaborado pelo autor.

# 3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2

As disciplinas MAG1 e MAG2 foram fundamentais para o aprofundamento na prática de modelagem de sistemas em um contexto ágil, oferecendo uma abordagem incremental e iterativa na construção de modelos de *software*. Ambas as disciplinas se articularam em torno de um projeto contínuo: a modelagem de um Sistema de Gestão de Condomínio, com a criação de diferentes artefatos em cada fase.

Na disciplina MAG1, o foco esteve na identificação e detalhamento dos requisitos funcionais por meio de artefatos de alto nível, como os Diagramas de Casos de Uso (níveis 1 e 2) (Figuras 5 e 6) e Histórias de Usuário (Figura 7 e 8), no formato "sendo... quero... para..." proposto por Cohn (2004). Essa fase foi essencial para compreender o escopo do sistema de forma colaborativa, com ênfase em comunicação clara com stakeholders, alinhada às práticas ágeis de priorização de funcionalidades e entrega de valor incremental.

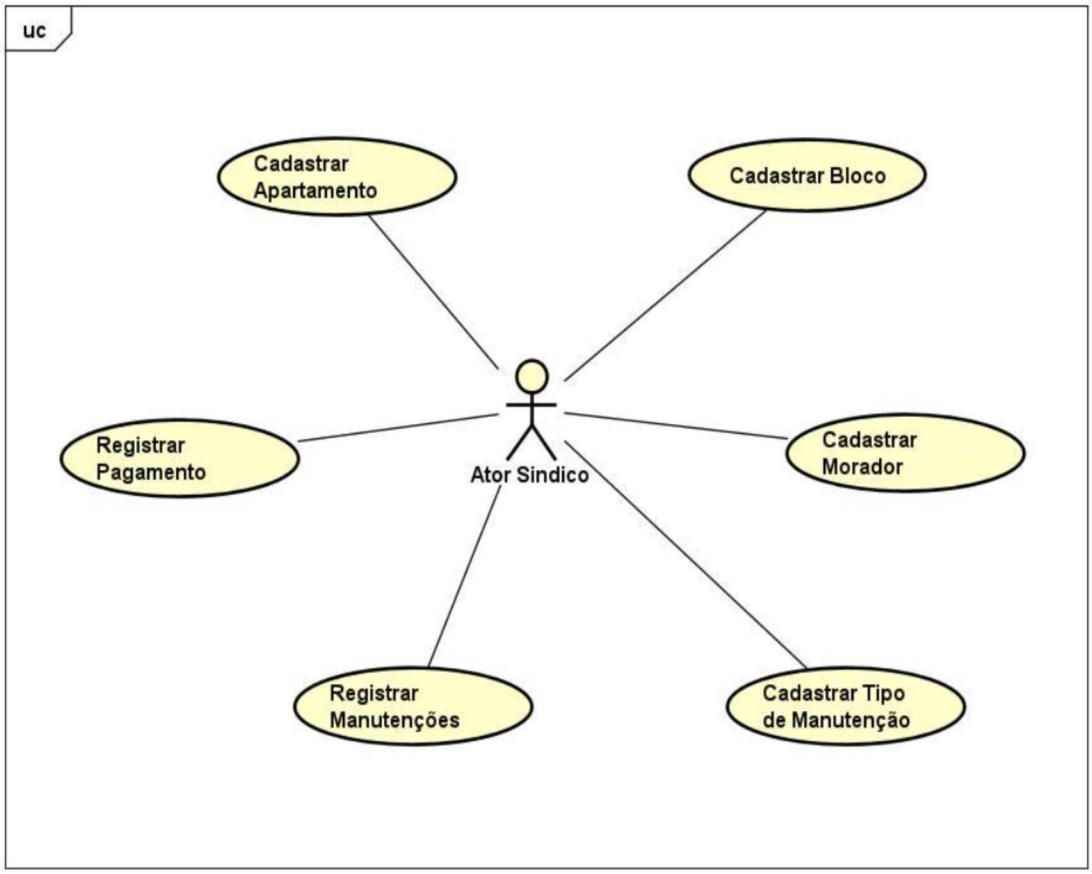
Já na disciplina MAG2, o projeto avançou para a modelagem estrutural e comportamental, com a produção de Diagramas de Sequência (Figura 9) para cada história de usuário e de Diagrama de Classes (Figura 10), incluindo o detalhamento das classes com atributos e estrutura do banco de dados relacional. Essa etapa permitiu a transição natural dos requisitos para um modelo técnico mais refinado, facilitando o desenvolvimento e testes subsequentes. Todos os diagramas foram feitos utilizando a ferramenta de modelagem UML *Astah*.

A importância dessas disciplinas no contexto ágil está na capacidade de modelar de forma enxuta, iterativa e evolutiva, evitando o excesso de documentação e focando nos elementos essenciais à construção incremental do *software*. As entidades e fluxos modelados neste projeto serviram como base estrutural para os sistemas desenvolvidos em WEB 1, WEB2 e BD, evidenciando como uma modelagem bem estruturada facilita a implementação técnica posterior.

Assim, MAG1 e MAG2 representaram um elo fundamental entre a engenharia de requisitos ágil e a implementação orientada a objetos, promovendo uma base sólida para o desenvolvimento técnico e conceitual dos projetos subsequentes. Como destaca Pressman (2016), a modelagem eficaz do sistema serve como alicerce para uma implementação bem estruturada, reforçando a importância de um planejamento cuidadoso para garantir a qualidade e a coerência do produto final.

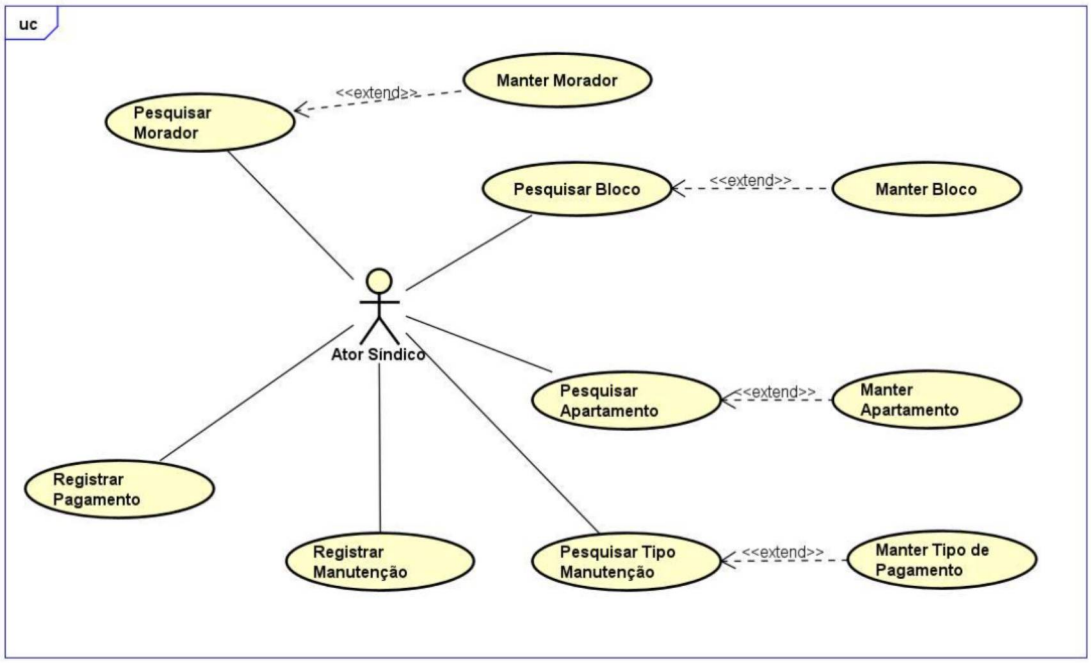
## 3.1 ARTEFATOS DO PROJETO

**Figura 5: Diagrama de Caso de Uso de Nível 1 de um sistema de gestão de condomínio feito utilizando a ferramenta *Astah*.**




Fonte: Elaborado pelo autor.

**Figura 6: Diagrama de Caso de Uso de Nível 2 de um sistema de gestão de condomínio feito utilizando a ferramenta *Astah*.**




Fonte: Elaborado pelo autor.

**Figura 7: Detalhamento de uma das História de Usuário, chamada de "Pesquisar Morador" utilizando a metodologia "Sendo, Quero, Para", além do desenho da tela e a lista de critérios de aceitação.**

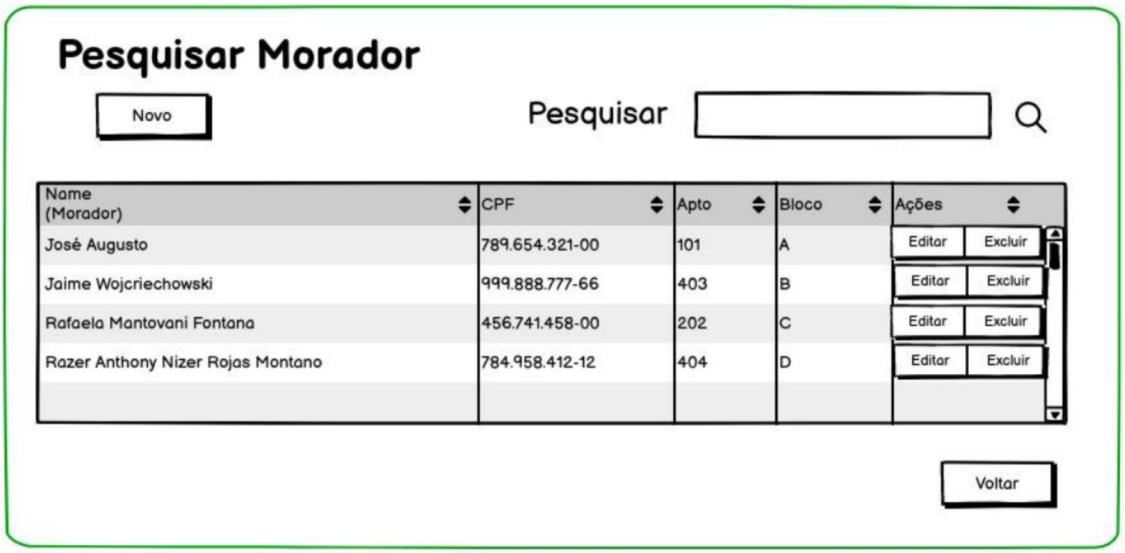
**HU001 – Pesquisar Morador**

**SENDO O Síndico**

**QUERO Pesquisar por moradores**

**PARA Fazer manutenções nos seus dados**

**DESENHO DA TELA**




**LISTA DE CRITÉRIOS DE ACEITAÇÃO**

1) Deve preencher a tela com todos os moradores cadastrados.
2) Deve permitir incluir um novo morador.
3) Deve permitir alterar dados de um morador cadastrado.
4) Deve permitir excluir um morador cadastrado.
5) Deve permitir consultar os dados de um morador cadastrado.
6) Deve pesquisar os moradores na tabela da tela.
7) Deve voltar à tela anterior.

Fonte: Elaborado pelo autor.

**Figura 8: Detalhamento dos critérios de aceitação de uma História de Usuário.**

**CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO**

1) Deve preencher a tela com todos os moradores cadastrados.

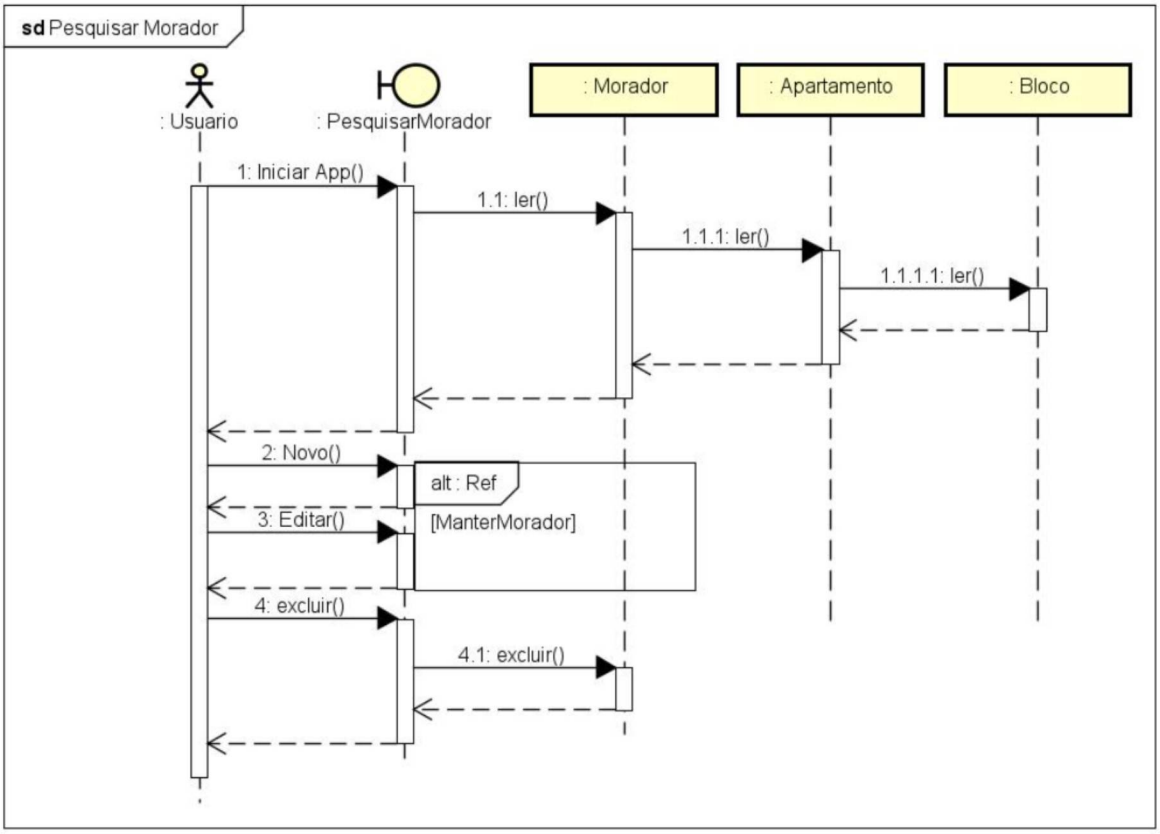
| **Dado que** | |
|---|---|
| **Quando** | A tela é apresentada |
| **Então** | A tabela da tela deve ser preenchida com todos os moradores cadastrados. |

2) Deve permitir incluir um novo morador.

| **Dado que** | Os moradores estão na tabela |
|---|---|
| **Quando** | O botão “Novo” é pressionado. |
| **Então** | O sistema chama a HU002 - Manter Morador passando o parâmetro “Novo”. |

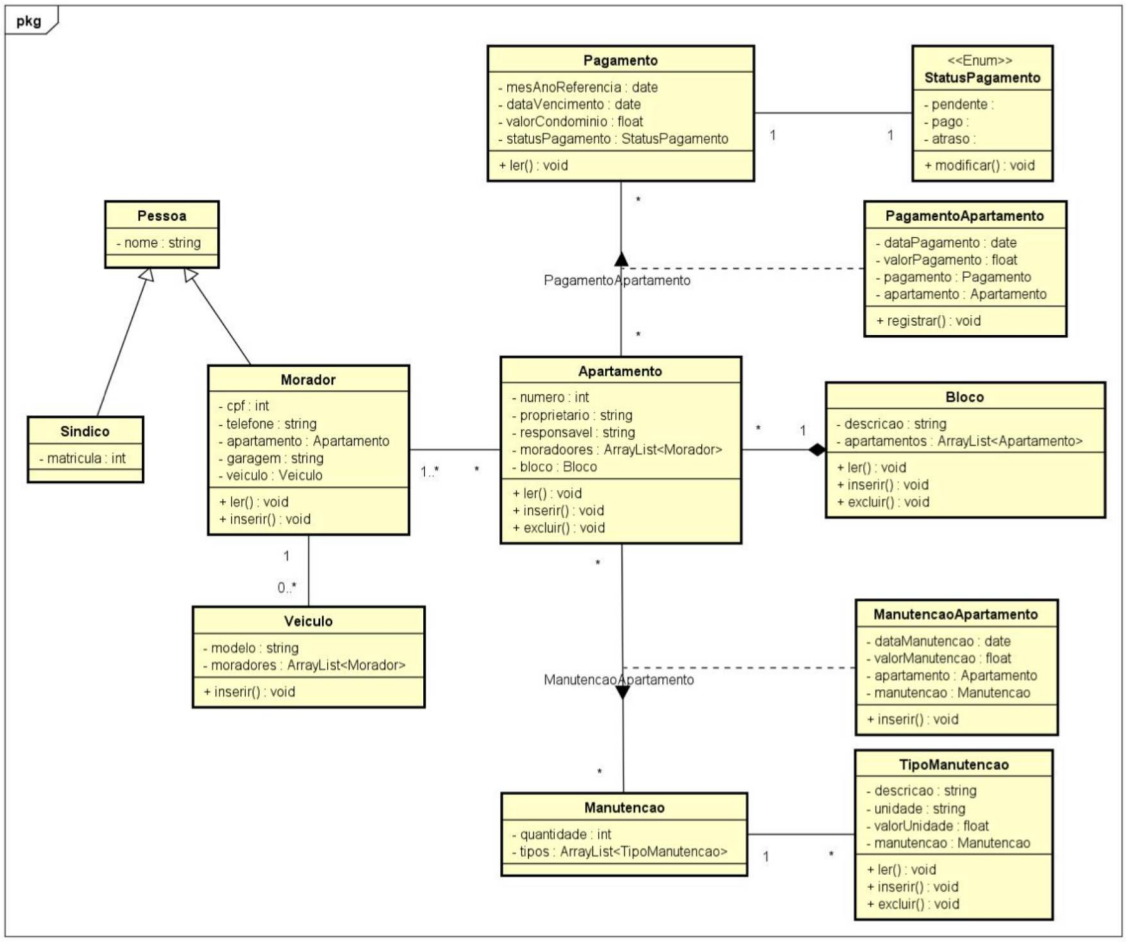
Fonte: Elaborado pelo autor.

**Figura 9: Diagrama de Sequência de uma História de Usuário feito utilizando a ferramenta *Astah*.**




Fonte: Elaborado pelo autor.

**Figura 10: Diagrama de Classes do sistema feito utilizando a ferramenta *Astah*.**




Fonte: Elaborado pelo autor.

# 4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2

As disciplinas GAP1 e GAP2 foram essenciais para o desenvolvimento da mentalidade de planejamento e acompanhamento contínuo em projetos ágeis, utilizando práticas reais de *Scrum* e *Kanban*. Ambas as disciplinas promoveram a experimentação prática da gestão de tempo, estimativas, priorização de *backlog* e medição de desempenho, alinhando-se diretamente à rotina de um projeto de desenvolvimento. Essa abordagem, que enfatiza a adoção de ciclos curtos e planejamento adaptativo, está em conformidade com as recomendações de Highsmith (2009), que reforça a importância da inovação contínua na gestão de produtos para o sucesso em ambientes ágeis.

Na disciplina GAP1, o projeto proposto foi elaborar um plano de *release* completo para um *software* idealizado pelo aluno, incluindo estimativas de esforço, organização por *sprints* e planejamento baseado em histórias de usuário. A atividade exigiu a compreensão e aplicação dos conceitos de velocidade de equipe, cálculo de pontos por *sprint* (Figura 11) e divisão do *backlog* em incrementos viáveis, simulando um projeto real (Figura 12). Essa prática foi fundamental para a construção de uma visão abrangente, estratégica e tática de um projeto ágil desde sua concepção.

Já em GAP2, a proposta foi ainda mais prática e dinâmica: a realização de um simulador de fluxo de trabalho no modelo *Kanban* utilizando o ambiente do *kanbanboardgame.com*. Durante os 35 dias simulados, o aluno atuou como gestor de um time de desenvolvimento, tomando decisões sobre priorização de tarefas, balanceamento de *WIP* (trabalho em progresso), gestão de gargalos e resposta a eventos inesperados (Figuras 13 e 14). A análise do CFD (*Cumulative Flow Diagram*) ao final do jogo (Figura 15) evidenciou os aprendizados sobre previsibilidade, estabilidade e *throughput* – aspectos centrais para a entrega contínua de valor em projetos reais.

Essas duas disciplinas ofereceram fundamentos concretos para o planejamento ágil e a adaptação constante, competências diretamente aplicadas nos projetos subsequentes da especialização. Mais do que planejamento formal, GAP1 e GAP2 ensinaram o aluno a agir e reagir com agilidade, tomando decisões baseadas em dados e *feedback* contínuo.

## 4.1 ARTEFATOS DO PROJETO

**Figura 11: Detalhamento da estimativa de pontos por HU.**

**Projeto Loja Virtual - Histórias de Usuário**

| Nome | Descrição | Estimativa em Pontos |
|---|---|---|
| HU001: Cadastro de Usuário | SENDO um usuário<br>QUERO criar uma conta no sistema<br>PARA acessar seus recursos | 2 |
| HU002: Edição de Dados Cadastrais | SENDO um usuário logado<br>QUERO editar meus dados cadastrais<br>PARA manter meus dados atualizados | 1 |
| HU003: Navegação por Produtos | SENDO um usuário logado<br>QUERO navegar pelos produtos disponíveis na loja<br>PARA iniciar uma compra | 1 |
| HU004: Adição de Produtos ao Carrinho de Compras | SENDO um usuário logado<br>QUERO quero adicionar produtos ao carrinho de compras<br>PARA realizar uma compra | 1 |
| HU005: Finalização da Compra | SENDO um usuário logado<br>QUERO finalizar a compra de produtos<br>PARA finalizar uma compra | 1 |
| HU006: Consulta do Status de Pedidos | SENDO um usuário logado<br>QUERO consultar o status dos meus pedidos<br>PARA acompanhar o status da minha compra | 1 |
| HU007: Rastreamento da Entrega de Pedidos | SENDO um usuário logado<br>QUERO rastrear a entrega dos meus pedidos<br>PARA acompanhar a entrega da minha compra. | 1 |
| HU008: Escolha do Método de Pagamento | SENDO um usuário logado<br>QUERO escolher um método de pagamento<br>PARA finalizar a compra | 2 |
| HU009: Cadastro de Produtos (Perfil de Administrador) | SENDO um usuário logado com perfil de administrador<br>QUERO cadastrar novos produtos no sistema<br>PARA atualizar os produtos da loja | 2 |

Fonte: Elaborado pelo autor.

**Figura 12: Detalhamento da *Sprint* incluindo as HU priorizadas.**

**Cálculo da Velocidade:**

| | | | |
|---|---|---|---|
| Horas disponíveis por dia: | **5 horas por dia** | Tamanho da Sprint: | **1 semana** |
| Horas disponíveis por Sprint: | 5 horas * 5 dias = **25 horas por semana** | Velocidade: | 25 horas por semana / 8 horas por ponto = **3 pontos por Sprint** |

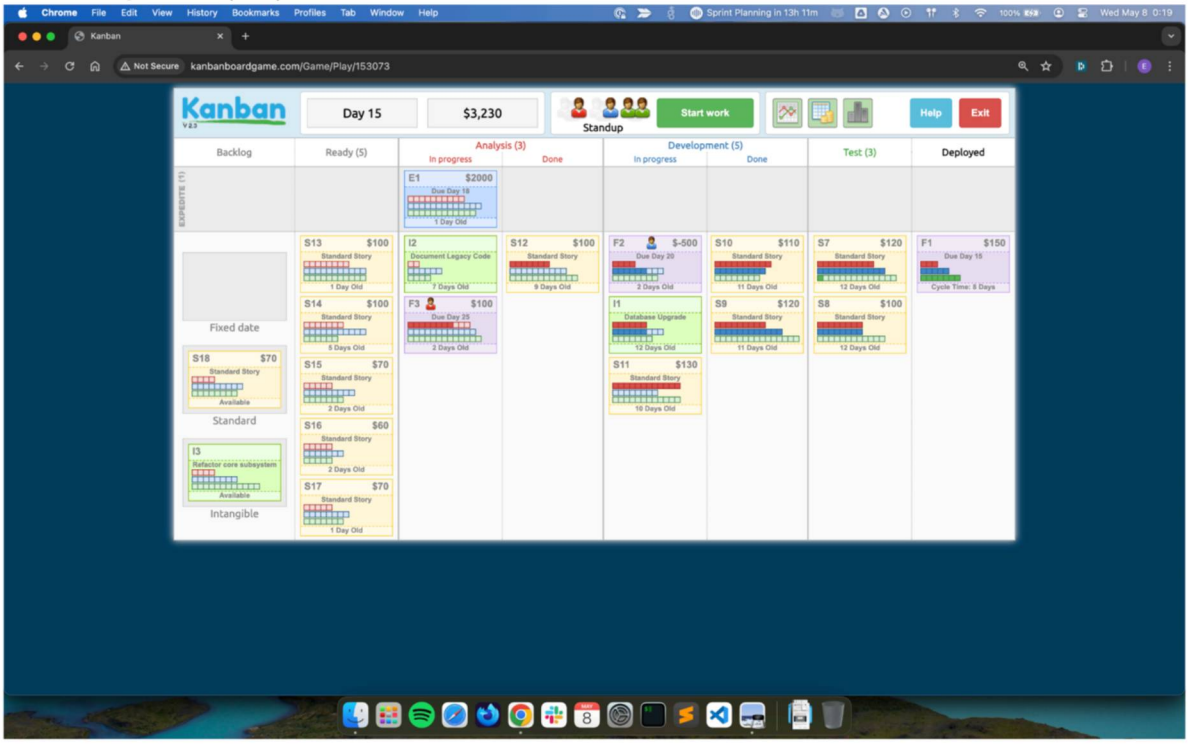
**Plano de Release:**

| Iteração/Sprint 1 | Iteração/Sprint 2 | Iteração/Sprint 3 | Iteração/Sprint 4 |
|---|---|---|---|
| Data Início: 06/05/2024 | Data Início: 13/05/2024 | Data Início: 20/05/2024 | Data Início: 27/05/2024 |
| Data Fim: 10/05/2024 | Data Fim: 17/05/2024 | Data Fim: 24/05/2024 | Data Fim: 31/05/2024 |
| HU001: Cadastro de Usuário<br><br>SENDO um usuário<br>QUERO criar uma conta no sistema<br>PARA acessar seus recursos<br>ESTIMATIVA 2 pontos | HU003: Navegação por Produtos<br><br>SENDO um usuário logado<br>QUERO navegar pelos produtos disponíveis na loja<br>PARA iniciar uma compra<br>ESTIMATIVA 1 ponto | HU006: Consulta do Status de Pedidos<br><br>SENDO um usuário logado<br>QUERO consultar o status dos meus pedidos<br>PARA acompanhar o status da minha compra<br>ESTIMATIVA 1 ponto | HU008: Escolha do Método de Pagamento - PARTE 2<br><br>SENDO um usuário logado<br>QUERO escolher um método de pagamento<br>PARA finalizar a compra<br>ESTIMATIVA 1 ponto |
| HU002: Edição de Dados Cadastrais<br><br>SENDO um usuário logado<br>QUERO editar meus dados cadastrais<br>PARA manter meus dados atualizados<br>ESTIMATIVA 1 ponto | HU004: Adição de Produtos ao Carrinho de Compras<br><br>SENDO um usuário logado<br>QUERO quero adicionar produtos ao carrinho de compras<br>PARA realizar uma compra<br>ESTIMATIVA 1 ponto | HU007: Rastreamento da Entrega de Pedidos<br><br>SENDO um usuário logado<br>QUERO rastrear a entrega dos meus pedidos<br>PARA acompanhar a entrega da minha compra.<br>ESTIMATIVA 1 ponto | HU009: Cadastro de Produtos (Perfil de Administrador)<br><br>SENDO um usuário logado com perfil de administrador<br>QUERO cadastrar novos produtos no sistema<br>PARA atualizar os produtos da loja<br>ESTIMATIVA 2 pontos |
| | HU005: Finalização da Compra<br><br>SENDO um usuário logado<br>QUERO finalizar a compra de produtos<br>PARA finalizar uma compra<br>ESTIMATIVA 1 ponto | HU008: Escolha do Método de Pagamento - PARTE 1<br><br>SENDO um usuário logado<br>QUERO escolher um método de pagamento<br>PARA finalizar a compra<br>ESTIMATIVA 1 ponto | |

Fonte: Elaborado pelo autor.

**Figura 13: Detalhamento da *Sprint* simulada utilizando o website *kanbanboardgame.com.***

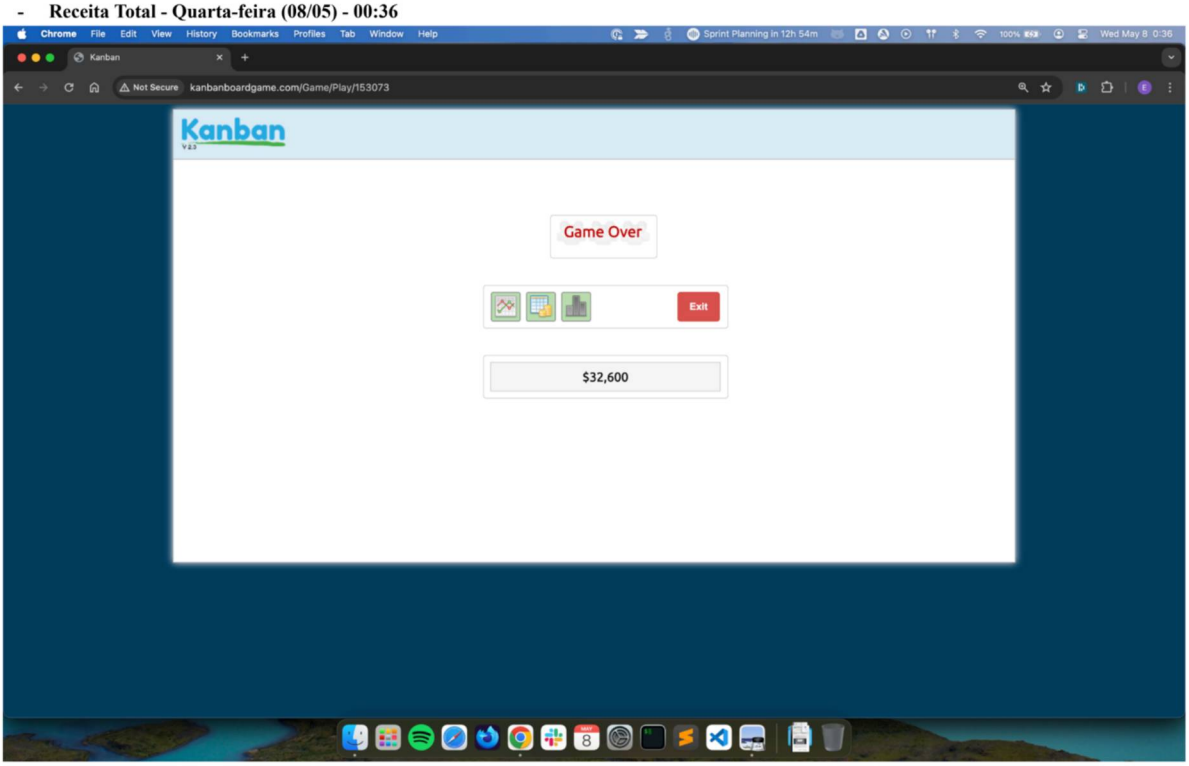
- **Dia 15 - Quarta-feira (08/05) - 00:19**




Fonte: Elaborado pelo autor.

**Figura 14: Valor financeiro fictício ganho após *Sprint* simulada utilizando o website *kanbanboardgame.com.***

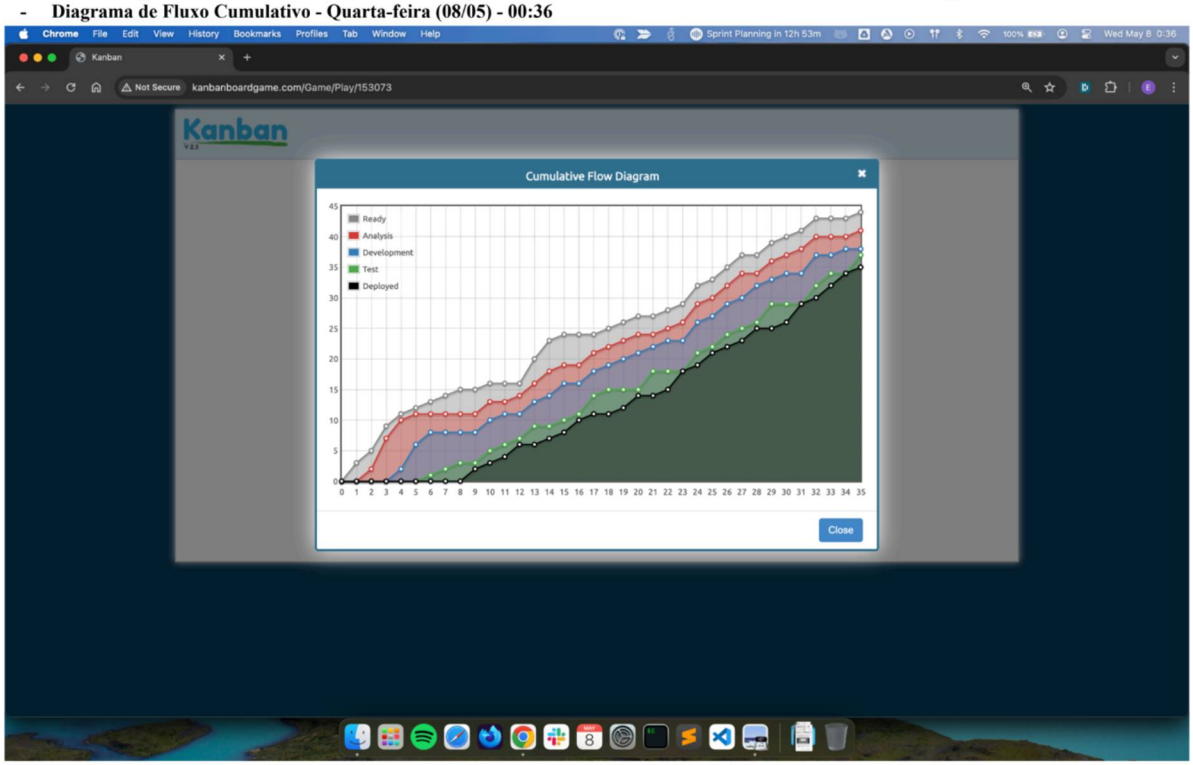
- **Receita Total - Quarta-feira (08/05) - 00:36**




Fonte: Elaborado pelo autor.

**Figura 15: CFD após simulação utilizando o website *kanbanboardgame.com.***

- **Diagrama de Fluxo Cumulativo - Quarta-feira (08/05) - 00:36**




Fonte: Elaborado pelo autor.

# 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

A disciplina INTRO teve como principal objetivo proporcionar aos alunos o primeiro contato prático com conceitos fundamentais de desenvolvimento de *software* orientado a objetos, bem como com testes automatizados, persistência em banco de dados e integração com sistemas reais. O projeto final da disciplina consistiu na implementação de um *backend* de um sistema bancário simplificado, simulando operações de cadastro de clientes, contas correntes e contas investimento.

Esse projeto permitiu aplicar conhecimentos essenciais de orientação a objetos, como herança, encapsulamento, polimorfismo e associação entre classes, além da prática com *Java* e uso de um banco de dados relacional (*MySQL*) a partir de um script DDL fornecido. O desenvolvimento foi orientado por 42 testes unitários já prontos, utilizando a biblioteca *JUnit*, e o objetivo era realizar correções no código para que seus testes unitários fossem aprovados, o que exigiu entendimento das regras de negócio, boa estruturação de código e domínio das boas práticas de desenvolvimento.

A experiência prática adquirida nesta disciplina foi essencial para a formação da base técnica dos projetos subsequentes, especialmente pela ênfase na validação contínua via testes automatizados e adoção de persistência em banco de dados relacional.

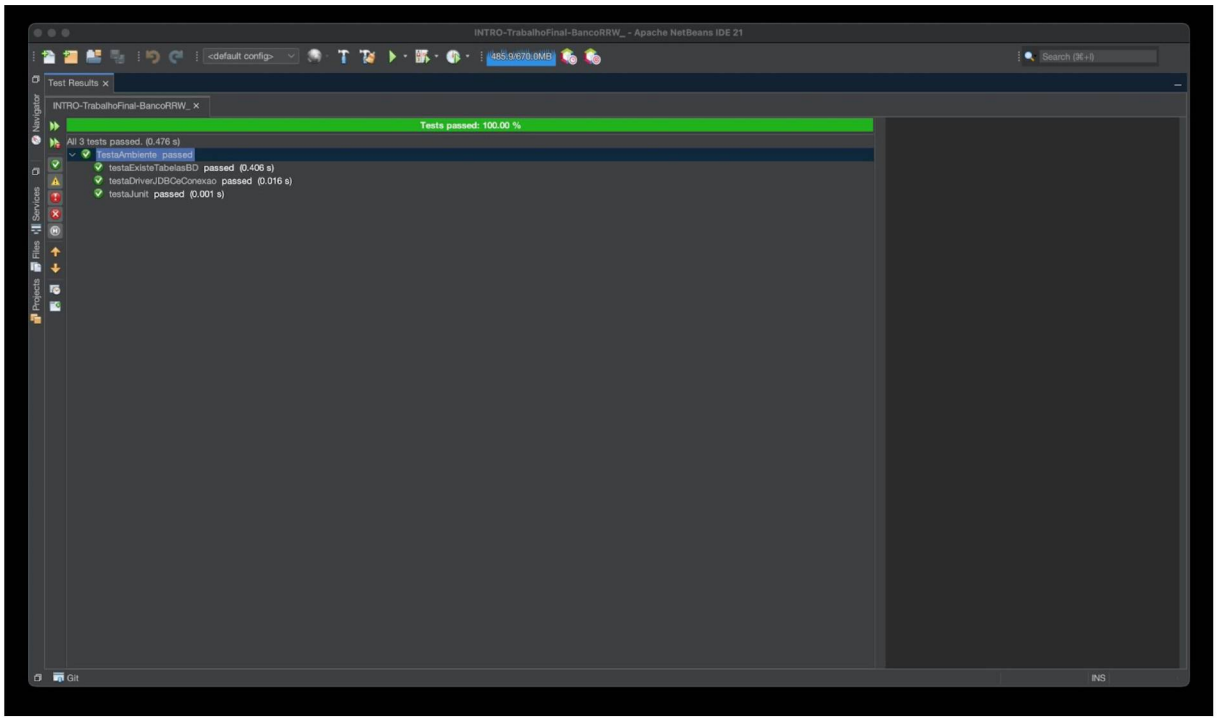
A entrega do projeto consistiu em um pacote com:

- As classes Java implementadas e reparadas, que atendiam aos requisitos e validavam os testes.
- Os arquivos do projeto completo do NetBeans, com todos os testes agora funcionando.
- E um PDF contendo uma página de evidência, com imagens da execução dos testes na IDE. (Figuras 16, 17 e 18)

Essa experiência representou o primeiro ciclo completo de desenvolvimento *backend* com qualidade validada, contribuindo significativamente para a visão prática da engenharia de *software* moderna.

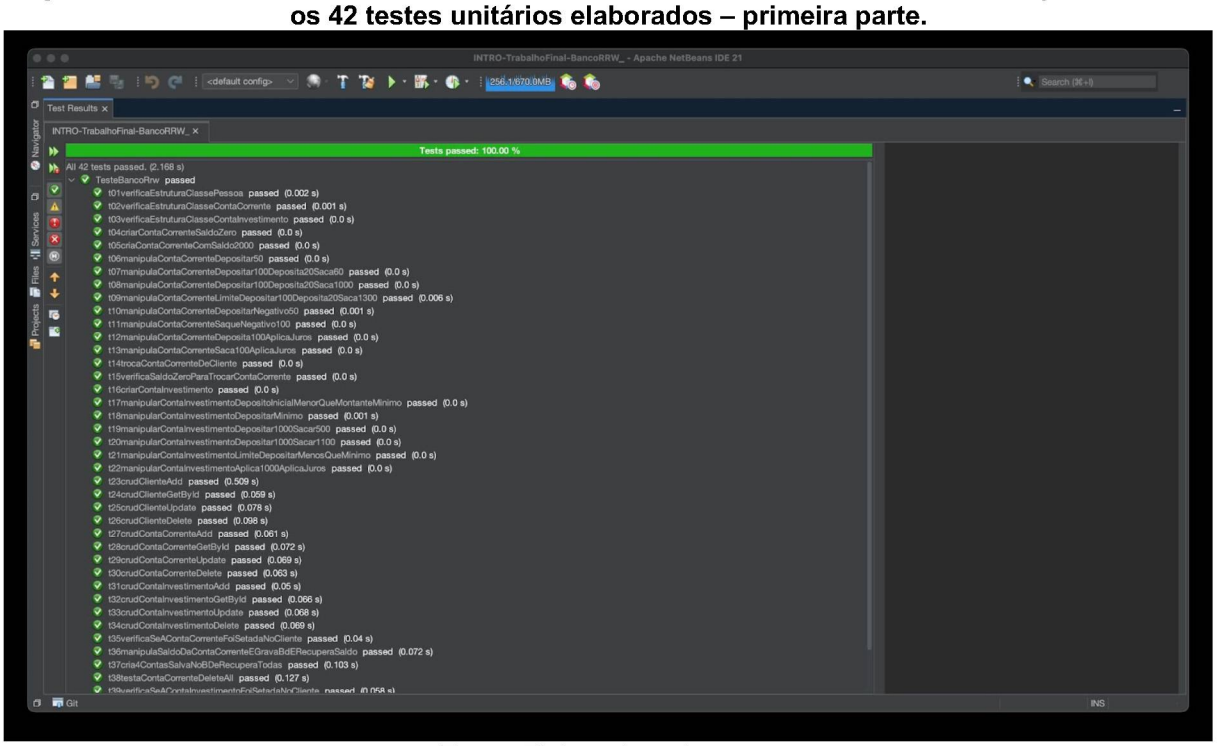
## 5.1 ARTEFATOS DO PROJETO

**Figura 16: Janela do ambiente na IDE NetBeans mostrando os testes de conexão com o banco de dados.**



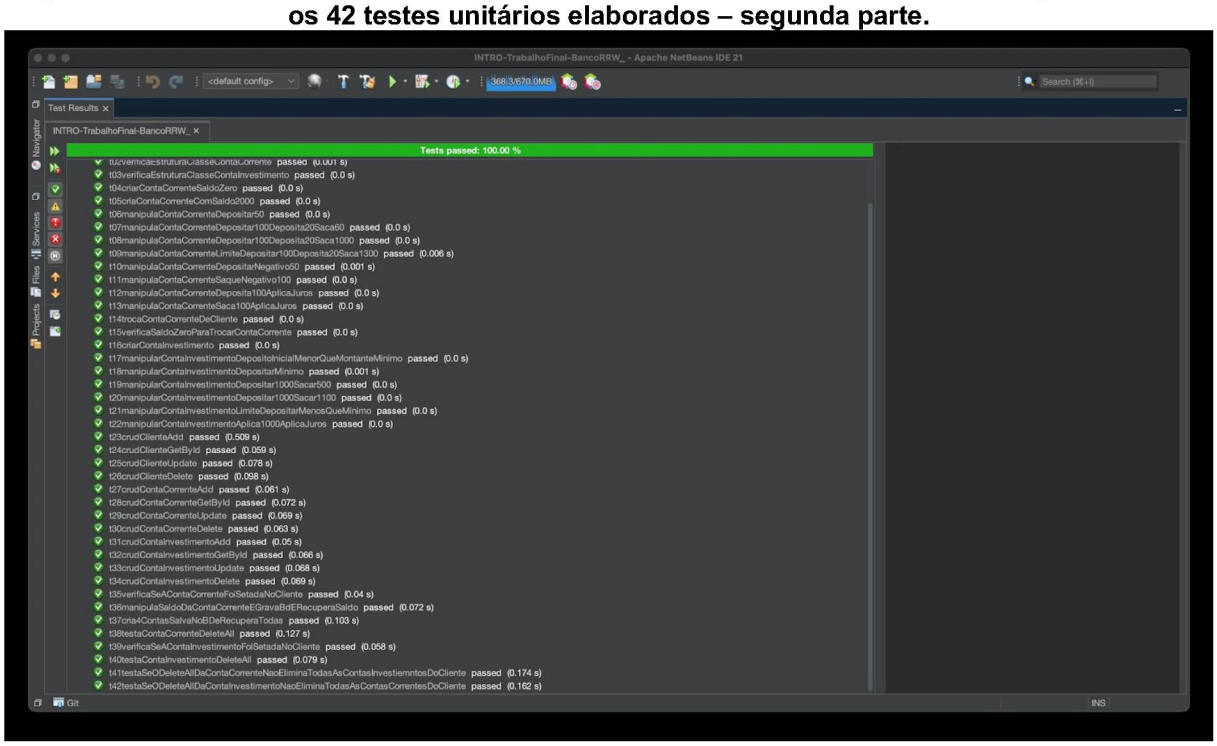

Fonte: Elaborado pelo autor.

**Figura 17: Janela do ambiente na IDE *NetBeans* mostrando o resultado da execução de todos os 42 testes unitários elaborados – primeira parte.**




Fonte: Elaborado pelo autor.

**Figura 18: Janela do ambiente na IDE *NetBeans* mostrando o resultado da execução de todos os 42 testes unitários elaborados – segunda parte.**




Fonte: Elaborado pelo autor.

# 6 DISCIPLINA: BD – BANCO DE DADOS

A disciplina de Banco de Dados teve como foco central o desenvolvimento de competências relacionadas à modelagem de dados, normalização, e implementação prática em SQL, pilares fundamentais para qualquer sistema de informação moderno. O projeto final da disciplina foi dividido em duas partes complementares: uma voltada à modelagem conceitual e lógica de um sistema real (controle de uma biblioteca), e outra dedicada à concepção de um projeto original, com modelagem completa e implementação via scripts SQL, ambos utilizando o sistema de gerenciamento de banco de dados *MySQL* e sua ferramenta de visualização *MySQL Workbench*.

Na primeira parte, foi solicitado o desenvolvimento de um modelo conceitual (entidade-relacionamento) (Figura 19) e do correspondente modelo lógico relacional (Figura 20), a partir dos requisitos funcionais de um sistema de controle de biblioteca. O projeto envolveu a representação correta de entidades como Obra, Editora, Usuário, Funcionário, Departamento, Empréstimo e Reserva, bem como o mapeamento adequado dos relacionamentos e suas cardinalidades.

Este exercício foi essencial para o entendimento das boas práticas de modelagem, identificação de chaves primárias e estrangeiras, além da definição correta dos tipos de relacionamentos (1x1, 1xN e NxN).

A segunda parte do projeto envolveu a criação de um sistema de banco de dados completo, com tema livre, desde que contemplasse todos os tipos de relacionamento (1x1, 1xN, NxN).

O tema escolhido foi um sistema de gerenciamento de eventos, no qual são armazenadas informações sobre eventos, seus participantes, organizadores e a associação entre eventos e participantes. Esse sistema foi modelado de forma a evidenciar diferentes tipos de relacionamentos, conforme exigido no enunciado:

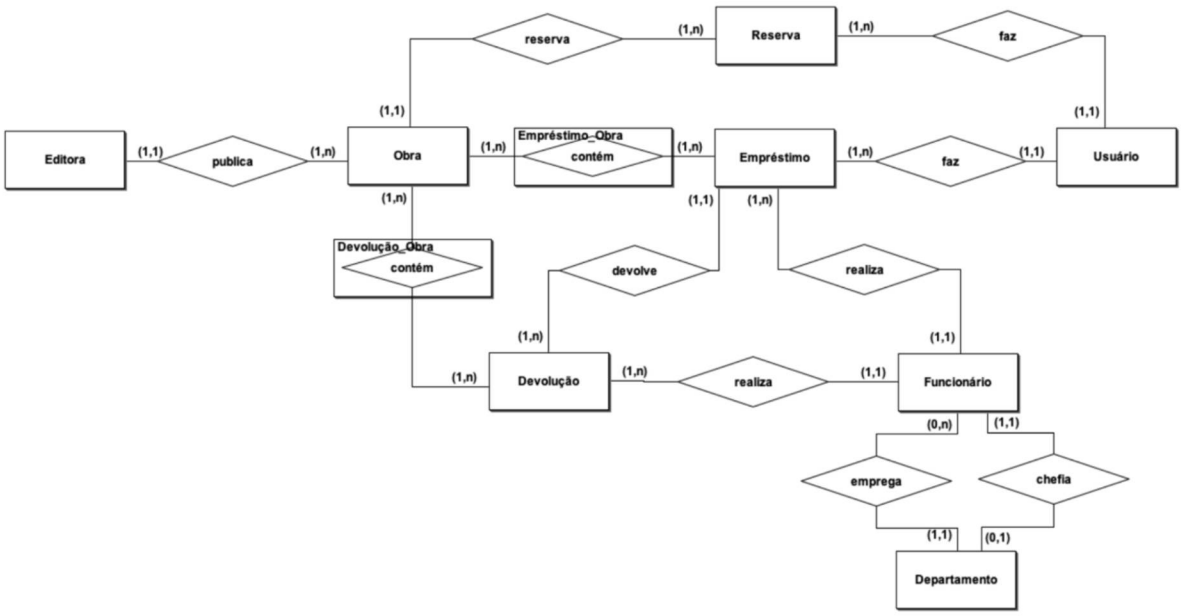
- Relacionamento 1x1: entre cada evento e seu respectivo endereço (cada evento ocorre em um endereço único).
- Relacionamento 1xN: entre organizador e eventos (um organizador pode organizar vários eventos).
- Relacionamento NxN: entre eventos e participantes (um evento pode ter vários participantes, e um participante pode estar em vários eventos).

Cada organizador possui um nome, telefone e e-mail. Os eventos cadastrados têm um nome, descrição, data e endereço. Os participantes, por sua vez, possuem nome, e-mail e telefone. O modelo foi implementado com um diagrama lógico (Figura 21), seguido por um script SQL completo contendo a criação das tabelas com chaves primárias e estrangeiras (Figura 22), além da inserção de dados de exemplo com no mínimo cinco registros por tabela e da consulta desses dados para asserção da correta criação do banco. (Figuras 23 e 24)

Este projeto reforçou a importância da modelagem adequada para garantir a integridade dos dados, além de demonstrar, na prática, como representar diferentes tipos de relacionamentos e como traduzi-los para estruturas relacionais no banco de dados. A experiência adquirida foi fundamental para os demais projetos do curso como em WEB2, que exigiram integração entre *backend*, banco de dados e interface do usuário.

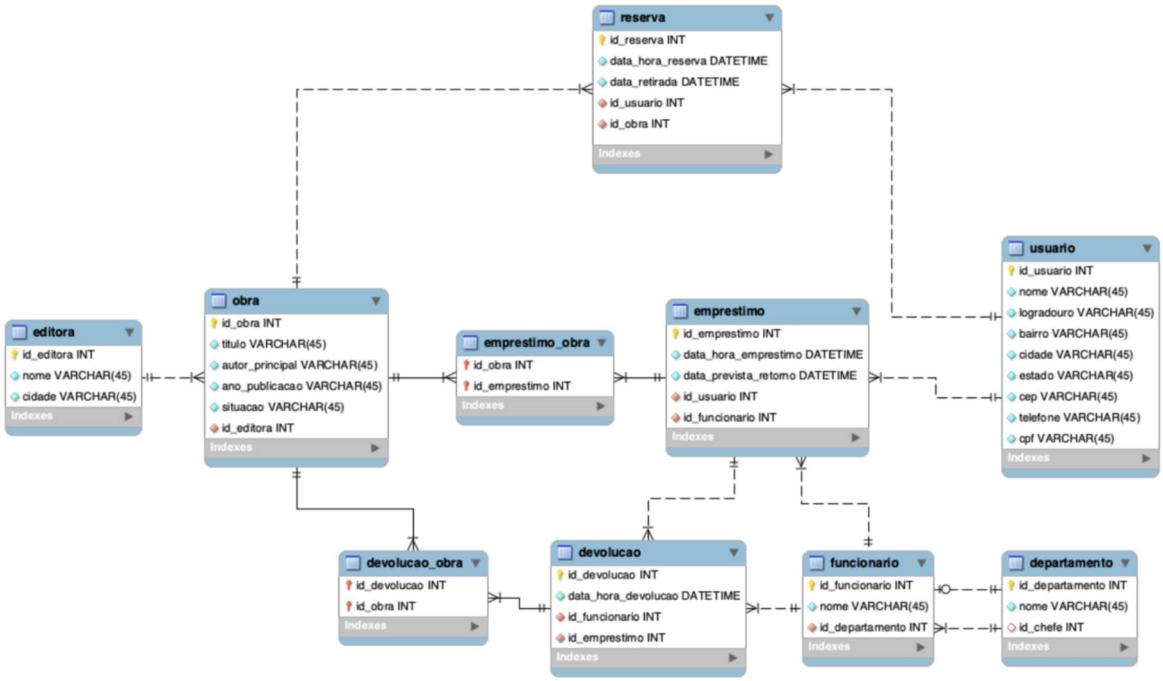
## 6.1 ARTEFATOS DO PROJETO

**Figura 19: Modelo entidade-relacionamento conceitual contendo somente as entidades e os relacionamentos a partir dos requisitos funcionais de um sistema de controle de biblioteca.**




Fonte: Elaborado pelo autor.

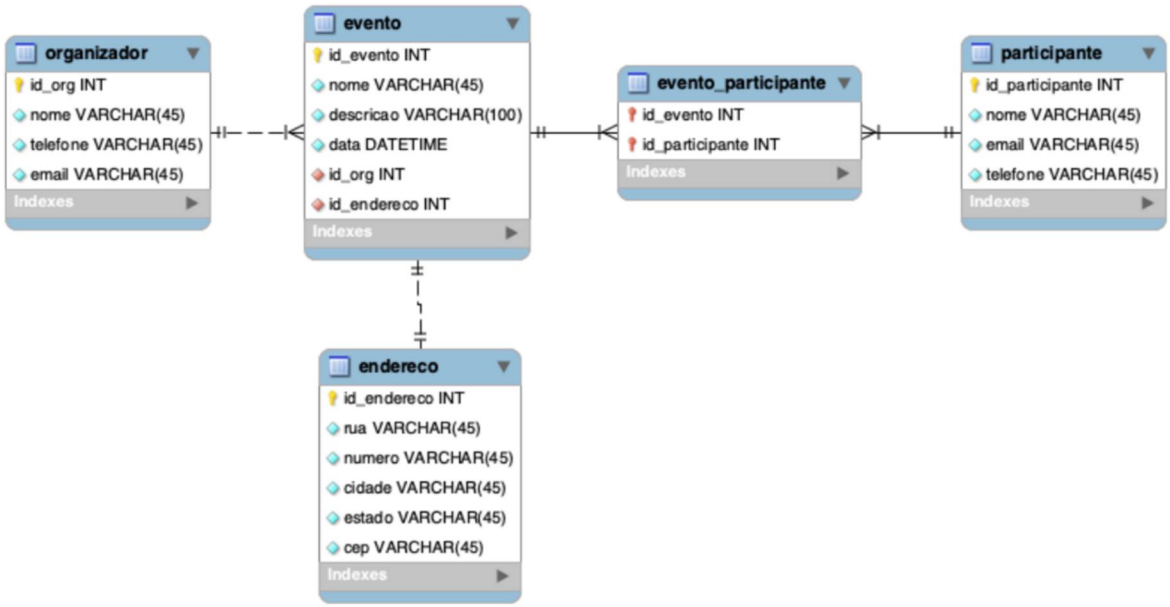
**Figura 20: Modelo lógico contendo tabelas, campos, chaves primárias e estrangeiras e relacionamentos a partir dos requisitos funcionais de um sistema de controle de biblioteca feito no *MySQL Workbench*.**




Fonte: Elaborado pelo autor.

**Figura 21: Diagrama lógico contendo tabelas, campos, chaves primárias e estrangeiras e relacionamentos do projeto de "gerenciamento de eventos" feito no *MySQL Workbench*.**

**Modelo Lógico:**



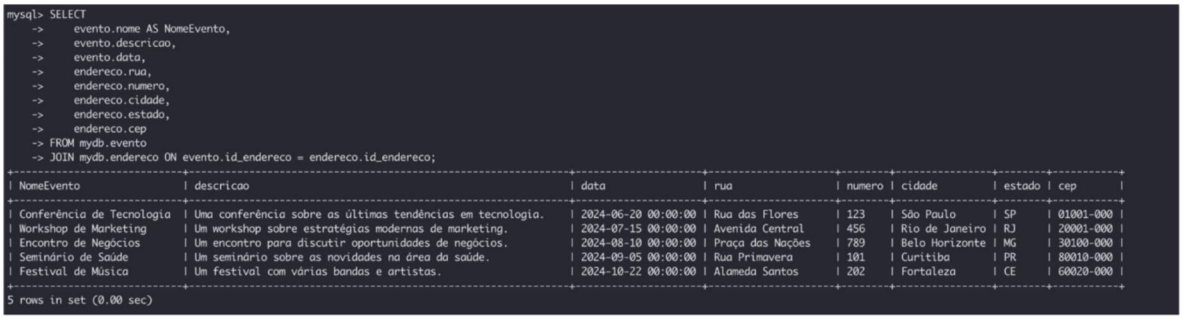

Fonte: Elaborado pelo autor.

**Figura 22: Parte do script de criação do banco contendo a criação das tabelas do projeto com chaves primárias e estrangeiras, gerado automaticamente pela ferramenta *MySQL Workbench*.**

```
-- -----------------------------------------------------
-- Table `mydb`.`organizador`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`organizador` (
  `id_org` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `telefone` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_org`),
  UNIQUE INDEX `telefone_UNIQUE` (`telefone` ASC) VISIBLE,
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,
  UNIQUE INDEX `id_org_UNIQUE` (`id_org` ASC) VISIBLE)
ENGINE = InnoDB;
```

Fonte: Elaborado pelo autor.

**Figura 23: Consulta exemplo “Selecionar todos os eventos e seus respectivos endereços” para asserção do banco criado.**



```
mysql> SELECT
    ->     evento.nome AS NomeEvento,
    ->     evento.descricao,
    ->     evento.data,
    ->     endereco.rua,
    ->     endereco.numero,
    ->     endereco.cidade,
    ->     endereco.estado,
    ->     endereco.cep
    -> FROM mydb.evento
    -> JOIN mydb.endereco ON evento.id_endereco = endereco.id_endereco;
+-------------------------+-----------------------------------------------------+---------------------+-----------------+--------+----------------+--------+----------+
| NomeEvento              | descricao                                           | data                | rua             | numero | cidade         | estado | cep      |
+-------------------------+-----------------------------------------------------+---------------------+-----------------+--------+----------------+--------+----------+
| Conferência de Tecnologia | Uma conferência sobre as últimas tendências em tecnologia. | 2024-06-20 00:00:00 | Rua das Flores  | 123    | São Paulo      | SP     | 01001-000 |
| Workshop de Marketing   | Um workshop sobre estratégias modernas de marketing.  | 2024-07-15 00:00:00 | Avenida Central | 456    | Rio de Janeiro | RJ     | 20001-000 |
| Encontro de Negócios    | Um encontro para discutir oportunidades de negócios.  | 2024-08-10 00:00:00 | Praça das Nações | 789   | Belo Horizonte | MG     | 30100-000 |
| Seminário de Saúde      | Um seminário sobre as novidades na área da saúde.     | 2024-09-05 00:00:00 | Rua Primavera   | 101    | Curitiba       | PR     | 80010-000 |
| Festival de Música      | Um festival com várias bandas e artistas.             | 2024-10-22 00:00:00 | Alameda Santos  | 202    | Fortaleza      | CE     | 60020-000 |
+-------------------------+-----------------------------------------------------+---------------------+-----------------+--------+----------------+--------+----------+
5 rows in set (0.00 sec)
```

Fonte: Elaborado pelo autor.

**Figura 24: Consulta exemplo “Selecionar todos os participantes do evento Conferência de Tecnologia” para asserção do banco de dados criado.**

```
mysql> SELECT
    ->     evento.nome AS NomeEvento,
    ->     participante.nome AS NomeParticipante,
    ->     participante.email,
    ->     participante.telefone
    -> FROM mydb.evento
    -> JOIN mydb.evento_participante ON evento.id_evento = evento_participante.id_evento
    -> JOIN mydb.participante ON evento_participante.id_participante = participante.id_participante
    -> WHERE evento.nome = 'Conferência de Tecnologia';
+---------------------------+-----------------+--------------------+-----------+
| NomeEvento                | NomeParticipante | email              | telefone  |
+---------------------------+-----------------+--------------------+-----------+
| Conferência de Tecnologia | Carlos Souza    | carlos@example.com | 5555-6666 |
| Conferência de Tecnologia | Ana Lima        | ana@example.com    | 7777-8888 |
| Conferência de Tecnologia | Pedro Alves     | pedro@example.com  | 9999-0000 |
+---------------------------+-----------------+--------------------+-----------+
3 rows in set (0.00 sec)
```

Fonte: Elaborado pelo autor.

# 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

A disciplina de AAP teve como principal objetivo desenvolver nos alunos práticas relacionadas à escrita de código limpo (*clean code*) conforme Martin (2019), organização modular de *software* e princípios de legibilidade, manutenibilidade e reuso. Foram abordados conceitos fundamentais como encapsulamento, nomes significativos, modularização de lógica, princípios *SOLID* e *DRY (Don't Repeat Yourself*), sempre com uma forte ênfase na prática. Também conforme Martin (2019) destaca, a aplicação rigorosa desses princípios é essencial para garantir a qualidade, a escalabilidade e a facilidade de manutenção do *software* ao longo do tempo.

A atividade final da disciplina consistiu na refatoração de um código de ordenação, o *Bubble Sort*, que foi entregue pelo professor em sua implementação original. O desafio era aplicar os conceitos discutidos ao longo do curso e transformar o código em uma versão mais legível, reutilizável e organizada, de acordo com os princípios de código limpo. Foram realizadas diversas melhorias no código (Figura 25), destacando-se as seguintes:

- Modularização da lógica de troca: a checagem de necessidade de troca e a troca em si foram movidas para dois novos métodos privados, promovendo clareza e reuso de lógica, além de reduzir a complexidade do método principal.
- Melhoria na nomenclatura: Alguns métodos, variáveis e outros parâmetros foram renomeados expressando de forma mais clara seu propósito, facilitando a compreensão do código por outros desenvolvedores.
- Encapsulamento adequado: Os métodos auxiliares foram definidos como privados, seguindo o princípio de encapsulamento da programação orientada a objetos, uma vez que não precisam ser acessados externamente à classe.
- Uso de estruturas mais legíveis: O laço for tradicional foi substituído por um *enhanced for loop*, tornando a leitura do código mais simples e elegante.
- Organização estrutural: A função *main* foi movida para o início do arquivo, permitindo que os métodos chamados apareçam na sequência do código, o que facilita o entendimento por parte de quem lê o programa de cima para baixo.

Essas alterações refletem uma preocupação prática com a qualidade de código, algo essencial para o trabalho em equipe e para a manutenção de sistemas reais, especialmente dentro de contextos ágeis.

## 7.1 ARTEFATOS DO PROJETO

**Figura 25: *Script de Bubble Sort* após diversas melhorias no código baseadas no conceito de código limpo.**

Código:

```
// Implementação otimizada em Java do Bubble sort
// Código extraído de https://www.geeksforgeeks.org/bubble-sort/
// Edição feita por Enzo Furlan para a disciplina de AAP - 23/7/2024
// Revisão baseada no feedback do professor - 31/07/2024

import java.io.*;

public class BubbleSortCleanCode {

    public static void main(String[] args) {
        int[] numeros = {64, 34, 25, 12, 22, 11, 90};
        bubbleSort(numeros);
        System.out.println("Array ordenado: ");
        printSortedArray(numeros);
    }

    public static void bubbleSort(int[] array) {
        int tamanhoIndexacao = array.length - 1;
        for (int i = 0; i < tamanhoIndexacao; i++) {
            if (!executarTroca(array, tamanhoIndexacao - i)) {
                break;
            }
        }
    }

    private static boolean executarTroca(int[] array, int indexFinal) {
        boolean trocado = false;
        for (int j = 0; j < indexFinal; j++) {
            if (array[j] > array[j + 1]) {
                trocar(array, j, j + 1);
                trocado = true;
            }
        }
        return trocado;
    }

    private static void trocar(int[] array, int primeiroElemento, int segundoElemento) {
        int temp = array[primeiroElemento];
        array[primeiroElemento] = array[segundoElemento];
        array[segundoElemento] = temp;
    }

    private static void printSortedArray(int[] array) {
        for (int numero : array) {
            System.out.print(numero + " ");
        }
        System.out.println();
    }
}
```

Fonte: https://www.geeksforgeeks.org/bubble-sort-algorithm/ com alterações do autor do trabalho.

# 8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2

As disciplinas de Desenvolvimento Web 1 (WEB1) e Desenvolvimento Web 2 (WEB2) proporcionaram a construção progressiva de uma aplicação web completa, começando com o *frontend* e utilizando armazenamento local do navegador, até a implementação de uma solução *full stack* com persistência em banco de dados e integração com *backend* desenvolvido em *Spring Boot*.

Na disciplina de WEB1, o projeto proposto foi a criação de dois CRUDs (sigla para *Create*, *Read*, *Update* e *Delete*) independentes: um para Alunos e outro para Cursos (Figura 26). O foco principal esteve no desenvolvimento do *frontend* e na utilização do *Local Storage* do navegador como mecanismo de persistência de dados, sem necessidade de autenticação, menus ou relacionamentos entre as entidades.

As funcionalidades implementadas para cada CRUD foram:

- Listagem: Exibição de todos os registros, com botões para criação, edição e exclusão. (Figura 28 e 30)
- Inserção: Formulário para cadastro de novos registros. (Figura 27 e 29)
- Alteração: Formulário com preenchimento automático dos dados para edição. (Figura 29)
- Remoção: Exclusão de registros com confirmação do usuário.

A interface foi desenvolvida com o uso do *Bootstrap*, garantindo responsividade e boa apresentação visual. A aplicação também implementou boas práticas de separação de responsabilidades no código e manipulação dinâmica dos elementos da interface.

Já na disciplina de Web 2, o projeto foi expandido e aprimorado com a introdução de um *backend* em *Spring Boot* e o uso de um banco de dados *PostgreSQL*, substituindo o *Local Storage* como mecanismo de persistência. Além disso, foi adicionado um terceiro CRUD, agora para Matrículas, estabelecendo relacionamentos entre Alunos e Cursos.

Cada matrícula representa um aluno inscrito em um curso em uma determinada data, com uma nota final atribuída.

O sistema passou a contar com:

- Um menu de navegação para acesso aos três CRUDs (Figura 32);

- Funcionalidades de listagem, inserção, edição, visualização em modal e remoção com confirmação (Figura 31 e 32);
- *Comboboxes* (*selects*) para seleção de Aluno e Curso no cadastro e edição de Matrículas (Figura 33);
- Uma arquitetura robusta com *backend* estruturado em camadas e uso de JPA para mapeamento das entidades no banco de dados.

A interface continuou utilizando *Bootstrap*, conforme exigido, mantendo a consistência visual entre as versões do sistema.

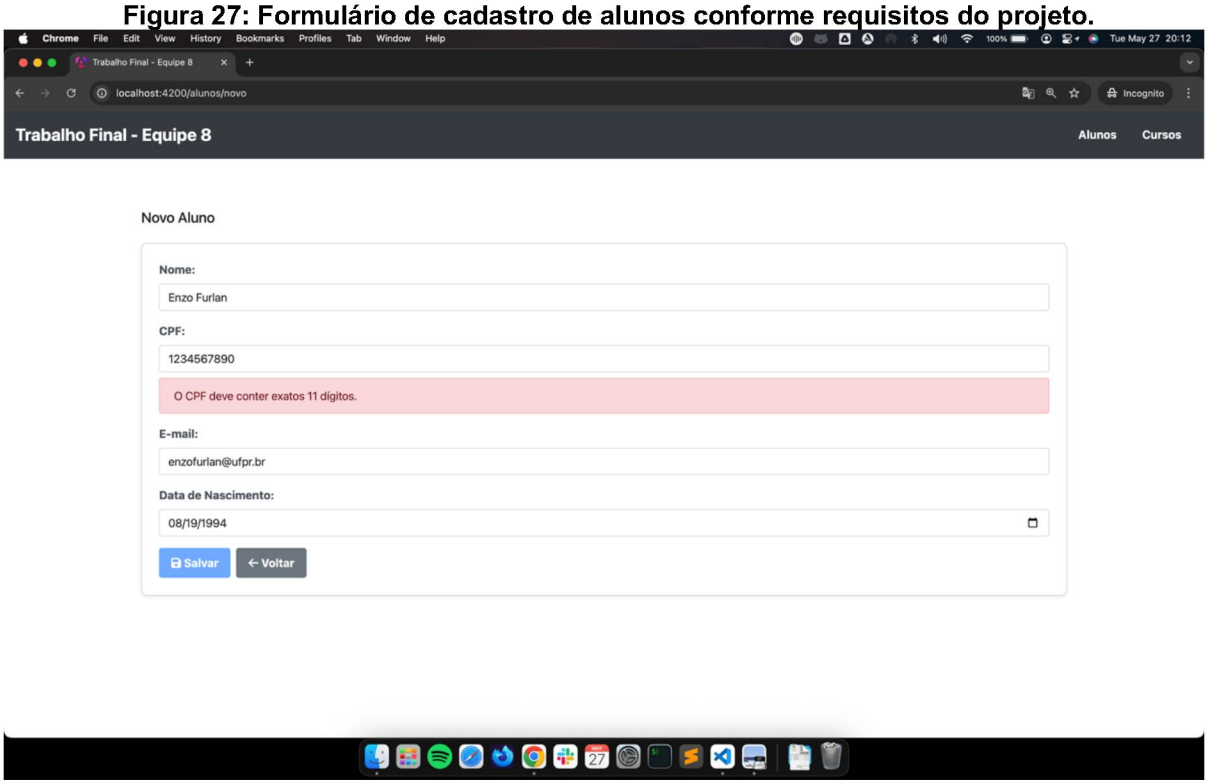
Essa evolução do projeto entre as duas disciplinas proporcionou uma vivência prática do ciclo completo de desenvolvimento web, desde a criação da interface e manipulação local de dados até a integração com banco de dados e API *RESTful*, aproximando o projeto das exigências do mercado de trabalho.

## 8.1 ARTEFATOS DO PROJETO

**Figura 26: Estrutura do código para o projeto de WEB1 utilizando Bootstrap e o Local Storage do navegador.**

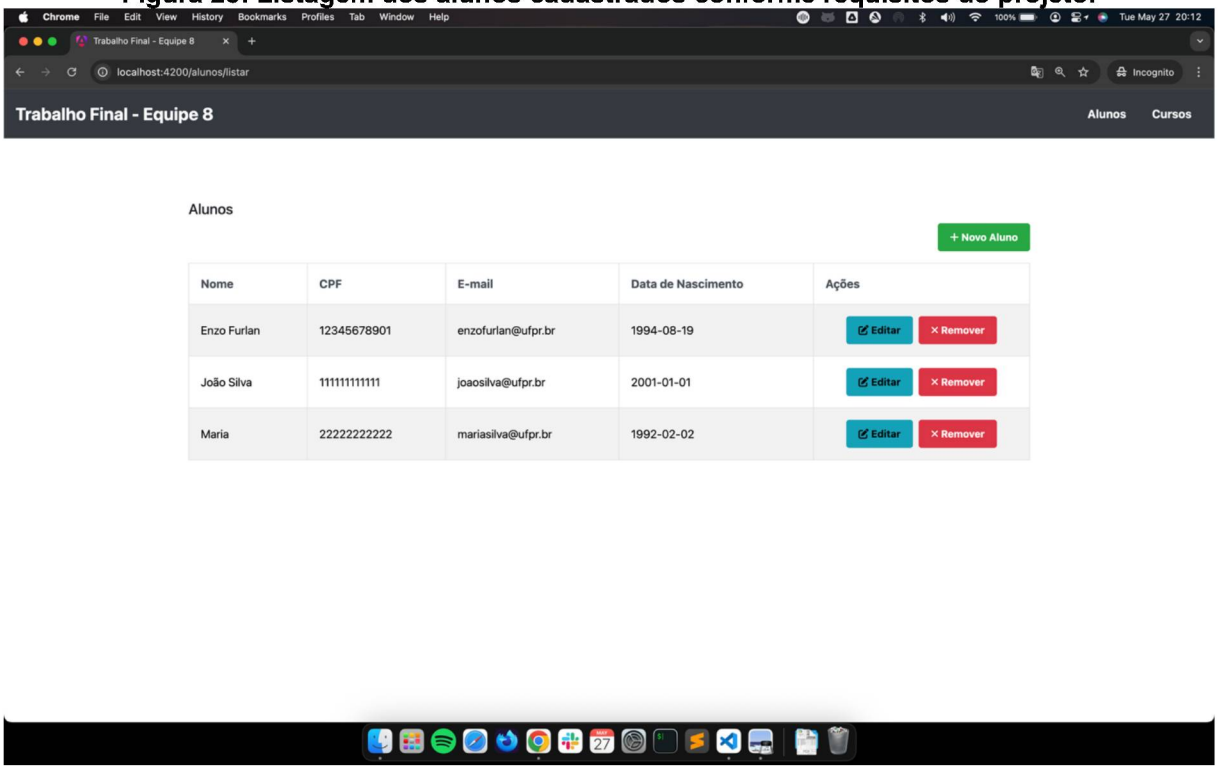

Fonte: Elaborado pelo autor.

**Figura 27: Formulário de cadastro de alunos conforme requisitos do projeto.**



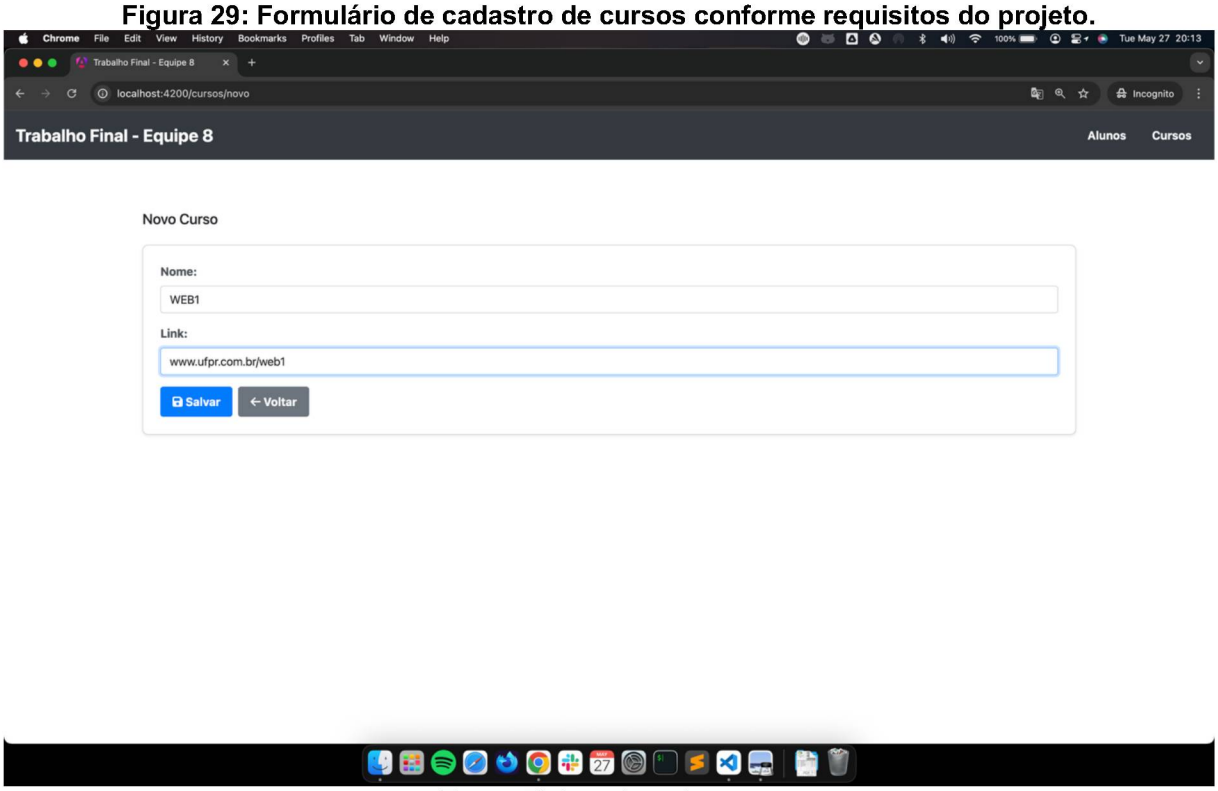

Fonte: Elaborado pelo autor.

**Figura 28: Listagem dos alunos cadastrados conforme requisitos do projeto.**



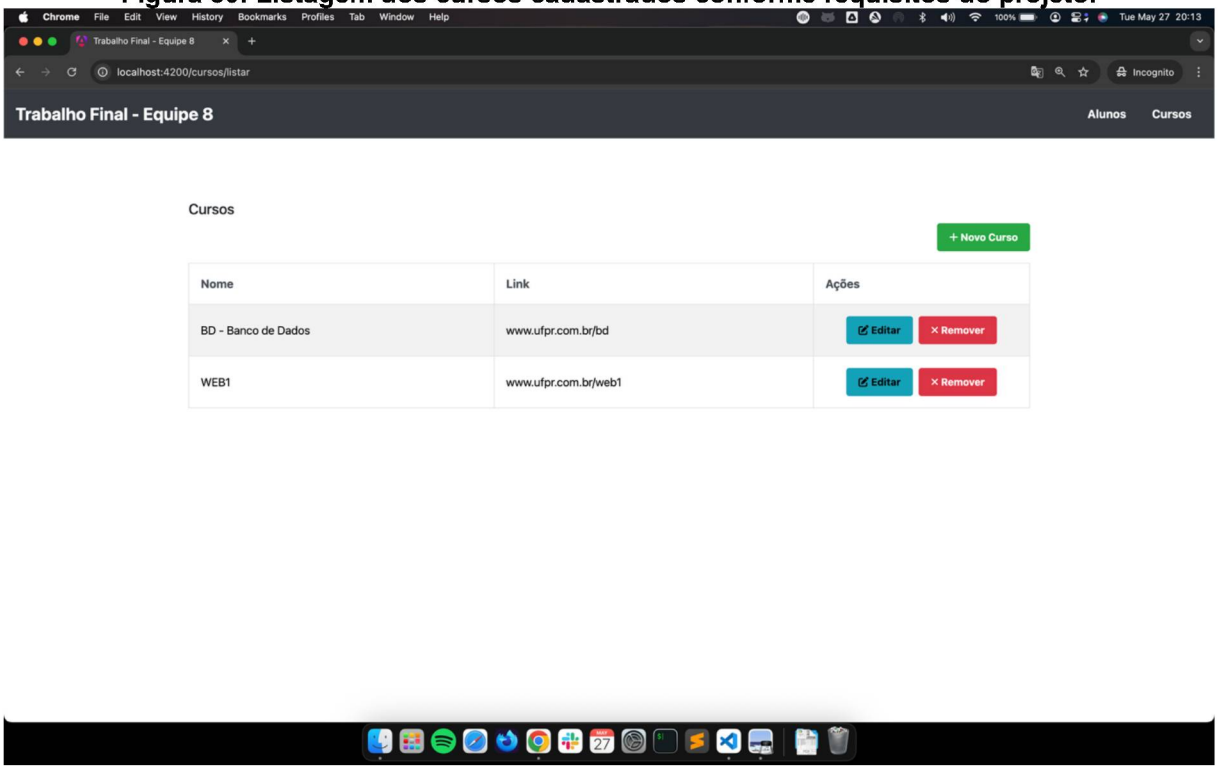

Fonte: Elaborado pelo autor.

**Figura 29: Formulário de cadastro de cursos conforme requisitos do projeto.**



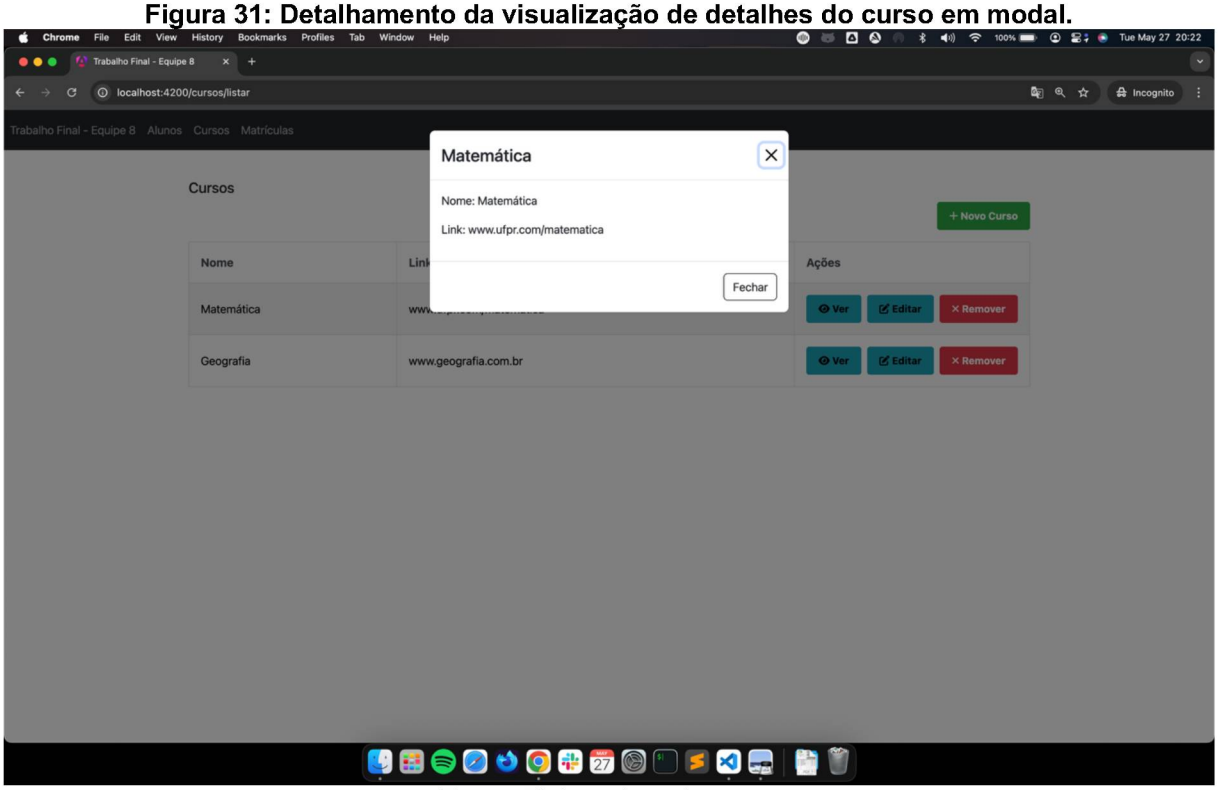

Fonte: Elaborado pelo autor.

**Figura 30: Listagem dos cursos cadastrados conforme requisitos do projeto.**



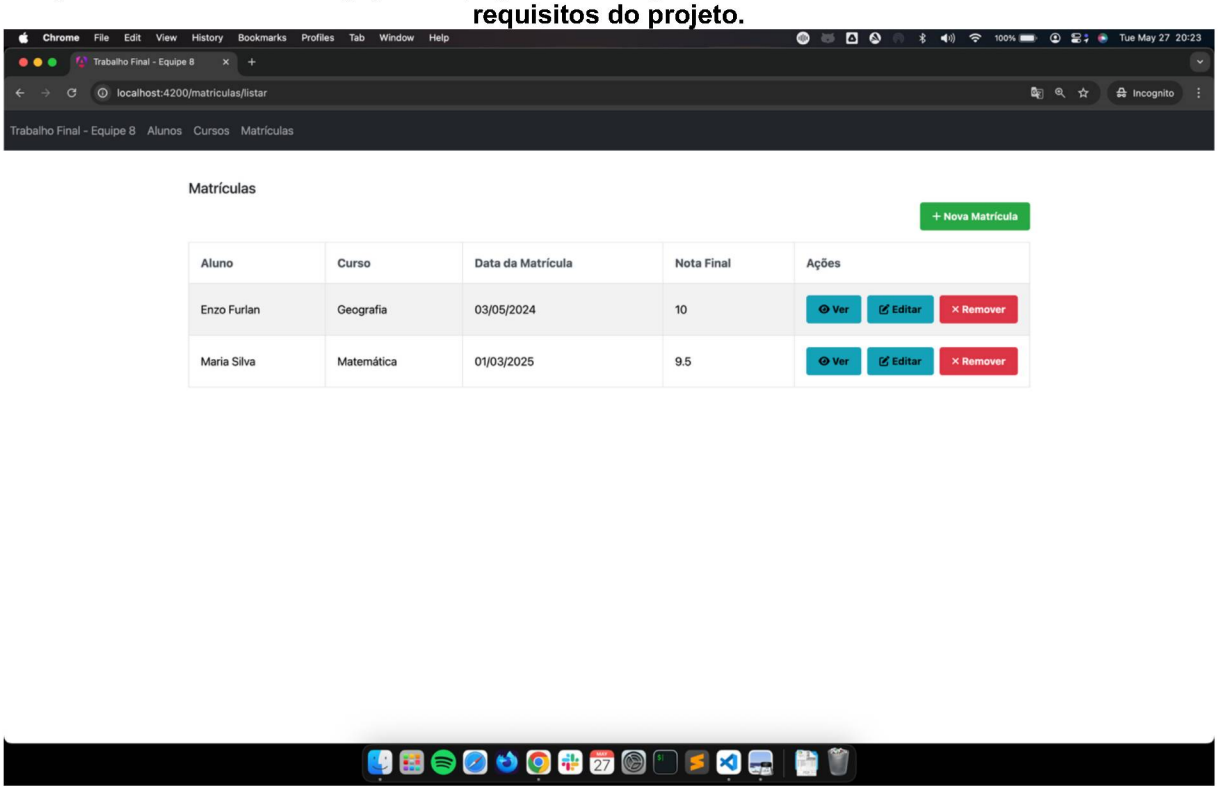

Fonte: Elaborado pelo autor.

**Figura 31: Detalhamento da visualização de detalhes do curso em modal.**



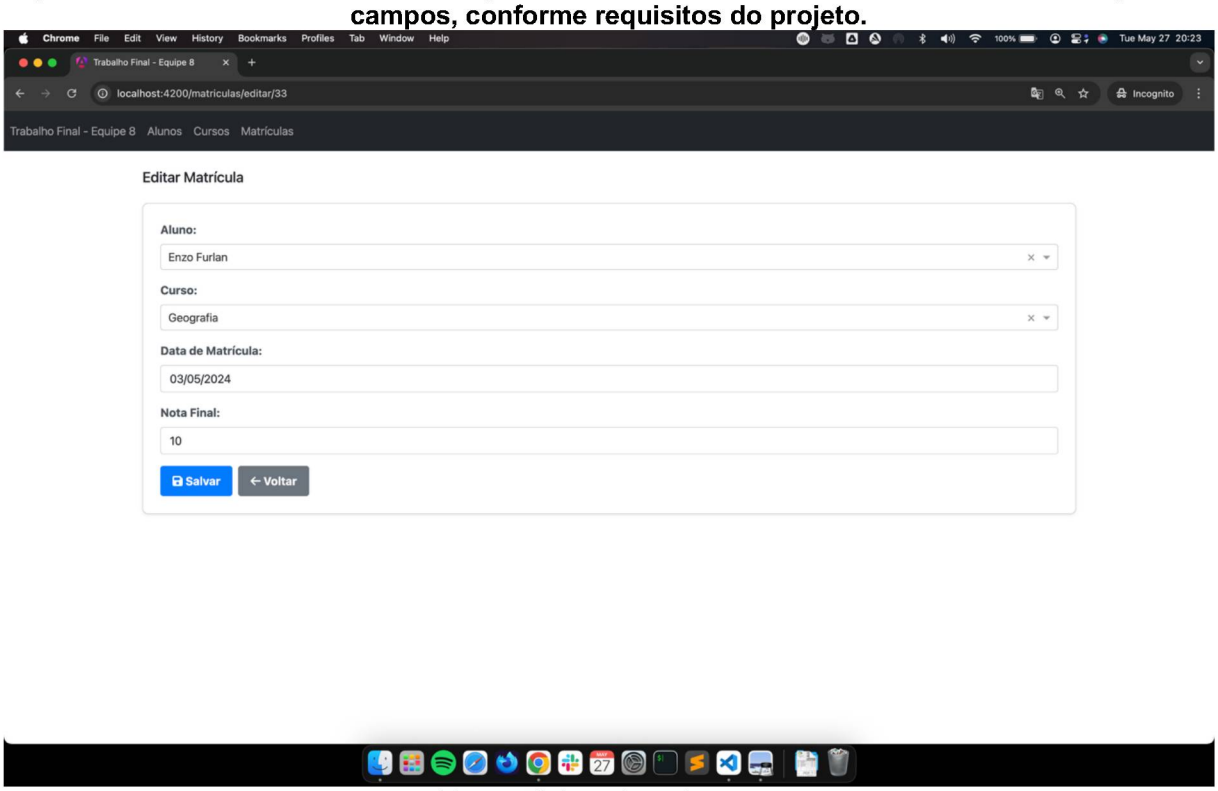

Fonte: Elaborado pelo autor.

**Figura 32: Menu de navegação na página de listagem das matrículas cadastradas conforme requisitos do projeto.**




Fonte: Elaborado pelo autor.

**Figura 33: Formulário de cadastro/edição de matrículas contendo *comboboxes* e validações de campos, conforme requisitos do projeto.**




Fonte: Elaborado pelo autor.

# 9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

A disciplina de UX teve como foco a aplicação prática de conceitos relacionados à experiência do usuário no desenvolvimento de interfaces intuitivas, acessíveis e esteticamente agradáveis. De acordo com Santos (2020), compreender o comportamento do usuário e aplicar ferramentas de UX são fundamentais para criar soluções mais eficazes e alinhadas às necessidades do público-alvo, destacando a importância de uma abordagem centrada no usuário no processo de criação e design. Nesse sentido, o projeto final consistiu na criação de protótipos de um aplicativo mobile, desenvolvido com a ferramenta *Figma*, com foco em navegação, *design* centrado no usuário e clareza de propósito.

O aplicativo desenvolvido foi chamado de "*Events*", um gerenciador de eventos sociais, acadêmicos e corporativos. Sua função principal é facilitar a criação, organização e acompanhamento de eventos pelos organizadores e participantes.

Funcionalidades principais:

- Cadastro de eventos com detalhes como título, data, horário, descrição, imagens e localização;
- Gerenciamento de sessões e participantes dentro de um evento;
- Descoberta de eventos próximos ao usuário com base em sua localização;
- Visualização de eventos em que o usuário está inscrito ou participou;
- Personalização do perfil e preferências de notificação.

O protótipo foi desenvolvido no Figma, contemplando cinco telas funcionais:

- Tela de Login: permite ao usuário acessar a plataforma via login ou realizar um novo cadastro. Contém campos para e-mail, senha e botões de ação (Figura 34);
- Página Principal: reúne todos os eventos disponíveis na plataforma. Possui busca por eventos, filtros, seção de eventos próximos, eventos inscritos, e eventos em destaque. Um botão “Novo Evento” permite criar eventos, e a foto do perfil no canto superior direciona para a tela de configurações do usuário (Figura 34);
- Tela de Criação de Novo Evento: tela única com campos para preenchimento dos detalhes do evento, como nome, descrição, imagens, local, data e hora (com seleção via calendário), e botão de salvar (Figura 35);

- Tela de Detalhes do Evento: mostra todas as informações do evento, como fotos, data, descrição, localização no mapa e opções de aceite/confirmação de presença (Figura 36);
- Tela de Perfil do Usuário: permite acessar preferências, editar perfil, gerenciar eventos criados ou inscritos e configurar notificações do app (Figura 36).

As cores predominantes no design são azuis e brancas, que oferecem um contraste forte e auxiliam na legibilidade. A cor azul transmite confiança e tranquilidade, além de manter o foco nas informações principais. Elementos vibrantes e ilustrações modernas foram usados para representar a energia de eventos ao vivo.

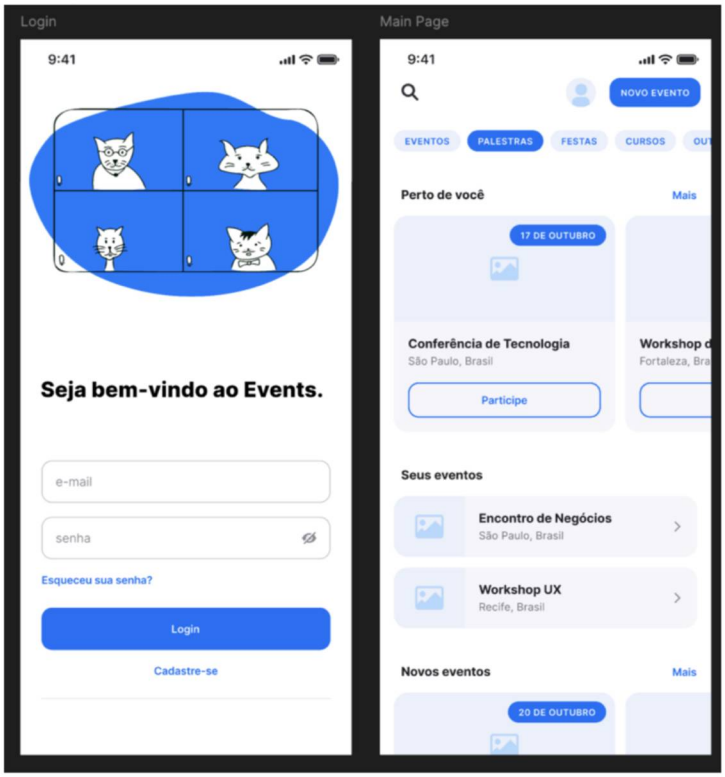
O layout priorizou a simplicidade e objetividade. A navegação é feita por cards e seções bem definidas, promovendo uma leitura agradável e intuitiva. Os botões são destacados e os elementos interativos são dispostos de forma ergonômica para o uso em dispositivos móveis.

Foi escolhida a fonte "Inter", que apresenta excelente legibilidade em diferentes tamanhos e resoluções. O uso do negrito e de diferentes cores auxilia na hierarquização visual das informações, facilitando a leitura rápida e promovendo uma interface moderna e funcional.

Esse projeto proporcionou uma visão prática dos processos de prototipação, design centrado no usuário, escolha de paletas e tipografia, validação com usuário e iteração com base em feedback real, aspectos fundamentais no desenvolvimento de produtos digitais bem-sucedidos.

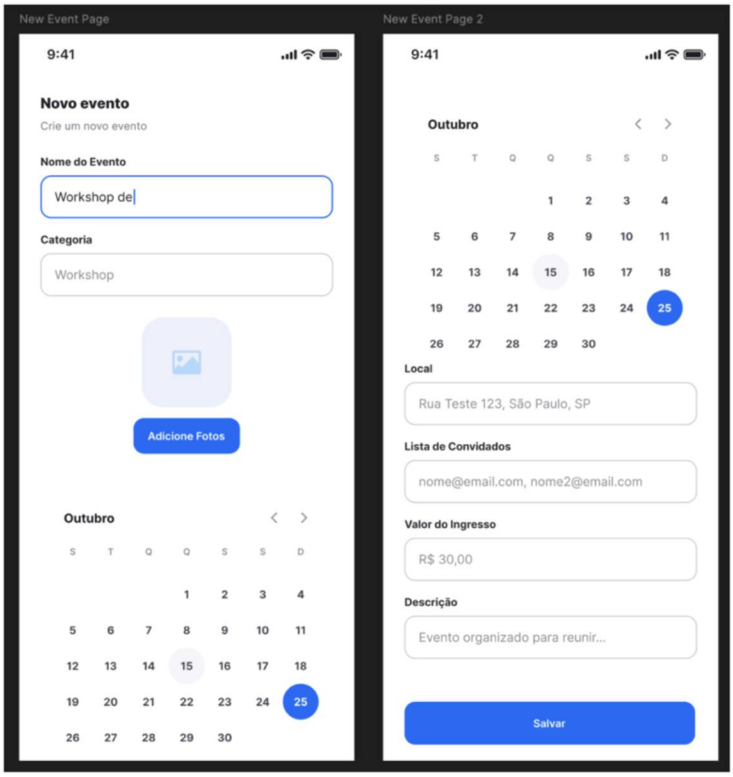
## 9.1 ARTEFATOS DO PROJETO

**Figura 34: Protótipos das telas de Login e Página Principal do aplicativo "*Events*", feitos utilizando a ferramenta *Figma*.**



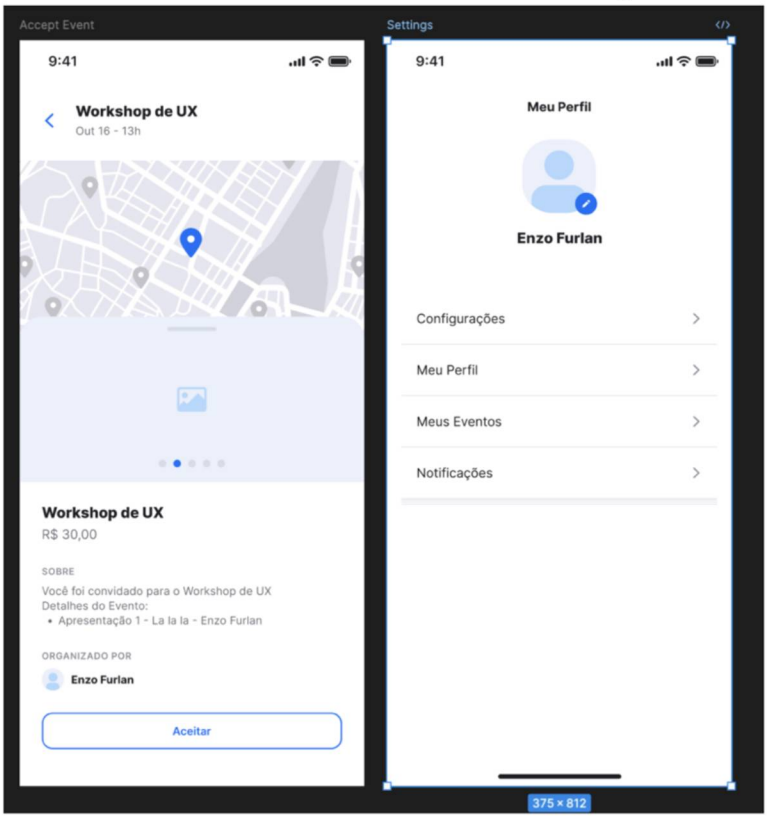

Fonte: Elaborado pelo autor.

**Figura 35: Protótipos das telas de Criação de Evento do aplicativo "*Events*", feitos utilizando a ferramenta *Figma*.**




Fonte: Elaborado pelo autor.

**Figura 36: Protótipos das telas de Detalhes de Evento e de Perfil do Usuário do aplicativo "*Events*", feitos utilizando a ferramenta *Figma*.**




Fonte: Elaborado pelo autor.

## 10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2

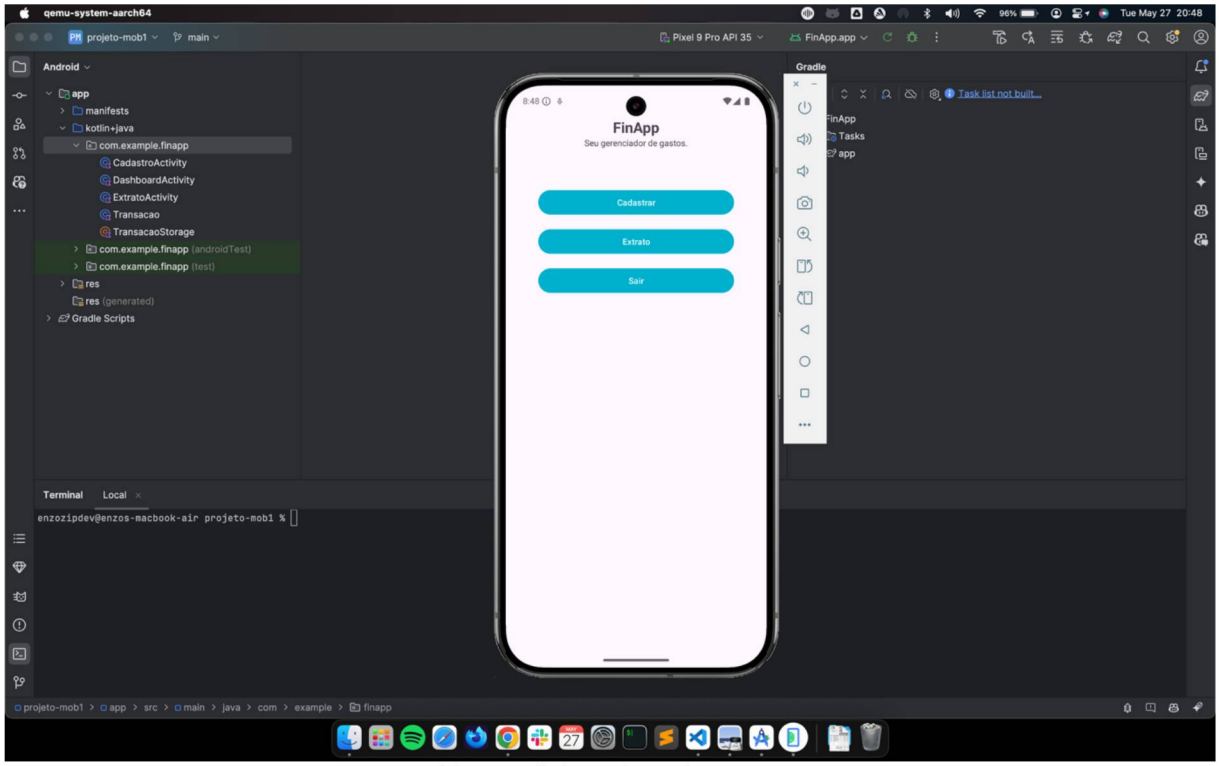
Os projetos desenvolvidos nas disciplinas MOB1 e MOB2 tiveram papel fundamental na consolidação dos conhecimentos práticos sobre o desenvolvimento de aplicativos móveis, utilizando *Android* com a API 28. Durante MOB1, o projeto proposto foi criar um aplicativo de controle financeiro pessoal (Figura 37), voltado para o cadastro e visualização de transações (créditos e débitos) do usuário (Figura 38 e 39). O projeto enfatizou a importância do *design* de interfaces intuitivas, uso de estruturas de dados em memória, navegação entre telas e manipulação básica de entradas. Já em MOB2, foi utilizado conceitos modernos como o consumo de APIs REST e o uso de corrotinas *Kotlin* para chamadas assíncronas, aplicando esses conhecimentos em um aplicativo baseado na *HP-API (Harry Potter),* que permite a busca (Figura 40) e exibição de dados como personagens (Figura 42), professores (Figura 41) e personagens por ID (Figura 43).

Ambos os projetos foram conduzidos de forma incremental e iterativa, em pequenos grupos, promovendo práticas do desenvolvimento ágil como colaboração contínua, ciclos curtos de entrega e adaptação constante. O planejamento e a execução dos aplicativos exigiram organização e divisão de tarefas, com integração contínua via *GitHub*, simulado ambientes reais de desenvolvimento ágil em equipes multidisciplinares.

Além disso, houve uma clara integração com outras disciplinas da especialização, como MAG1 e MAG2 (na definição das funcionalidades do app), TEST (na verificação do comportamento das telas e entradas), e UX (na construção de interfaces funcionais e agradáveis ao usuário). Assim, os projetos de MOB1 e MOB2 foram essenciais para aplicar na prática os conceitos vistos nas demais disciplinas, fortalecendo uma visão completa do ciclo de desenvolvimento ágil de *software* em plataformas móveis.

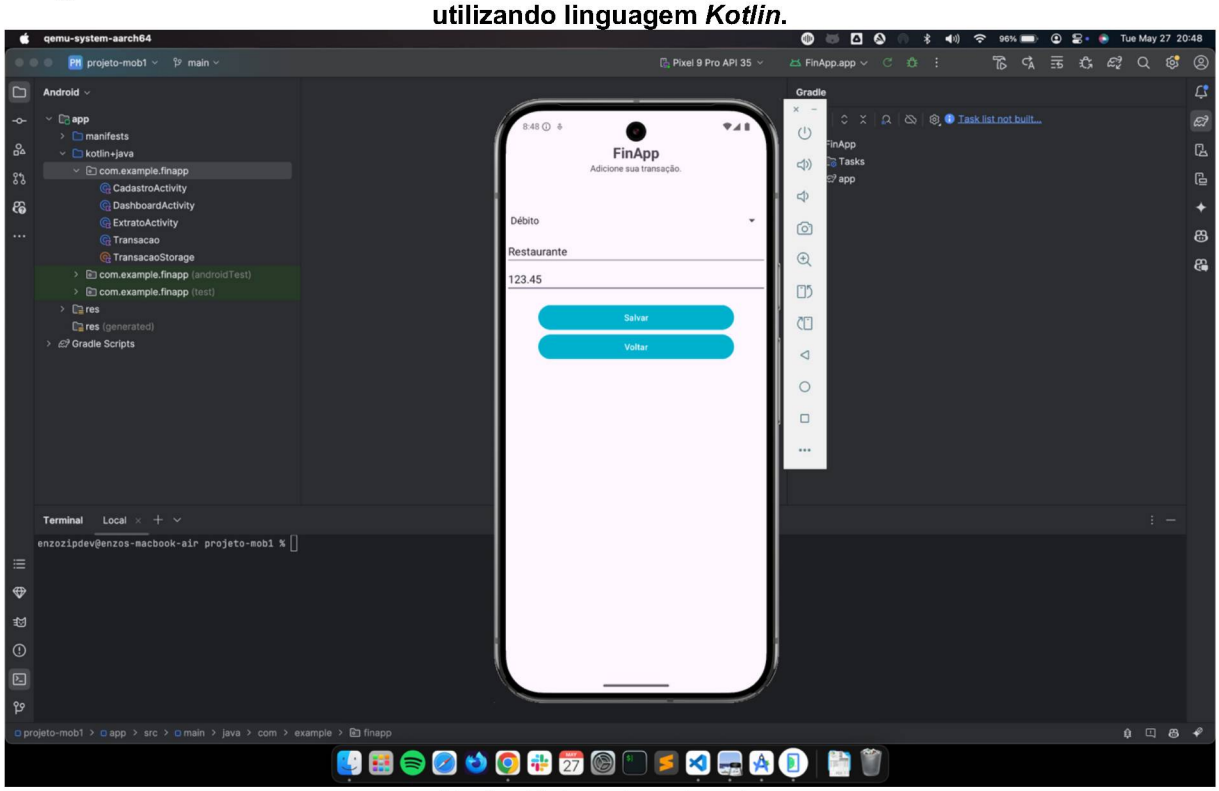
## 10.1 ARTEFATOS DO PROJETO

**Figura 37: Tela de menu do aplicativo de controle financeiro criado utilizando linguagem *Kotlin*.**



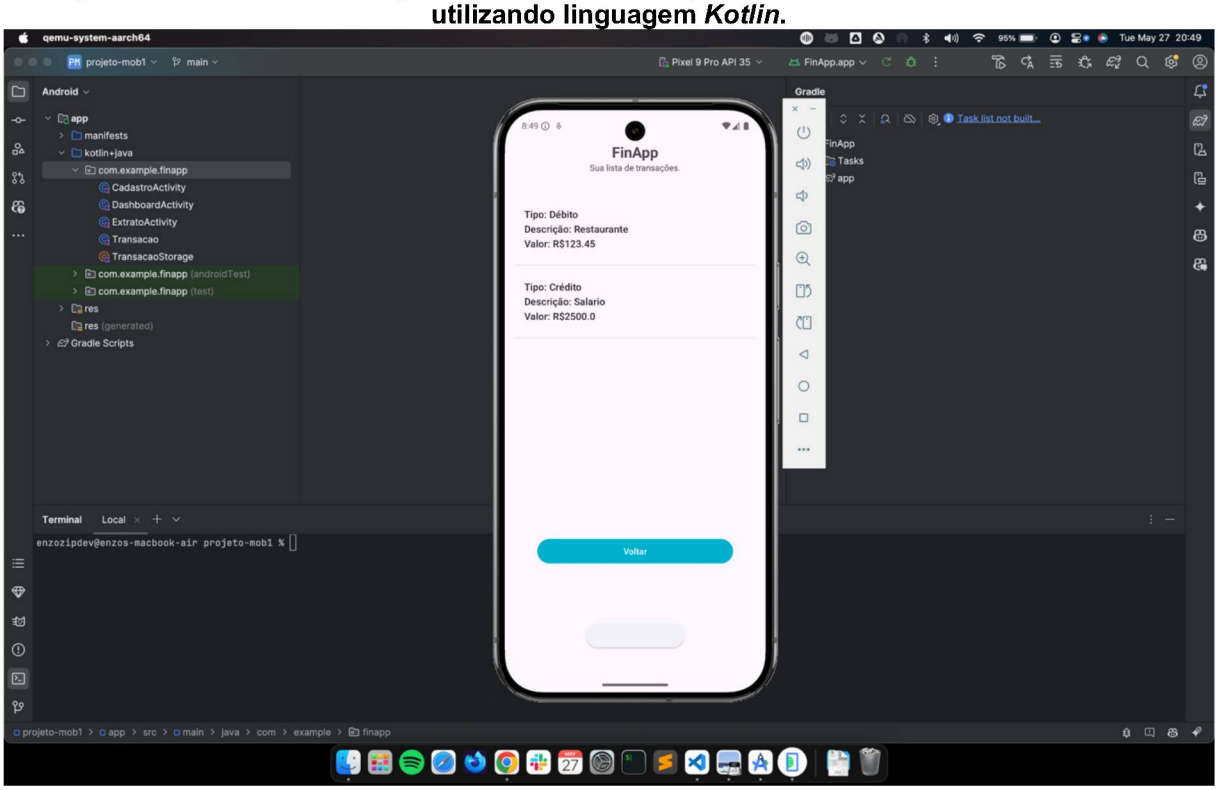

Fonte: Elaborado pelo autor.

**Figura 38: Tela de cadastro de crédito ou débito do aplicativo de controle financeiro criado utilizando linguagem *Kotlin*.**



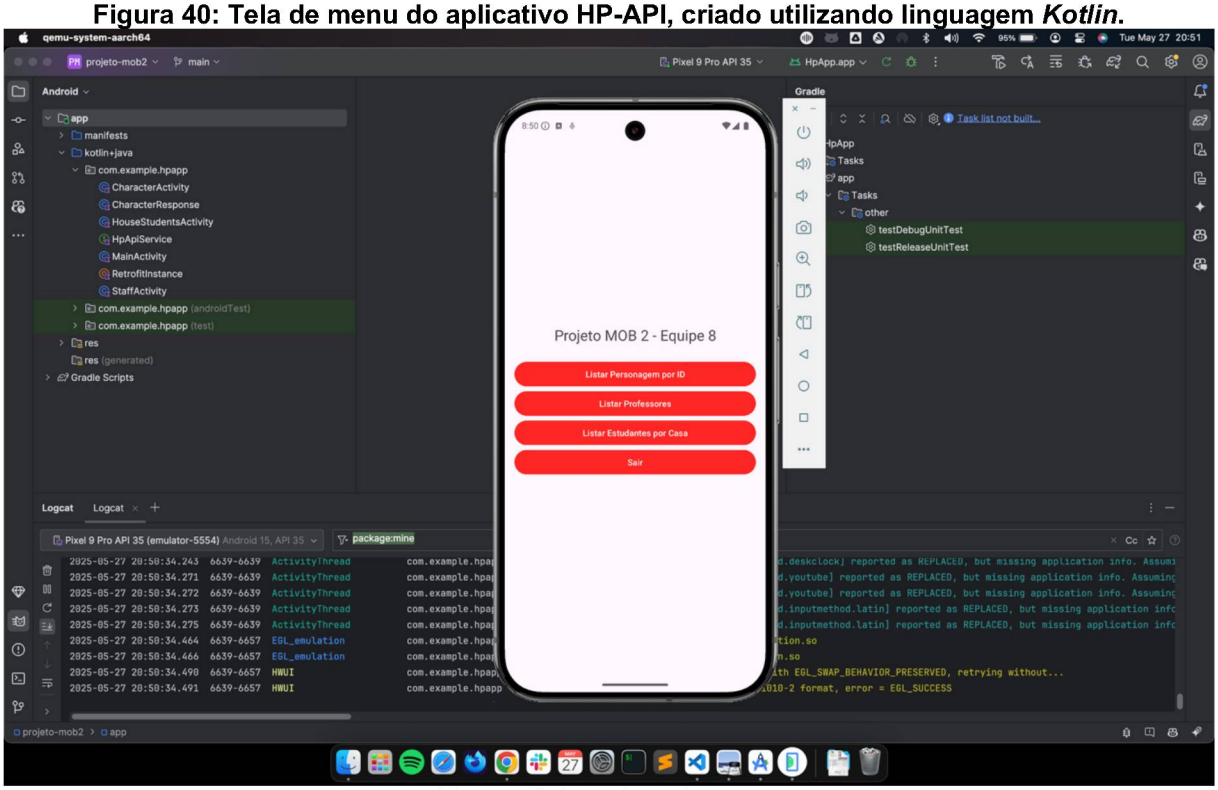

Fonte: Elaborado pelo autor.

**Figura 39: Tela de visualização de transações do aplicativo de controle financeiro criado utilizando linguagem *Kotlin*.**



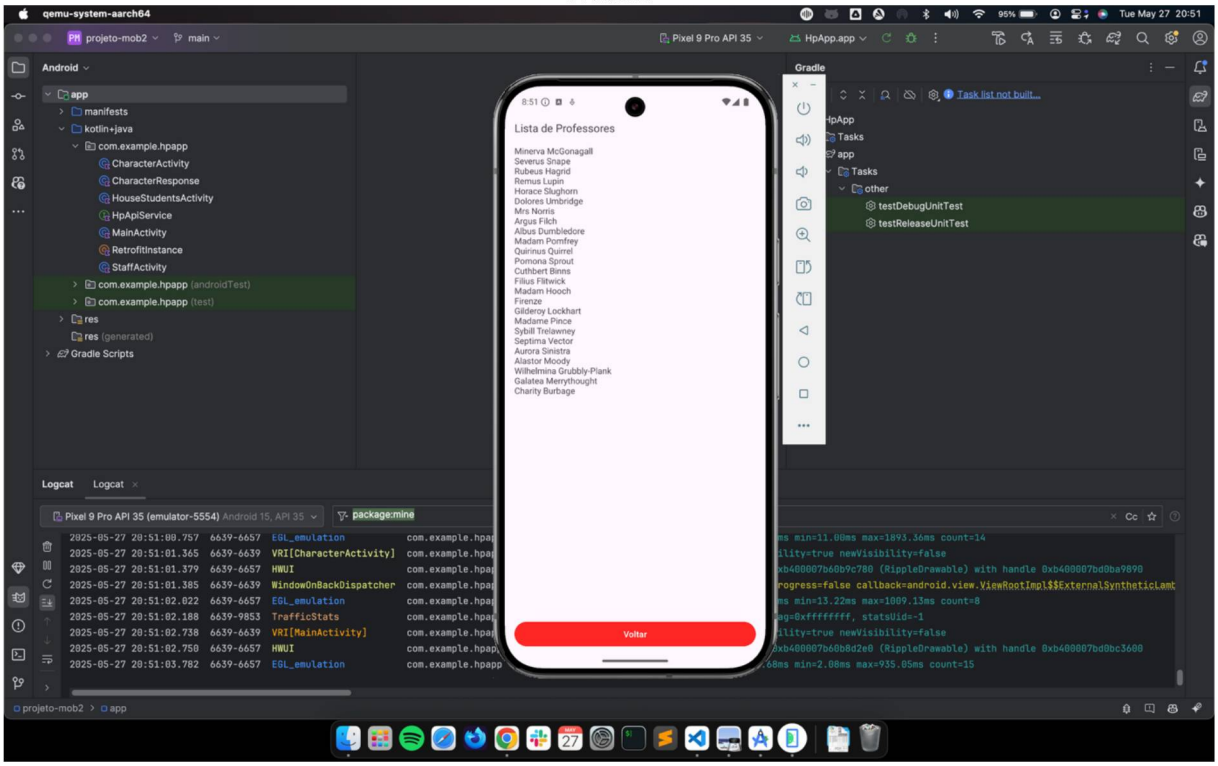

Fonte: Elaborado pelo autor.

**Figura 40: Tela de menu do aplicativo HP-API, criado utilizando linguagem *Kotlin*.**



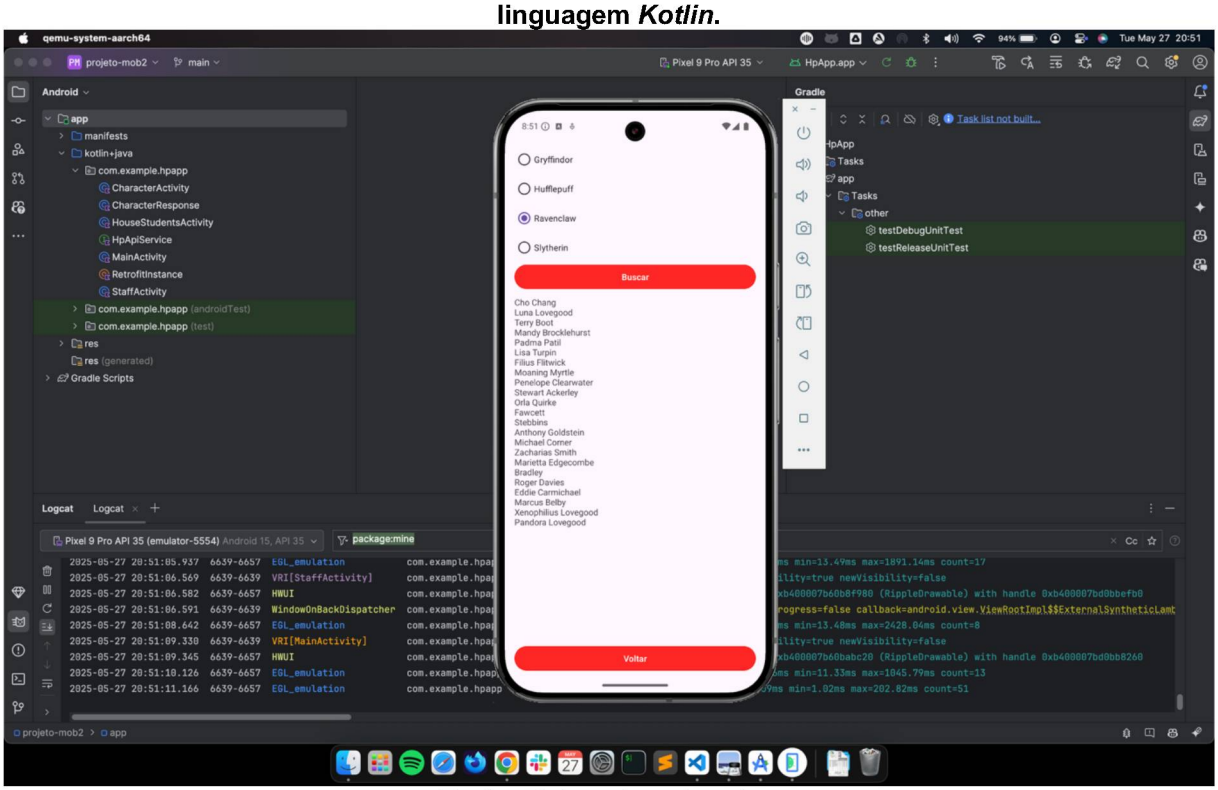

Fonte: Elaborado pelo autor.

**Figura 41: Tela de listagem dos professores do aplicativo HP-API, criado utilizando linguagem *Kotlin*.**



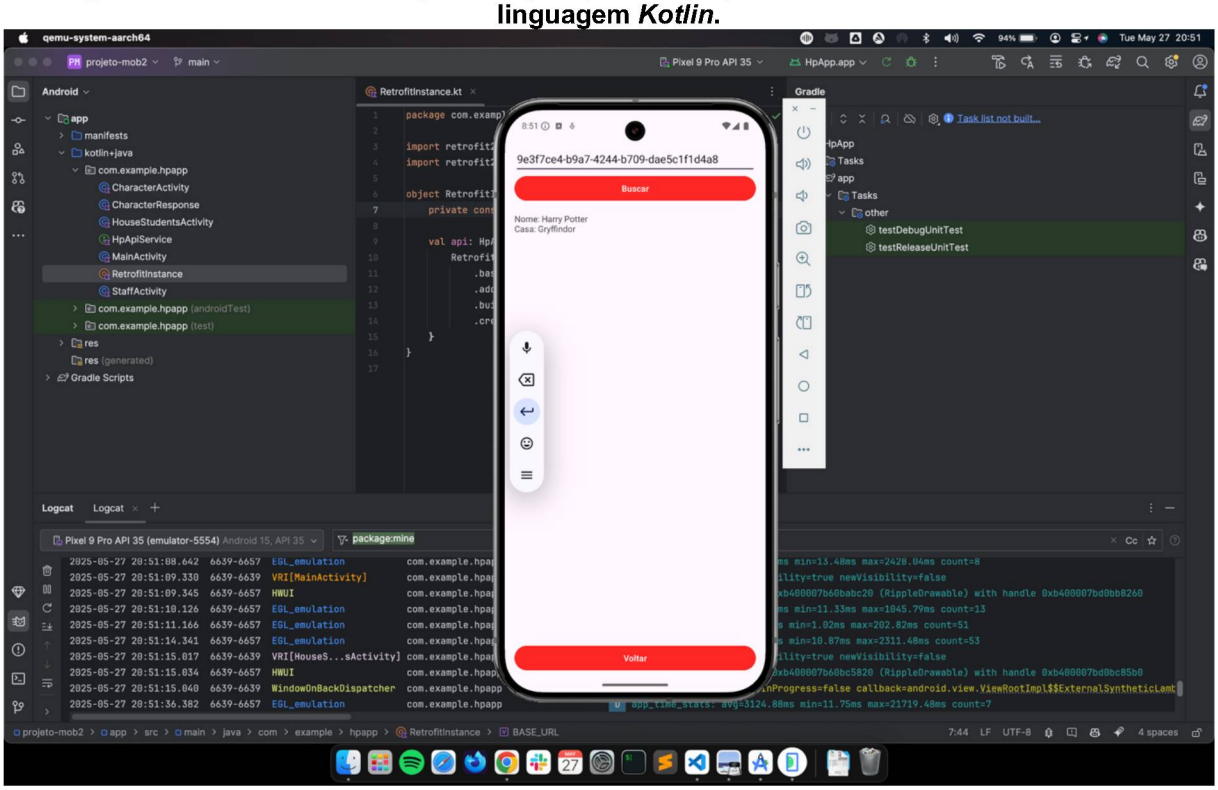

Fonte: Elaborado pelo autor.

**Figura 42: Tela de listagem de personagens por casa do aplicativo HP-API, criado utilizando linguagem *Kotlin*.**




Fonte: Elaborado pelo autor.

**Figura 43: Tela de busca de personagem por ID do aplicativo HP-API, criado utilizando linguagem *Kotlin*.**




Fonte: Elaborado pelo autor.

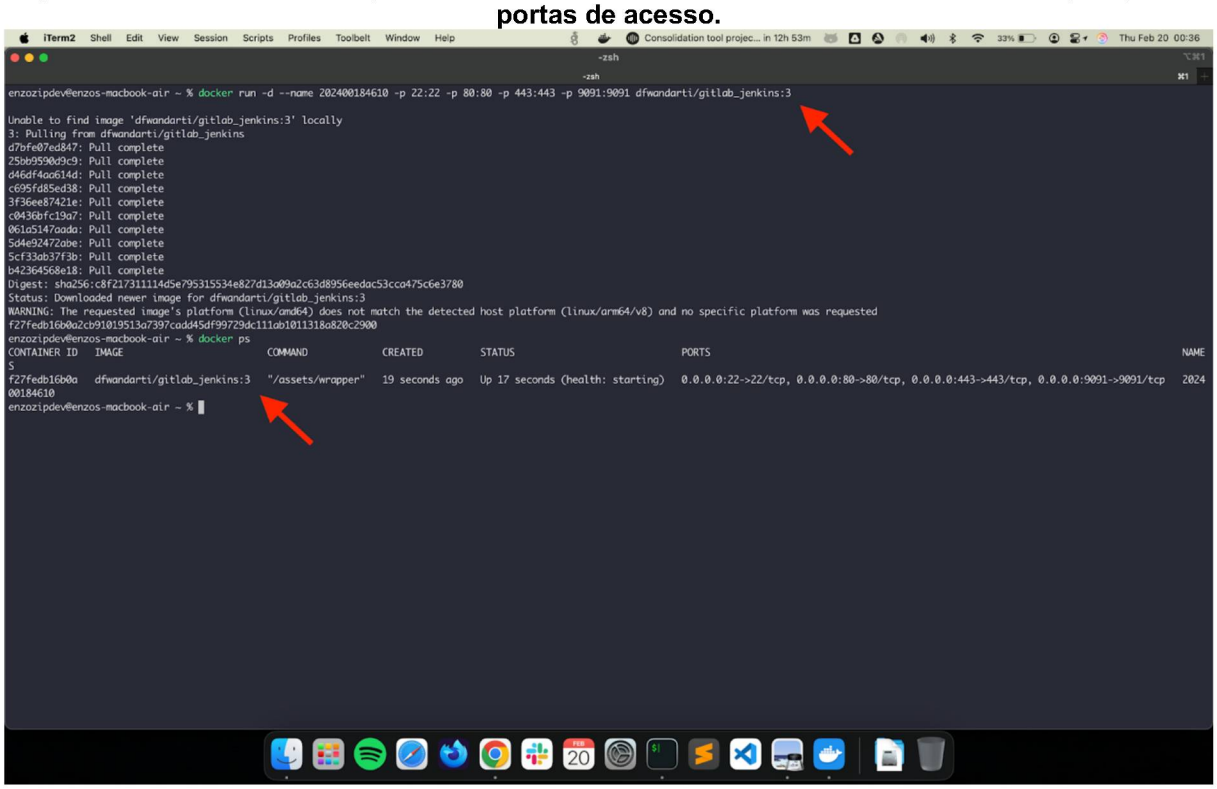
# 11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)

A disciplina de *DEVOPS* teve como objetivo principal aproximar o desenvolvimento e operações, promovendo agilidade com estabilidade, o que é fundamental para o funcionamento de ambientes de desenvolvimento modernos (Kim et al., 2016). Nesse sentido, o projeto final foi a criação de um ambiente de hospedagem da ferramenta *GitLab* em um container *Docker*. Para isso, os alunos realizaram o download e a execução da imagem personalizada (contendo uma instância do cliente de *git GitLab*), configurando o container com o nome da matrícula e publicando portas específicas nele (Figura 44). Após a execução, foi necessário acessar a interface do *GitLab* via navegador (Figura 46), utilizando o usuário *root* e a senha localizada dentro do container (Figura 45), e realizar operações práticas como *commit* e *push* de alterações em um repositório *Git* já existente na plataforma (Figura 47).

Esse projeto foi fundamental para demonstrar, na prática, a execução do ambiente em um container Docker e proporcionou uma introdução sólida à conteinerização e ao isolamento de ambientes, competências essenciais para o desenvolvimento moderno e escalável de *software*. Além disso, mostrou como configurar e utilizar um ambiente de controle de versão como o *GitLab*, ferramenta amplamente adotada em equipes ágeis

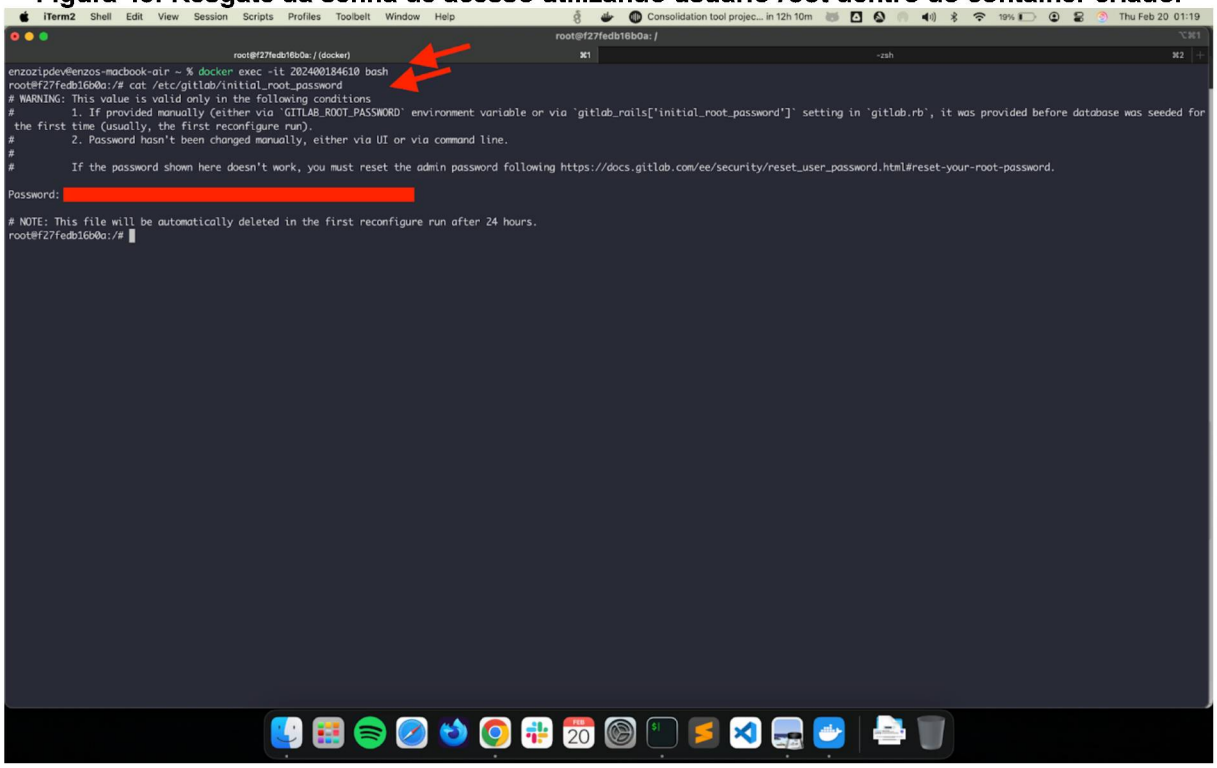
## 11.1 ARTEFATOS DO PROJETO

**Figura 44: Download da imagem personalizada utilizando *Docker,* incluindo a configuração de portas de acesso.**



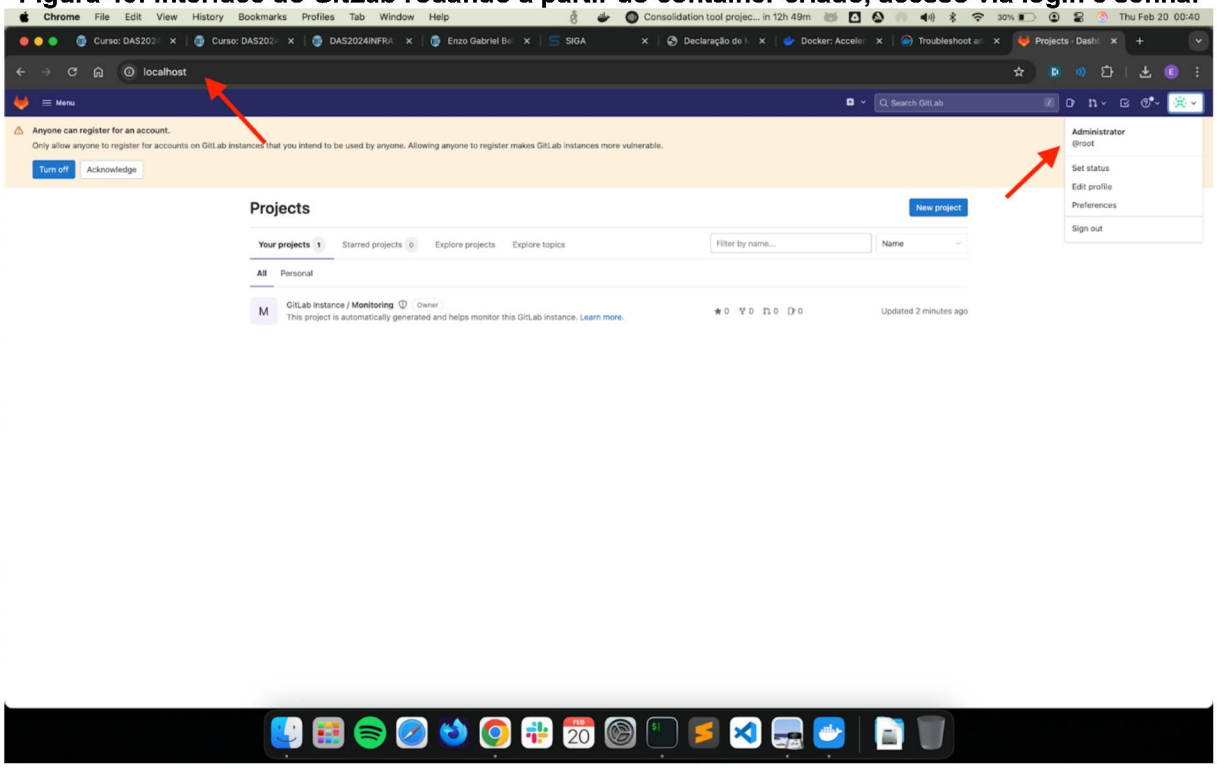

Fonte: Elaborado pelo autor.

**Figura 45: Resgate da senha de acesso utilizando usuário *root* dentro do container criado.**



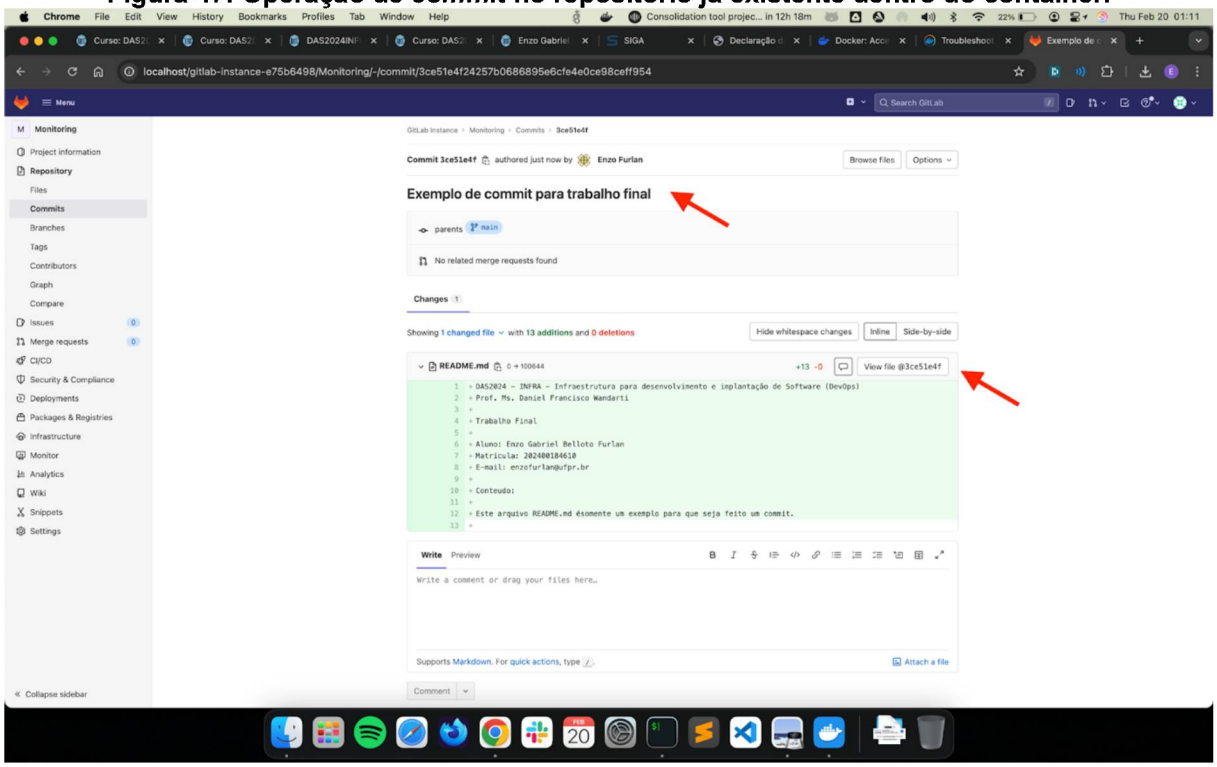

Fonte: Elaborado pelo autor.

**Figura 46: Interface do *GitLab* rodando a partir do container criado, acesso via login e senha.**




Fonte: Elaborado pelo autor.

**Figura 47: Operação de *commit* no repositório já existente dentro do container.**




Fonte: Elaborado pelo autor.

## 12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

Em TEST, o objetivo foi aplicar, na prática, conceitos fundamentais de automação de testes de interface, desenvolvendo um simples teste automatizado (Figura 48) que preenchesse campos em uma aplicação web com os dados dos integrantes do grupo (nome e matrícula) (Figura 49), e validasse a inserção correta dessas informações na interface. O projeto final da disciplina consistiu na criação de um *script* automatizado de teste utilizando o *framework Playwright* com a linguagem *JavaScript*.

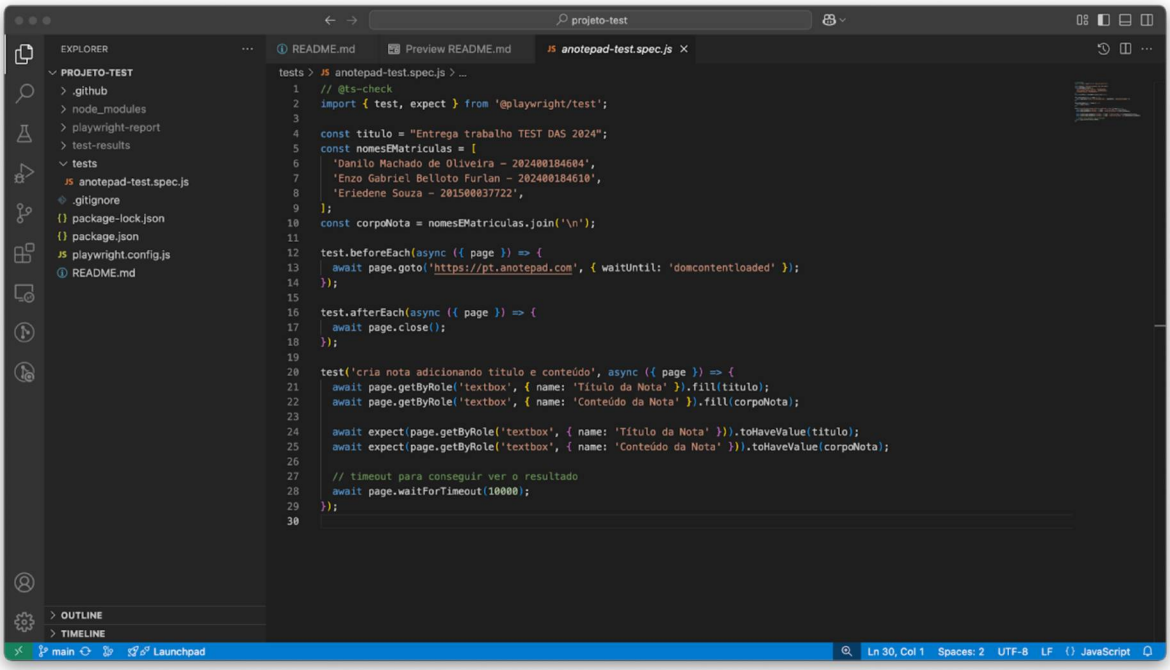
O teste foi estruturado com boas práticas, utilizando os blocos *beforeEach* e *afterEach* para configurar e limpar o ambiente de teste, além de utilizar asserções baseadas em variáveis constantes para garantir reprodutibilidade e clareza. Durante o desenvolvimento, foi identificado um desafio com o evento padrão de carregamento da página (*load*), que atrasava a execução e causava falhas por *timeout*. Para resolver essa limitação, o evento de espera foi alterado, garantindo que o teste prosseguisse assim que o conteúdo básico da página estivesse disponível, uma decisão técnica alinhada com o pensamento crítico exigido no desenvolvimento ágil.

A realização desse trabalho foi essencial para reforçar a importância dos testes automatizados como parte do ciclo de desenvolvimento ágil de *software*. Segundo o *BrowserStack* (2023), a automação de testes de interface permite simular interações do usuário, como cliques e preenchimento de formulários, para verificar a funcionalidade e a usabilidade do sistema, garantindo uma experiência consistente e eficiente para o usuário final.

Portanto, essa atividade destacou não apenas a parte técnica da automação, mas também seu papel estratégico dentro de um fluxo de desenvolvimento moderno, escalável e centrado na entrega de valor com qualidade.

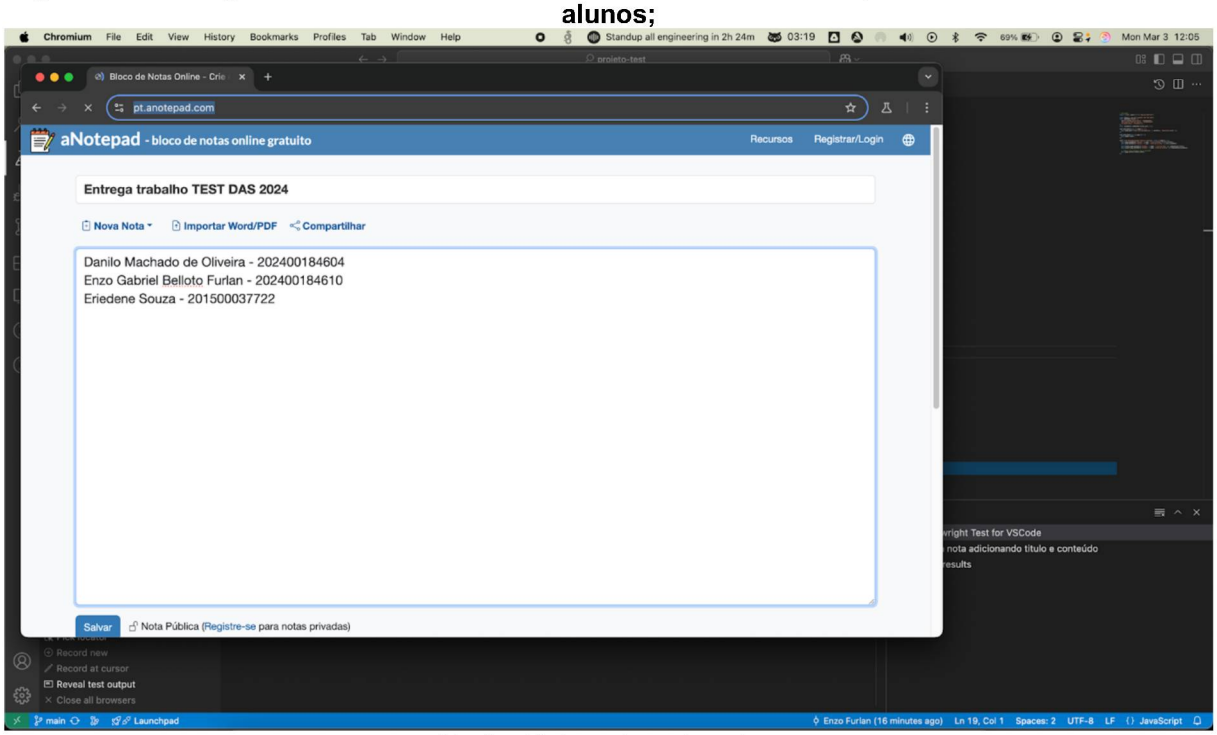
## 12.1 ARTEFATOS DO PROJETO

**Figura 48: *Script* em *Javascript* contendo um teste de interface utilizando o *framework Playwright*.**




Fonte: Elaborado pelo autor.

**Figura 49: Execução do teste automatizado acessando o website e preenchendo os dados dos alunos;**




Fonte: Elaborado pelo autor.

# 13 CONCLUSÃO

Este memorial chamado de "Desenvolvimento *Full Stack* com Práticas Ágeis" consolidou os principais projetos desenvolvidos ao longo da especialização em Desenvolvimento Ágil de *Software*, apresentando uma trajetória prática e multidisciplinar voltada à construção de soluções com base em princípios ágeis. Cada disciplina contribuiu para o amadurecimento técnico e metodológico, desde a fundamentação teórica dos métodos ágeis até a entrega de aplicações completas com foco em valor, qualidade e adaptabilidade.

Os projetos abrangeram diversas áreas: modelagem ágil de sistemas (MADS, MAG1 e MAG2), planejamento e gestão de projetos com Scrum e *Kanban* (GAP1 e GAP2), desenvolvimento orientado a objetos e testes automatizados com *Java* (INTRO), modelagem e implementação de banco de dados relacionais (BD), aplicação de princípios de código limpo (AAP), construção de sistemas *web* e *mobile* com camadas bem definidas (WEB1, WEB2, MOB1 e MOB2), design centrado no usuário (UX), automação de testes de interface (TEST) e configuração de ambientes com *Docker* e *GitLab* (INFRA).

Durante o percurso, observou-se que o desenvolvimento ágil vai além de métodos e ferramentas: ele requer mudança de mentalidade, colaboração constante e entrega contínua de valor. Os ciclos iterativos, os feedbacks rápidos e a integração entre áreas foram elementos centrais para o sucesso dos projetos.

Entre os principais desafios enfrentados na adoção prática da agilidade, destacam-se: o equilíbrio entre documentação enxuta e clareza na comunicação técnica; a adaptação a ferramentas e tecnologias diversas em curtos períodos; a manutenção da coesão entre integrantes de um mesmo projeto; e a garantia de qualidade e estabilidade mesmo com entregas frequentes.

Ainda assim, os resultados obtidos demonstram a viabilidade e os benefícios da abordagem ágil em ambientes acadêmicos e profissionais. A especialização não apenas transmitiu conhecimento técnico, mas também proporcionou vivência prática em situações reais de projeto, formando um profissional mais preparado, adaptável e focado em soluções eficazes.

## 14 REFERÊNCIAS

BAHREHVAR, Majid; MOSHIRPOUR, Mohammad. Full-stack Development and Soft Skills: An Agile-based Learning Framework. **Proceedings of the Canadian Engineering Education Association (CEEA)**, *[S. l.]*, 2022. Disponível em: https://ojs.library.queensu.ca/index.php/PCEEA/article/view/15844. Acesso em: 5 jun. 2025. https://doi.org/10.24908/pceea.vi.15844.

BECK, K. et al. **Manifesto para Desenvolvimento Ágil de Software**, 2001. Disponível em: https://agilemanifesto.org/iso/ptbr/manifesto.html. Acesso em: 27 maio 2025.

BROWSERSTACK. **Automated UI testing: Benefits, challenges & solution**. 2023. Não Paginado. Disponível em: https://www.browserstack.com/guide/what-is-automated-ui-testing. Acesso em: 3 jun. 2025.

COHN, M. **User stories applied: For agile software development**. Boston: Addison-Wesley Professional, 2004. Acesso em: 3 jun. 2025.

HIGHSMITH, Jim. **Agile Project Management: Creating Innovative Products**. 2. ed. Boston: Addison-Wesley, 2009. Acesso em: 3 jun. 2025.

KIM, Gene et al. **The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations**. 2. ed. IT Revolution Press, 2016. Acesso em: 3 jun. 2025.

MARTIN, R. C. **Clean code: A handbook of agile software craftsmanship**. Filadélfia, PA, USA: Prentice Hall, 2009. Acesso em: 3 jun. 2025.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: McGraw-Hill Brasil, 2016. Acesso em: 3 jun. 2025.

SANTOS, Brenda Carolina dos. **UX Design – um guia prático para iniciantes**. 2020. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Informática para Negócios) – Faculdade de Tecnologia de São José do Rio Preto, São José do Rio Preto, 2020.

STRODE, D.; DINGSØYR, T.; LINDSJORN, Y. A teamwork effectiveness model for agile software development. **Empirical Software Engineer**, v. 27, n. 56, 2022. Disponível em: https://link.springer.com/article/10.1007/s10664-021-10115-0. Acesso em: 5 jun. 2025. https://doi.org/10.1007/s10664-021-10115-0.