

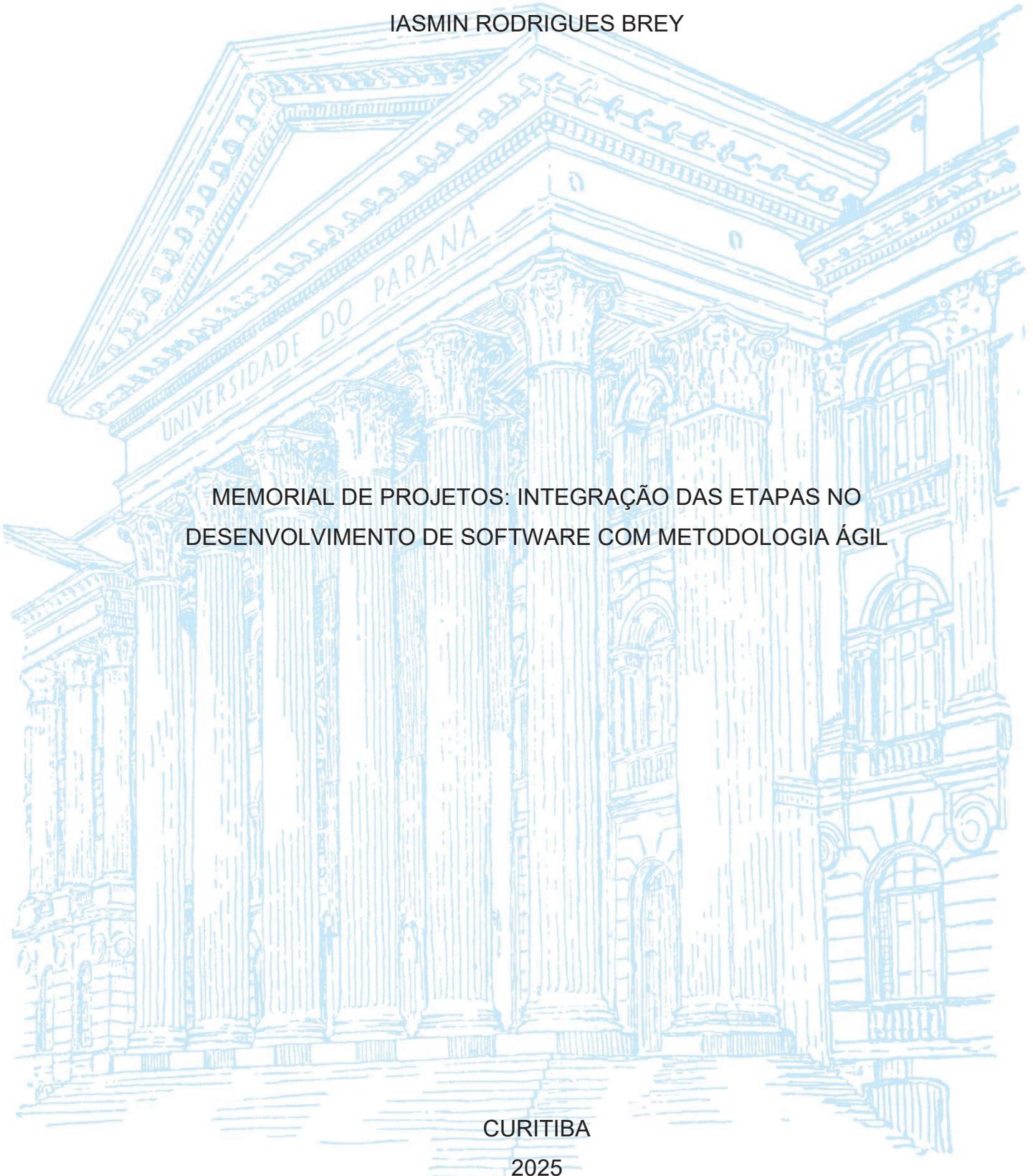
UNIVERSIDADE FEDERAL DO PARANÁ

IASMIN RODRIGUES BREY

MEMORIAL DE PROJETOS: INTEGRAÇÃO DAS ETAPAS NO  
DESENVOLVIMENTO DE SOFTWARE COM METODOLOGIA ÁGIL

CURITIBA

2025



IASMIN RODRIGUES BREY

MEMORIAL DE PROJETOS: INTEGRAÇÃO DAS ETAPAS NO  
DESENVOLVIMENTO DE SOFTWARE COM METODOLOGIA ÁGIL

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: Profa. Dra. Rafaela Mantovani Fontana

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 **IASMIN RODRIGUES BREY**, intitulada: **MEMORIAL DE PROJETOS: INTEGRAÇÃO DAS ETAPAS NO DESENVOLVIMENTO DE SOFTWARE COM METODOLOGIA ÁGIL**, que após terem inquirido a aluna 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, 15 de Agosto de 2025.

RAFAELA MANTOVANI FONTANA  
Presidente da Banca Examinadora

JAIMÉ WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

## RESUMO

O objetivo deste memorial é apresentar a integração das disciplinas estudadas ao longo do curso, evidenciando como cada uma contribuiu para a formação de uma abordagem ágil no desenvolvimento de software. Foram abordadas quinze disciplinas, incluindo Métodos Ágeis, Modelagem Ágil, Gerenciamento de Projetos, Desenvolvimento Web, UX, Desenvolvimento Mobile, Infraestrutura DevOps e Testes Automatizados. Os artefatos apresentados, como mapas mentais, planos de release, sistemas de gestão e protótipos, demonstraram a aplicação prática dos conceitos ágeis, promovendo a colaboração e a adaptação contínua. A interligação entre as disciplinas permitiu a construção de soluções que priorizam a experiência do usuário e a entrega de valor, refletindo a essência do desenvolvimento ágil. A experiência adquirida ao longo do curso não apenas consolidou conhecimentos técnicos, mas também desenvolveu uma mentalidade ágil, essencial para enfrentar os desafios do mercado de trabalho.

Palavras-chave: desenvolvimento ágil, integração, colaboração, experiência do usuário.

## **ABSTRACT**

The objective of this memorial is to present the integration of the disciplines studied throughout the course, highlighting how each contributed to the formation of an agile approach to software development. Fifteen disciplines were covered, including Agile Methods, Agile Software Modeling, Agile Project Management, Web Development, UX, Mobile Development, DevOps Infrastructure, and Automated Testing. The artifacts presented, such as mind maps, release plans, management systems, and prototypes, demonstrated the practical application of agile concepts, promoting collaboration and continuous adaptation. The interconnection between the disciplines allowed for the construction of solutions that prioritize user experience and value delivery, reflecting the essence of agile development. The experience gained throughout the course not only consolidated technical knowledge but also developed an agile mindset, essential for facing the challenges of the job market.

Keywords: agile development, integration, collaboration, user experience.

## SUMÁRIO

<b>1 PARECER TÉCNICO.....</b>	<b>7</b>
<b>2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....</b>	<b>10</b>
2.1 ARTEFATOS DO PROJETO.....	11
<b>3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2.....</b>	<b>14</b>
3.1 ARTEFATOS DO PROJETO – MAG1 PT1 .....	14
3.2 ARTEFATOS DO PROJETO – MAG1 PT2 .....	15
3.3 ARTEFATOS DO PROJETO – MAG2.....	18
<b>4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2 .....</b>	<b>18</b>
4.1 ARTEFATOS DO PROJETO – GAP1 .....	20
4.2 ARTEFATOS DO PROJETO – GAP2 .....	20
<b>5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....</b>	<b>21</b>
5.1 ARTEFATOS DO PROJETO.....	23
<b>6 DISCIPLINA: BD – BANCO DE DADOS.....</b>	<b>24</b>
6.1 ARTEFATOS DO PROJETO – QUESTÃO 1 .....	25
6.2 ARTEFATOS DO PROJETO – QUESTÃO 2 .....	26
<b>7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO .....</b>	<b>28</b>
7.1 ARTEFATOS DO PROJETO.....	29
<b>8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2 .....</b>	<b>30</b>
8.1 ARTEFATOS DO PROJETO.....	31
<b>9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE.....</b>	<b>33</b>
9.1 ARTEFATOS DO PROJETO.....	34
<b>10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....</b>	<b>35</b>
<b>11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS) .....</b>	<b>37</b>
11.1 ARTEFATOS DO PROJETO.....	38
<b>12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS .....</b>	<b>39</b>
12.1 ARTEFATOS DO PROJETO.....	40
<b>13 CONCLUSÃO .....</b>	<b>41</b>
<b>14 REFERÊNCIAS.....</b>	<b>43</b>

## 1 PARECER TÉCNICO

O percurso de aprendizado que compôs este parecer técnico incluiu quinze disciplinas essenciais, formando uma base sólida para a aplicação do desenvolvimento ágil na prática. Na disciplina de Métodos Ágeis para Desenvolvimento de Software (MADS), foi requerido a construção de mapas mentais que sintetizam os princípios fundamentais da agilidade, como Scrum, XP e Lean, além de compreender como aplicar a agilidade em diferentes contextos de desenvolvimento.

As disciplinas de Modelagem Ágil de Software (MAG1 e MAG2) permitiram aplicar na prática os conceitos estudados, por meio da modelagem de um sistema de gestão de condomínios. Evidenciando a importância da simplicidade, conforme o Princípio 10 do Manifesto Ágil, de Beck et al. (2001), que afirma: "Simplicidade--a arte de maximizar a quantidade de trabalho não realizado--é essencial". A modelagem ágil não apenas esclareceu os requisitos de forma eficaz, demonstrando como deve funcionar no sistema, mas também contribuiu para evitar riscos de retrabalho por meio de um entendimento mais preciso das necessidades.

No âmbito do Gerenciamento Ágil de Projetos (GAP1 e GAP2), desenvolveu-se um plano de release que materializou o conceito de sprints iterativos para releases incrementais, conforme os princípios propostos por Sutherland (2014), que enfatiza que equipes que operam em ciclos curtos e iterativos tendem a ser mais adaptáveis às mudanças e mais eficazes na entrega contínua de valor. Foram aplicados cálculos de velocidade da equipe e a priorizar o backlog do produto, o que evidenciou a importância da adaptação e do feedback contínuo, permitindo respostas rápidas às mudanças nos requisitos.

A base técnica foi consolidada na disciplina de Introdução à Programação (INTRO), por meio da implementação de um sistema bancário. Seguindo os princípios defendidos por Crispin e Gregory (2009), que destacam que os testes automatizados devem ser parte integrante do processo de desenvolvimento ágil, a prática de escrever testes desde o início reforçou a qualidade do código e a confiança nas entregas. Essa abordagem também se alinha ao segundo princípio do Manifesto Ágil, que valoriza a aceitação de mudanças de requisitos mesmo em fases avançadas do desenvolvimento, como forma de promover vantagem competitiva ao cliente. O trecho

reforça como os testes automatizados ajudam a garantir que novas funcionalidades possam ser adicionadas ou alteradas sem comprometer a integridade do sistema.

Na disciplina de Banco de Dados (BD), a modelagem de um sistema bibliotecário mostrou que assim como nas metodologias ágeis, onde o desenvolvimento é realizado em sprints curtos e incrementais, a modelagem e a implementação de bancos de dados podem ser conduzidas de forma iterativa. Essa abordagem permite ajustes no esquema do banco de dados com base no feedback contínuo e nas necessidades emergentes do projeto.

Os aspectos de qualidade foram aprofundados em Aspectos Ágeis de Programação (AAP), com foco em algoritmos de ordenação e na aplicação do princípio de código limpo, conforme proposto por Martin (2008). Essa abordagem está alinhada ao nono princípio do Manifesto Ágil, que destaca que a atenção contínua à excelência técnica e ao bom design potencializa a agilidade. Nesse contexto, a prática de refatoração contínua favoreceu a legibilidade do código e o sincronismo entre os desenvolvedores.

As disciplinas de Desenvolvimento Web (WEB1 e WEB2) com foco em operações CRUDs utilizando Spring Boot, demonstraram como os frameworks ágeis promovem a adaptação contínua, integrando front-end e back-end, conforme destaca Highsmith (2002). O uso de tecnologias modernas e frameworks ágeis contribui para o desenvolvimento de funcionalidades de forma mais rápida e eficiente, reduzindo o tempo de entrega.

Na disciplina de UX no Desenvolvimento Ágil, a prototipagem do sistema de gestão de condomínios validou a ideia de que o design centrado no usuário, conforme argumenta Norman (2013), é um dos pilares da agilidade. A ênfase na experiência do usuário garantiu que as soluções desenvolvidas fossem funcionais, intuitivas e agradáveis, alinhando-se aos princípios ágeis de entrega de valor com foco na interface visual.

Já nos projetos desenvolvidos na disciplina de Mobile (MOB1 e MOB2) consistiram na criação de dois aplicativos: um voltado para funcionalidades financeiras e outro para consulta a uma API. Durante o desenvolvimento, os testes de usabilidade foram fundamentais para garantir que as soluções fossem funcionais e agradáveis, refletindo a abordagem centrada no usuário, com foco nas funcionalidades.

Na disciplina de Infraestrutura DevOps (INFRA), a implementação de CI/CD com Docker materializou o conceito apresentado por Humble e Farley (2010),

segundo o qual a automação é a base da entrega contínua. A automação de processos de integração e entrega acelerou o ciclo de desenvolvimento e garantiu a qualidade e a consistência das entregas.

Finalmente, em Testes Automatizados (TEST), a construção de testes reforçou a tese defendida por Jeffries et al. (2000), segundo a qual é essencial testar cedo e com frequência, completando o ciclo ágil. A prática de automação de testes contribuiu para a economia de tempo e a redução de riscos de falhas em produção.

Essa integração multidisciplinar demonstra como diferentes conhecimentos se complementam no ecossistema ágil. Segundo Kaur e Malik (2020), a verdadeira agilidade resulta da sinergia entre pessoas, processos e tecnologias. A experiência adquirida transcende a técnica, moldando uma mentalidade ágil capaz de enfrentar os desafios dinâmicos do desenvolvimento moderno de software.

A crescente necessidade de adaptação às demandas aceleradas do mercado é um aspecto crucial. As equipes de desenvolvimento contemporâneas são estruturadas com processos ágeis que permitem reações rápidas às mudanças constantes, priorizando a melhoria contínua da experiência do usuário e a resposta eficaz às solicitações dos consumidores. Carmona (2023) enfatiza essa realidade ao afirmar que as empresas precisam garantir que cada pessoa na equipe compreenda a velocidade do mundo do desenvolvimento de softwares, adaptando-se rapidamente à cultura e aprendendo a ser mais criativas.

Essa visão ressalta a importância de cultivar uma mentalidade ágil, na qual criatividade e flexibilidade se tornam essenciais para o sucesso em um ambiente de desenvolvimento dinâmico e em constante evolução. A capacidade de inovar e se ajustar rapidamente não apenas melhora a eficiência das equipes, mas também garante que os produtos finais atendam às expectativas dos usuários.

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

O projeto desenvolvido na disciplina Métodos Ágeis de Desenvolvimento de Software teve como objetivo a criação de um mapa mental (Figura 1) que sintetizasse os principais conceitos relacionados aos processos de software, aos princípios ágeis e aos métodos de desenvolvimento. Essa atividade proporcionou uma base teórica consistente e evidenciou a importância de estruturar o conhecimento de forma visual, facilitando tanto a aprendizagem quanto a retenção de informações.

Ao explorar os fundamentos do desenvolvimento ágil — como colaboração, flexibilidade e entrega contínua de valor ao cliente — o mapa mental permitiu uma compreensão integrada de abordagens como o Scrum (Figura 2), Manifesto Ágil (Figura 3), Extreme Programming – XP (Figura 4) e Lean Software Development (Figura 5). Visualizar essas metodologias em conjunto revelou como cada uma pode ser aplicada em diferentes contextos, reforçando a ideia de que não existe uma solução única para todos os projetos, e que a escolha da abordagem deve considerar as necessidades específicas de cada equipe e produto.

O projeto também promoveu uma reflexão sobre a capacidade de adaptação e resposta rápida às mudanças, características essenciais do pensamento ágil. A habilidade de representar informações complexas de forma clara e acessível mostrou-se valiosa para o trabalho em equipe, contribuindo para uma comunicação mais eficiente e uma colaboração mais produtiva entre os integrantes.

Por fim, os conhecimentos adquiridos nesta disciplina representam um marco fundamental para o desenvolvimento acadêmico e profissional ao longo do curso. A compreensão dos métodos ágeis e sua aplicação prática servem como base conceitual e estratégica para disciplinas futuras, como Gerenciamento Ágil de Projetos, Modelagem de Software, Testes, entre outras. Essa preparação inicial fortalece a capacidade de análise crítica, tomada de decisão e adaptação a diferentes cenários de desenvolvimento, elementos indispensáveis para quem deseja atuar com excelência na área.

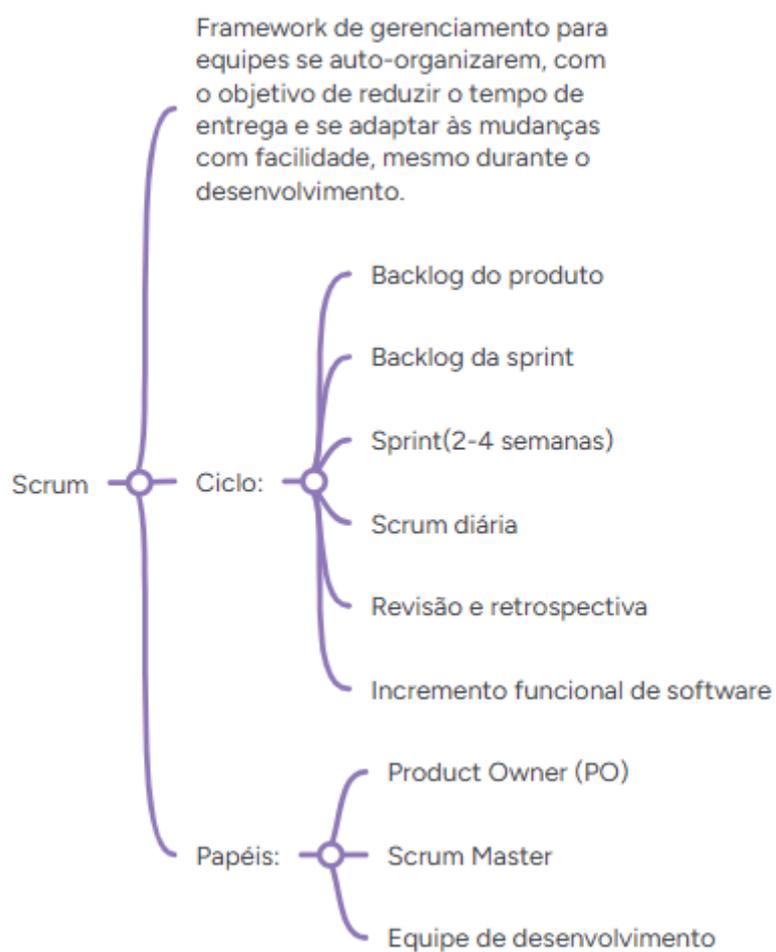
## 2.1 ARTEFATOS DO PROJETO

FIGURA 1 – MAPA MENTAL



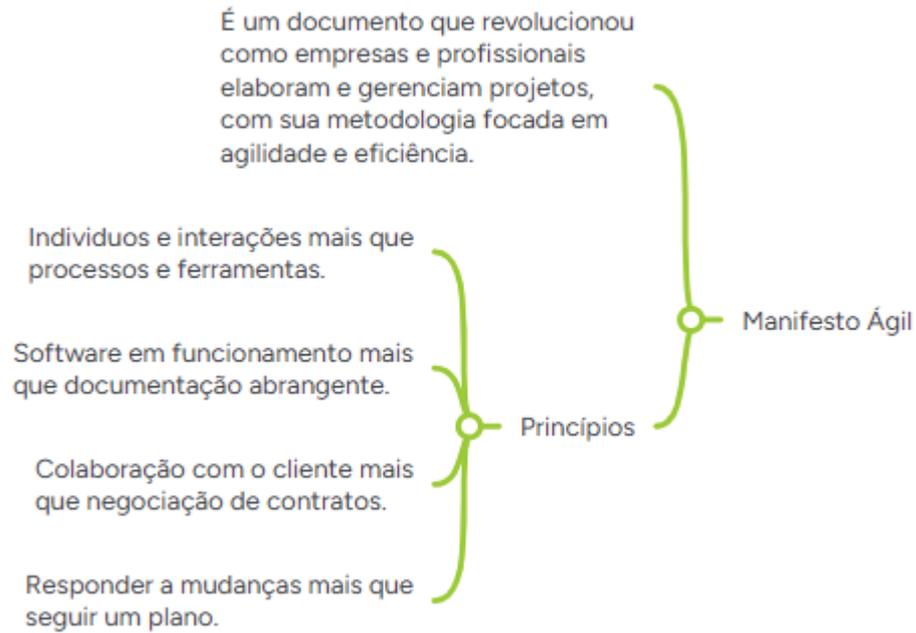
FONTE: A autora (2024).

FIGURA 2 – TÓPICO SCRUM DO MAPA MENTAL



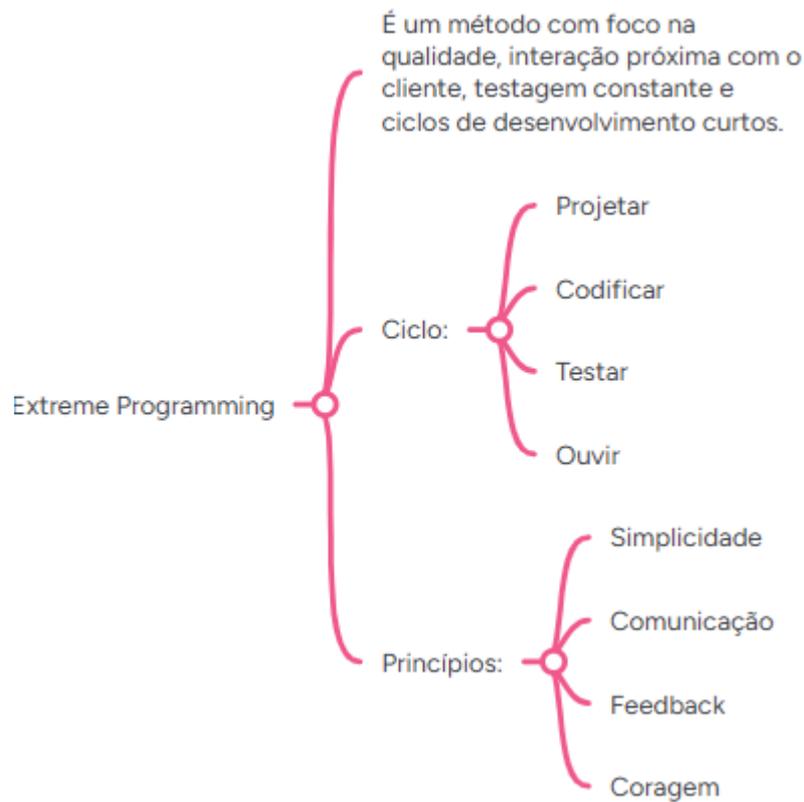
FONTE: A autora (2024).

FIGURA 3 – TÓPICO MANIFESTO ÁGIL DO MAPA MENTAL



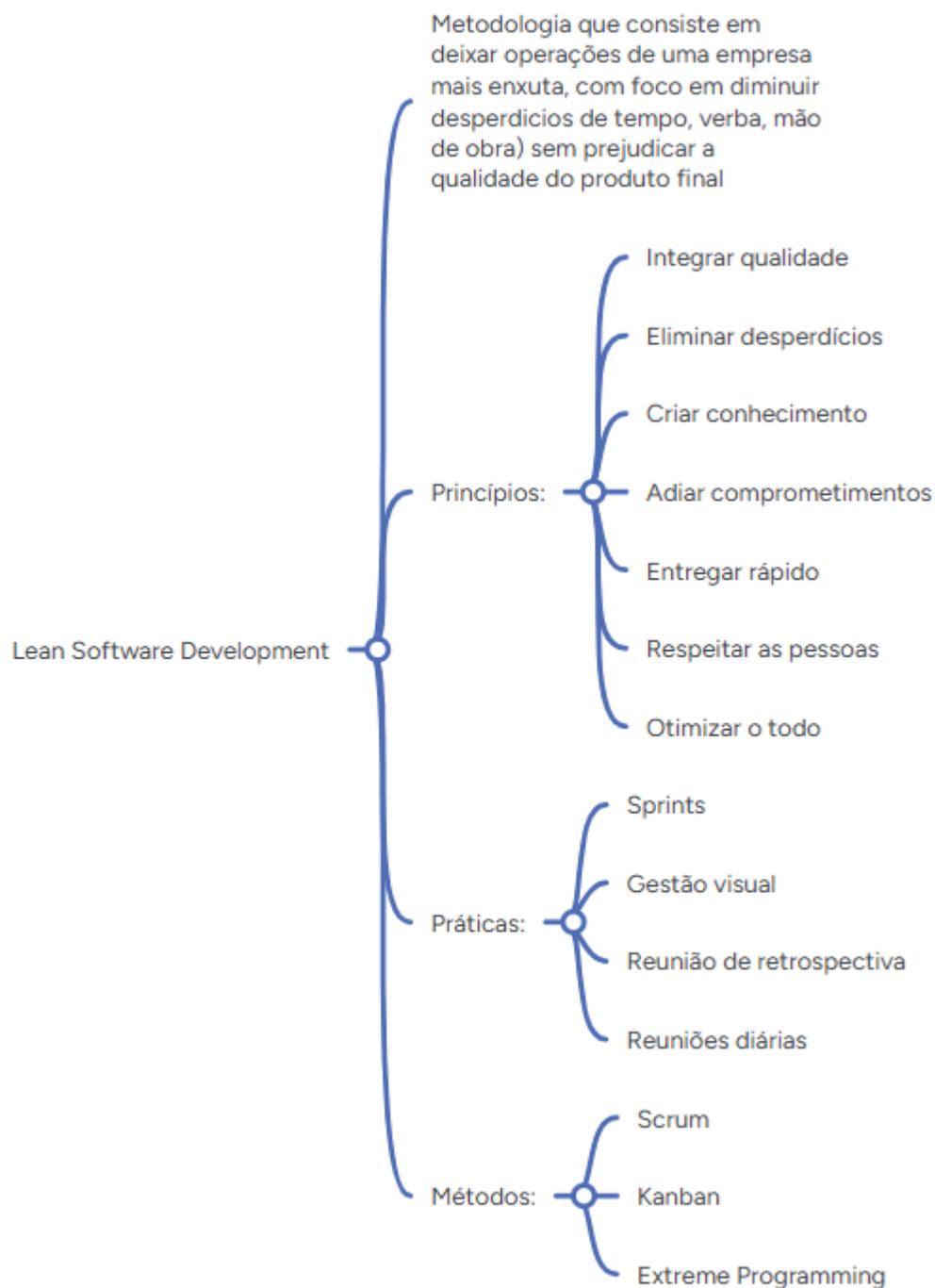
FONTE: A autora (2024).

FIGURA 4 – TÓPICO EXTREME PROGRAMMING DO MAPA MENTAL



FONTE: A autora (2024).

FIGURA 5 – TÓPICO LEAN SOFTWARE DEVELOPMENT DO MAPA MENTAL



FONTE: A autora (2024).

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

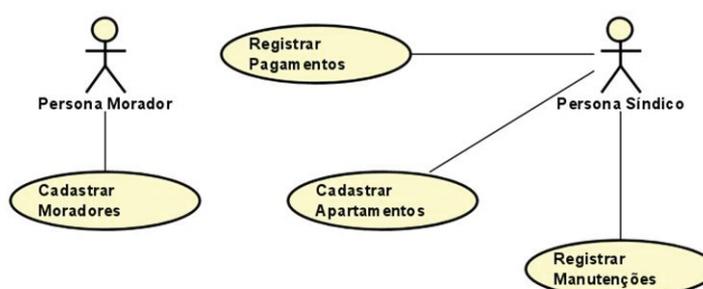
O projeto desenvolvido nas disciplinas de Modelagem Ágil de Software 1 e 2 consistiu na modelagem de um Sistema de Gestão de Condomínio, com o objetivo de automatizar e otimizar os serviços administrativos relacionados à gestão de condomínios. Este projeto foi estruturado em duas partes, nas quais foram definidos requisitos como cadastro de apartamentos e moradores, registro de pagamentos e manutenções.

A aplicação prática dos princípios ágeis foi um aspecto central deste projeto, enfatizando a colaboração, a flexibilidade e a entrega contínua de valor. Ao longo do desenvolvimento, foram realizadas entregas de artefatos, como Diagrama de Caso de Uso e Histórias de Usuário, que foram avaliados e discutidos para permitir a continuidade do trabalho com o Diagrama de Classes, classes com atributos associados, Diagramas de Sequência e todas as Histórias de Usuário. Essa abordagem possibilitou a compreensão da importância de um processo iterativo e adaptativo, onde o feedback e a melhoria contínua são fundamentais.

Os documentos gerados são essenciais para a compreensão do funcionamento do sistema, abrangendo desde os requisitos iniciais até os detalhes de implementação. Isso garante que qualquer equipe de desenvolvimento possa dar continuidade ao projeto com base nos artefatos produzidos.

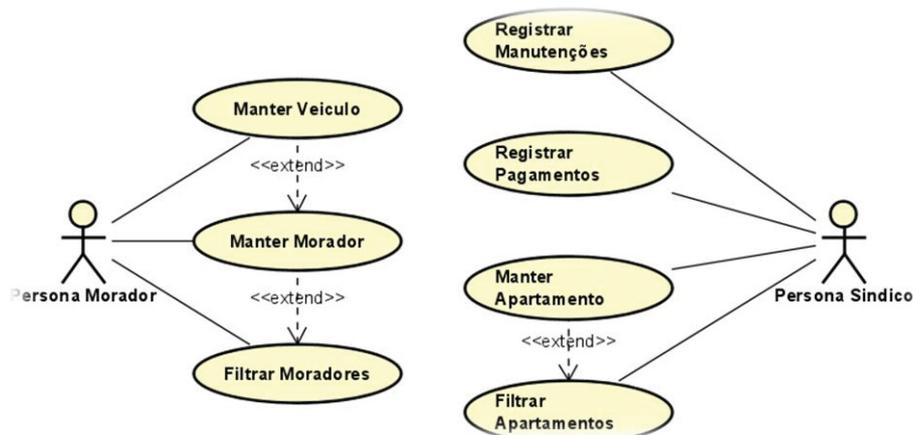
#### 3.1 ARTEFATOS DO PROJETO – MAG1 PT1

FIGURA 6 – DIAGRAMA DE CASO DE USO DE NÍVEL 1



FONTE: A autora (2024).

FIGURA 7 – DIAGRAMA DE CASO DE USO DE NÍVEL 2



FONTE: A autora (2024).

### 3.2 ARTEFATOS DO PROJETO – MAG1 PT2

#### História de Usuário

#### HU006 – Registrar Pagamentos

**SENDO** o Síndico

**QUERO** registrar os pagamentos

**PARA** manter o controle dos pagamentos

FIGURA 8 – DIAGRAMA DE CASO DE USO DE NÍVEL 1

### Registrar Pagamentos

Apartamento:  ▼

Morador:  Preço:

Data Vencimento:  Data Pagamento:

Mês referencial:  Ano:

FONTE: A autora (2024).

#### LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve preencher o campo morador sendo ele o responsável;
- 2) Deve preencher o campo preço;
- 3) Não deve salvar se não houver registro de pagamento do mês anterior;
- 4) Deve salvar os dados do registro de pagamento;
- 5) Deve voltar à tela anterior;

## CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

1) Deve preencher o campo morador sendo ele o responsável;

<b>Dado que</b>	O apartamento foi selecionado
<b>Quando</b>	O cursor sai do campo apartamento
<b>Então</b>	O sistema busca os dados na tabela morador que possua o apartamento correspondente e com o campo responsável = “true”

2) Deve preencher o campo preço;

<b>Dado que</b>	O apartamento foi selecionado
<b>Quando</b>	O cursor sai do campo apartamento
<b>Então</b>	O sistema busca o valor do condomínio na tabela apartamentos e preenche o campo preço

3) Não deve salvar se não houver registro de pagamento do mês anterior;

<b>Dado que</b>	Os campos foram preenchidos
<b>Quando</b>	O botão Salvar é pressionado
<b>Então</b>	O sistema verifica se há um registro de pagamento desse apartamento no ano correspondente e mês anterior. Apresenta a mensagem “Não é possível completar um novo registro com pagamentos pendentes, regularize o pagamento referente ao mês de <mesAnterior>” ou “Dados salvos com sucesso” e retorna à tela anterior

4) Deve salvar os dados do registro de pagamento;

<b>Dado que</b>	Os campos foram preenchidos
<b>Quando</b>	O botão Salvar é pressionado
<b>Então</b>	O sistema salva os dados, apresenta a mensagem “Dados salvos com sucesso” e retorna à tela anterior

5) Deve voltar à tela anterior;

<b>Dado que</b>	O registro de pagamento esta sendo elaborado
<b>Quando</b>	O botão “Voltar” é pressionado
<b>Então</b>	O sistema volta para a tela anterior

## OBSERVAÇÕES TÉCNICAS

- Usar CSS e HTML para desenvolver as telas;
- Utilizar os componentes do primeNg – Angular;
- Usar TypeScript para as chamar as funções;

## CRITÉRIOS DE ACEITAÇÃO PARA TESTES

1) Deve preencher o campo morador sendo ele o responsável;

<b>Dado que</b>	O apartamento “301” foi selecionado
<b>Quando</b>	O cursor sai do campo apartamento
<b>Então</b>	O sistema busca os dados na tabela morador com o apartamento “301” e com o campo responsável = “true”: “Erica Silva”

2) Deve preencher o campo preço;

<b>Dado que</b>	O apartamento “301” foi selecionado
<b>Quando</b>	O cursor sai do campo apartamento
<b>Então</b>	O sistema busca o valor do condomínio na tabela apartamentos e preenche o campo preço: “R\$1700,00”

3) Não deve salvar se não houver registro de pagamento do mês anterior;

<b>Dado que</b>	Os campos foram preenchidos, mes=maio
<b>Quando</b>	O botão Salvar é pressionado
<b>Então</b>	O sistema busca no banco o registro de pagamento deste morador do mes=abril. Se não existir deve apresentar a mensagem: “Não é possível completar um novo registro com pagamentos pendentes, regularize o pagamento referente ao mês de abril” ou se for encontrado: “Dados salvos com sucesso” e retorna à tela anterior

4) Deve salvar os dados do registro de pagamento;

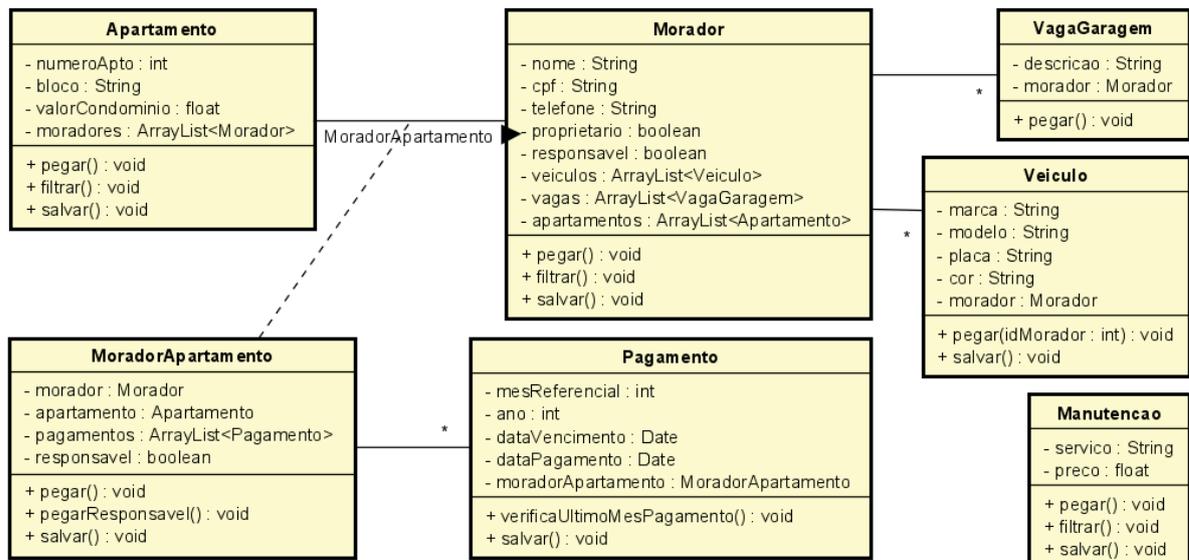
<b>Dado que</b>	Todos os campos estão preenchidos corretamente
<b>Quando</b>	O botão “Salvar” é pressionado
<b>Então</b>	O sistema salva os dados, apresenta a mensagem “Dados salvos com sucesso” e retorna à tela anterior

5) Deve voltar à tela anterior;

<b>Dado que</b>	O registro de pagamento esta sendo elaborado
<b>Quando</b>	O botão “Voltar” é pressionado
<b>Então</b>	O sistema volta para a tela anterior

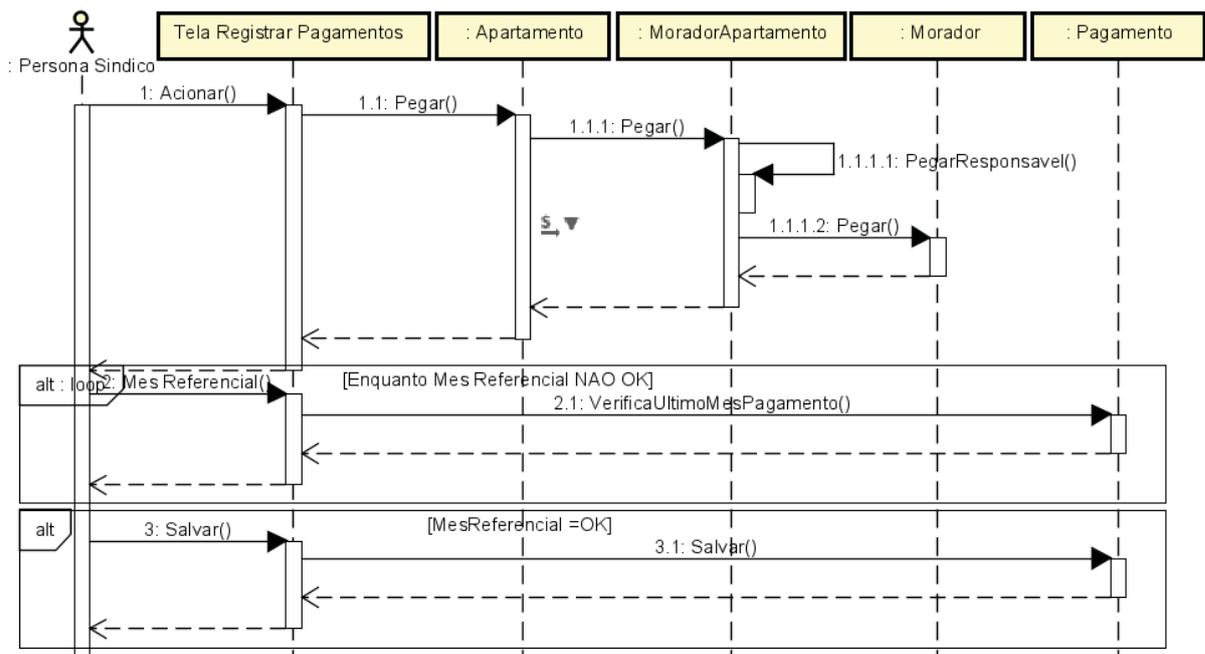
### 3.3 ARTEFATOS DO PROJETO – MAG2

FIGURA 9 – DIAGRAMA DE CLASSES



FONTE: A autora (2024).

FIGURA 10 – DIAGRAMA DE SEQUÊNCIA: HU006-REGISTRAR PAGAMENTOS



FONTE: A autora (2024).

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

O projeto desenvolvido nas disciplinas de Gerenciamento Ágil de Projetos de Software 1 e 2 consistiu na elaboração de um plano de release para um software, com foco na aplicação prática dos princípios ágeis. A atividade permitiu uma compreensão aprofundada das metodologias ágeis, destacando a importância da colaboração, da flexibilidade e da entrega contínua de valor, que são fundamentais no desenvolvimento ágil de software.

A estrutura do projeto foi baseada em um ciclo de desenvolvimento que envolveu o cálculo da velocidade, a definição de sprints e a elaboração de histórias de usuário. A velocidade foi calculada considerando que 1 ponto equivale a 8 horas de trabalho, permitindo uma estimativa realista do tempo necessário para a conclusão das tarefas. As histórias de usuário foram formuladas seguindo o formato SENDO/QUERO/PARA, o que garantiu clareza na identificação das necessidades dos usuários.

A prática de iterações curtas e incrementais possibilitou ajustes rápidos, refletindo a adaptabilidade do projeto às necessidades emergentes. A entrega contínua de valor foi assegurada por meio da priorização das funcionalidades mais relevantes, garantindo que o software atendesse às expectativas dos usuários desde as primeiras sprints.

O conhecimento técnico desenvolvido ao longo do projeto apresenta aplicabilidade em diversos contextos profissionais. Em ambientes de trabalho, a capacidade de adaptação a mudanças e a colaboração eficaz entre membros da equipe são consideradas competências essenciais. A ênfase na comunicação e no feedback contínuo contribuiu para a entrega de projetos com qualidade e dentro dos prazos estabelecidos, enquanto a auto-organização e a busca por melhoria contínua favorecem ambientes produtivos e inovadores.

Essa abordagem representa um modelo eficaz para o desenvolvimento de soluções tecnológicas inovadoras, alinhando-se às exigências do mercado de trabalho contemporâneo. A experiência proporcionada pelo projeto fortalece a formação de profissionais aptos a atuar em contextos dinâmicos e colaborativos.

### 4.1 ARTEFATOS DO PROJETO – GAP1

FIGURA 11 – PLANO DE RELEASE

**Cálculo da Velocidade:**

Horas disponíveis por dia:	5	Tamanho da Sprint:	2 Semanas (10 dias)
Horas disponíveis por Sprint:	50 horas/sprint	Velocidade:(50 horas/sprint) / 8 (1ponto)	6

**Plano de Release:**

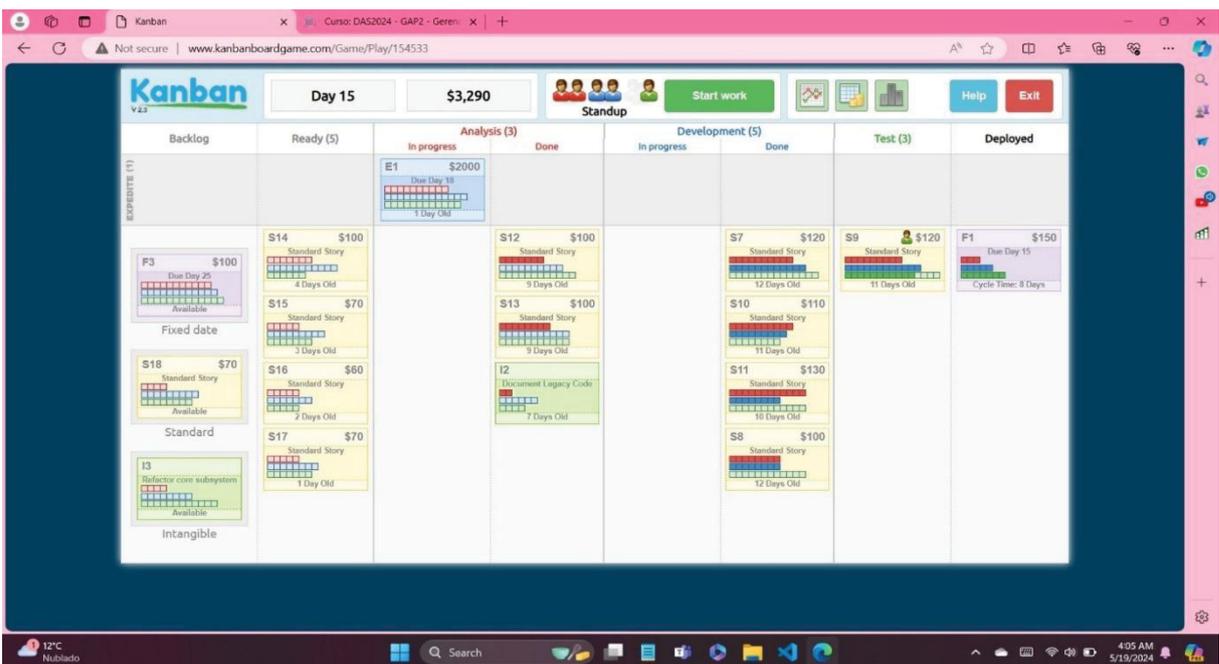
Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3
Data Início: 06/05/2024 Data Fim: 17/05/2024	Data Início: 20/05/2024 Data Fim: 31/05/2024	Data Início: 03/06/2024 Data Fim: 14/06/2024
< HU002 – Manter Apartamento > SENDO o Síndico QUERO Manter os dados do apartamento PARA Manter seus dados atualizados ESTIMATIVA (4)	<HU004 – Manter Morador > SENDO o Morador QUERO Manter os dados do morador PARA Manter seus dados atualizados ESTIMATIVA (4)	<HU005 – Manter Veículo > SENDO o Morador QUERO Manter os dados do veículo PARA Manter seus dados atualizados ESTIMATIVA (6)
< HU001 – Filtrar Apartamentos > SENDO o Síndico QUERO Filtrar os apartamentos PARA Fazer alterações nos seus dados ESTIMATIVA (2)	<HU003 – Filtrar Moradores > SENDO o Morador QUERO Filtrar os moradores cadastrados PARA Fazer alterações nos seus dados ESTIMATIVA (2)	

Iteração/Sprint 4	Iteração/Sprint 5
Data Início: 17/06/2024 Data Fim: 28/06/2024	Data Início: 01/07/2024 Data Fim: 12/07/2024
<HU006 – Registrar Pagamentos > SENDO o Síndico QUERO Registrar os pagamentos PARA Manter o controle dos pagamentos ESTIMATIVA (5)	<HU007 – Registrar Manutenções > SENDO o Síndico QUERO Registrar as manutenções no condomínio PARA Manter o controle dos gastos ESTIMATIVA (6)

FONTE: A autora (2024).

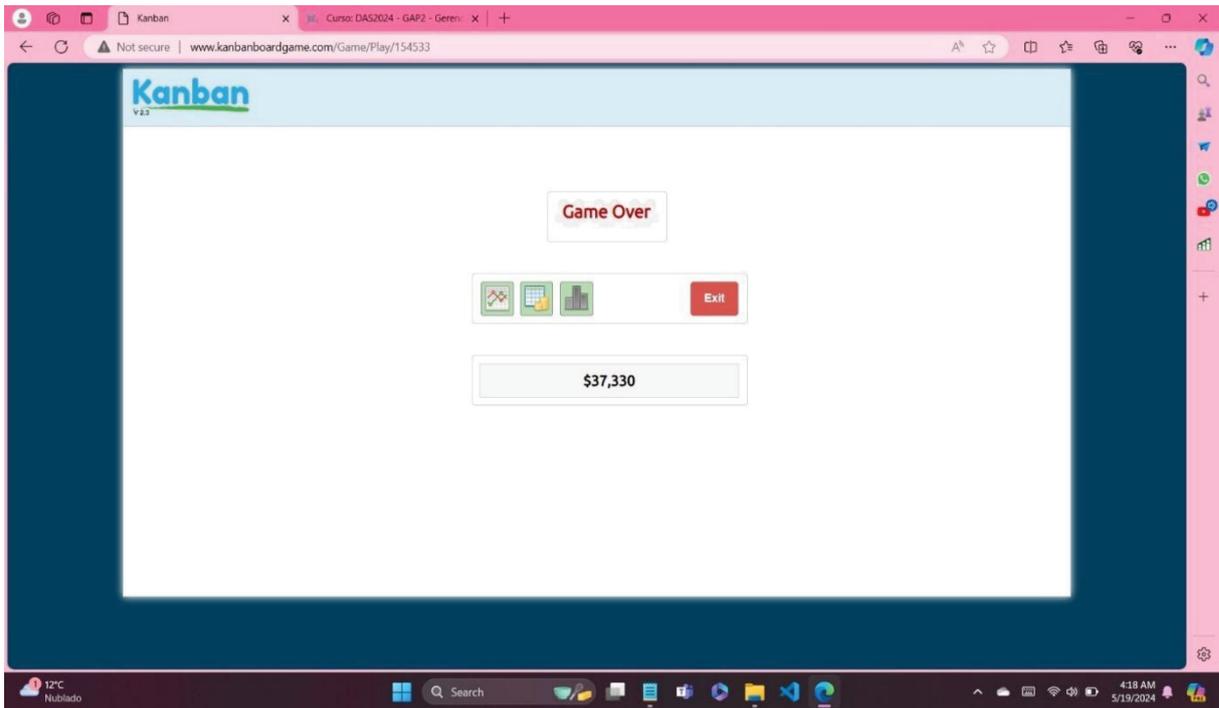
### 4.2 ARTEFATOS DO PROJETO – GAP2

FIGURA 12 – PRINT DE TELA DO DIA 15: A EXECUÇÃO DO JOGO KANBANBOARDGAME.COM



FONTE: A autora (2024).

FIGURA 13 – PRINT DE TELA DO RESULTADO FINAL COM A RECEITA FINAL OBTIDA



FONTE: A autora (2024).

FIGURA 14 – PRINT DE TELA DO CFD, NO FINAL DA EXECUÇÃO DO JOGO



FONTE: A autora (2024).

## 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

O projeto desenvolvido na disciplina de Introdução à Programação consistiu na implementação do backend de um sistema bancário simplificado, focado no controle de cadastro de clientes, contas correntes e contas de investimento. Atendendo às necessidades básicas de um banco, permitindo a gestão eficiente de informações financeiras.

Para a execução do projeto, foram fornecidos recursos essenciais, incluindo um diagrama de classes que orientou o desenvolvimento do sistema, um projeto no NetBeans com testes unitários já escritos e um script DDL para a criação das tabelas do banco de dados no MySQL. Esses elementos foram fundamentais para garantir que o desenvolvimento fosse estruturado e que as funcionalidades fossem implementadas de acordo com as especificações.

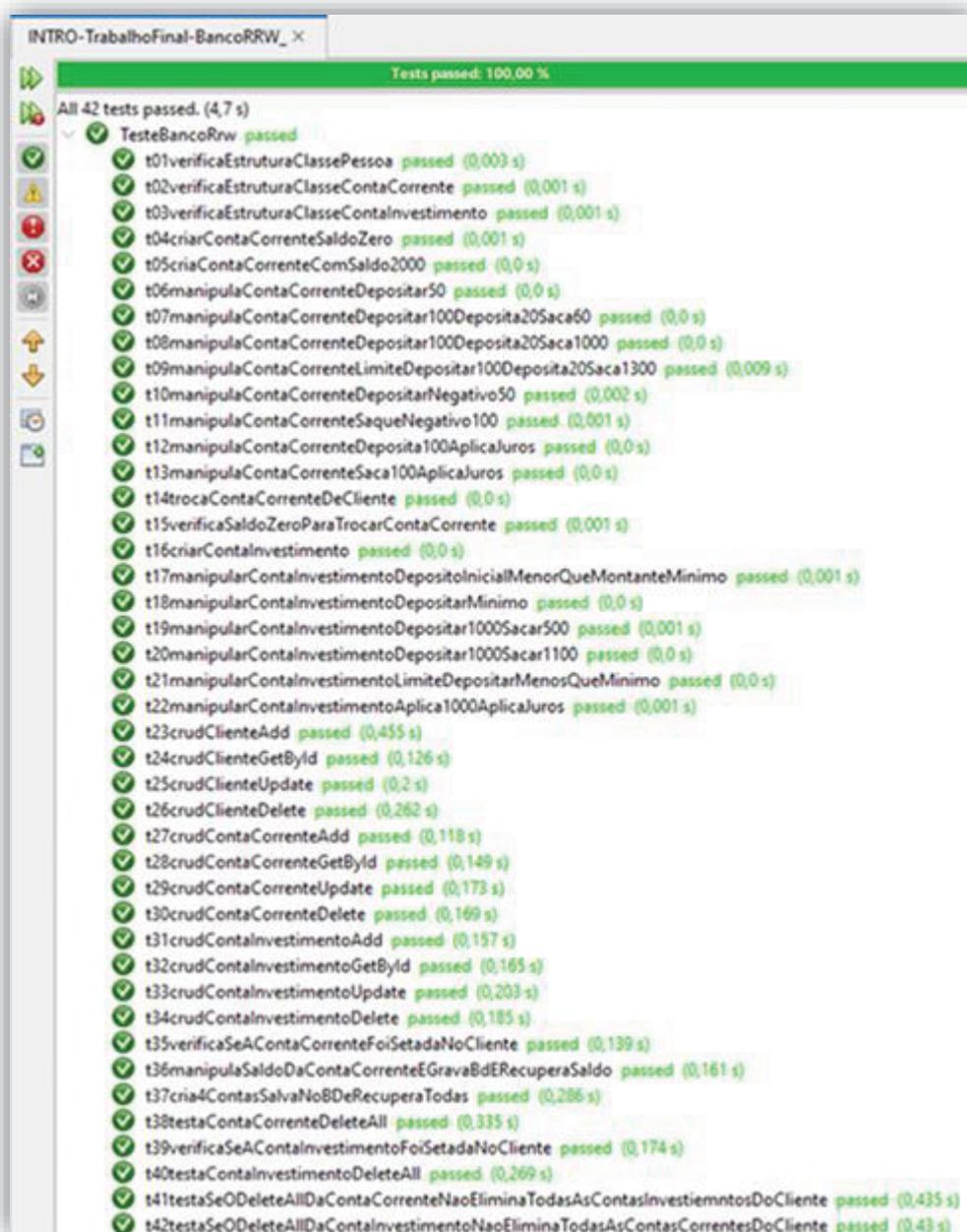
A implementação das classes do sistema foi realizada com o objetivo de garantir que todos os testes unitários ficassem em estado "verde", ou seja, que todos os 42 casos de teste fossem bem-sucedidos. A integração entre o backend e o banco de dados foi realizada com cuidado, assegurando que as operações de CRUD (criação, leitura, atualização e exclusão) fossem implementadas corretamente.

O projeto proporcionou uma base sólida para o desenvolvimento de sistemas, enfatizando a importância da estruturação do código, da persistência de dados e da realização de testes. A implementação do backend reforçou conceitos fundamentais de programação orientada a objetos e evidenciou como o desenvolvimento guiado por testes unitários desde o início do projeto contribui diretamente para a qualidade do software.

Essa abordagem permite identificar e corrigir erros com agilidade, assegura que novas funcionalidades possam ser incorporadas sem comprometer as existentes e oferece maior confiabilidade ao sistema como um todo. Trata-se de uma experiência essencial para aplicações futuras em ambientes de desenvolvimento, preparando o terreno para enfrentar desafios mais complexos com segurança, organização e eficiência.

## 5.1 ARTEFATOS DO PROJETO

FIGURA 15 – CASOS DE TESTE JUNIT



FONTE: A autora (2024).

## 6 DISCIPLINA: BD – BANCO DE DADOS

O projeto desenvolvido na disciplina de Banco de Dados consistiu na modelagem e implementação de um sistema de controle de biblioteca, abrangendo o cadastro de obras, usuários, funcionários, empréstimos e reservas. Este projeto foi fundamental para a compreensão dos conceitos de modelagem de dados, normalização, e a utilização de SQL para a criação e manipulação de bancos de dados relacionais.

Durante o desenvolvimento do projeto, foram elaborados diversos componentes essenciais. Inicialmente, foi criado um modelo entidade-relacionamento que definiu as entidades principais como Obra, Usuário, Funcionário, Departamento, Empréstimo e Reserva, além de seus atributos e relacionamentos. A partir desse modelo, foi gerado um diagrama lógico que traduziu as entidades em tabelas, especificando chaves primárias e estrangeiras, garantindo a integridade referencial dos dados.

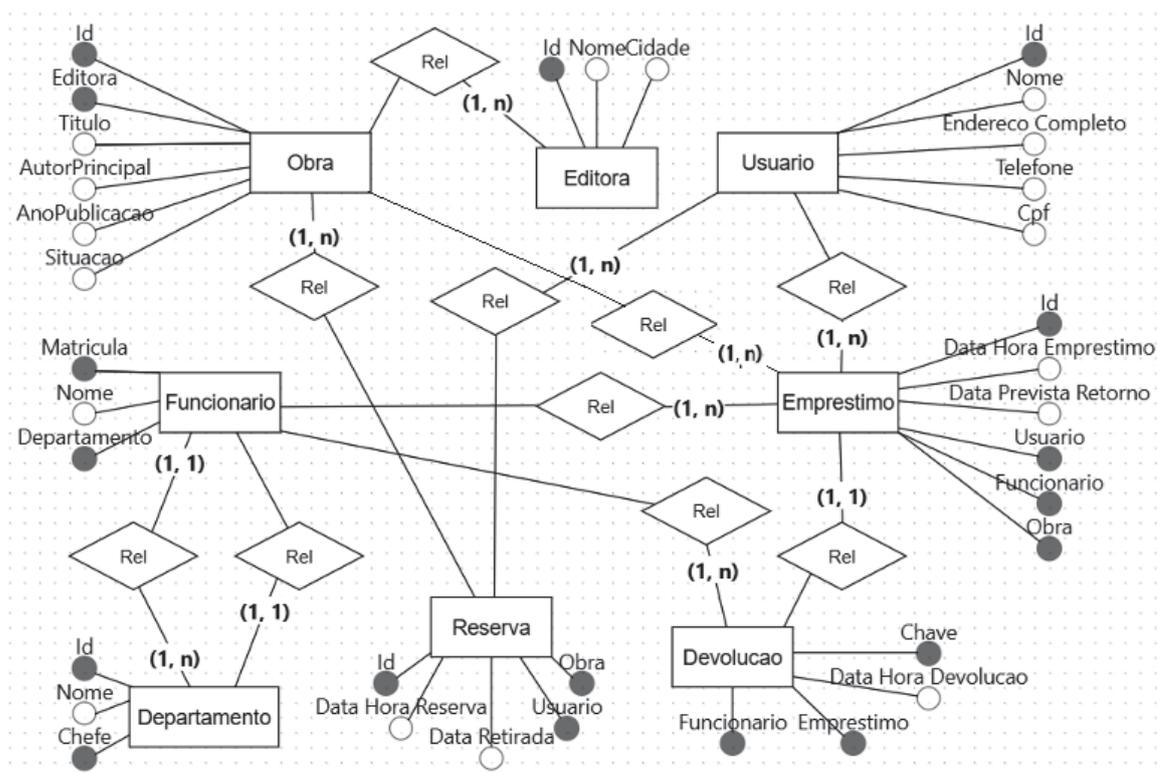
A implementação prática incluiu a criação de um banco de dados no MySQL, onde foram executados scripts SQL para a criação das tabelas e inserção de dados de exemplo. O sistema foi projetado para permitir operações de CRUD (criação, leitura, atualização e exclusão) para cada uma das entidades, facilitando a gestão das informações da biblioteca.

A importância deste projeto no desenvolvimento ágil de software reside na sua capacidade de promover uma abordagem iterativa e incremental. A modelagem de dados permite que os requisitos do sistema sejam traduzidos em estruturas de dados que podem ser facilmente adaptadas conforme novas necessidades surgem. Isso é especialmente relevante em ambientes ágeis, onde a flexibilidade e a capacidade de resposta a mudanças são cruciais. A implementação de um banco de dados eficiente garante que as informações sejam armazenadas de forma organizada e acessível, facilitando a entrega contínua de valor ao usuário final.

Além disso, o projeto de Banco de Dados se integrou de maneira significativa com as disciplinas de Introdução à Programação e Modelagem Ágil de Software. A programação do backend do sistema depende diretamente da estrutura de dados definida no modelo de banco de dados, enquanto a modelagem ágil ajuda a identificar os requisitos e as funcionalidades necessárias para atender às expectativas dos usuários.

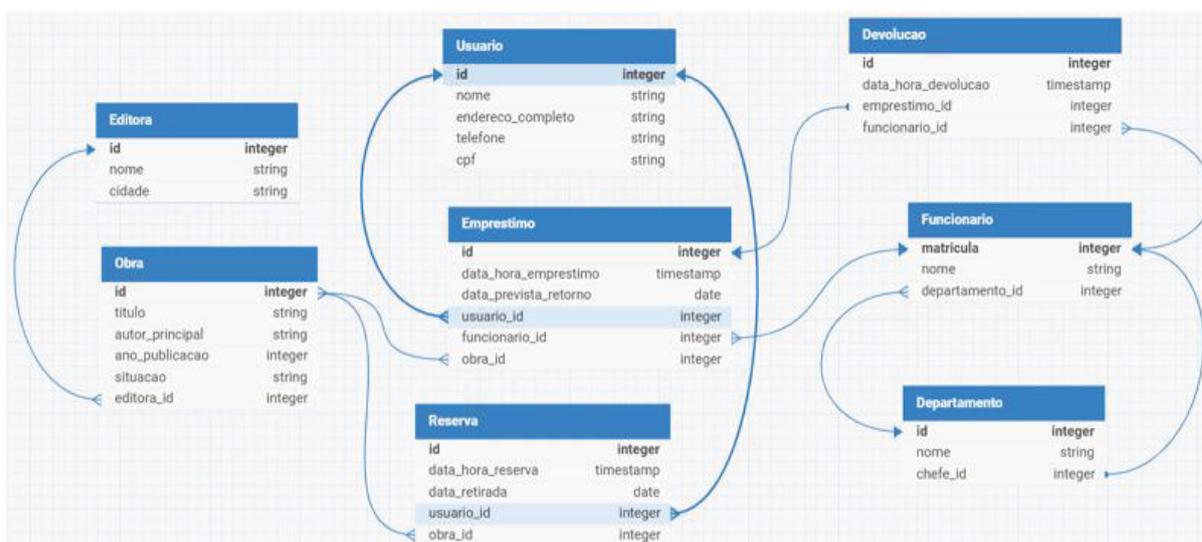
## 6.1 ARTEFATOS DO PROJETO – QUESTÃO 1

FIGURA 16 – MODELO ENTIDADE-RELACIONAMENTO CONCEITUAL



FONTE: A autora (2024).

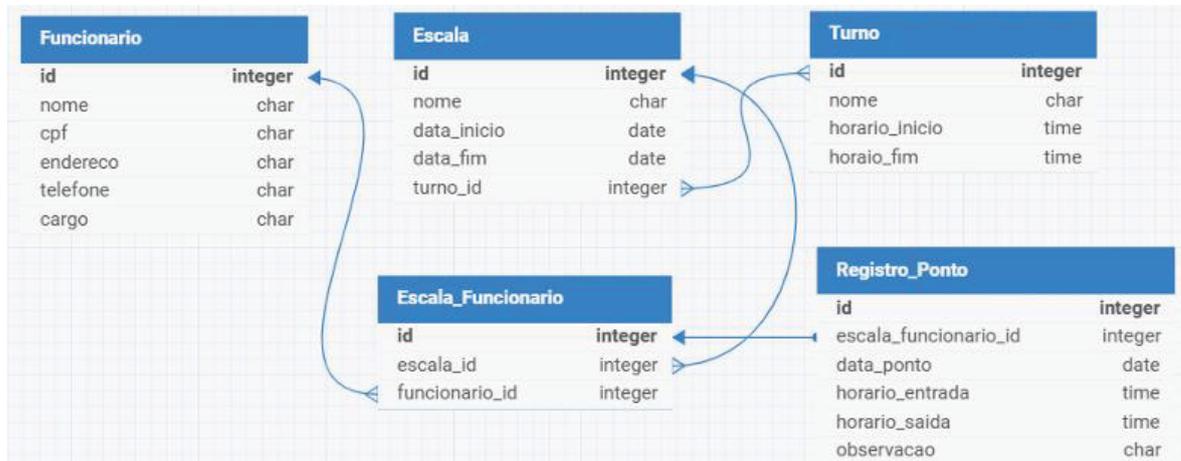
FIGURA 17 – MODELO LÓGICO: DIAGRAMA DE ENTIDADE-RELACIONAMENTO



FONTE: A autora (2024).

## 6.2 ARTEFATOS DO PROJETO – QUESTÃO 2

FIGURA 18 – MODELO LÓGICO: REGISTRO DE JORNADA DE UM CURSO PROFISSIONALIZANTE



FONTE: A autora (2024).

FIGURA 19 – SCRIPT SQL: CRIAÇÃO DE TABELAS

```

CREATE TABLE public.funcionario (
  id int8 NOT NULL,  nome varchar(255) NULL,
  cpf varchar(255) NOT NULL,
  endereco varchar(255) NULL,
  telefone varchar(255) NULL,
  cargo varchar(255) NULL,
  CONSTRAINT funcionario_pk PRIMARY KEY (id),
  CONSTRAINT funcionario_unique UNIQUE (cpf));

CREATE TABLE public.turno (
  id int8 NOT NULL,
  nome varchar(255) NULL,
  horario_inicio time NULL,
  horario_fim time NULL,
  CONSTRAINT turno_pk PRIMARY KEY (id));

CREATE TABLE public.escala (
  id int8 NOT NULL,
  nome varchar(255) NULL,
  data_inicio date NULL,
  data_fim date NULL,
  turno_id int8 NULL,
  CONSTRAINT escala_pk PRIMARY KEY (id),
  CONSTRAINT turno_fk FOREIGN KEY (turno_id) REFERENCES public.turno(id));

CREATE TABLE public.escala_funcionario (
  id int8 NOT NULL,
  escala_id int8 NULL,
  funcionario_id int8 NULL,
  CONSTRAINT escala_funcionario_pk PRIMARY KEY (id),
  CONSTRAINT escala_fk FOREIGN KEY (escala_id) REFERENCES public.escala(id),
  CONSTRAINT funcionario_fk FOREIGN KEY (funcionario_id) REFERENCES public.funcionario(id));

CREATE TABLE public.registro_ponto (
  id int8 NOT NULL,
  data_ponto date NULL,
  horario_entrada time NULL,
  horario_saida time NULL,
  observacoes varchar(1024) NULL,
  escala_funcionario_id int8 NULL,
  CONSTRAINT registro_ponto_pk PRIMARY KEY (id),
  CONSTRAINT escala_funcionario_fk FOREIGN KEY (escala_funcionario_id) REFERENCES public.escala_funcionario(id));

```

FONTE: A autora (2024).  
 FIGURA 20 – SCRIPT SQL: INSERÇÃO DE DADOS

```
CREATE SEQUENCE serial5 START 1000;
INSERT INTO public.registro_ponto (id, data_ponto, horario_entrada, horario_saida, escala_funcionario_id) VALUES
((SELECT nextval('serial5')), '2024-07-01', '07:58', '12:01', 100),
((SELECT nextval('serial5')), '2024-07-01', '12:59', '17:02', 101),
((SELECT nextval('serial5')), '2024-07-05', '08:00', '12:00', 102),
((SELECT nextval('serial5')), '2024-07-05', '14:00', '18:05', 103),
((SELECT nextval('serial5')), '2024-07-06', '09:58', '14:01', 104),
((SELECT nextval('serial5')), '2024-07-06', '09:59', '14:00', 106),
((SELECT nextval('serial5')), '2024-07-06', '15:01', '19:00', 105),
((SELECT nextval('serial5')), '2024-07-06', '15:00', '18:57', 107),
((SELECT nextval('serial5')), '2024-07-01', '07:54', '12:00', 108),
((SELECT nextval('serial5')), '2024-07-01', '08:00', '12:01', 109);
```

FONTE: A autora (2024).

FIGURA 21 – SCRIPT SQL: VIZUALIZAÇÃO DOS DADOS COM RELACIONAMENTOS

	id	data_ponto	horario_entrada	horario_saida	observacoes	escala_funcionario_id
1	1,000	2024-07-01	07:58:00	12:01:00	[NULL]	100
2	1,001	2024-07-01	12:59:00	17:02:00	[NULL]	101
3	1,002	2024-07-05	08:00:00	12:00:00	[NULL]	102
4	1,003	2024-07-05	14:00:00	18:05:00	[NULL]	103
5	1,004	2024-07-06	09:58:00	14:01:00	[NULL]	104
6	1,005	2024-07-06	09:59:00	14:00:00	[NULL]	106
7	1,006	2024-07-06	15:01:00	19:00:00	[NULL]	105
8	1,007	2024-07-06	15:00:00	18:57:00	[NULL]	107
9	1,008	2024-07-01	07:54:00	12:00:00	[NULL]	108
10	1,009	2024-07-01	08:00:00	12:01:00	[NULL]	109

FONTE: A autora (2024).

## 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

O projeto desenvolvido na disciplina de Aspectos Ágeis de Programação teve como foco a implementação de algoritmos de ordenação, com ênfase no Bubble Sort.. Através da prática de desenvolvimento, pode se aplicar conceitos teóricos em um contexto prático. A prática de refatoração do código são essenciais para garantir a qualidade do software, permitindo que identifiquem e corrijam problemas de forma ágil. Durante o projeto, diversas mudanças foram implementadas para melhorar a qualidade do código. Entre as principais alterações, destacam-se:

- **Colocar o método main no início da classe:** Essa mudança melhora a legibilidade, permitindo que o ponto de entrada do programa seja facilmente identificado.
- **Renomear variáveis para inglês e torná-las mais descritivas:** Variáveis como array foram renomeadas para numbersArray, facilitando a compreensão do código.
- **Remover comentários desnecessários:** Comentários que não agregavam valor foram eliminados, tornando o código mais limpo e autoexplicativo.
- **Declarar as variáveis o mais próximo possível de onde serão usadas:** Essa prática melhora a clareza e a manutenção do código.
- **Reduzir os parâmetros dos métodos:** O número de parâmetros foi minimizado, tornando os métodos mais fáceis de entender e usar.
- **Dividir o método bubbleSort em dois:** Essa mudança aumentou a modularidade do código, permitindo que cada método tivesse uma única responsabilidade, podendo ser aproveitado em outras funções.
- **Identar as chaves {} conforme combinado com a equipe:** A consistência na formatação do código é importante para a legibilidade e colaboração.
- **Simplificar o código == false para !:** Essa simplificação torna o código mais conciso e fácil de ler.
- **Simplificar o código de impressão:** O loop de impressão foi alterado para um formato mais limpo e legível, utilizando um loop aprimorado.
- **Remover o método swapPositions:** Essa remoção simplificou o código, mantendo todas as linhas abaixo de 80 caracteres e evitando complexidade desnecessária.

Este projeto também se integrou de maneira significativa com outras disciplinas, como Introdução à Programação e Banco de Dados. A programação de algoritmos de ordenação está diretamente relacionada à manipulação de dados, e a compreensão de como os dados são organizados, nomeados e acessados é crucial para o desenvolvimento de sistemas eficientes. Essas alterações contribuíram para um código mais limpo, legível e eficiente, alinhando-se aos princípios de *Clean Code* (Martin, 2008), que enfatizam a clareza, simplicidade e responsabilidade na escrita de software.

## 7.1 ARTEFATOS DO PROJETO

FIGURA 22 – VERSÃO CLEAN CODE DO BUBBLE SORT

```

C:\Users> iasmi > OneDrive > Documents > Study > Pos > Trabalhos Entregues > Trabalho AAP.java > ...
1 // Implementação otimizada em Java do Bubble sort
2 // Código extraído de https://www.geeksforgeeks.org/bubble-sort/
3
4 import java.io.*;
5
6 class BubbleSort {
7     Run | Debug
8     public static void main(String args[]){
9         int numbersArray[] = { 64, 34, 25, 12, 22, 11, 90 };
10        bubbleSort(numbersArray);
11        System.out.println(x:"Array ordenado: ");
12        printArray(numbersArray);
13    }
14
15    static void bubbleSort(int numbersArray[]){
16        int sizeArray = numbersArray.length;
17        boolean swapped;
18        for(int i = 0; i < sizeArray - 1; i++) {
19            swapped = false;
20            for (int j = 0; j < sizeArray - i - 1; j++) {
21                if(numbersArray[j] > numbersArray[j + 1]){
22                    swapPositions(numbersArray, j, j + 1);
23                    swapped = true;
24                }
25            }
26            // If no elements have been swapped, break the loop
27            if(!swapped) break;
28        }
29    }
30
31    static public void swapPositions(int numbersArray[], int firstId, int secondId){
32        int valueOffFirstId = numbersArray[firstId];
33        numbersArray[firstId] = numbersArray[secondId];
34        numbersArray[secondId] = valueOffFirstId;
35    }
36
37    static void printArray(int numbersArray[]){
38        for(int number : numbersArray){
39            System.out.print(number + " ");
40        }
41        System.out.println();
42    }
43 }

```

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

O projeto desenvolvido nas disciplinas de Desenvolvimento Web 1 e 2 consistiu na criação de um sistema de gerenciamento educacional, que implementou dois CRUDs (Create, Read, Update, Delete) na primeira parte e três CRUDs na segunda parte, abrangendo Alunos, Cursos e Matrículas. Este projeto foi fundamental para a aplicação prática dos conceitos de desenvolvimento ágil de software, permitindo que os alunos experimentassem a construção de aplicações web de forma iterativa e colaborativa.

Na primeira parte do projeto, foi desenvolvido um CRUD para gerenciar informações de Alunos e Cursos, utilizando Local Storage para armazenar os dados. A implementação incluiu telas para listagem, inserção, alteração e remoção de registros, proporcionando uma experiência prática na manipulação de dados. O uso de frameworks como Bootstrap ou Material Design para o desenvolvimento das interfaces garantiu que as telas fossem responsivas e visualmente agradáveis, facilitando a interação do usuário.

Na segunda parte do projeto, expandiu o sistema para incluir um CRUD de Matrículas, que relaciona Alunos e Cursos. Essa etapa exigiu a implementação de um back-end em Spring Boot com acesso ao banco de dados PostgreSQL, permitindo a aplicação de conceitos de persistência de dados e integração entre front-end e back-end. A criação de tabelas para Alunos, Cursos e Matrículas no banco de dados foi essencial para garantir a integridade e a organização das informações.

A integração com disciplinas como Banco de Dados, Introdução à Programação e Desenvolvimento Web foi essencial para consolidar o aprendizado técnico. A prática com banco de dados relacionais permitiu compreender a estrutura e manipulação eficiente de dados, enquanto a implementação de um sistema completo reforçou a lógica de programação e a organização do código. Já o desenvolvimento de aplicações web trouxe experiência concreta com tecnologias voltadas à interface e à comunicação entre camadas do sistema. Essa combinação de conhecimentos teóricos e práticos resultou em uma formação mais sólida, preparando para desafios reais no desenvolvimento de software.

## 8.1 ARTEFATOS DO PROJETO

FIGURA 23 – TELA DE UPDATE ALUNO

FONTE: A autora (2024).

FIGURA 24 – TELA DE UPDATE CURSO

FONTE: A autora (2024).

FIGURA 25 – TELA DE LISTAGEM ALUNOS

Nome	CPF	E-mail	data nascimento
joao	123456789-0	joao@gmail.com	07/08/2000

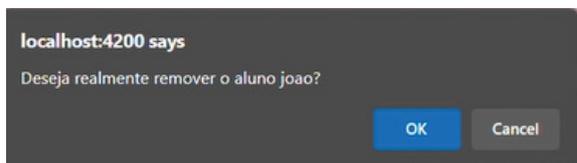
FONTE: A autora (2024).

FIGURA 26 – TELA DE LISTAGEM CURSOS

Nome	Link
desenvolvimento humano	http/linkvideoaula

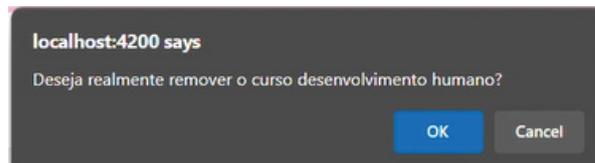
FONTE: A autora (2024).

FIGURA 27 – POP-UP DE DELETAR ALUNO



FONTE: A autora (2024).

FIGURA 28 – POP-UP DE DELETAR CURSO



FONTE: A autora (2024).

FIGURA 29 – SQL: CRIAÇÃO DAS TABELAS NO BANCO DE DADOS PELO VSC

```

1  create database crud;
2
3  create table tb_aluno (
4      id integer primary key generated always as identity,
5      nome varchar(100) not null,
6      cpf varchar(14) not null,
7      email varchar(100) not null,
8      data_nascimento varchar(10) not null
9  );
10
11 create table tb_curso (
12     id integer primary key generated always as identity,
13     nome varchar(100) not null,
14     link varchar(255) not null
15 );
16
17 create table tb_matricula (
18     id integer primary key generated always as identity,
19     aluno_id integer REFERENCES tb_aluno(id) on delete cascade not null,
20     curso_id integer references tb_curso(id) on delete cascade not null,
21     data_matricula varchar(10) not null,
22     nota decimal(5, 2) not null,
23     unique(aluno_id, curso_id)
24 );
25
26 insert into tb_aluno (nome, cpf, email, data_nascimento)
27 values ('Arthur', '111.222.333-44', 'arthur01@email.com', '02/06/2000')
28
29 insert into tb_curso (nome, link)
30 values ('Web 2', 'https://web2.aulas/10')
31
32 insert into tb_matricula (aluno_id, curso_id, data_matricula, nota)
33 values (1, 1, '15/01/2025', 9.8)
34

```

FONTE: A autora (2024).

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

O projeto desenvolvido na disciplina de UX no Desenvolvimento Ágil consistiu na criação de um software para cadastrar moradores de um condomínio residencial em um novo sistema de acesso eletrônico de um condomínio residencial. O síndico é o principal usuário da aplicação, sendo responsável por registrar apartamentos, moradores e veículos, e sua jornada de navegação foi cuidadosamente mapeada para garantir eficiência e clareza.

A navegação do síndico começa pela tela principal, onde ele pode adicionar novos apartamentos, moradores e veículos. Cada ação realizada gera um feedback visual imediato, reforçando a interatividade e o controle sobre o sistema. O design adotou a cor azul como elemento central, transmitindo sensações de inovação e segurança, enquanto os ícones foram desenvolvidos com base nas diretrizes do Material Design, garantindo familiaridade e facilidade de uso.

A interface foi pensada para manter consistência em fontes, cores e layout, promovendo uma experiência fluida e intuitiva. A tela inicial destaca as informações mais relevantes, com um menu lateral no canto superior esquerdo que facilita a navegação. No lado direito, um ícone identifica o usuário logado, oferecendo transparência sobre a sessão ativa. Desde o início, o projeto priorizou a usabilidade e a experiência do usuário, aliando princípios de design à metodologia ágil para entregar uma solução eficiente e centrada nas necessidades reais de síndicos e moradores.

A colaboração entre disciplinas como Modelagem Ágil de Software e Desenvolvimento de Software foi fundamental para atender não apenas aos requisitos funcionais, mas também para criar uma interface agradável, intuitiva, flexível e iterativa, essenciais em ambientes de desenvolvimento ágil.

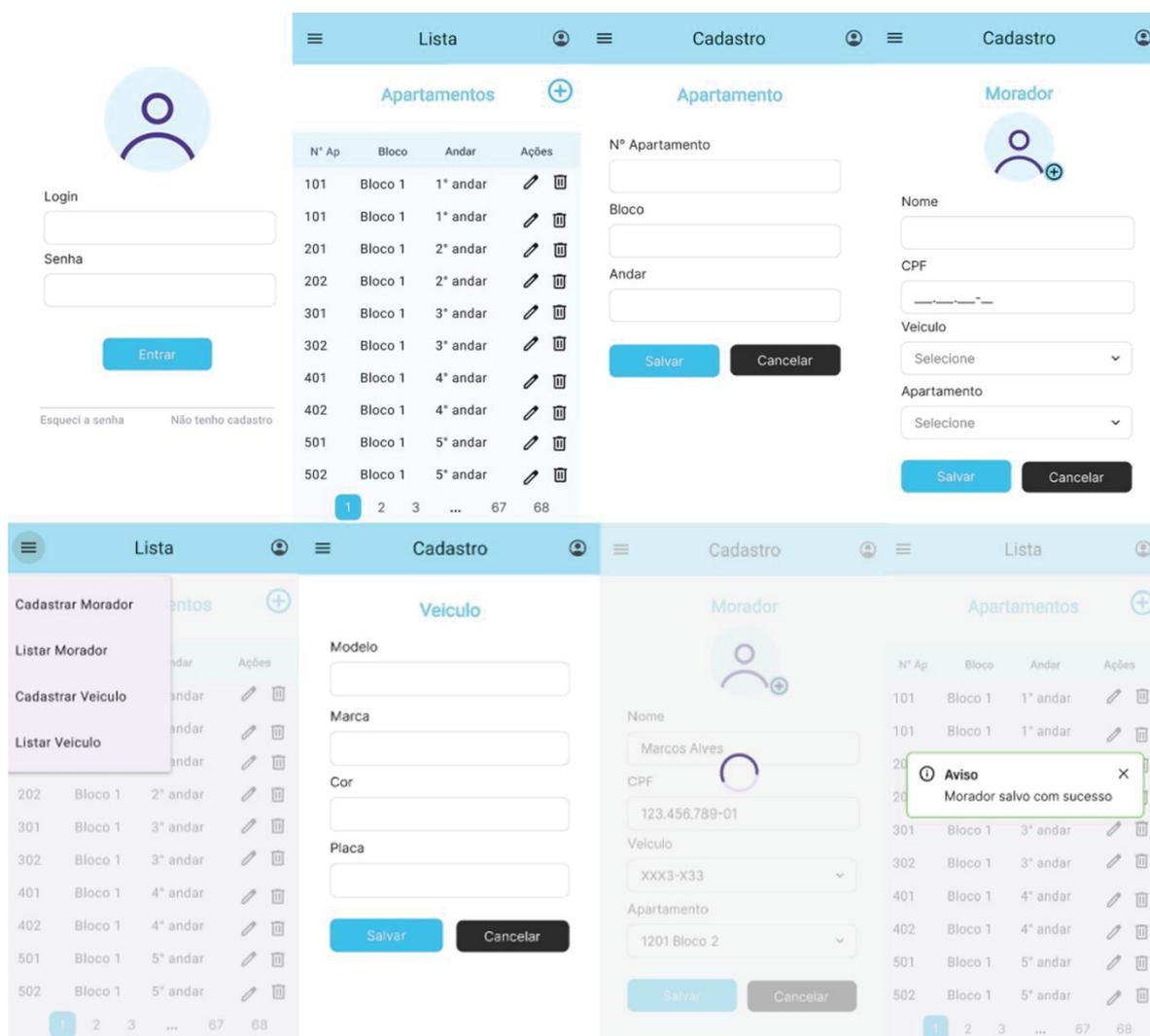
A prototipagem de telas foi um dos pilares do projeto, desempenhando papel essencial no refinamento da interface. Essa abordagem possibilitou testes antecipados, identificação de falhas de usabilidade e validação de escolhas visuais antes da implementação definitiva. Além de facilitar melhorias, a prototipagem promoverá o alinhamento entre desenvolvedores, designers e usuários finais, fortalecendo a comunicação e reduzindo retrabalho.

Os conhecimentos adquiridos nesta disciplina servem como base sólida para etapas posteriores do desenvolvimento de sistemas. Antes mesmo de iniciar disciplinas como Introdução à Programação, Desenvolvimento Mobile ou Banco de

Dados, é essencial compreender como o usuário interage com a interface e quais informações são realmente relevantes. Essa compreensão orienta decisões técnicas futuras, como a estruturação de tabelas, definição de fluxos lógicos e implementação de funcionalidades. Assim, a disciplina de UX atua como elo entre o planejamento visual e a execução técnica, promovendo uma visão integrada e estratégica do desenvolvimento de software.

## 9.1 ARTEFATOS DO PROJETO

FIGURA 30 – HISTÓRIA DE USUÁRIO COM PROTÓTIPO DE TELAS



FONTE: A autora (2024).

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

O projeto desenvolvido nas disciplinas de Desenvolvimento Mobile 1 e 2 consistiu na criação de dois aplicativos distintos, cada um com objetivos específicos, mas ambos focados em proporcionar soluções práticas e eficientes para os usuários. O primeiro aplicativo, voltado para a gestão financeira, permitiu que os usuários cadastrassem e organizassem seus gastos e ganhos, enquanto o segundo aplicativo consultou a HP-API, permitindo acesso a informações sobre personagens do universo Harry Potter.

### **Desenvolvimento do Aplicativo FinTech**

O primeiro projeto foi um aplicativo projetado para ajudar os usuários a gerenciar suas finanças pessoais. O desenvolvimento começou com a definição dos requisitos funcionais e não funcionais, que incluíam a criação de uma interface intuitiva e a capacidade de registrar transações financeiras de forma simples.

1. Activity Principal - Dashboard: A interface principal foi projetada para ser clara e acessível, com botões que permitiam ao usuário navegar facilmente entre as funcionalidades do aplicativo, como cadastro de transações e visualização de extratos. A disposição equilibrada dos botões foi uma prioridade para garantir uma experiência de usuário fluida.

2. Activity Cadastro de Operações: Esta funcionalidade permitiu que os usuários registrassem suas transações financeiras, escolhendo entre débitos e créditos, e inserindo uma descrição e um valor. A decisão de armazenar os dados em memória, sem persistência, foi uma escolha deliberada para simplificar o desenvolvimento inicial e focar na funcionalidade básica do aplicativo.

3. Activity Extrato: A lista de transações foi implementada utilizando o template padrão do Android. Cada célula da lista exibia o tipo de transação, a descrição e o valor, facilitando a visualização das finanças do usuário.

### **Desenvolvimento do Aplicativo de Consulta à Harry Potter-API**

O segundo projeto envolveu a criação de um aplicativo que interagiu com a HP-API, demonstrando o uso de web services e endpoints. O desenvolvimento deste aplicativo também começou com a definição clara dos requisitos.

1. Activity Principal - Dashboard: A interface principal apresentava botões que permitiam ao usuário acessar diferentes funcionalidades, como listar personagens por ID, professores da escola e estudantes de cada casa.

2. Activity Listar um Personagem Específico: Esta funcionalidade permitiu que os usuários buscassem informações sobre um personagem específico, utilizando o endpoint da API. A implementação de chamadas assíncronas com corrotinas garantiu que a interface permanecesse responsiva enquanto os dados eram carregados.

3. Activity Listar os Professores da Escola: A consulta aos professores foi implementada de forma a exibir todos os nomes em um TextView.

4. Activity Listar os Estudantes de uma Casa: Esta funcionalidade foi enriquecida com a adição de Radio Buttons, permitindo que os usuários escolhessem uma casa específica para listar os estudantes. Essa interação melhorou a experiência do usuário e demonstrou a capacidade de criar interfaces dinâmicas.

A prototipagem e os testes de usabilidade foram elementos cruciais no desenvolvimento dos aplicativos. A criação de interfaces intuitivas e a realização de testes com usuários reais garantiram que as soluções fossem não apenas funcionais, mas também agradáveis e fáceis de usar. Essa abordagem centrada no usuário é um dos pilares do desenvolvimento ágil, pois assegura que o produto final atenda às expectativas e necessidades do público-alvo.

A integração deste projeto com outras disciplinas, como Design de UX no Desenvolvimento Ágil, Modelagem Ágil de Software e Introdução a Programação, foi fundamental para a formação de uma visão holística do desenvolvimento de software. Podendo aplicar conceitos de design centrado no usuário, técnicas de prototipagem e práticas de programação limpa, resultando em aplicativos que não apenas funcionam bem, mas também oferecem uma experiência de usuário satisfatória. No entanto, como se tratava de uma disciplina opcional, o projeto não foi concluído integralmente. Ainda assim, a experiência proporcionou uma base sólida para futuros aprimoramentos e aplicações

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

O projeto desenvolvido na disciplina de Infraestrutura para Desenvolvimento e Implantação de Software (DevOps) teve como foco a criação de um ambiente de integração contínua utilizando Docker, GitLab e Jenkins. O objetivo principal foi proporcionar uma compreensão prática das ferramentas e práticas que suportam o desenvolvimento ágil de software, dando ênfase à importância da automação e da colaboração entre equipes.

O projeto teve início com o download e execução da imagem `dfwandarti/gitlab_jenkins:3` no Docker. O container foi nomeado com a matrícula do aluno, garantindo uma identificação única conforme especificado. As portas 22 (SSH), 80 (HTTP), 443 (HTTPS) e 9091 (Jenkins) foram publicadas para permitir o acesso às interfaces do GitLab e Jenkins. A Figura 31 apresenta o comando utilizado e o container em execução, evidenciando a configuração inicial do ambiente.

Após a execução do container, foi realizado o login no GitLab utilizando o usuário `root` e a senha localizada no arquivo `/etc/gitlab/initial_root_password`, acessado diretamente dentro do container. Essa etapa foi essencial para compreender o gerenciamento de repositórios e permissões em um ambiente de controle de versão. A Figura 32 mostra a interface do GitLab acessada via navegador, confirmando o sucesso do login.

Em seguida, foi realizado um `commit` e, posteriormente, um `push` em um projeto hospedado no GitLab, praticando o fluxo de trabalho típico de desenvolvimento. A captura de tela do log do Git — exibindo o `commit` — está apresentada na Figura 33, sendo uma parte importante da documentação do processo e evidenciando a integração entre o GitLab e o Jenkins.

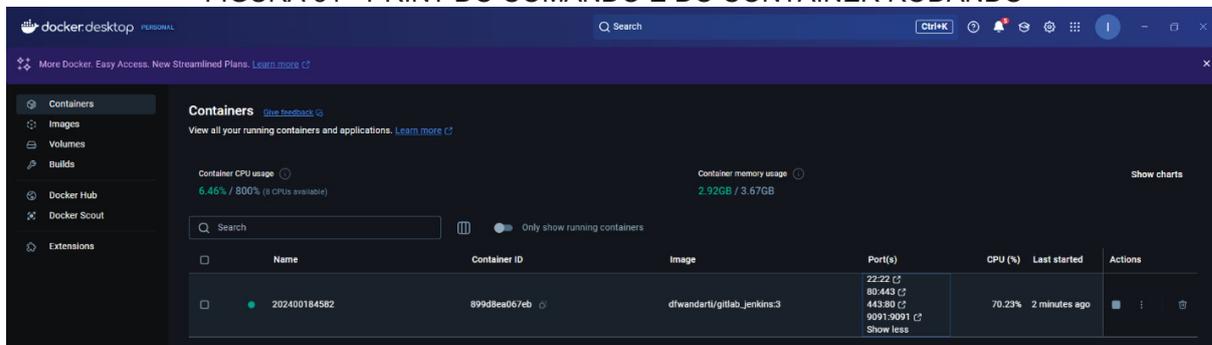
A importância deste projeto no contexto do desenvolvimento ágil de software é inegável. O DevOps promove uma cultura de colaboração entre as equipes de desenvolvimento e operações, permitindo que mudanças sejam implementadas de forma rápida e eficiente. A automação de processos, como a integração contínua e a entrega contínua (CI/CD), reduz o tempo de feedback e aumenta a qualidade do software, alinhando-se perfeitamente aos princípios ágeis.

Além disso, a utilização de containers Docker facilita a criação de ambientes consistentes e reproduzíveis, eliminando problemas relacionados à compatibilidade

de versões. Fundamentais em um ambiente ágil, onde a flexibilidade e a capacidade de adaptação são essenciais para o sucesso do projeto.

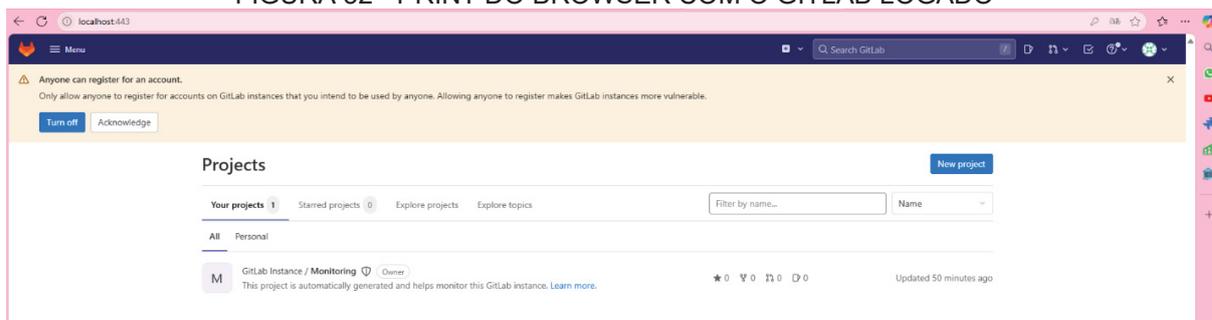
## 11.1 ARTEFATOS DO PROJETO

FIGURA 31 - PRINT DO COMANDO E DO CONTAINER RODANDO



FONTE: A autora (2024).

FIGURA 32 - PRINT DO BROWSER COM O GITLAB LOGADO



FONTE: A autora (2024).

FIGURA 33 - PRINT DO LOG COM O COMMIT

```

Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA> git clone http://localhost:443/gitlab-instance-204afbe4/Monitoring.git
Cloning into 'Monitoring'...
warning: You appear to have cloned an empty repository.
PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA> cd Monitoring
PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA\Monitoring> git switch -c main
Switched to a new branch 'main'

PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA\Monitoring> New-Item -Path . -Name 'README.md' -ItemType 'File'

Diretório: C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA\Monitoring

Mode                LastWriteTime         Length Name
----                -
-a-----          3/9/2025   3:32 AM             0 README.md

PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA\Monitoring> git add README.md
PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA\Monitoring> git commit -m "add README"
[main (root-commit) e13c68c] add README
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\Users\iasmi\OneDrive\Documents\Study\Pos\Auxiliares\INFRA\Monitoring> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://localhost:443/gitlab-instance-204afbe4/Monitoring.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

FONTE: A autora (2024).

## 12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

O projeto desenvolvido na disciplina de Testes Automatizados teve como objetivo a criação de um programa em Java que escreve informações em um arquivo de texto utilizando a plataforma Anotepad. O foco principal foi entender a importância dos testes automatizados no ciclo de desenvolvimento de software.

O projeto começou com a definição dos requisitos, que incluíam a criação de um código que gerasse um arquivo de texto com o título "Entrega trabalho TEST DAS 2024" e que incluísse os nomes e matrículas dos integrantes do grupo. A implementação foi realizada em Java, utilizando classes e métodos para manipulação de arquivos. O código foi estruturado de forma a garantir que, ao ser executado, o programa criasse um arquivo no Anotepad com as informações solicitadas.

O código apresentado na Figura 34 cria um arquivo chamado "*anotepad.txt*" e escreve o título e os nomes dos integrantes do grupo. A utilização das classes `BufferedWriter` e `FileWriter` permite uma manipulação eficiente do arquivo, garantindo que as informações sejam gravadas corretamente. O resultado da execução, ou seja, o conteúdo gerado no arquivo *anotepad.txt*, pode ser visualizado na Figura 35.

Os testes automatizados são fundamentais para garantir a qualidade do código, permitindo que os desenvolvedores identifiquem e corrijam erros rapidamente. Em um ambiente ágil, onde as iterações são rápidas e as mudanças são frequentes, a capacidade de realizar testes automatizados ajuda a manter a confiança na base de código e a acelerar o ciclo de desenvolvimento.

Além disso, a prática de escrever código que gera saídas específicas, como neste projeto, reforça a importância de validar as funcionalidades implementadas. A automação de testes não apenas economiza tempo, mas também reduz o risco de falhas em produção, alinhando-se aos princípios ágeis de entrega contínua e feedback rápido.

A integração deste projeto com outras disciplinas, como Introdução à Programação e Aspectos Ágeis de Programação, foi essencial para aplicar conceitos de design de software, práticas de codificação limpa e técnicas de teste, resultando em um entendimento mais profundo de como cada aspecto do desenvolvimento se interconecta.

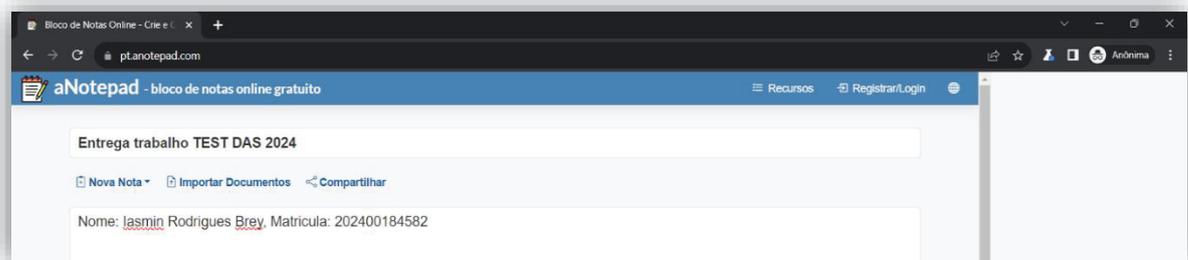
## 12.1 ARTEFATOS DO PROJETO

FIGURA 34 – CÓDIGO DE GERAÇÃO DE UM ANOTEPAD

```
Main.java | x
C: > Users > Iasmi > OneDrive > Documents > Study > Pos > Auxiliares > Test > src > main > java > com > example > Main.java > ...
1  package com.example;
2
3  import com.microsoft.playwright.*;
4
5  public class Main {
6      public static void main(String[] args) {
7          try (Playwright playwright = Playwright.create()) {
8              Browser browser = playwright.chromium().launch(new BrowserType.LaunchOptions().setHeadless(false));
9              BrowserContext context = browser.newContext();
10             Page page = context.newPage();
11
12             page.navigate("https://pt.anotepad.com", new Page.NavigateOptions().setTimeout(60000));
13             page.fill("#edit_title", "Entrega trabalho TEST DAS 2024");
14             page.fill("#edit_textarea", "Nome: Iasmin Rodrigues Brey, Matricula: 202400184582");
15             page.waitForSelector("button.save-note", new Page.WaitForSelectorOptions().setTimeout(60000));
16             page.click("button.save-note");
17
18
19             System.out.println(x:"Nota criada com sucesso!");
20         }
21     }
22 }
```

FONTE: A autora (2024).

FIGURA 35 – RESULTADO NO ANOTEPAD.COM



FONTE: A autora (2024).

## 13 CONCLUSÃO

Este parecer técnico oferece uma análise abrangente da integração das disciplinas estudadas ao longo do curso de Pós-Graduação em Desenvolvimento Ágil de Software, destacando como os princípios da metodologia ágil são aplicados, na prática, no desenvolvimento de software. Por meio de experiências em áreas como Métodos Ágeis, Modelagem Ágil, Gerenciamento de Projetos, Desenvolvimento Web, UX, Desenvolvimento Mobile, Infraestrutura DevOps e Testes Automatizados, ficou claro como cada disciplina contribui para a construção de um ecossistema coeso e colaborativo, essencial para a entrega contínua de valor. Essa integração é vital, pois permite que as equipes trabalhem em sinergia, aproveitando as habilidades e os conhecimentos de cada membro para alcançar resultados mais eficazes.

Os projetos desenvolvidos ao longo do curso evidenciaram a importância da adaptação e da iteração, características fundamentais do desenvolvimento ágil. Segundo Cohn (2004), uma das principais vantagens dessa abordagem em relação aos métodos tradicionais é a capacidade de responder com agilidade às mudanças nas necessidades dos usuários e do mercado. A colaboração entre equipes multidisciplinares, a ênfase na experiência do usuário e a automação de processos foram elementos-chave que possibilitaram a criação de soluções robustas e alinhadas às necessidades dos usuários. Além disso, a prática de testes automatizados e a implementação de CI/CD mostraram-se cruciais para garantir a qualidade e a confiabilidade das entregas, refletindo a essência do pensamento ágil. Humble e Farley (2010) argumentam que essas práticas contribuem significativamente para a eficiência do desenvolvimento e fortalecem a confiança nas entregas, permitindo que as equipes foquem em inovação e melhoria contínua.

Entretanto, a adoção prática do desenvolvimento ágil de software enfrenta desafios significativos. Um dos principais obstáculos é a resistência à mudança cultural dentro das organizações, onde equipes acostumadas a métodos tradicionais podem ter dificuldades em se adaptar a novas práticas e mentalidades. Martin (2008) aponta que essa resistência tende a se intensificar quando não há uma compreensão clara dos benefícios da abordagem ágil, o que pode resultar em uma aplicação superficial e limitada, incapaz de explorar todo o seu potencial. Além disso, a falta de treinamento adequado e a compreensão limitada dos princípios ágeis podem levar a

implementações que não atendem às expectativas, prejudicando a eficácia do processo (Behrens et al., 2021).

Outro desafio identificado é a gestão das expectativas dos stakeholders, que muitas vezes esperam resultados rápidos sem compreender a natureza iterativa do desenvolvimento ágil. Pichler e Hohmann (2016) enfatizam que manter uma comunicação constante e transparente com os envolvidos é essencial para garantir que suas necessidades sejam atendidas e que estejam cientes do progresso e das limitações do projeto.

Por fim, a integração de ferramentas e tecnologias que suportem práticas ágeis, como automação de testes e integração contínua, pode ser complexa e exigir investimentos significativos. Kim et al. (2016) destacam que a construção de uma infraestrutura adequada para essas práticas não apenas otimiza o processo de desenvolvimento, mas também favorece um ambiente de trabalho mais colaborativo e voltado à inovação.

Em suma, a experiência adquirida ao longo do curso e a reflexão sobre os desafios da metodologia ágil proporcionam uma base sólida para enfrentar os desafios do mercado de trabalho. A capacidade de adaptação, a colaboração e o foco em valor são habilidades essenciais que, quando bem aplicadas, podem transformar a forma como o software é desenvolvido e entregue, promovendo um ambiente de inovação e excelência. Adotar uma mentalidade ágil não é apenas uma vantagem competitiva, mas uma necessidade em um mundo em constante mudança, onde a habilidade de se adaptar rapidamente às novas demandas é crucial para o sucesso.

## 14 REFERÊNCIAS

- BECK, Kent et al. Manifesto Ágil: princípios por trás do manifesto. **Agile Manifesto**, 2001. Disponível em: <<https://agilemanifesto.org/iso/ptbr/principles.html>>. Acesso em: 9 ago. 2025.
- BEHRENS, Andy; OFORI, Martinson; NOTEBOOM, Cherie; BISHOP, Dave. A systematic literature review: how agile is agile project management? **Issues in Information Systems**, v. 22, n. 3, 2021, p. 298–316. Disponível em: <[https://doi.org/10.48009/3\\_iis\\_2021\\_298-316](https://doi.org/10.48009/3_iis_2021_298-316)>. Acesso em: 9 ago. 2025.
- CARMONA, Rodolfo. Os 10 desafios mais comuns no desenvolvimento de softwares. **WK Technology**, 2023. Disponível em: <<https://wktechnology.com.br/desafios-no-desenvolvimento-de-softwares/>>. Acesso em: 9 ago. 2025.
- COHN, Mike. *User Stories Applied: For Agile Software Development*. **Addison-Wesley**, Boston, 2004.
- CRISPIN, Lisa; GREGORY, Janet. *Agile Testing: A Practical Guide for Testers and Agile Teams*. **Addison-Wesley**, Boston, 2009.
- HIGHSMITH, James A. *Agile Software Development Ecosystems*. **Addison-Wesley**, Boston, 2002.
- HUMBLE, Jez; FARLEY, David. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. **Bookman**, Porto Alegre, 2014.
- JEFFRIES, Ron; ANDERSON, Ann; HENDRICKSON, Chet. *Extreme Programming Installed*. **Addison-Wesley**, Boston, 2000.
- KIM, Gene; BEHR, Kevin; SPAFFORD, George; HUMBLE, Jez. *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. **IT Revolution Press**, Portland, 2016.
- MARTIN, Robert C. *Clean Code: A Handbook of Agile Software Craftsmanship*. **Prentice Hall**, Boston, 2008.
- NORMAN, Donald A. *The Design of Everyday Things*. **Rocco**, Rio de Janeiro, 2013.
- PICHLER, Roman; HOHMANN, Luke. *Agile Product Management with Scrum: Creating Products that Customers Love*. **Addison-Wesley**, Boston, 2016.
- SUTHERLAND, Jeff. *Scrum: The Art of Doing Twice the Work in Half the Time*. **Leya**, Rio de Janeiro, 2014.