

UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME ZANINI MOREIRA

MEMORIAL DE PROJETOS: INTEGRAÇÃO DE TECNOLOGIAS E DESAFIOS NA
APLICAÇÃO PRÁTICA DO DESENVOLVIMENTO ÁGIL DE SOFTWARE

CURITIBA

2025

GUILHERME ZANINI MOREIRA

MEMORIAL DE PROJETOS: INTEGRAÇÃO DE TECNOLOGIAS E DESAFIOS NA
APLICAÇÃO PRÁTICA DO DESENVOLVIMENTO ÁGIL DE SOFTWARE

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

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 **GUILHERME ZANINI MOREIRA**, intitulada: **MEMORIAL DE PROJETOS: INTEGRAÇÃO DE TECNOLOGIAS E DESAFIOS NA APLICAÇÃO PRÁTICA DO DESENVOLVIMENTO ÁGIL DE SOFTWARE**, 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, 26 de Agosto de 2025.



JAIME WOJCIECHOWSKI

Presidente da Banca Examinadora



RAFAELA MANTOVANI FONTANA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Este memorial de projetos é um registro dos trabalhos desenvolvidos em um curso de pós-graduação em Desenvolvimento Ágil de Software. Ele abrange desde conceitos básicos de programação e modelagem de sistemas até a aplicação de práticas avançadas de desenvolvimento web, banco de dados, usabilidade e automação de testes. Os projetos demonstram a integração progressiva entre a teoria e a prática, utilizando tecnologias como Angular, Spring Boot, PostgreSQL, MySQL e Playwright. O conjunto dessas atividades evidencia uma evolução técnica sólida, adaptabilidade a diferentes contextos e o uso de boas práticas de engenharia de software. Em última análise, o documento apresenta um portfólio coeso, que está alinhado com as exigências contemporâneas do mercado de desenvolvimento.

Palavras-chave: desenvolvimento ágil, modelagem de software, banco de dados, usabilidade

ABSTRACT

This document is a record of the work carried out in a graduate program in Agile Software Development. It covers everything from fundamental concepts in programming and systems modeling to the application of advanced practices in web development, databases, usability, and test automation. The projects demonstrate the progressive integration of theory and practice, using technologies such as Angular, Spring Boot, PostgreSQL, MySQL, and Playwright. Taken together, these activities showcase solid technical growth, adaptability to different contexts, and the use of software engineering best practices. Ultimately, the document presents a cohesive portfolio that aligns with the contemporary demands of the development market.

Keywords: agile development, software modeling, database, usability

SUMÁRIO

1.PARECER TÉCNICO.....	7
2.DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....	10
2.1.ARTEFATOS DO PROJETO	11
3.DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2	15
3.1.ARTEFATOS DO PROJETO.....	16
4.DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2	20
4.1.ARTEFATOS DO PROJETO.....	21
5.DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO	22
5.1.ARTEFATOS DO PROJETO.....	23
6.DISCIPLINA: BD – BANCO DE DADOS	24
6.1.ARTEFATOS DO PROJETO.....	25
7.DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO	27
7.1.ARTEFATOS DO PROJETO.....	28
8.DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2.....	29
8.1.ARTEFATOS DO PROJETO.....	30
9.DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE	32
9.1.ARTEFATOS DO PROJETO.....	33
10.DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2	36
10.1.ARTEFATOS DO PROJETO	37
11.DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)	41
11.1.ARTEFATOS DO PROJETO	42
12.DISCIPLINA: TEST – TESTES AUTOMATIZADOS	44
12.1.ARTEFATOS DO PROJETO.....	45
13.CONCLUSÃO	46
14.REFERÊNCIAS	47

1. PARECER TÉCNICO

O conjunto de trabalhos apresentados ao longo das disciplinas demonstra uma evolução consistente e progressiva na aplicação de conceitos, ferramentas e metodologias de desenvolvimento ágil de software (Highsmith, 2009). Desde atividades iniciais de menor complexidade até a entrega de sistemas completos, as implementações refletiram a integração entre teoria e prática, envolvendo modelagem, desenvolvimento, testes, gestão de projetos e usabilidade.

Sob o ponto de vista tecnológico, há um claro domínio de múltiplas linguagens e frameworks, incluindo Angular para construção de interfaces modernas e responsivas, Spring Boot para desenvolvimento de back-end robusto, PostgreSQL e MySQL para gestão de dados (Coronel; Morris, 2018), além de ferramentas especializadas como Playwright para automação de testes e ambientes de simulação como o Kanban Board Game (Anderson; Carmichael, 2016).

As disciplinas de Desenvolvimento Web 1 e 2 ilustram de forma evidente a progressão técnica. Inicialmente, foi desenvolvido um sistema Angular com persistência em Local Storage, contemplando operações CRUD para Aluno e Curso. Na sequência, o projeto foi ampliado com a adição de um módulo de Matrículas, a substituição do armazenamento local por um banco de dados PostgreSQL e a implementação de um back-end em Spring Boot, com integração via API REST. Essa transição demonstrou compreensão sobre arquitetura em camadas, comunicação entre sistemas e práticas modernas de integração, apoiadas por integração/entrega contínua para reduzir risco e aumentar previsibilidade (Humble; Farley, 2010).

Na área de modelagem, trabalhos como Modelagem Ágil de Software I e II e MADS – Métodos Ágeis de Desenvolvimento de Software evidenciam a capacidade de representar requisitos e funcionalidades de forma visual e estruturada. Foram elaborados Diagramas de Caso de Uso, Diagramas de Classes, Diagramas de Sequência, todos alinhados a práticas ágeis que privilegiam clareza, objetividade e adaptação rápida (Ambler; Lines, 2012). Essa abordagem permitiu que as modelagens fossem não apenas documentos técnicos, mas também ferramentas eficazes de comunicação dentro de um contexto colaborativo.

O trabalho de Banco de Dados consolidou habilidades na modelagem conceitual e lógica, bem como na implementação prática de estruturas relacionais. Foram produzidos scripts SQL completos, com definição de chaves primárias e estrangeiras, regras de integridade e povoamento inicial para teste de relacionamentos do tipo 1x1, 1xN e NxN. Essa prática demonstrou atenção à normalização, à integridade referencial e à performance, de acordo com diretrizes consolidadas de projeto de bancos de dados (Coronel; Morris, 2018).

Nas disciplinas Gerenciamento Ágil de Projetos I e II, foi possível observar a aplicação concreta de técnicas de planejamento e acompanhamento de projetos. O plano de release detalhado e a execução de um ciclo de 35 dias no Kanban Board Game permitiram experimentar, de forma prática, conceitos como limitação de trabalho em progresso, análise de gargalos e uso de métricas visuais, como o Cumulative Flow Diagram, para a melhoria contínua do fluxo de trabalho (Anderson; Carmichael, 2016; Highsmith, 2009).

O trabalho de UX mostrou a preocupação em alinhar requisitos funcionais e experiência do usuário. A concepção de um produto digital, a elaboração de telas, a justificativa de escolhas visuais e tipográficas e a coleta de feedback junto a um possível usuário refletem a aplicação de princípios de design centrado no ser humano, garantindo que a solução proposta tenha aderência e usabilidade no contexto de uso (Norman, 2013).

Na disciplina Aspectos Ágeis de Programação, a refatoração de um algoritmo utilizando princípios de clean code evidenciou atenção à legibilidade, manutenibilidade e clareza de intenção no código, fatores essenciais para a longevidade e evolução de um sistema (Martin, 2008). Já no projeto para a disciplina de testes automatizados, a automação de interações em um editor online com Playwright mostrou a capacidade de integrar desenvolvimento web com automação de testes, agregando valor ao processo de verificação de requisitos e execução de fluxos automatizados por meio de práticas associadas ao TDD e ao XP (Beck, 2003; Beck; Andres, 2004).

No conjunto, os trabalhos indicam maturidade técnica, domínio de práticas ágeis e habilidade de aplicar conhecimentos de forma integrada, do planejamento e modelagem até a entrega e validação final. A experiência adquirida em múltiplos contextos e tecnologias, aliada à abordagem iterativa e adaptativa, reforça a

prontidão para atuar em projetos reais que demandem colaboração, flexibilidade e foco em resultados, em consonância com princípios do ágil, modelagem, gestão visual do fluxo de trabalho, engenharia orientada a testes e entrega contínua (Highsmith, 2009; Ambler; Lines, 2012; Anderson; Carmichael, 2016; Humble; Farley, 2010; Martin, 2008; Beck, 2003; Beck; Andres, 2004; Coronel; Morris, 2018; Norman, 2013).

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

O trabalho teve como objetivo a elaboração de um mapa mental abrangendo os conceitos principais relacionados a processos de software, princípios ágeis e métodos ágeis de desenvolvimento. A proposta visou sintetizar visualmente os tópicos essenciais, proporcionando uma visão clara e integrada das práticas e fundamentos da área.

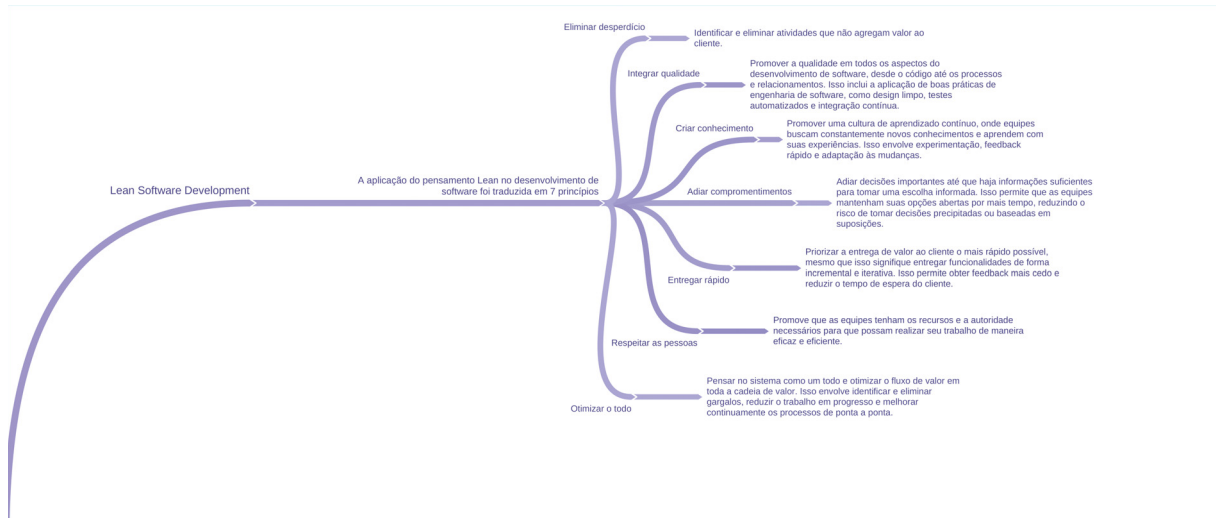
O mapa mental abordou, no mínimo, os seguintes itens: Processo de Software, Modelos Tradicionais, Manifesto Ágil, Princípios Ágeis, Lean Software Development, Scrum, Extreme Programming (XP), Kanban e Entrega Contínua de Software. Cada um desses tópicos foi detalhado com dois ou mais subtópicos, contendo conceitos e explicações sobre suas etapas, atividades, princípios e processos específicos.

A elaboração do mapa exigiu o entendimento das diferenças entre modelos tradicionais e ágeis, bem como da aplicação prática das metodologias no contexto do desenvolvimento moderno. Além disso, a representação visual favoreceu a organização hierárquica das ideias, reforçando a compreensão dos conceitos de forma rápida e acessível.

Este trabalho está alinhado ao desenvolvimento ágil por promover a visualização e comunicação eficaz entre membros de uma equipe e partes interessadas. Como afirma Highsmith (2009, p. 23), “a agilidade não é apenas sobre processos, mas sobre a capacidade de responder rapidamente a mudanças e manter o foco no valor entregue ao cliente”

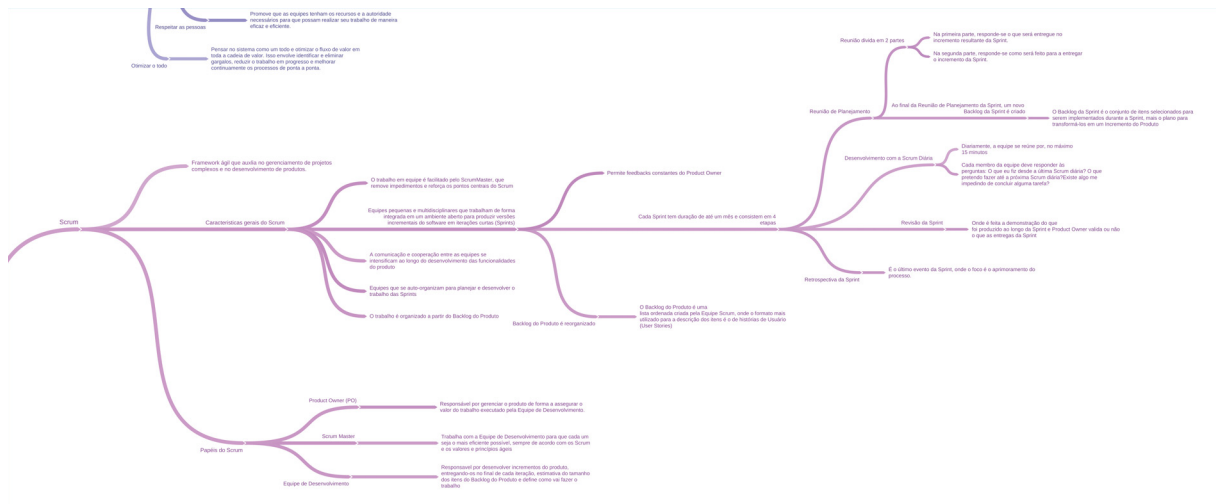
2.1.ARTEFATOS DO PROJETO

FIGURA 1 - Mapa mental (Lean Software Development)



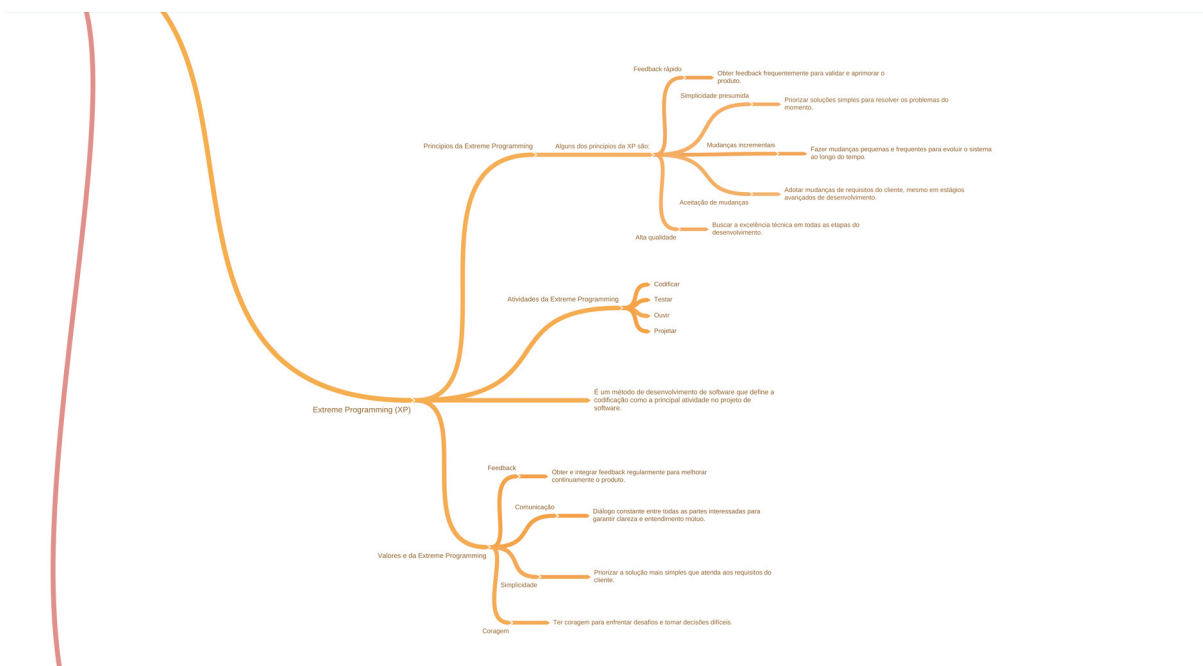
Fonte: O autor (2025)

FIGURA 2 - Mapa mental (Scrum)



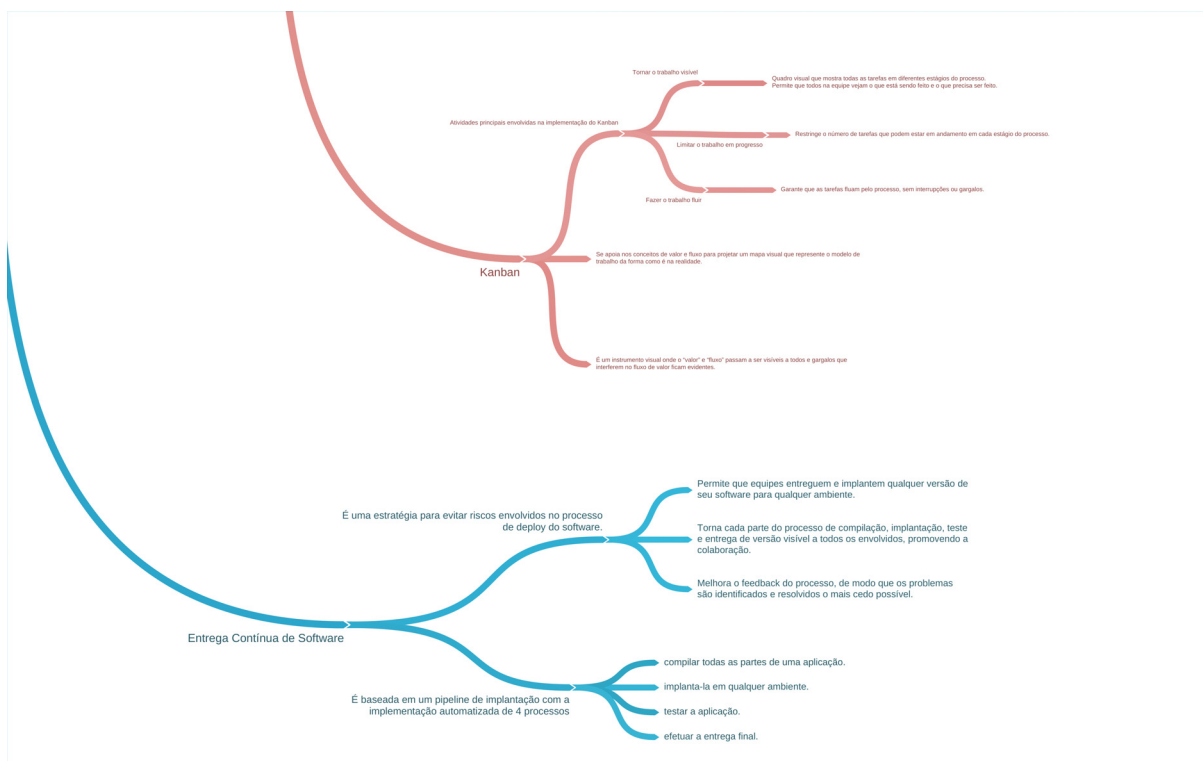
Fonte: O autor (2025)

FIGURA 3 - Mapa mental (Extreme Programming - XP)



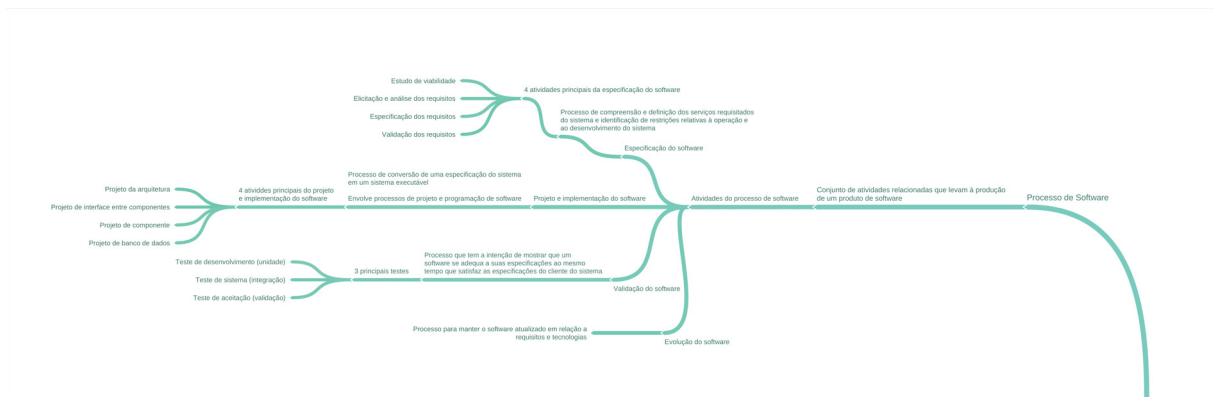
Fonte: O autor (2025)

FIGURA 4 - Mapa mental (Kanban e Entrega Contínua de Software)



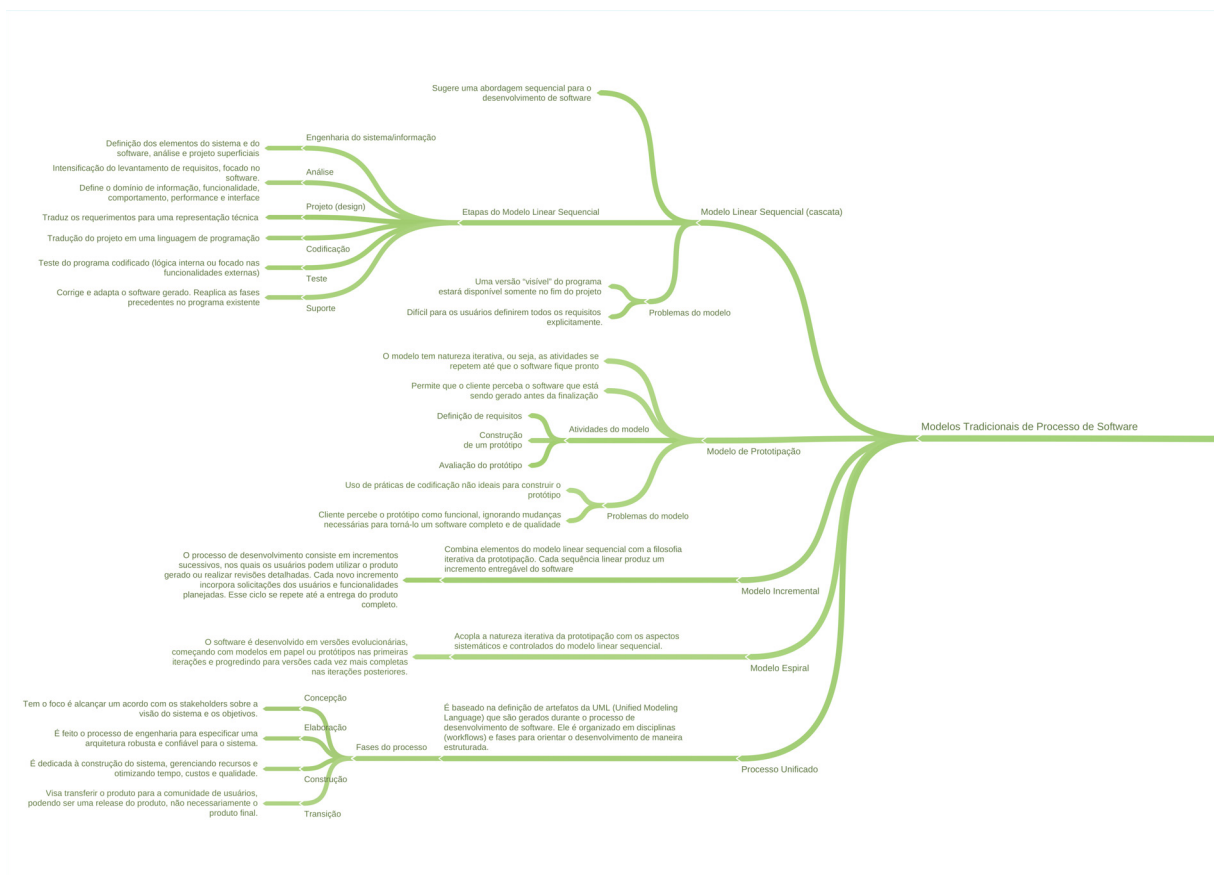
Fonte: O autor (2025)

FIGURA 5 - Mapa mental (processo de software)



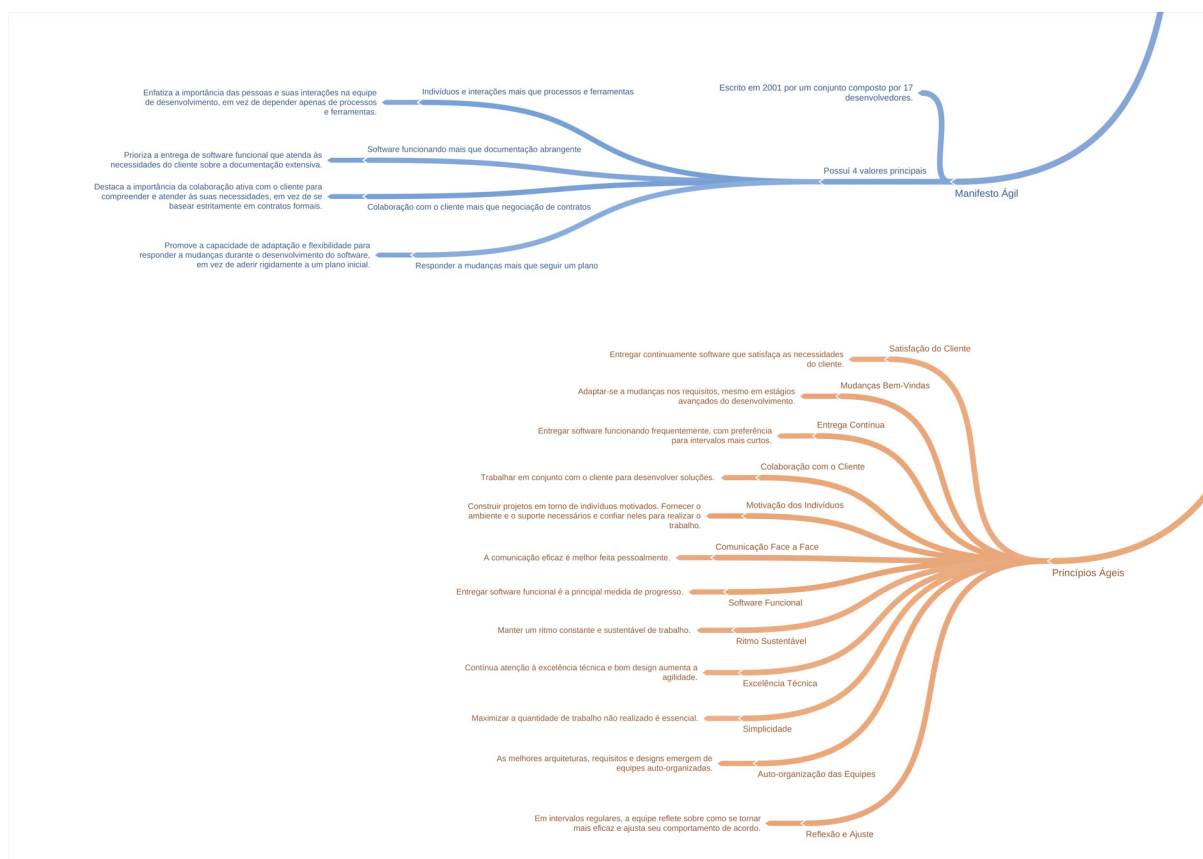
Fonte: O autor (2025)

FIGURA 6 - Mapa mental (modelos tradicionais de processo de software)



Fonte: O autor (2025)

FIGURA 7 - Mapa mental (manifesto ágil e princípios ágeis)



Fonte: O autor (2025)

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

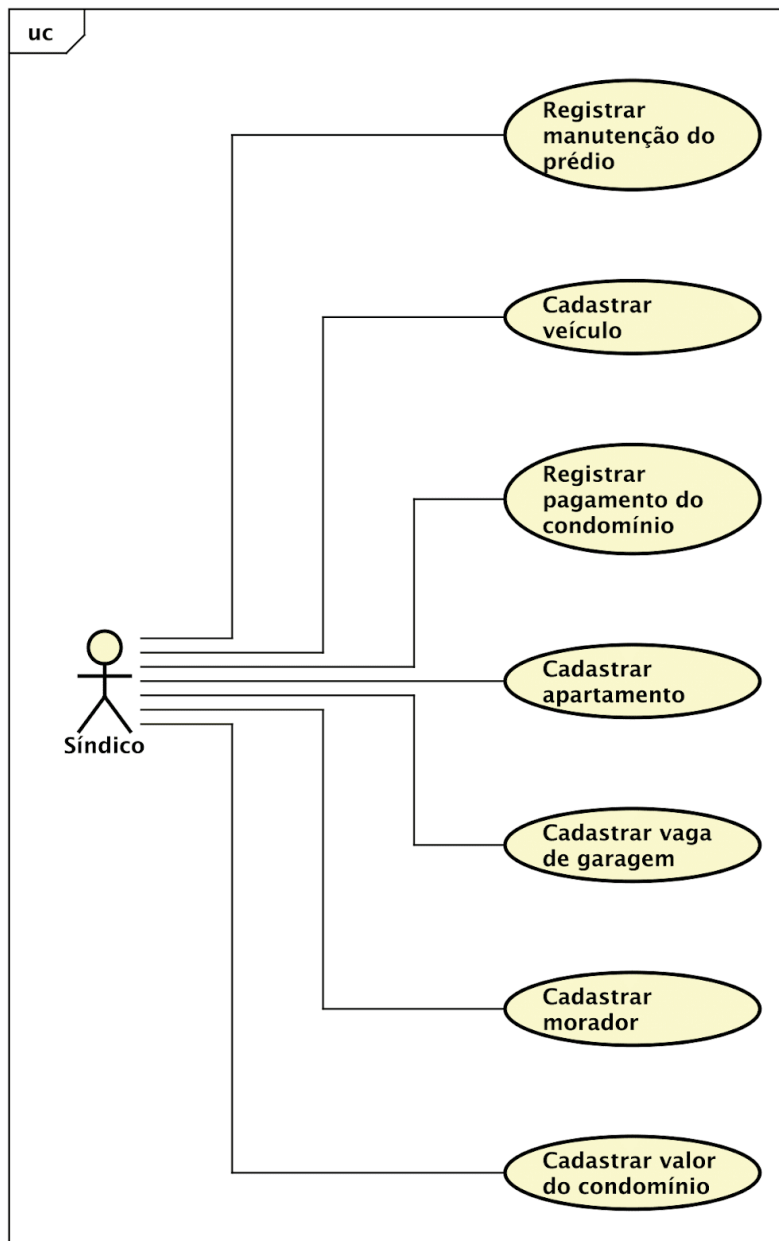
O trabalho teve como objetivo modelar um Sistema de Gestão de Condomínio, integrando as visões funcional e estrutural do sistema de forma alinhada a práticas ágeis. Na primeira etapa, referente à Modelagem Ágil de Software I – Visão Funcional, foram elaborados os Diagramas de Caso de Uso nos Níveis 1 e 2, representando de forma visual as interações entre os atores e as funcionalidades principais. Esses diagramas detalharam o fluxo de uso do sistema e facilitaram a compreensão do escopo. Também foram criadas Histórias de Usuário completas, descrevendo os objetivos, as necessidades e os critérios de aceitação para funcionalidades como cadastro de apartamentos e moradores, registro de pagamentos e controle de manutenções.

Na segunda etapa, correspondente à Modelagem Ágil de Software II – Visão Estrutural, a modelagem foi aprofundada com a criação do Diagrama de Classes, contendo atributos e relacionamentos que representam a estrutura lógica do sistema. Foi definido também o modelo de tabelas do banco de dados, garantindo coerência entre a modelagem e a persistência das informações. Além disso, foram elaborados Diagramas de Sequência para todas as Histórias de Usuário, demonstrando de forma clara a interação entre objetos e atores ao longo do tempo, desde a entrada de dados até o processamento e resposta do sistema.

A união dessas duas etapas permitiu que o sistema fosse compreendido tanto no aspecto de o que precisa ser feito quanto de como será estruturado e executado. Essa abordagem está alinhada ao conceito de modelagem ágil, que, segundo Ambler e Lines (2012, p. 21), “é uma prática iterativa e colaborativa que visa criar modelos suficientemente bons para guiar o desenvolvimento, evitando o excesso de documentação e favorecendo a adaptação”.

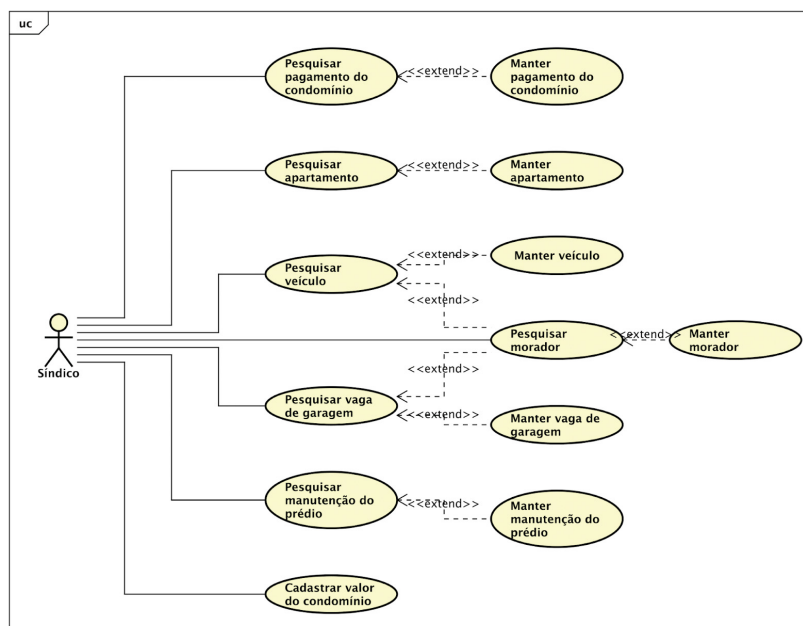
3.1.ARTEFATOS DO PROJETO

FIGURA 8 - Diagrama de Caso de Uso Nível 1



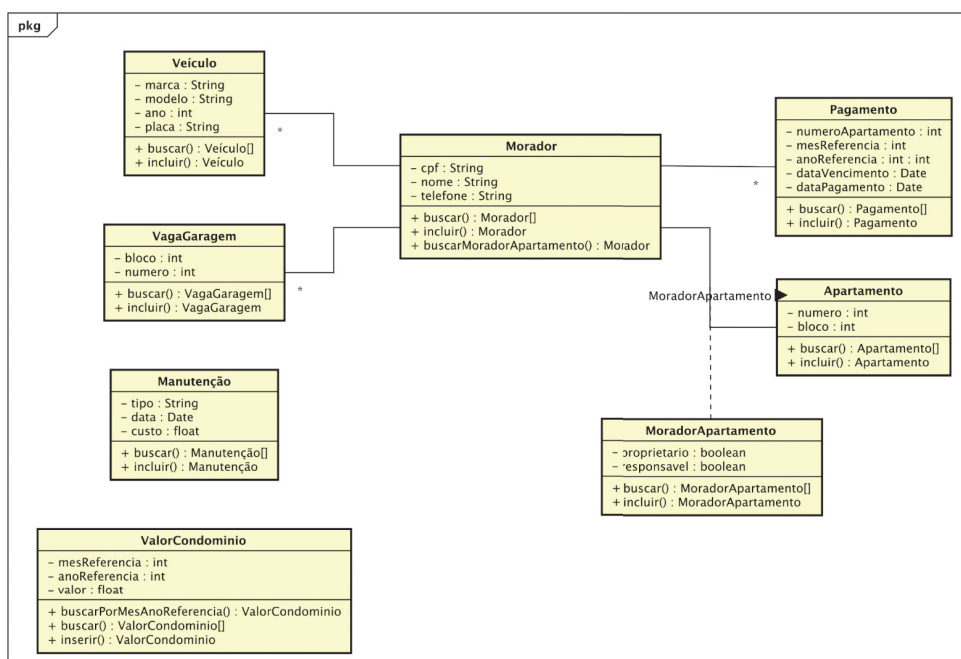
Fonte: O autor (2025)

FIGURA 9 - Diagrama de Caso de Uso Nível 2



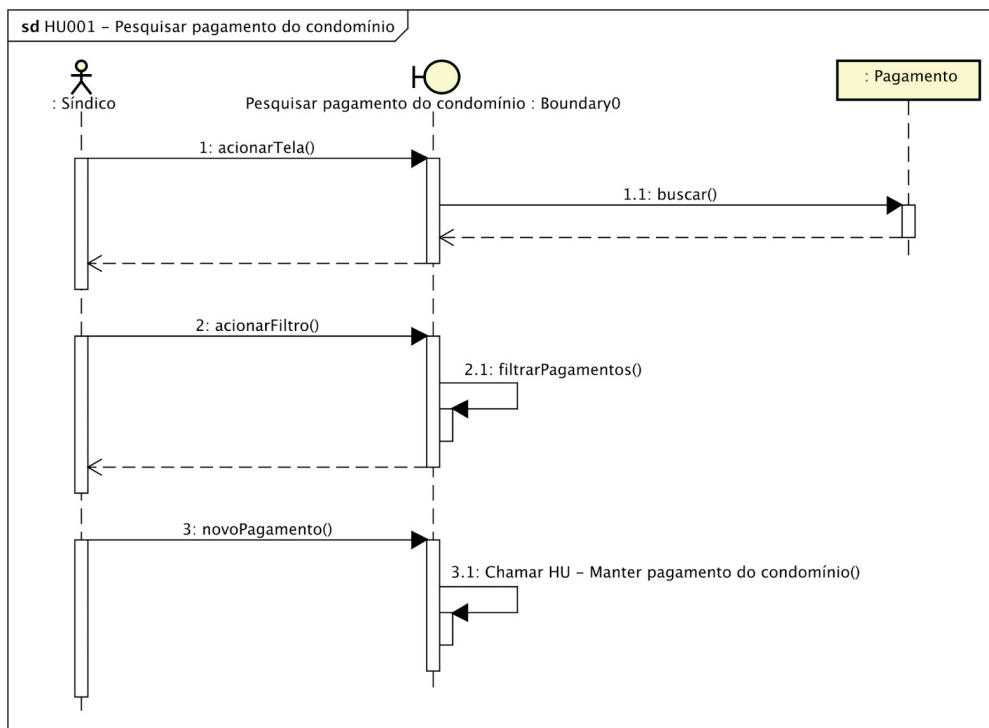
Fonte: O autor (2025)

FIGURA 10 - Diagrama de Classes



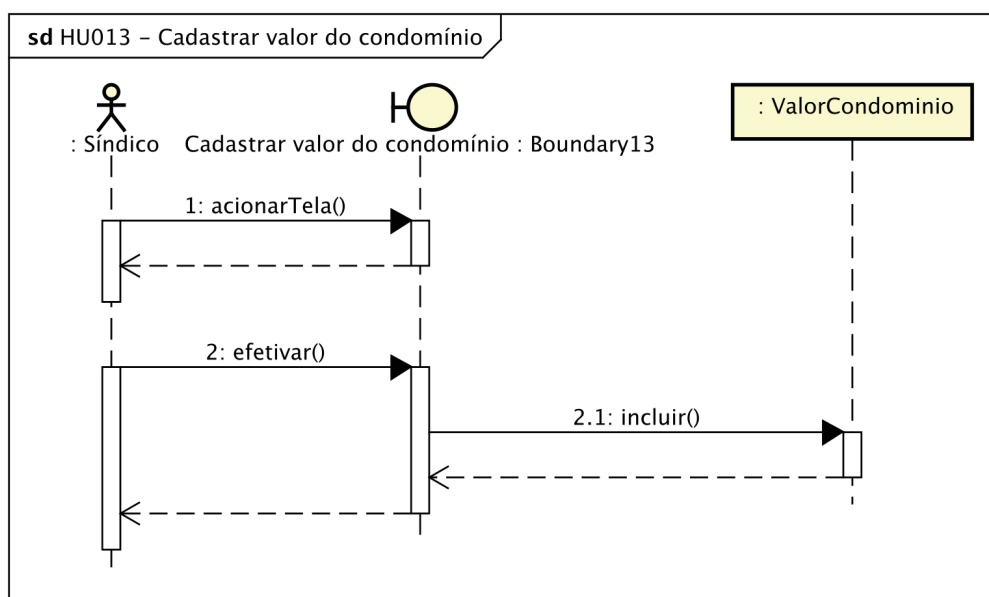
Fonte: O autor (2025)

FIGURA 11 - História de usuário



Fonte: O autor (2025)

FIGURA 12 - História de usuário



Fonte: O autor (2025)

FIGURA 13 - Desenho da tela “Cadastrar manutenção”

Cadastrar manutenção

Tipo de manutenção 

Data 

R\$

Salvar

Fonte: O autor (2025)

FIGURA 14 - Desenho da tela “Lista de veículos”

Lista de veículos Novo veículo

Morador	Marca	Modelo	Placa	Ano	
Sofia Pereira	Audi	A4	ABC1D23	2019	 
Ana Silva	BMW	M140i	XYZ4E56	2020	 
João Santos	Volkswagen	Golf	LMN7F89	2017	 
Maria Oliveira	Jeep	Commander	QRS0G12	2022	 
Pedro Almeida	Chevrolet	Onix	UVW3H45	2023	 
Laura Martins	Ford	Fusion	IJK6I78	2017	 
Beatriz Sousa	Fiat	500	STU9J01	2016	 

Fonte: O autor (2025)

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

O conjunto de trabalhos teve como objetivo aplicar, de forma prática, conceitos e técnicas de gerenciamento ágil de projetos, desde o planejamento inicial até a execução de um ciclo completo de desenvolvimento simulado.

Na disciplina Gerenciamento Ágil de Projetos I, foi elaborado um plano de release para o desenvolvimento de um software, considerando que todo o trabalho seria executado por uma única pessoa. O plano incluiu o cálculo da velocidade de desenvolvimento, com base na equivalência de 1 ponto para 8 horas de trabalho, e a distribuição das histórias de usuário em sprints com datas definidas de início e fim. As histórias foram escritas no formato *Sendo/Quero/Para*, acompanhadas de suas estimativas, garantindo clareza nos objetivos e viabilidade no planejamento.

Já na disciplina Gerenciamento Ágil de Projetos II, o foco foi a execução prática de um ciclo de 35 dias no ambiente simulado *kanbanboardgame.com*, aplicando princípios e aprendizados da metodologia Kanban. A execução incluiu a captura de métricas e evidências visuais em três momentos: dia 15 de execução, tela final com o total de receita obtida e o *Cumulative Flow Diagram* (CFD) gerado pela ferramenta. Essa prática permitiu observar, na prática, a importância do fluxo contínuo, da limitação de trabalho em progresso e da análise de gargalos no processo.

A integração das duas disciplinas proporcionou uma visão completa, unindo planejamento estruturado e execução adaptativa, reforçando a importância da flexibilidade e da transparência no acompanhamento do projeto. Como afirmam Anderson e Carmichael (2016, p. 12), “o Kanban promove melhorias evolutivas no processo, tornando visível o trabalho e possibilitando decisões baseadas em dados”.

4.1.ARTEFATOS DO PROJETO

FIGURA 15 - Plano de Release

Gerenciamento Ágil de Projetos I
Profa. Dra. Rafaela Mantovani Fontana
Template para o Plano de Release

Cálculo da Velocidade:			
Horas disponíveis por dia:	5 horas	Tamanho da Sprint:	2 semanas (10 dias úteis)
Horas disponíveis por Sprint:	50 horas	Velocidade:	6 pontos

Plano de Release:			
Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint 4
Data Início: 06/05/2024	Data Início: 20/05/2024	Data Início: 03/06/2024	Data Início: 17/06/2024
Data Fim: 17/05/2024	Data Fim: 31/05/2024	Data Fim: 14/06/2024	Data Fim: 28/06/2024
Cadastro de Serviço para Prestadores SENDO o prestador de serviço QUERO preencher um formulário com informações sobre meu serviço PARA disponibilizar meus serviços para os usuários interessados ESTIMATIVA (2 pontos)	Avaliação de Serviço SENDO o usuário QUERO poder atribuir uma classificação e escrever um comentário sobre a prestação de serviço PARA ajudar outros usuários tomarem decisões e fornecer feedback ao prestador de serviço ESTIMATIVA (1 ponto)	Gerenciamento de Favoritos SENDO o usuário QUERO poder adicionar e remover serviços da minha lista de favoritos PARA facilitar a busca por serviços que já me interessaram anteriormente ESTIMATIVA (1 ponto)	Compartilhamento de Fotos SENDO o usuário. QUERO poder fazer upload de fotos na avaliação do serviço PARA ajudar outros usuários a tomar decisões ESTIMATIVA (1 ponto)
Cadastro de Usuário SENDO um novo usuário QUERO poder me cadastrar no aplicativo PARA poder acessar os serviços oferecidos na plataforma ESTIMATIVA (1 ponto)	Notificações de Serviço SENDO o usuário QUERO ser notificado sobre confirmação de agendamento, atualizações de status e lembretes de serviço PARA ficar informado e não perder nenhum compromisso ESTIMATIVA (1 ponto)	Agendamento Recorrente SENDO o usuário QUERO poder selecionar a frequência desejada e as datas de início e término para agendar automaticamente serviços repetitivos PARA facilitar o agendamento de serviços que necessitam de periodicidade regular ESTIMATIVA (2 pontos)	Histórico de Serviços SENDO o usuário QUERO poder acessar informações como datas, prestadores, serviços contratados, avaliações e pagamentos relacionados a cada serviço PARA revisar meu histórico de serviços anteriores e facilitar a reagendamento ou referência futura ESTIMATIVA (1 ponto)

Gerenciamento Ágil de Projetos I
Profa. Dra. Rafaela Mantovani Fontana
Template para o Plano de Release

Pesquisa por Serviços SENDO o usuário QUERO poder filtrar os serviços por categoria, localização e preço PARA encontrar o serviços que melhor atende às minhas necessidades ESTIMATIVA (2 pontos)	Filtragem Avançada de Serviços SENDO o usuário QUERO poder filtrar por critérios adicionais, como avaliação do prestador, distância e disponibilidade de horário PARA encontrar o serviço mais adequado às minhas preferências e necessidades específicas ESTIMATIVA (2 pontos)	Sincronização de Calendário SENDO o usuário QUERO que as datas e horários dos serviços agendados sejam adicionados ao meu calendário do smartphone ou computador PARA ter uma visão integrada de todos os meus compromissos e serviços agendados ESTIMATIVA (2 pontos)	Chat entre Usuário e Prestador de Serviço SENDO o usuário ou prestador de serviço QUERO poder me comunicar diretamente com o outro através de um sistema de chat integrado PARA discutir detalhes do serviço, esclarecer dúvidas e resolver problemas ESTIMATIVA (2 pontos)
Visualização do perfil dos prestadores de serviço SENDO o usuário QUERO ver as informações como avaliações, especialidades, preço e disponibilidade PARA tomar decisão sobre quem contratar ESTIMATIVA (1 ponto)	Integração de Pagamento SENDO o usuário QUERO poder inserir os dados do meu cartão de crédito e confirmar o pagamento de forma segura PARA facilitar o processo de pagamento ESTIMATIVA (2 pontos)	Notificações Personalizadas SENDO o usuário QUERO poder escolher quais tipos de notificações desejo receber e como desejo recebê-las (notificação push, e-mail ou SMS) PARA receber apenas notificações importantes e relevantes de acordo com as preferências ESTIMATIVA (1 ponto)	Relatórios de Desempenho SENDO o prestador de serviço QUERO poder visualizar métricas como número de agendamentos, avaliações dos clientes, receita gerada e taxa de conversão PARA avaliar o desempenho dos meus serviços, identificar áreas de melhoria e tomar decisões ESTIMATIVA (2 pontos)

Fonte: O autor (2025)

ESTIMATIVA (1 ponto)	ESTIMATIVA (2 pontos)	ESTIMATIVA (1 ponto)	ESTIMATIVA (2 pontos)
Desenvolvimento de funcionalidades básicas do sistema.	Implementação de funcionalidades avançadas e testes de integração.	Desenvolvimento de funcionalidades básicas do sistema.	Implementação de funcionalidades avançadas e testes de integração.

5. DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

O trabalho teve como objetivo a implementação do backend de um sistema bancário simplificado para controle de cadastro de clientes, contas correntes e contas de investimento. A especificação inclui um diagrama de classes para desenvolvimento, um projeto no NetBeans contendo 42 casos de teste unitários escritos em JUnit e um script DDL para criação das tabelas no MySQL.

A tarefa consistiu em implementar as classes do sistema de forma que pelo menos 40 dos 42 testes ficassem simultaneamente no estado “verde”, assegurando o funcionamento correto das funcionalidades previstas. A solução exigiu a utilização de um banco de dados MySQL para a persistência, seguindo a modelagem fornecida, e a integração com a aplicação desenvolvida no NetBeans.

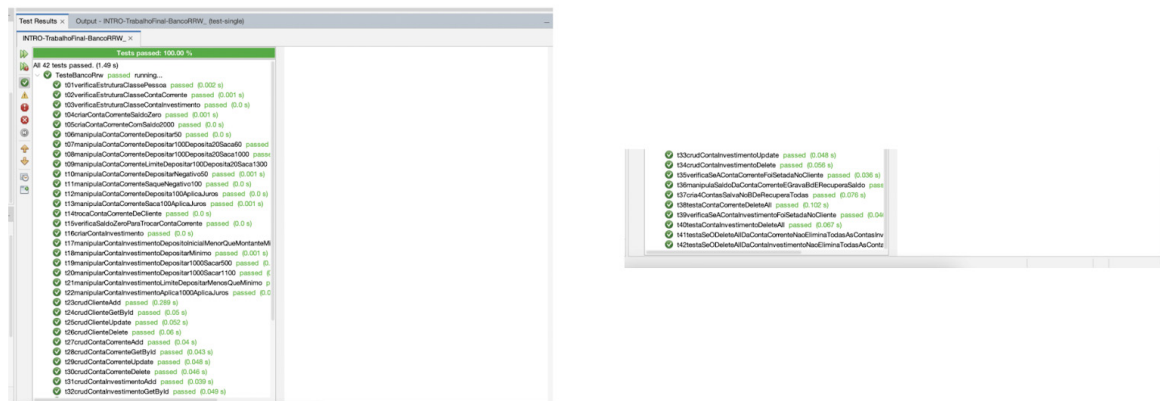
O processo de desenvolvimento envolve compreender o diagrama de classes, aplicar conceitos de programação orientada a objetos, implementar operações de cadastro e movimentação bancária, e garantir a integridade dos dados. A validação do sistema foi feita por meio dos testes unitários, que permitiram identificar falhas e confirmar a conformidade com os requisitos.

Esse projeto reforçou a importância de unir práticas de desenvolvimento estruturado a estratégias de teste automatizado, alinhando-se ao pensamento ágil. Como afirma Beck (2003, p. 55), “testes automatizados permitem modificar o código com confiança, sabendo que erros serão detectados rapidamente”.

5.1.ARTEFATOS DO PROJETO

FIGURA 16 - Resultado dos testes automatizados

Introdução à Programação
Universidade Federal do Paraná
Especialização em Desenvolvimento Ágil de Software
Guilherme Zanini Moreira



Fonte: O autor (2025)

6. DISCIPLINA: BD – BANCO DE DADOS

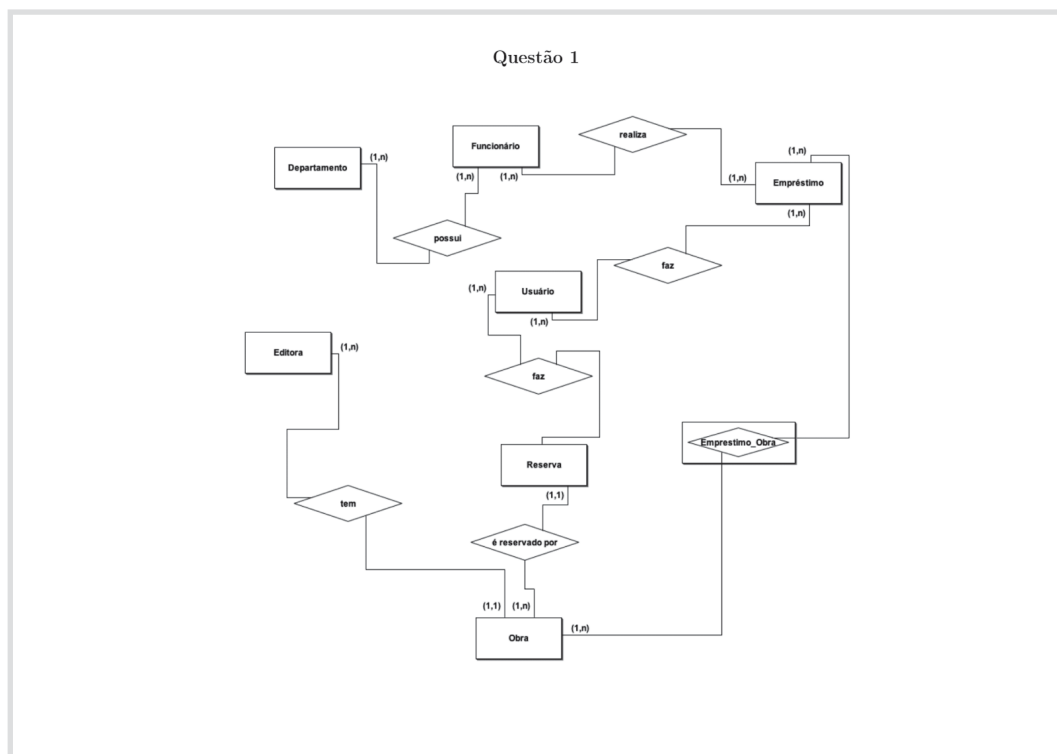
O trabalho teve como objetivo aplicar conceitos de modelagem e implementação de bancos de dados relacionais, dividindo-se em duas partes principais. Na primeira, foi realizada a modelagem de um sistema de controle de biblioteca, contemplando o Modelo Entidade-Relacionamento Conceitual, com entidades e relacionamentos, e o Modelo Lógico, no qual as entidades foram transformadas em tabelas com definição de campos, chaves primárias, chaves estrangeiras e relacionamentos. O modelo considerou requisitos como o cadastro de obras, editoras, usuários, funcionários, departamentos, empréstimos e reservas, evidenciando regras de negócio como vínculos obrigatórios e restrições de integridade.

Na segunda parte, foi proposta a criação de um modelo lógico e implementação em SQL para um tema escolhido, exigindo a presença de relacionamentos 1x1, 1xN e NxN. Após a modelagem, foi gerado o script SQL para criação das tabelas, com todas as restrições e constraints necessárias, e realizado o povoamento inicial com ao menos cinco registros por tabela, permitindo a verificação dos diferentes tipos de relacionamentos.

O desenvolvimento integrou teoria e prática, reforçando a importância da modelagem como base para a consistência e desempenho do banco de dados. A implementação prática em SQL permitiu validar as estruturas criadas e demonstrar a aplicação dos conceitos no atendimento de requisitos funcionais e de integridade. Como destacam Coronel e Morris (2018, p. 55), “a qualidade do banco de dados está diretamente ligada à precisão do seu modelo e à clareza na definição das suas entidades e relacionamentos”.

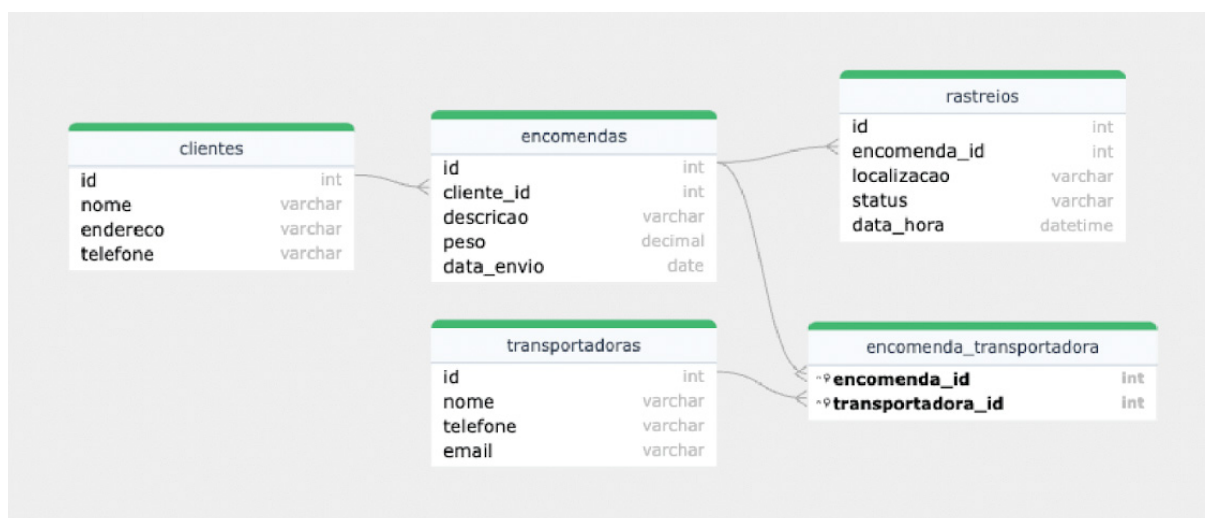
6.1. ARTEFATOS DO PROJETO

FIGURA 17 - Modelo Entidade-Relacionamento



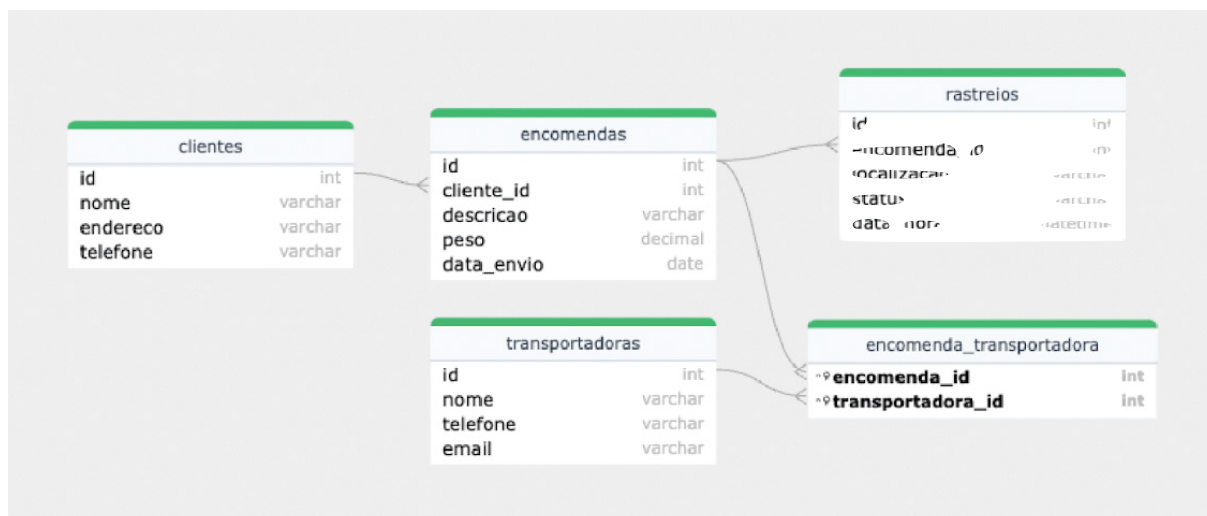
Fonte: O autor (2025)

FIGURA 18 - Modelo Lógico



Fonte: O autor (2025)

FIGURA 19 - Modelo Lógico



Fonte: O autor (2025)

7. DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

O trabalho teve como objetivo refatorar um código de exemplo do algoritmo Bubble Sort, aplicando princípios de *clean code* para torná-lo mais claro, legível e de fácil manutenção. A proposta reforçou a importância da escrita de código que seja simples de compreender, evitando complexidade desnecessária e facilitando a colaboração em equipe.

A aplicação de práticas de código limpo está diretamente alinhada aos valores do desenvolvimento ágil, pois reduz a probabilidade de erros, facilita testes e permite que ajustes sejam feitos com maior segurança e agilidade. Além disso, contribui para que o software se mantenha sustentável ao longo do tempo, mesmo diante de mudanças nos requisitos.

Como destaca Martin (2009, p. 13), “um código limpo é aquele que é fácil de ler, fácil de entender e fácil de alterar”, evidenciando que a clareza e a simplicidade são elementos essenciais para a qualidade de um projeto de software.

7.1. ARTEFATOS DO PROJETO

FIGURA 20 - Descrição das alterações e código

Renomeação de variáveis

- `trocado` foi renomeado para `isSwapped`.
- `arr` foi renomeado para `array`.
- `n` foi renomeado para `length`.

Extração de métodos

- O código de troca de elementos foi movido para o método `swap`.
- O código de iteração de bolha foi movido para o método `sortIteration`.

Uso de Enhanced For Loop

- O método `printArray` foi atualizado para usar um *enhanced for loop*, assim, não sendo necessário passar o segundo parâmetro (*size*).

```

1  class BubbleSort {
2
3      public static void main(String[] args) {
4          int[] array = { 64, 34, 25, 12, 22, 11, 90 };
5          System.out.println("Array antes da ordenação:");
6          printArray(array);
7
8          bubbleSort(array);
9
10         System.out.println("Array após a ordenação:");
11         printArray(array);
12     }
13
14     static void bubbleSort(int[] array) {
15         int length = array.length;
16         for (int i = 0; i < length - 1; i++) {
17             if (!sortIteration(array, length - i - 1)) {
18                 break;
19             }
20         }
21     }
22
23     private static boolean sortIteration(int[] array, int end) {
24         boolean isSwapped = false;
25         for (int i = 0; i < end; i++) {
26
27             if (array[j] > array[j + 1]) {
28                 swap(array, j, j + 1);
29                 isSwapped = true;
30             }
31         }
32         return isSwapped;
33     }
34
35     private static void swap(int[] array, int index1, int index2) {
36         int temp = array[index1];
37         array[index1] = array[index2];
38         array[index2] = temp;
39     }
40
41     static void printArray(int[] array) {
42         for (int value : array) {
43             System.out.print(value + " ");
44         }
45         System.out.println();
46     }
47 }

```

Fonte: O autor (2025)

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

Os trabalhos finais tiveram como objetivo o desenvolvimento de aplicações completas em Angular, evoluindo de um projeto simples com persistência local para uma aplicação integrada a um back-end em Spring Boot com banco de dados PostgreSQL.

Na disciplina Desenvolvimento Web 1, foi criada uma aplicação contendo dois CRUDs: um para gerenciamento de alunos e outro para cursos. Cada CRUD implementou funcionalidades de listagem, inserção, edição e remoção de registros, utilizando Local Storage para persistência de dados. A interface foi desenvolvida com Bootstrap ou Material Design, priorizando responsividade e clareza visual.

Na disciplina Desenvolvimento Web 2, o projeto foi expandido para incluir um terceiro CRUD, referente às matrículas, integrando dados de alunos e cursos. Foi implementado um back-end em Spring Boot, com acesso a um banco de dados PostgreSQL, eliminando o uso de Local Storage e garantindo persistência real. As operações foram disponibilizadas via API REST, e a interface manteve o uso de Bootstrap ou Material Design para consistência visual.

Essa progressão entre as duas disciplinas possibilitou a integração de conhecimentos de front-end e back-end, fortalecendo a compreensão sobre o desenvolvimento de aplicações completas e integradas.

8.1.ARTEFATOS DO PROJETO

FIGURA 21 - Tela "Lista de Cursos"

Alunos Cursos

Lista de Cursos

ID	Nome	Link	Ações
1753497841696	Desenvolvimento WEB I	https://ufprvirtual.ufpr.br/	 

[Adicionar Curso](#)

Fonte: O autor (2025)

FIGURA 22 - Formulário para cadastro de aluno

Alunos Cursos

Adicionar Aluno

Nome

CPF

Email

Data de Nascimento

[Salvar](#) [Cancelar](#)

Fonte: O autor (2025)

FIGURA 23 - Formulário para cadastro de curso

Alunos Cursos

Adicionar Curso

Nome

Link

[Salvar](#) [Cancelar](#)

Fonte: O autor (2025)

FIGURA 24 - Formulário para cadastro de matricula

Alunos Cursos Matrículas

Adicionar Matricula

Aluno

Curso

Data de Matricula

Nota

0

Salvar

Cancelar

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

O projeto consistiu no desenvolvimento de um produto digital cujo objetivo foi atender a uma necessidade específica previamente identificada. A função principal do produto foi definida com base na análise do problema e na proposta de valor para o público-alvo, justificando seu desenvolvimento como uma solução prática e acessível.

O trabalho contemplou a entrega de cinco telas prontas, elaboradas de forma a representar o fluxo principal de uso do produto. Essas telas foram projetadas considerando hierarquia visual, usabilidade e consistência, garantindo que a navegação fosse clara e intuitiva.

As escolhas de design, incluindo cores, tipografia, espaçamentos e elementos visuais, foram realizadas com base em princípios de legibilidade, contraste e identidade visual. A paleta de cores buscou transmitir o tom e a personalidade da marca, enquanto a tipografia foi escolhida para oferecer clareza na leitura e coerência com o estilo do produto. O layout seguiu padrões responsivos, garantindo adaptação a diferentes dispositivos.

Como parte do processo, as telas foram apresentadas a um possível usuário, que recebeu explicações sobre a função principal do produto, as opções de interação e a navegação. Durante essa etapa, foi coletado feedback sobre clareza, facilidade de uso e atratividade visual. As observações obtidas foram registradas e analisadas, servindo de base para ajustes e melhorias no design final.

Esse trabalho reforça a importância da experiência do usuário como fator central no desenvolvimento de soluções digitais. Como afirma Norman (2013, p. 9), “o design centrado no ser humano começa com uma compreensão profunda das pessoas para quem o produto está sendo projetado”. Assim, o projeto não apenas entregou um protótipo visual, mas também incorporou o ciclo iterativo de concepção, validação e refinamento característico do desenvolvimento ágil e orientado ao usuário.

9.1. ARTEFATOS DO PROJETO

FIGURA 25 - Descrição do aplicativo e telas de cadastro de usuário e login

UX no Desenvolvimento Ágil de Software

Universidade Federal do Paraná

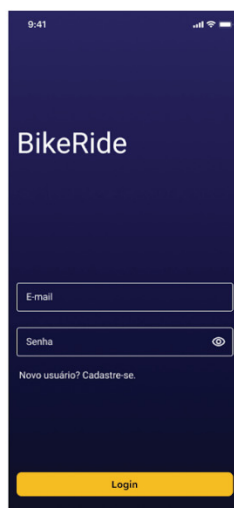
Especialização em Desenvolvimento Ágil de Software

Guilherme Zanini Moreira

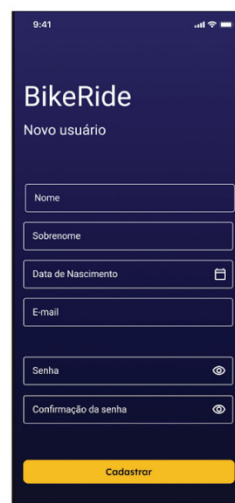
Explicação do Produto

- **Nome do Produto:** BikeRide
- **Função:** BikeRide é um aplicativo que facilita a criação e o gerenciamento de passeios de bicicleta em grupo. Os usuários podem organizar passeios, convidar outros ciclistas, compartilhar rotas através de arquivos GPX, e se inscrever em passeios organizados por outros. Além disso, há uma funcionalidade de chat integrada para que os participantes possam se comunicar antes, durante e depois do passeio.
- **Justificativa para o Desenvolvimento:** Muitos ciclistas gostam de explorar novas rotas e conhecer outros entusiastas do esporte, mas têm dificuldades em organizar passeios em grupo de maneira eficiente. BikeRide visa simplificar a organização desses passeios e criar uma comunidade engajada de ciclistas que compartilham suas aventuras.
- **Público-alvo:** Ciclistas amadores e profissionais que desejam participar de passeios em grupo, explorar novas rotas e se conectar com outros ciclistas na sua região.

Telas

A tela de login do aplicativo BikeRide possui um fundo escuro com o nome 'BikeRide' em branco no topo. Abaixo, há campos de entrada para 'E-mail' e 'Senha'. O campo de senha possui um ícone de olho para alternar a visibilidade. Abaixo dos campos, há o texto 'Novo usuário? Cadastre-se.' e um botão amarelo 'Login' no rodapé.

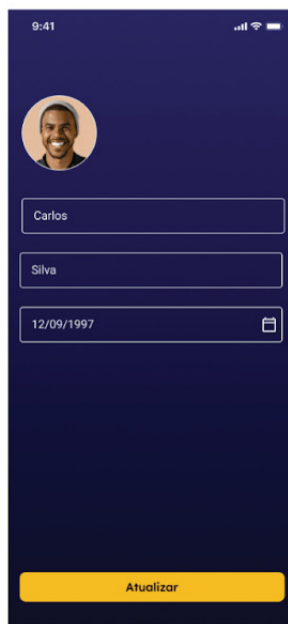
(a) Login

A tela de cadastro de usuário do aplicativo BikeRide possui um fundo escuro com o nome 'BikeRide' e o subtítulo 'Novo usuário' em branco no topo. Abaixo, há campos de entrada para 'Nome', 'Sobrenome', 'Data de Nascimento' (com ícone de calendário), 'E-mail', 'Senha' e 'Confirmação da senha'. Os campos de senha possuem ícones de olho para alternar a visibilidade. Um botão amarelo 'Cadastrar' está no rodapé.

(b) Cadastro de usuário

Fonte: O autor (2025)

FIGURA 26 - Tela para atualizar dados do usuário e escolhas de design e funcionalidades



(g) Atualizar perfil

1 Escolhas de Design e Funcionalidades

1.1 Cores Primárias e Paleta de Cores

Cor Primária: #3331A3

Esta cor, um tom de azul escuro e intenso, será usada para os elementos principais do aplicativo, como cabeçalhos, botões de ação primária e ícones principais. O azul transmite confiança, segurança e profissionalismo, sendo ideal para criar uma sensação de comunidade e organização.

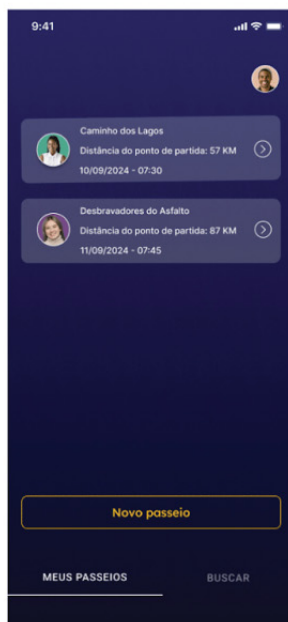
1.1.1 Cores de Texto

- **Branco:** #FFFFFF - Usado para o texto em cima de fundos escuros. O contraste do branco sobre o azul escuro da cor primária cria uma leitura clara e nítida, melhorando a usabilidade.
- **Preto:** #000000 - Utilizado para textos sobre botão primário com fundo na cor de destaque.

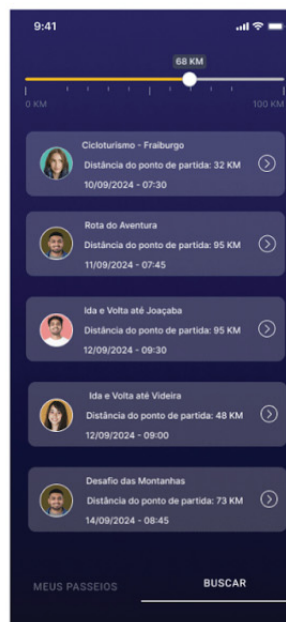
Cor de Destaque (Accent Color): #FFC107

Essa cor é um tom de amarelo que será usada para destacar elementos interativos, como botões primários e o slider. O amarelo é vibrante e atrai a atenção do usuário, ajudando a guiar a interação com o app.

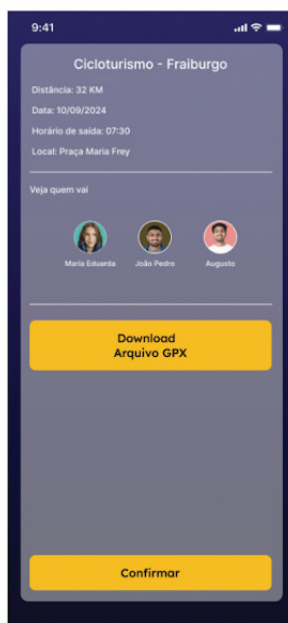
FIGURA 27 - Telas do aplicativo



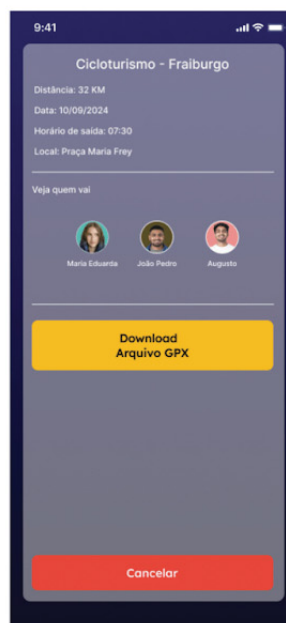
(c) Meus passeios



(d) Buscar passeios



(e) Detalhes do passeio



(f) Buscar passeios

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

Os trabalhos finais das disciplinas Desenvolvimento Mobile I e Desenvolvimento Mobile II tiveram como foco a criação de aplicativos Android que aplicam conceitos fundamentais da programação móvel, desde a manipulação de layouts e atividades até o consumo de web services.

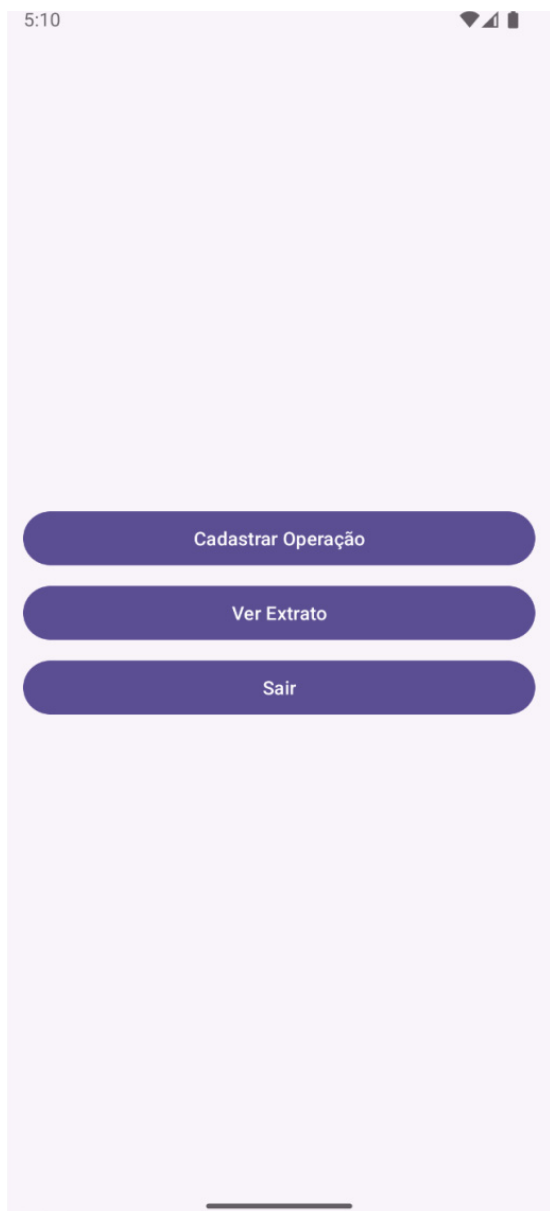
No Desenvolvimento Mobile I, foi criado o FinApp, um aplicativo voltado para a gestão financeira pessoal. Ele permite o cadastro de transações classificadas como débito ou crédito, com descrição e valor, armazenadas em memória. A aplicação apresenta um Dashboard com botões para acessar o cadastro de operações, o extrato e encerrar o aplicativo. O Cadastro de operações possibilita inserir novas transações, enquanto o Extrato exibe todas as operações cadastradas com tipo, descrição e valor.

Já no Desenvolvimento Mobile II, foi desenvolvido um aplicativo integrado à HP-API, explorando corrotinas e comunicação com serviços externos. O Dashboard inicial oferece botões para listar personagem por ID, listar professores, listar estudantes de uma casa e sair. A função de listagem por ID retorna o nome e a casa do personagem, enquanto a listagem de professores e estudantes apresenta, respectivamente, o corpo docente e os alunos de uma casa selecionada pelo usuário.

Ambos os projetos se alinham a práticas do desenvolvimento ágil de software, pois foram desenvolvidos em pequenas iterações, permitindo testes e ajustes constantes. O primeiro reforçou conceitos de interface e armazenamento temporário de dados, enquanto o segundo evidenciou a importância da integração com APIs externas. Como afirma Sommerville (2011, p. 59), “a prototipagem rápida e a entrega incremental são práticas que reduzem riscos e melhoram a comunicação com o usuário”. Além disso, a utilização de chamadas assíncronas e testes frequentes está em conformidade com o que Beck e Andres (2004, p. 45) destacam: “a integração frequente e a automação de testes asseguram que mudanças possam ser incorporadas com confiança”.

10.1. ARTEFATOS DO PROJETO

FIGURA 28 - Tela inicial



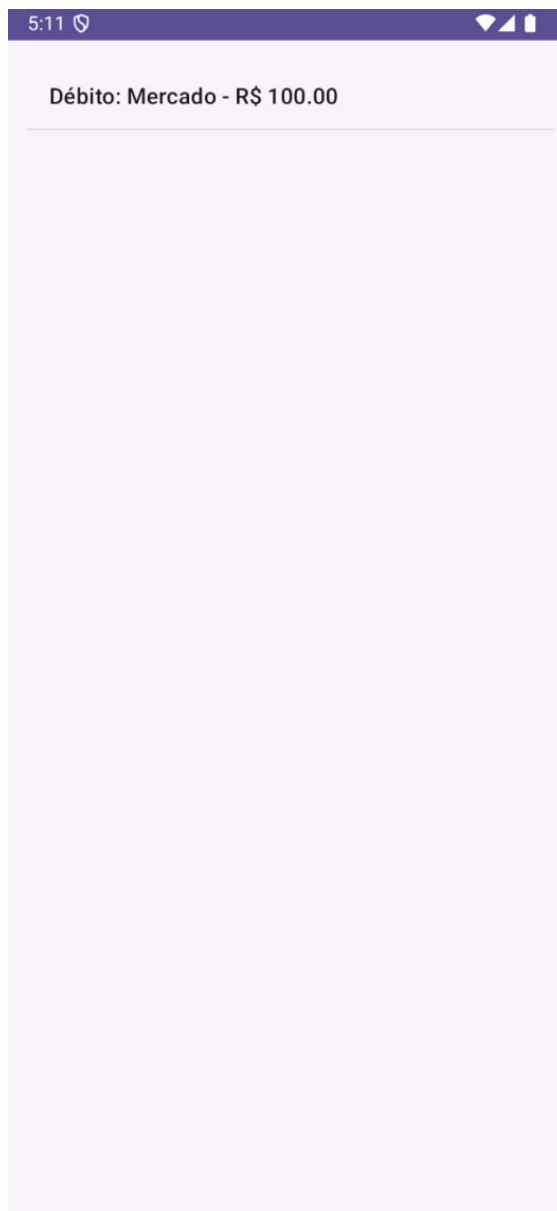
Fonte: O autor (2025)

FIGURA 29 - Tela de cadastro

A imagem mostra a tela de cadastro de uma operação. No topo, há uma barra de status com o horário 5:11 e ícones de conexão. O fundo da tela é branco. Abaixo da barra de status, há dois campos de entrada com rótulos "Descrição" e "Valor" em cinza. Abaixo desses campos, há duas opções de seleção: "Crédito" com um botão de rádio desativado e "Débito" com um botão de rádio ativado. Na parte inferior, há um botão retangular com cantos arredondados em um tom de verde-água, rotulado "Salvar".

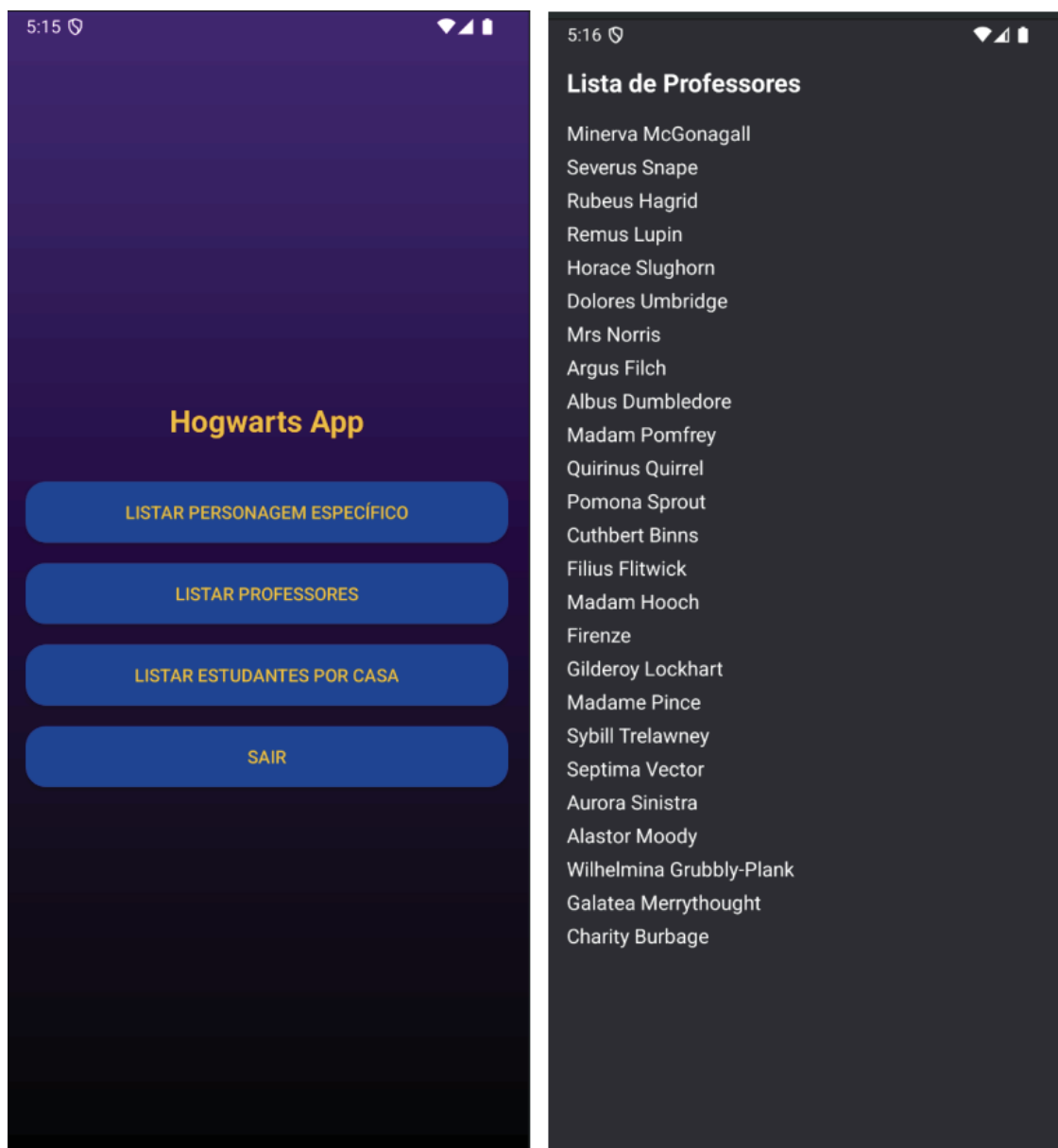
Fonte: O autor (2025)

FIGURA 30 - Lista de transações



Fonte: O autor (2025)

FIGURA 31 - Menu e tela “Lista de Professores”



Fonte: O autor (2025)

FIGURA 32 - Seleção de casas e tela com dados do estudante

5:16

Seleção de casas e tela com dados do estudante

Seleção de casas

☐ Gryffindor

☐ Slytherin

☒ Hufflepuff

☐ Ravenclaw

Buscar Estudantes

Cedric Diggory

Newt Scamander

Fat Friar

Hannah Abbott

Susan Bones

Justin Finch-Fletchley

Pomona Sprout

Ernie Macmillan

Eleanor Branstone

Owen Cauldwell

Laura Madley

Kevin Whitby

Summers

Nymphadora Tonks

Rose Zeller

Summerby

Leanne

Cadwallader

Ted Lupin



Harry Potter

Casa: Gryffindor

Espécie: human

Gênero: male

Data de Nascimento: 31-07-1980

Patrono: stag

Ator: Daniel Radcliffe

Ancestralidade: half-blood

Cor dos Olhos: green

Cor do Cabelo: black

Varinha: Madeira - holly,
Núcleo - phoenix tail feather,
Comprimento - 11.0"

Fonte: O autor (2025)

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

O projeto tem como objetivo configurar e executar, via Docker, a imagem `dfwandarti/gitlab_jenkins:3`, nomeando o container com o número da matrícula e publicando as portas 22, 80, 443 e 9091. Após a execução, deve-se capturar um print screen do comando utilizado e do container em funcionamento.

Em seguida, é necessário acessar o GitLab utilizando o usuário `root`, cuja senha inicial encontra-se no arquivo `/etc/gitlab/initial_root_password` dentro do container. Após o login, deve-se capturar outro *print screen* mostrando o acesso no navegador.

A etapa final consiste em realizar um commit e um push no projeto já presente no repositório do Git, e registrar, por meio de um *print screen*, o log do Git evidenciando a realização do commit.

Este trabalho envolve conhecimentos práticos de infraestrutura aplicada ao desenvolvimento de software, ao exigir a configuração de serviços de versionamento e integração contínua em ambiente containerizado. Está alinhado aos princípios do desenvolvimento ágil, pois ferramentas como GitLab e Jenkins são amplamente utilizadas para entrega contínua (Continuous Delivery) e integração contínua (Continuous Integration). Segundo Humble e Farley (2011, p. 6), “a entrega contínua possibilita que o software seja implantado em produção de forma rápida, sustentável e com baixo risco”, destacando a importância desse tipo de prática no ciclo moderno de desenvolvimento.

Dessa forma, o projeto consolida competências essenciais para ambientes de desenvolvimento ágil, combinando práticas de controle de versão, automação e gerenciamento de infraestrutura, reforçando a integração entre codificação, testes e implantação.

11.1. ARTEFATOS DO PROJETO

FIGURA 33 - Comandos executados e tela do GtLab

Infraestrutura para desenvolvimento e implantação de Software (DevOps)

Universidade Federal do Paraná
Especialização em Desenvolvimento Ágil de Software
Guilherme Zanini Moreira

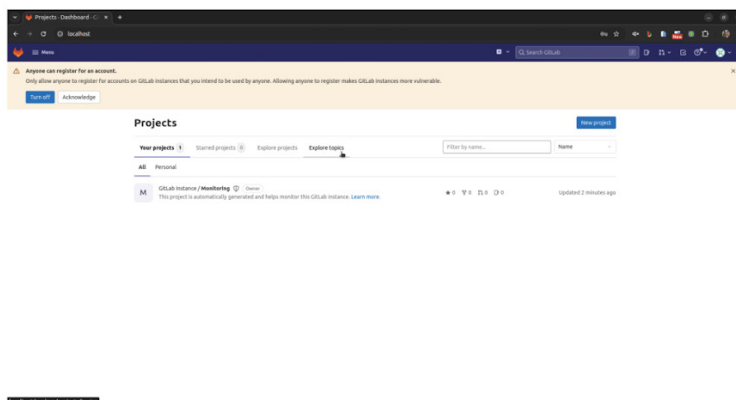


```

guilherme@guilherme:~$ docker run -d --name gitlab --hostname gitlab --shm-size 128m --ulimit memlock=-1 --ulimit stack=65536 --restart=always --privileged --network=host --p 22:22 --p 80:80 --p 443:443 --p 9091:9091 \
guilherme@guilherme:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS
gitlab         guilherme@guilherme:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS
gitlab         guilherme@guilherme:~$

```

(1) Comandos - iniciar execução do container e *docker ps*



(2) Screenshot do navegador após o login no GitLab

Fonte: O autor (2025)

FIGURA 34 - Comando e tela do GitLab

DAS - INFRA

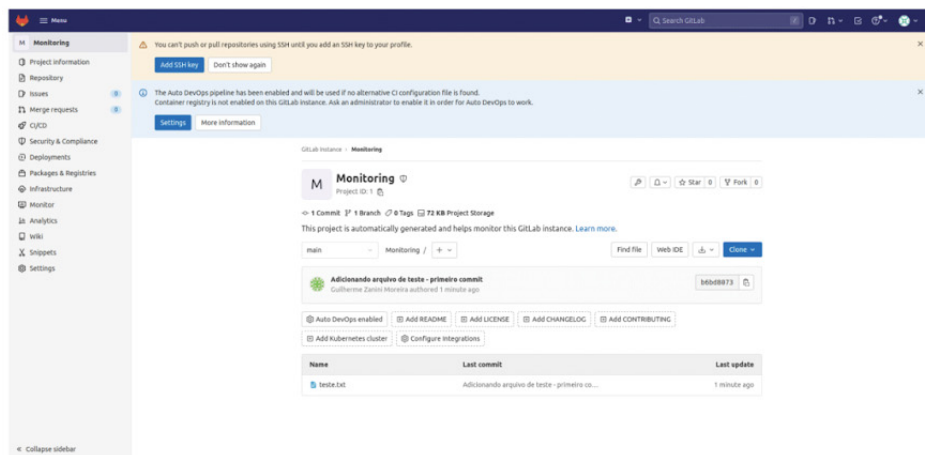
Guilherme Zanini Moreira

```

guilherme@guilherme:~/Monitoring
guilherme in guilherme on main [?]
> git clone http://localhost/gitlab-instance-f7539248/Monitoring.git
Cloning into 'Monitoring'...
Username for 'http://localhost': root
Password for 'http://root@localhost':
Warning: You appear to have cloned an empty repository.
guilherme in guilherme on main [?] took 8s
> cd Monitoring
guilherme in Monitoring on main
> ls
guilherme in Monitoring on main
> echo "Teste de commit" > teste.txt
guilherme in Monitoring on main [?]
> git add teste.txt
guilherme in Monitoring on main [*]
> git commit -m "Adicionando arquivo de teste - primeiro commit"
[main (root-commit) b6bd807] Adicionando arquivo de teste - primeiro commit
1 file changed, 1 insertion(+)
 create mode 100644 teste.txt
guilherme in Monitoring on main
> git push origin main
Username for 'http://localhost': root
Password for 'http://root@localhost':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 272 bytes | 272.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://localhost/gitlab-instance-f7539248/Monitoring.git
 * [new branch]      main -> main

```

(3) Clone do repositório *Monitoring*, criação de novo arquivo, adicionado o arquivo para o estágio *staging* do Git, comando para criar commit com mensagem e comando para *push* para a branch *main*



(4) Screenshot após o commit

12. DISCIPLINA: TEST – TESTES AUTOMATIZADOS

O projeto consiste em desenvolver um código capaz de acessar o editor de texto online Anotepad e criar automaticamente uma nova nota com o título "Entrega trabalho TEST DAS 2024". No corpo da nota, são inseridos os nomes e matrículas dos integrantes do trabalho, sendo que, em caso de grupos, todos os participantes devem ser listados.

Para a implementação, foi utilizado o Playwright, ferramenta que permite a automação de navegadores e a simulação de interações de forma rápida e precisa. Com ela, foi possível reproduzir todo o fluxo de uso do Anotepad: abrir o site, preencher o título e o corpo da nota, e salvar o conteúdo.

A proposta está alinhada a práticas do desenvolvimento ágil de software, pois envolve pequenos ciclos de implementação, execução de testes frequentes e ajustes rápidos até alcançar o resultado esperado. Segundo Beck e Andres (2004, p. 55), “testes automatizados são a base para a evolução segura e contínua de um sistema”, evidenciando sua relevância no ciclo ágil. Além disso, o trabalho conecta-se diretamente à área de desenvolvimento web, ao exigir a compreensão da estrutura e do comportamento dos elementos HTML manipulados pelo script de automação.

Assim, o projeto reforça habilidades práticas ligadas à automação de tarefas, à interação com aplicações web e à aplicação de testes como parte essencial do processo de desenvolvimento moderno de software.

12.1. ARTEFATOS DO PROJETO

FIGURA 35 - Código para testes com Playwright

```
package br.ufpr;

import com.microsoft.playwright.*;
import com.microsoft.playwright.options.WaitUntilState;

public class Main {
    public static void main(String[] args) {
        try (Playwright playwright = Playwright.create()) {
            Browser browser = playwright
                .chromium()
                .launch(new BrowserType.LaunchOptions().setHeadless(false));

            Page page = browser.newPage();

            // Acessa o site do Anotepad
            page.navigate("https://pt.anotepad.com", new Page.NavigateOptions().setWaitUntil(WaitUntilState.DOMCONTENTLOADED)
                .setTimeout(30000));

            page.fill("#edit_title", "Entrega trabalho TEST DAS 2024");

            String conteudo = "Nome: Guilherme Zanini Moreira\nMatricula: 40001016398E1";
            page.fill("#edit_textarea", conteudo);
        }
    }
}
```

Fonte: O autor (2025)

13. CONCLUSÃO

O conjunto de trabalhos apresentados ao longo das disciplinas evidencia uma trajetória consistente de aprendizado e aplicação prática dos princípios do desenvolvimento ágil de software. Foram exploradas múltiplas dimensões da engenharia de software, desde a concepção e modelagem até a implementação, testes, gestão de projetos e avaliação da experiência do usuário. Essa diversidade de atividades permitiu consolidar habilidades técnicas e metodológicas, integrando conhecimentos de front-end, back-end, banco de dados, automação e design centrado no usuário.

Os resultados obtidos demonstram que a aplicação dos métodos ágeis proporciona maior adaptabilidade, comunicação clara entre equipes e entrega contínua de valor. Ao longo das atividades, práticas como clean code, integração contínua, automação de testes e uso de frameworks modernos se mostraram fundamentais para garantir qualidade e eficiência no desenvolvimento.

Entretanto, a adoção prática do desenvolvimento ágil de software apresenta desafios que merecem atenção. Entre eles, destacam-se a necessidade de maturidade organizacional para sustentar processos iterativos, a dificuldade de alinhar expectativas de stakeholders com ciclos curtos de entrega, a importância de manter documentação suficiente sem comprometer a agilidade e a exigência de uma cultura colaborativa para que as cerimônias e práticas ágeis gerem resultados efetivos. Além disso, a integração entre diferentes tecnologias e equipes requer padrões claros de comunicação e arquitetura para evitar inconsistências e retrabalho.

Em síntese, os trabalhos aqui reunidos não apenas consolidam competências técnicas, mas também reforçam a compreensão sobre os benefícios e limitações da aplicação do ágil no contexto real. A experiência adquirida configura um portfólio sólido, refletindo tanto a capacidade de entregar soluções de valor quanto a consciência crítica sobre os fatores que influenciam a eficácia das práticas ágeis na indústria de software.

14. REFERÊNCIAS

ANDERSON, D.; CARMICHAEL, A. Essential Kanban Condensed. Sequim: Lean Kanban University Press, 2016.

AMBLER, S.; LINES, M. Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. Upper Saddle River: IBM Press, 2012.

BECK, K. Test Driven Development: By Example. Boston: Addison-Wesley, 2003

BECK, K.; ANDRES, C. Extreme Programming Explained: Embrace Change. 2. ed. Boston: Addison-Wesley, 2004.

CORONEL, C.; MORRIS, S. Database Systems: Design, Implementation, & Management. 13. ed. Boston: Cengage Learning, 2018.

HIGHSMITH, J. Agile Project Management: Creating Innovative Products. 2. ed. Boston: Addison-Wesley, 2009.

HUMBLE, J.; FARLEY, D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Upper Saddle River: Addison-Wesley, 2011.

MARTIN, R. C. Código Limpo: Habilidades Práticas do Agile Software. 1. ed. São Paulo: Alta Books, 2009.

NORMAN, D. O design do dia a dia. Rio de Janeiro: Rocco, 2013.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson, 2011.