

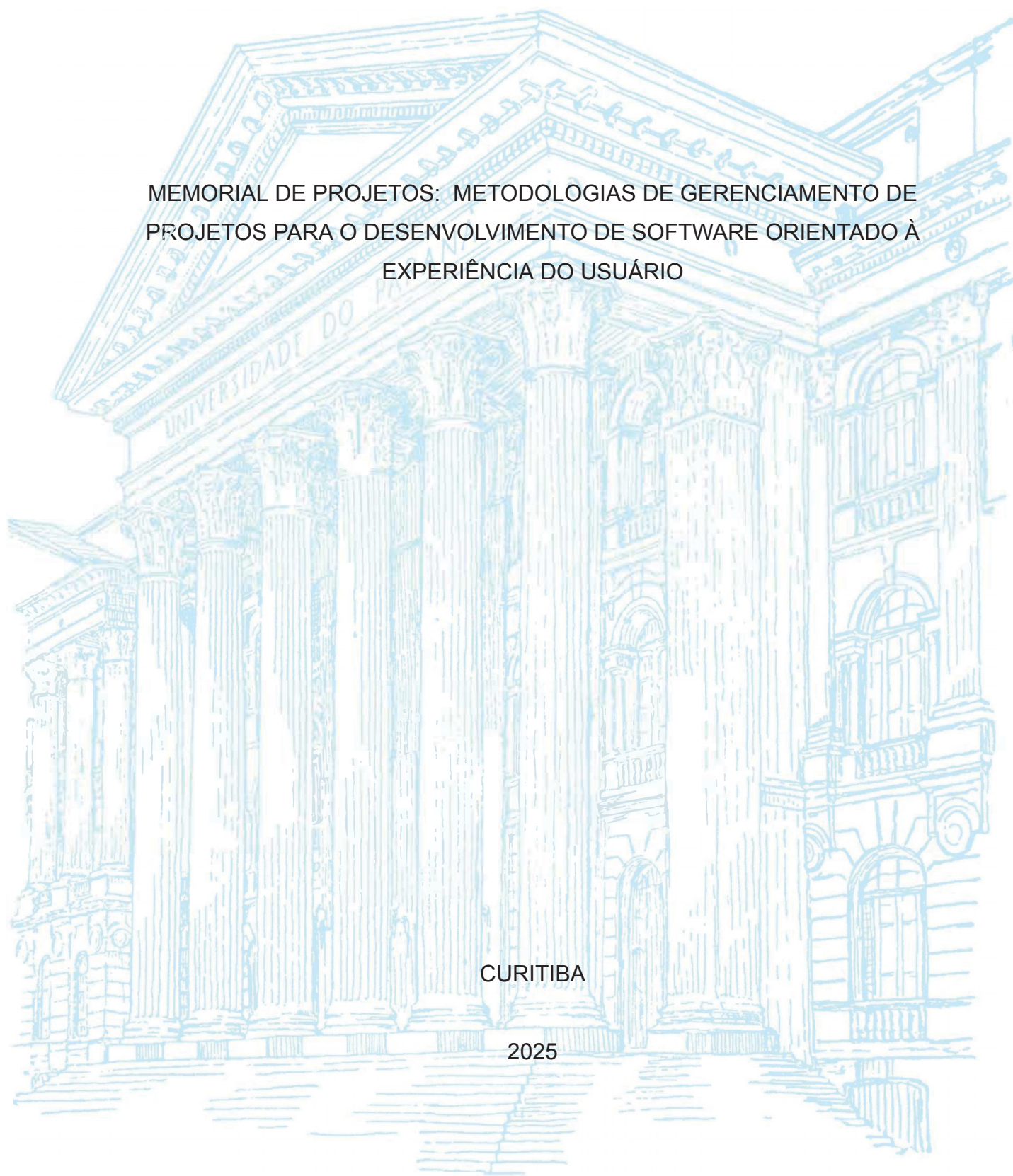
UNIVERSIDADE FEDERAL DO PARANÁ

VALÉRIA ROMÃO PASQUALINI NERIO

MEMORIAL DE PROJETOS: METODOLOGIAS DE GERENCIAMENTO DE  
PROJETOS PARA O DESENVOLVIMENTO DE SOFTWARE ORIENTADO À  
EXPERIÊNCIA DO USUÁRIO

CURITIBA

2025



VALÉRIA ROMÃO PASQUALINI NERIO

MEMORIAL DE PROJETOS: METODOLOGIAS DE GERENCIAMENTO DE  
PROJETOS PARA O DESENVOLVIMENTO DE SOFTWARE ORIENTADO À  
EXPERIÊNCIA DO USUÁRIO

Memorial de Projetos apresentado ao curso de Especialização 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. Razer Anthom Nizer Rojas Montañó

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 **VALERIA ROMAO PASQUALINI NERIO**, intitulada: **MEMORIAL DE PROJETOS: METODOLOGIAS DE GERENCIAMENTO DE PROJETOS PARA O DESENVOLVIMENTO DE SOFTWARE ORIENTADO À EXPERIÊNCIA DO USUÁRIO**, 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, 23 de Outubro de 2025.



RAZER ANTHOM NIZER ROJAS MONTAÑO  
Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

## RESUMO

O presente memorial descreve e integra os projetos desenvolvidos ao longo do curso de Especialização em Desenvolvimento Ágil de Software da Universidade Federal do Paraná. O trabalho aborda as práticas que abrangem o ciclo de vida do software, incluindo modelagem, gerenciamento, desenvolvimento e infraestrutura. A análise deste portfólio resulta em um parecer técnico que propõe que a eficácia do desenvolvimento ágil pode estar associada à integração entre metodologias de gerenciamento de projetos e um foco contínuo na experiência do usuário (UX). Na realização de projetos que envolveu a validação do usuário, compreende-se que, enquanto as práticas ágeis fornecem a estrutura para entregas eficientes, otimizando o fluxo e a velocidade, o UX pode contribuir com a direção estratégica, favorecendo que o produto final, por meio de prototipagem e testes com usuários, atenda às necessidades reais do cliente e gere valor de negócio. Dessa forma, evidencia-se que a agilidade não reside unicamente na aplicação de frameworks, mas também na internalização de seus princípios e no aprendizado contínuo dos profissionais. Por fim, para a adoção prática desta abordagem, sugere-se a integração do UX de forma contínua no ritmo dos sprints, considerando que trata-se de um processo colaborativo e que demanda foco em métricas de valor.

**Palavras-chave:** gestão de projetos de tecnologia; desenvolvimento de software; experiência do usuário; métricas de valor; portfólio.

## **ABSTRACT**

This memorial describes and integrates the projects developed throughout the Specialization course in Agile Software Development at the Federal University of Paraná. The work addresses practices spanning the software lifecycle, including modeling, management, development, and infrastructure. The analysis of this portfolio results in a technical report proposing that agile development effectiveness may be associated with the integration of project management methodologies and a continuous focus on user experience (UX). Based on the projects involving user validation, it is understood that while agile practices provide the framework for efficient deliveries by optimizing flow and velocity, UX can contribute strategic direction. This contribution helps ensure the final product, through prototyping and user testing, meets real customer needs and generates business value. Thus, it becomes evident that agility does not lie solely in the application of frameworks but also in the internalization of its principles and in the continuous learning of professionals. Finally, for the practical adoption of this approach, the continuous integration of UX within the sprint cadence is suggested, recognizing it as a collaborative process that demands a focus on value-driven metrics.

**Keywords:** technology project management; software development; user experience; value-driven metrics; portfolio.

## SUMÁRIO

|                                                                                                           |           |
|-----------------------------------------------------------------------------------------------------------|-----------|
| <b>1 PARECER TÉCNICO.....</b>                                                                             | <b>7</b>  |
| <b>2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....</b>                           | <b>9</b>  |
| 2.1 ARTEFATOS DO PROJETO.....                                                                             | 10        |
| <b>3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2.....</b>                                  | <b>14</b> |
| 3.1 ARTEFATOS DO PROJETO.....                                                                             | 15        |
| <b>4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2.....</b>                  | <b>18</b> |
| 4.1 ARTEFATOS DO PROJETO.....                                                                             | 19        |
| <b>5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....</b>                                                | <b>22</b> |
| 5.1 ARTEFATOS DO PROJETO.....                                                                             | 23        |
| <b>6 DISCIPLINA: BD – BANCO DE DADOS.....</b>                                                             | <b>24</b> |
| 6.1 ARTEFATOS DO PROJETO.....                                                                             | 25        |
| <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>31</b> |
| 8.1 ARTEFATOS DO PROJETO.....                                                                             | 32        |
| <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>38</b> |
| 10.1 ARTEFATOS DO PROJETO.....                                                                            | 39        |
| <b>11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS).....</b> | <b>41</b> |
| 11.1 ARTEFATOS DO PROJETO.....                                                                            | 42        |
| <b>12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS.....</b>                                                    | <b>44</b> |
| 12.1 ARTEFATOS DO PROJETO.....                                                                            | 45        |
| <b>13 CONCLUSÃO.....</b>                                                                                  | <b>47</b> |
| <b>REFERÊNCIAS.....</b>                                                                                   | <b>48</b> |

## 1 PARECER TÉCNICO

Este parecer técnico descreve os projetos realizados durante o curso de Especialização em Desenvolvimento Ágil de Software ao decorrer das disciplinas e realização das atividades acadêmicas. Dessa forma, neste documento há descrições sobre como estes projetos se integram conceitualmente ou na prática, no âmbito do desenvolvimento ágil de software.

O enfoque deste parecer técnico está na análise de metodologias de gerenciamento de projetos para o desenvolvimento de software, considerando o ciclo de vida do projeto e com foco na experiência do usuário.

Inicialmente, dentre as metodologias inerentes ao gerenciamento de projetos, o Plano de *Release*, que detalha o escopo e as entregas previstas, materializa a transição da gestão tradicional, focada em escopo fechado, para a gestão ágil, focada em valor e adaptação. O Plano possui intrínseca relação com a teoria do Scrum, cujo objetivo é transformar uma visão de produto num plano de ação tangível, iterativo e centrado no cliente, que é a base para iniciar um projeto de desenvolvimento ágil (Pereira *et al*, 2007).

Nesse sentido, segundo Spies (2013, p. 27), uma vez que o conteúdo da release é planejado, é fundamental acompanhá-lo continuamente. Sugerindo-se que o plano seja revisado a cada iteração para monitorar o progresso e permitir ações corretivas caso desvios importantes sejam identificados.

No projeto prático de Gerenciamento Ágil de Projetos de Software I, foram desenvolvidos os conhecimentos de Scrum para conceber um software e elaborar seu plano de lançamento. A tarefa exigiu a criação de um Plano de *Release* detalhado com o cálculo da velocidade da equipe, estabelecendo a capacidade de trabalho por *sprint*, incluindo o número de *sprints*, com datas de início e fim, e populando cada uma delas com histórias de usuário.

Dessa forma, o gerenciamento de projetos vai além da fase de planejamento, envolvendo um esforço contínuo para otimizar o fluxo de trabalho e assegurar a qualidade em todas as etapas. Para isso, ao utilizar Kanban e métricas, o profissional se torna capaz de identificar gargalos, prever entregas com maior precisão e tomar decisões baseadas em dados, garantindo não apenas a entrega do produto, mas a sustentabilidade e a excelência técnica do desenvolvimento a longo prazo (Lopes, 2024, p. 13).

Na disciplina de Gerenciamento Ágil de Projetos de Software II, foi realizada uma prática simulada de gestão de fluxo utilizando a plataforma online *Kanban Simulator* (Kanban, 2025). A atividade consistiu em executar um ciclo de desenvolvimento de 35 dias, gerenciando tarefas com base nos conceitos de Kanban, como otimização de fluxo e gerenciamento de gargalos. Essa abordagem prática alinha-se ao conceito de *Serious Games* (SGs), que são ferramentas projetadas especificamente para fins educacionais ou de simulação, em vez de mero entretenimento (Hernández-Reveles; Fernandez; Romero Moyano, 2024).

A apresentação de resultados concretos, baseou-se em desenvolver um relatório contendo as capturas de tela do quadro Kanban no 15º dia de simulação, da tela final com o resultado financeiro do projeto e do Diagrama de Fluxo Acumulado (CFD), evidenciando a capacidade de se gerir um projeto orientado a fluxo contínuo. Adicionalmente, elaborou-se um plano de gestão da qualidade, identificando como lidar com a denominada Dívida Técnica e quais práticas de automação de testes podem ser adotadas para a melhoria nos processos de trabalho.

Na perspectiva técnica e de mercado, a transição do PMBOK 6ª para a 7ª edição reflete uma maturação das disciplinas de gerenciamento de projetos I e II. A abordagem prescritiva da 6ª edição (PMI, 2017), embora robusta para ambientes preditivos, mostrava-se cada vez mais desalinhada com a volatilidade e complexidade dos projetos de software. A 7ª edição (PMI, 2021), portanto, não deve ser vista apenas como uma atualização, mas como um parecer do próprio PMI de que o foco deve ser na entrega de valor e na adaptação ao contexto, independentemente do método preditivo, ágil ou híbrido. Logo, o novo guia confere ao gerente de projetos a flexibilidade necessária para customizar sua abordagem, tornando-o mais um estrategista de valor do que um executor de processos.

Portanto, compreende-se que todos os projetos foram cruciais para que um especialista consiga não apenas aplicar métodos específicos, mas também para que possa analisar criticamente diferentes contextos e adaptar ou mesclar práticas ágeis, compreendendo os princípios fundamentais que devem guiar suas decisões no ambiente de desenvolvimento de software e que atenda as necessidades reais dos clientes.

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

O projeto da disciplina de Métodos Ágeis de Desenvolvimento de Software (MADS), teve como objetivo a elaboração de um mapa mental e visa representar visualmente os principais conceitos relacionados aos processos de software, princípios e métodos ágeis.

A importância deste projeto reside na sua capacidade de se estruturar e conectar um complexo e amplo conhecimento que é fundamental para a prática do desenvolvimento ágil. Dessa forma, ao abordar desde os modelos tradicionais até as práticas modernas como a *Entrega Contínua*, foi possível visualizar a evolução das metodologias e entender as razões que motivaram o surgimento do movimento ágil.

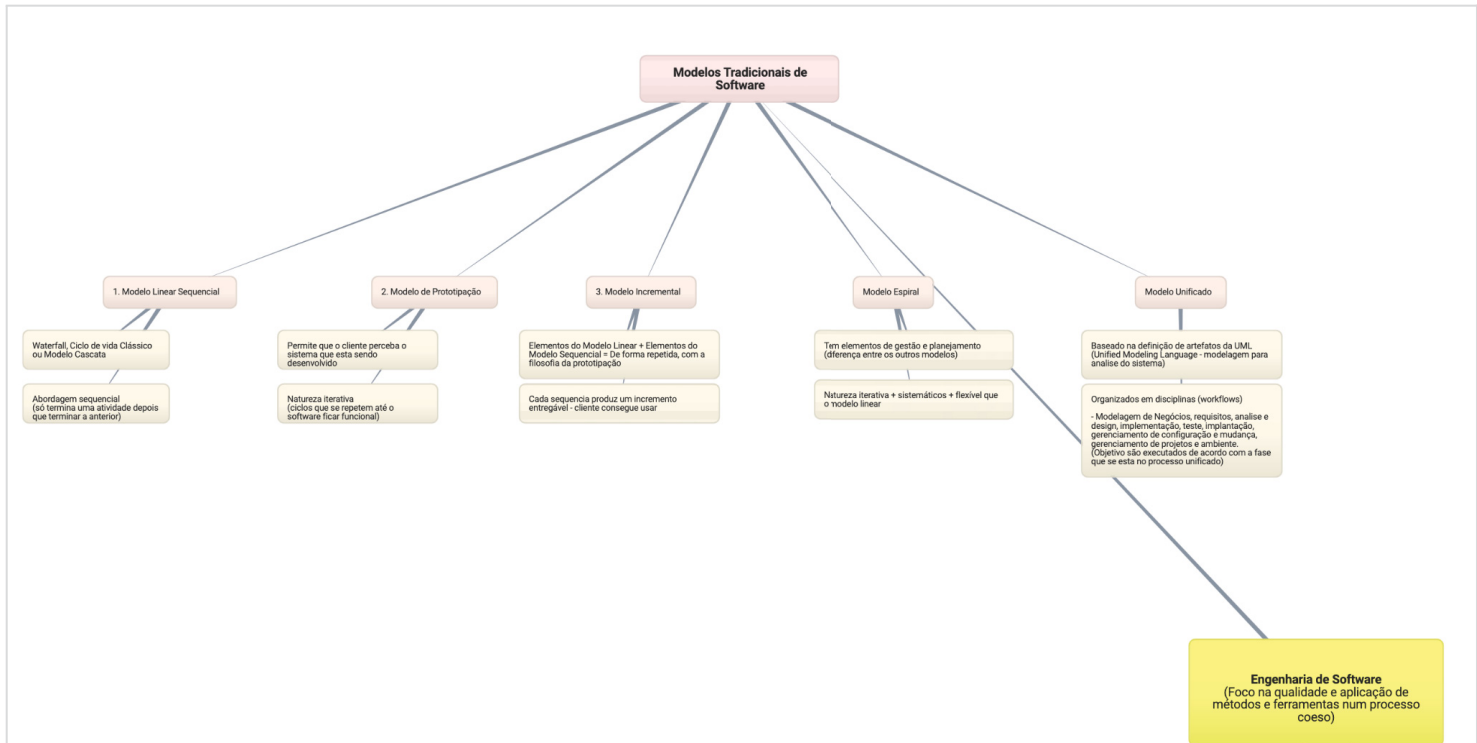
Segundo Sato (2007, p. 13) o desenvolvimento de software é uma atividade intrinsecamente humana e criativa, distante da previsibilidade de uma linha de produção em massa. Segundo ele, o sucesso depende fundamentalmente da comunicação e da colaboração entre as pessoas, ressaltando-se a natureza exploratória do processo, comparando-a à criação de um novo produto ou de uma receita inédita, com um caminho de aprendizado contínuo marcado por tentativas e erros.

Foi a partir dessa compreensão que emergiram os *Métodos Ágeis de Desenvolvimento de Software*, que propõem uma abordagem focada na interação entre os indivíduos e na entrega de valor concreto, em detrimento de uma documentação exaustiva e processos burocráticos, a fim de produzir software de qualidade.

Durante a realização da disciplina foi abordado constantemente que software funcionando é a medida do progresso, cuja maior prioridade é, portanto, a satisfação do cliente, alcançada através da entrega rápida e contínua de software de qualidade. Assim, ao se utilizar de *frameworks* como Scrum, *Extreme Programming* e Kanban, juntamente com os princípios do Manifesto Ágil e do Lean, permite-se consolidar as práticas sobre as cerimônias, papéis, artefatos e valores subjacentes (Gonzalez, 2021).

## 2.1 ARTEFATOS DO PROJETO

FIGURA 1 – MAPA MENTAL EM MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE (MADS)



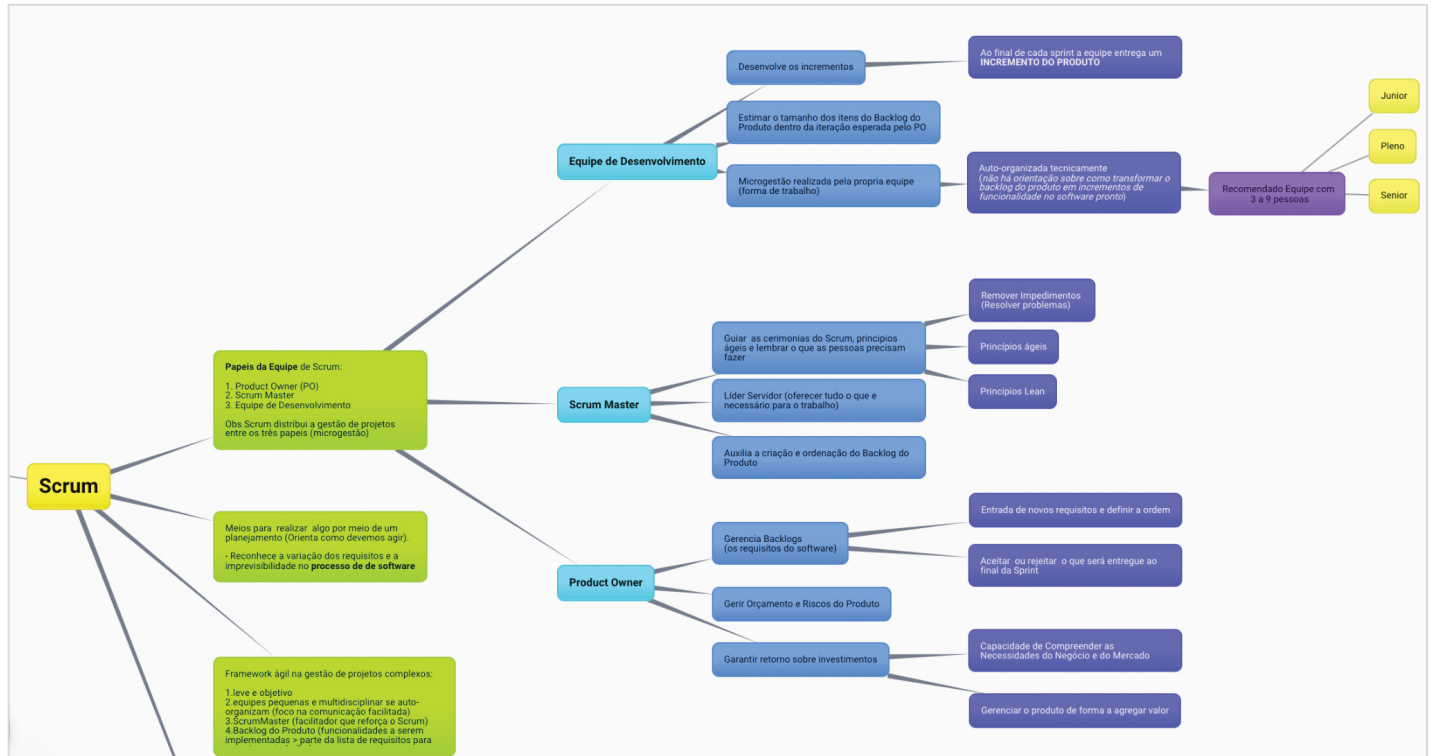
FONTE: Adaptado de Cohn, 2025.

FIGURA 2 – MAPA MENTAL EM MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE (MADS) - ENGENHARIA DE SOFTWARE



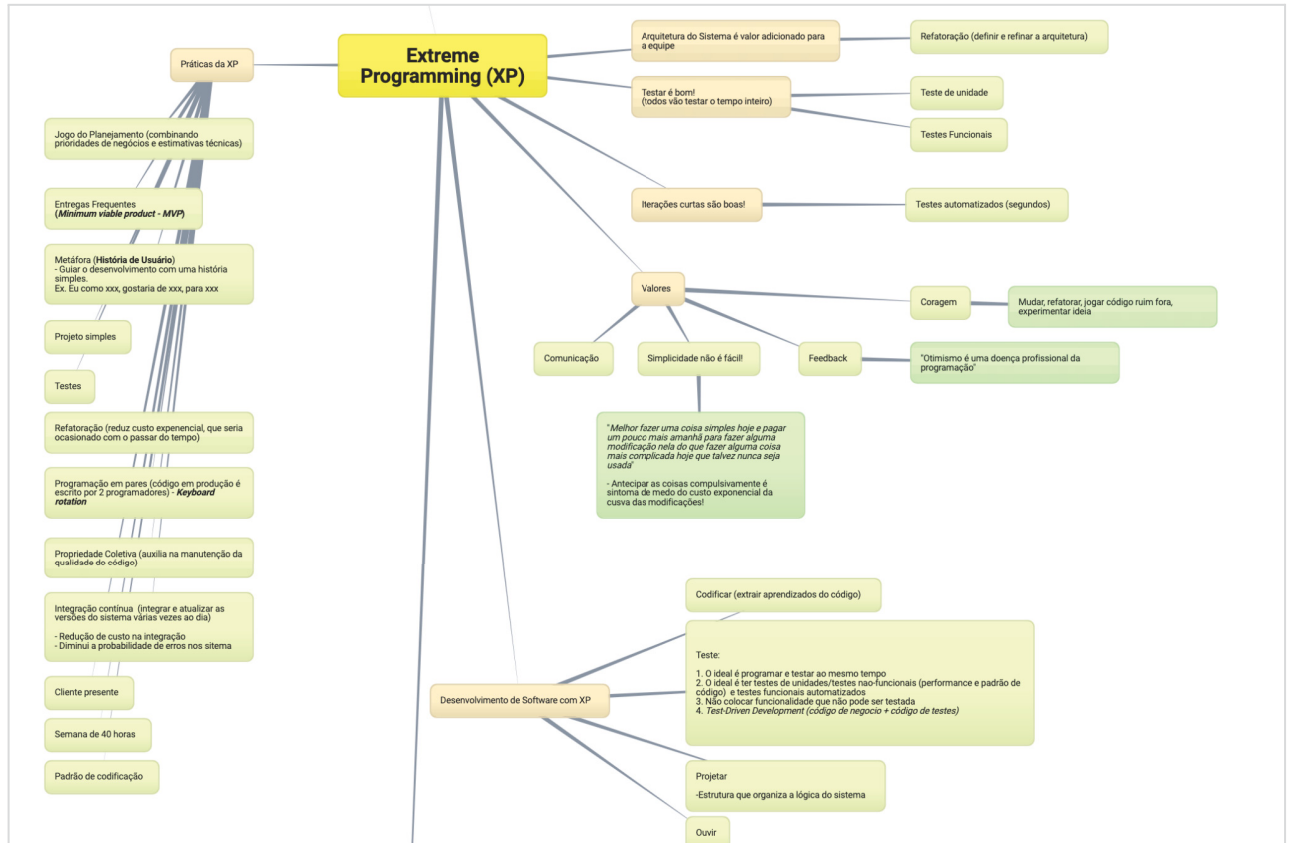
FONTE: Adaptado de Cohn, 2025.

FIGURA 3 – MAPA MENTAL EM MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE (MADS) - SCRUM



FONTE: Adaptado de Cohn, 2025.

FIGURA 4 – MAPA MENTAL EM MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE (MADS) - *EXTREME PROGRAMMING*



FONTE: Adaptado de Cohn, 2025.

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

Os projetos das disciplinas de Modelagem Ágil de Software I (Visão Funcional) e Modelagem Ágil de Software II (Visão Estrutural) são sequenciais e complementares, bem como foram projetados para fornecer uma compreensão da modelagem dentro do contexto de desenvolvimento ágil.

O projeto desta primeira disciplina focou em capturar e definir *o que* o sistema deve fazer do ponto de vista do usuário e do negócio, com o objetivo de garantir que a equipe de desenvolvimento construa o produto correto. Enquanto que o segundo projeto estruturou *como* detalhar a arquitetura interna e o comportamento do software para que os desenvolvedores possam construí-lo de forma coesa, escalável e de acordo com as funcionalidades definidas anteriormente, utilizando-se de diagramas UML que representam a estrutura e a dinâmica do sistema (Booch *et al*, 2006).

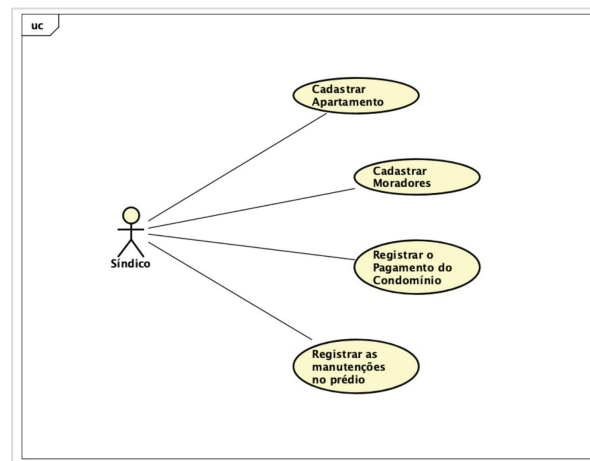
Dessa forma, as *Histórias de Usuário* e os *Casos de Uso* definidos em MAG1 foram o ponto de partida para a modelagem em MAG2. O primeiro projeto consistiu em modelar as funcionalidades de um *Sistema de Gestão de Condomínio* a partir da perspectiva do usuário. E, com base nas solicitações iniciais do síndico, como os cadastros de apartamentos, moradores, pagamentos e manutenções, a equipe desenvolveu os artefatos funcionais. Isso incluiu a criação de *Diagramas de Caso de Uso* (níveis 1 e 2) para visualizar as interações do síndico com o sistema e *Histórias de Usuário* completas, detalhando cada requisito com critérios de aceitação, protótipos de tela e cenários de teste.

Ao passo que, o segundo projeto, focou em como o sistema seria construído internamente para atender aos requisitos funcionais previamente definidos. A tarefa exigia a elaboração de artefatos que descrevem a arquitetura e o comportamento do software, como o *Diagrama de Classes*, para definir a estrutura de dados com entidades como Apartamento, Morador e Boleto, e os Diagramas de Sequência para cada História de Usuário, ilustrando a interação entre os diferentes componentes do sistema ao executar uma função.

Assim, a consolidação, integrando os artefatos da Visão Funcional com os diagramas da Visão Estrutural, proporcionam uma visão completa da modelagem do sistema.

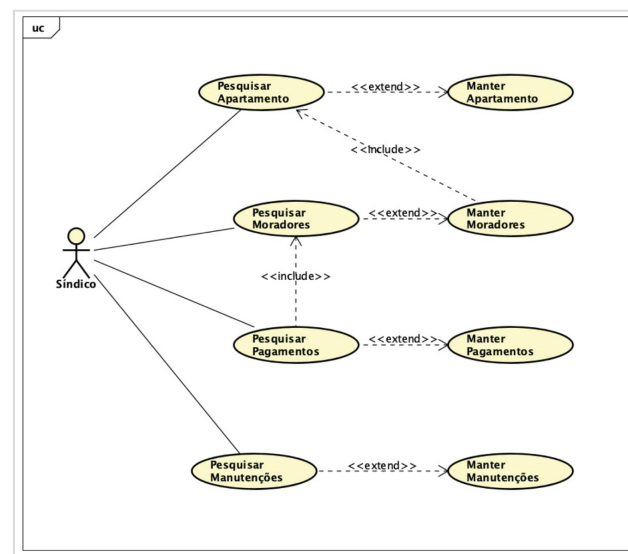
### 3.1 ARTEFATOS DO PROJETO

FIGURA 5 – MODELAGEM ÁGIL DE SOFTWARE (VISÃO FUNCIONAL) -  
DIAGRAMA DE CASO DE USO NÍVEL 1



FONTE: Autora (2025)

FIGURA 6 – MODELAGEM ÁGIL DE SOFTWARE (VISÃO FUNCIONAL) -  
DIAGRAMA DE CASO DE USO NÍVEL 2



FONTE: Autora (2025)

FIGURA 7 – MODELAGEM ÁGIL DE SOFTWARE (VISÃO FUNCIONAL) - DESENHO DA TELA

Consultar Apartamento

Tipo de Filtragem

Filtro

Filtrar

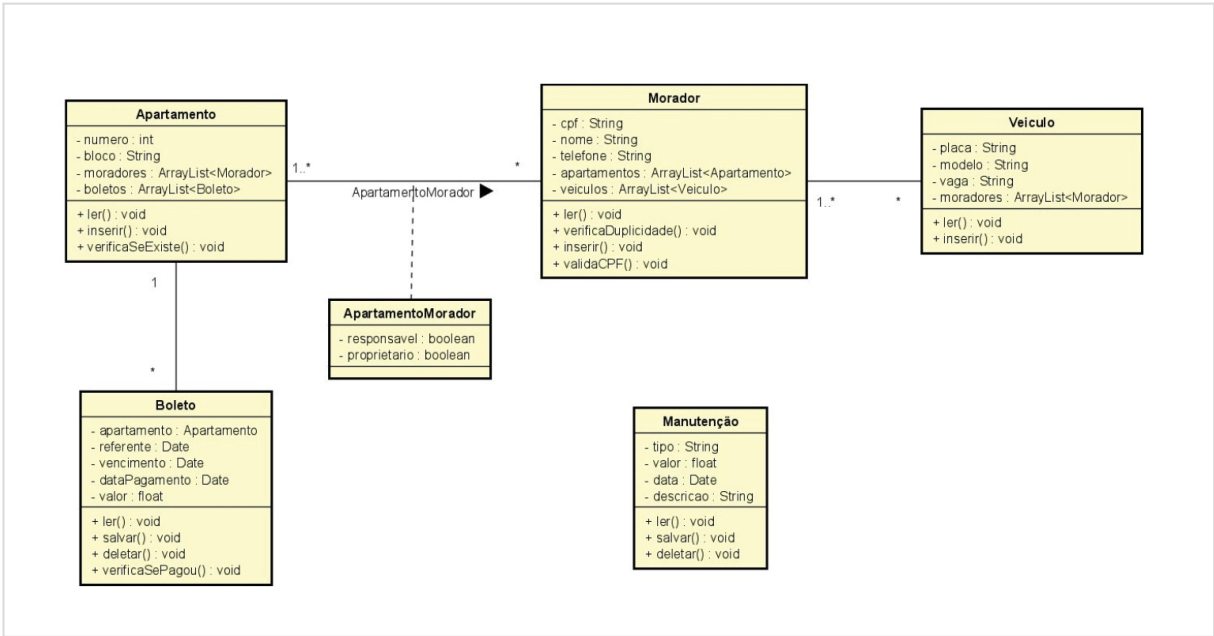
| Apartamento | CPF            | Morador       | Telefone       | Última Atualização |                                      |
|-------------|----------------|---------------|----------------|--------------------|--------------------------------------|
| 101         | xxx.xxx.xxx-00 | Fulano de Tal | (11) 0000-0000 | xx/xx/xxxx         | <div>Editar</div> <div>Deletar</div> |

Inserir Novo Apartamento

Retornar

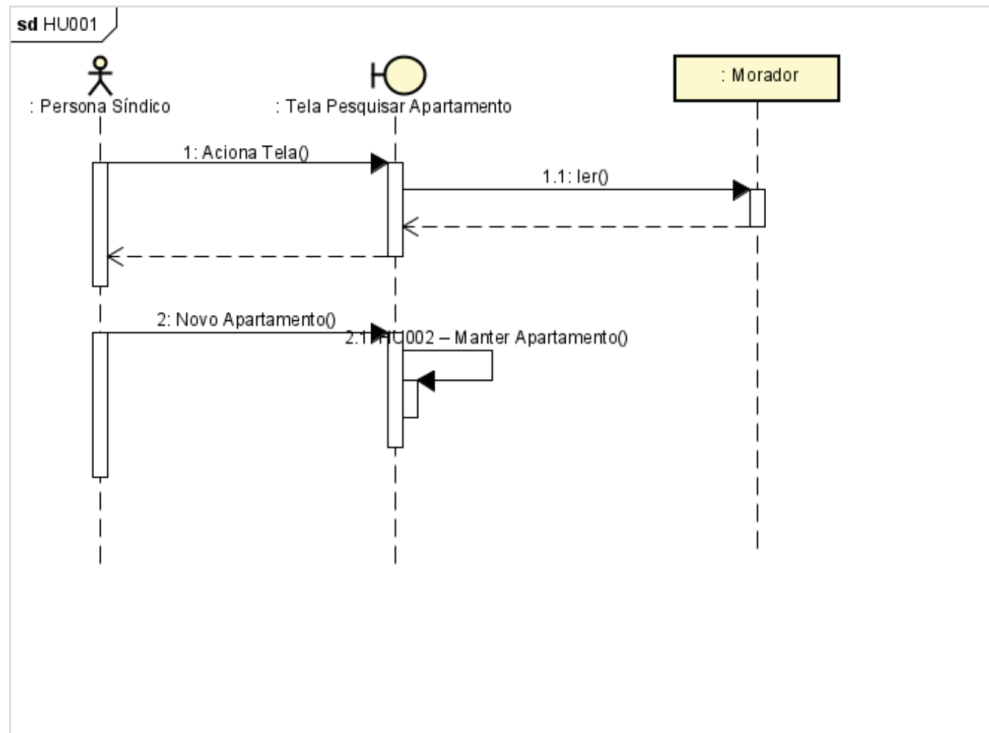
FONTE: Autora (2025)

FIGURA 8 – MODELAGEM ÁGIL DE SOFTWARE (VISÃO ESTRUTURAL) - DIAGRAMA DE CLASSES COM ATRIBUTOS ASSOCIADOS



FONTE: Autora (2025)

FIGURA 9 – MODELAGEM ÁGIL DE SOFTWARE (VISÃO ESTRUTURAL) -  
DIAGRAMA DE SEQUÊNCIAS PARA PESQUISAR APARTAMENTO



FONTE: Autora (2025)

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

As disciplinas de Gerenciamento Ágil de Projetos de Software I e II foram executadas de forma complementar entre si, resultando em projetos práticos que simulam o ciclo de vida completo de um projeto ágil. O objetivo foi capacitar o profissional a não apenas aplicar um *framework*, mas a gerir, medir e otimizar o fluxo de entrega de valor de um produto de software, conforme as bases dos Métodos Ágeis para o Software.

Inicialmente, o primeiro projeto, concentrou-se com base nos conceitos de concepção de produto de Design Thinking, Lean Inception e do *framework* Scrum para criar um plano de projeto completo (Pereira *et al*, 2007). A tarefa central foi executar um escopo de produto definido e desenvolver os artefatos essenciais para o planejamento: criar um *Mapa de Funcionalidades* para dar uma visão geral do produto, derivar e escrever Histórias de Usuário detalhadas para compor o *Product Backlog*, conduzir uma sessão simulada de *Planning Poker* para estimar o esforço e, por fim, elaborar um *Plano de Release* inicial, que agrupa as histórias em entregas de valor e define um cronograma macro do *Sistema de Gestão de Condomínios* (desenvolvidos em Modelagem Ágil de Software I e II).

No segundo projeto, utilizando-se dos conhecimentos sobre Kanban, métricas e gestão da qualidade, foram desenvolvidos os artefatos para gerir o fluxo de trabalho contínuo (Gonzalez, 2021). O projeto envolveu desenhar um Quadro Kanban customizado para a equipe, definindo colunas de fluxo e políticas definidas, estabelecer limites de *Work In Progress (WIP)* para otimizar a entrega, e definir um conjunto de métricas de fluxo como *Lead Time*, *Cycle Time* e *Throughput*, bem como relatórios visuais por meio do *Cumulative Flow Diagram* para monitorar a saúde do projeto.

Por fim, abordou-se a evolução do guia de gerenciamento de projetos do *Project Management Institute* (PMI), denominado *Guide to the Project Management Body of Knowledge* (PMBOK) (PMI, 2021). Destacando-se os conceitos da 6ª edição que é prescritiva e organizada em áreas de conhecimento e grupos de processos e avançou-se para a 7ª edição do PMBOK, representando uma mudança de paradigma, pois ela flexibiliza a abordagem processual por uma estrutura baseada

em 12 princípios de entrega de valor e 8 domínios de desempenho, sendo mais adaptável.

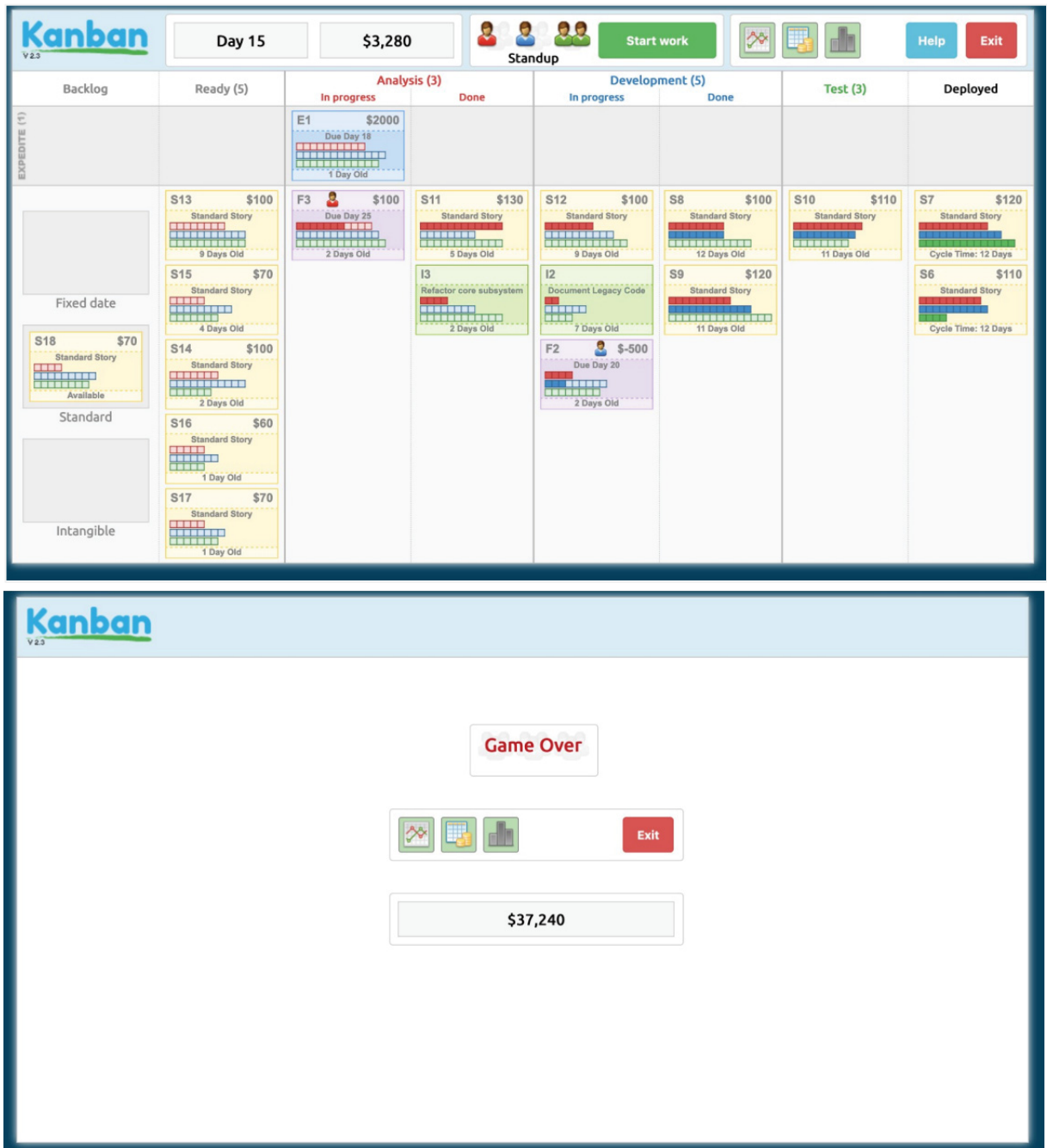
4.1 ARTEFATOS DO PROJETO

FIGURA 10 – GERENCIAMENTO ÁGIL DE PROJETOS I (PLANO DE *RELEASE*) - SISTEMA DE GESTÃO DE CONDOMÍNIOS

|                                                                                                                                                                                                   |                                                                                                                                                                                                       |                                                                                                                                                                                            |                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Gerenciamento Ágil de Projetos I<br>Profa. Dra. Rafaela Mantovani Fontana<br>Sistema de Gestão de Condomínio<br>Plano de Release                                                                  |                                                                                                                                                                                                       |                                                                                                                                                                                            |                                                                                                                                                                                                                                                                      |
| Cálculo da Velocidade:                                                                                                                                                                            |                                                                                                                                                                                                       |                                                                                                                                                                                            |                                                                                                                                                                                                                                                                      |
| Horas disponíveis por dia:                                                                                                                                                                        | 8 horas                                                                                                                                                                                               | Tamanho da Sprint:                                                                                                                                                                         | 2 semanas                                                                                                                                                                                                                                                            |
| Horas disponíveis por Sprint:                                                                                                                                                                     | 80 horas (8 horas/dia * 10 dias úteis por sprint)                                                                                                                                                     | Velocidade:                                                                                                                                                                                | 10 pontos por sprint (80 horas por sprint / 8 horas por ponto)                                                                                                                                                                                                       |
| Plano de Release:                                                                                                                                                                                 |                                                                                                                                                                                                       |                                                                                                                                                                                            |                                                                                                                                                                                                                                                                      |
| Iteração/Sprint 1                                                                                                                                                                                 | Iteração/Sprint 2                                                                                                                                                                                     | Iteração/Sprint 3                                                                                                                                                                          | Iteração/Sprint 4                                                                                                                                                                                                                                                    |
| Data Início:01/06/2024                                                                                                                                                                            | Data Início:17/06/2024                                                                                                                                                                                | Data Início:01/07/2024                                                                                                                                                                     | Data Início:15/07/2024                                                                                                                                                                                                                                               |
| Data Fim:14/06/2024                                                                                                                                                                               | Data Fim:30/06/2024                                                                                                                                                                                   | Data Fim: 14/07/2024                                                                                                                                                                       | Data Fim:28/07/2024                                                                                                                                                                                                                                                  |
| <HU001: <b>Pesquisar Apartamento</b> ><br>SENDO: O síndico<br>QUERO: Verificar os apartamentos cadastrados<br>PARA: Ter o registro de todos os apartamentos do condomínio<br>ESTIMATIVA: 3 pontos | <HU003: <b>Pesquisar Moradores</b> ><br>SENDO: O síndico<br>QUERO: Verificar os moradores cadastrados<br>PARA: Ter o registro de todos os moradores integrantes do condomínio<br>ESTIMATIVA: 2 pontos | <HU005: <b>Pesquisar Pagamentos</b> ><br>SENDO: O síndico<br>QUERO: Verificar os pagamentos efetuados<br>PARA: Ter o registro de todos os pagamentos do condomínio<br>ESTIMATIVA: 3 pontos | <HU007: <b>Pesquisar Manutenções</b> ><br>SENDO: O síndico<br>QUERO: Verificar as manutenções já realizadas no condomínio<br>PARA: Verificar todas as manutenções já realizadas no condomínio, assim como suas datas, para uma melhor gestão<br>ESTIMATIVA: 2 pontos |
| <HU002: <b>Manter Apartamento</b> ><br>SENDO: O síndico<br>QUERO: Registrar um novo apartamento no sistema<br>PARA: Poder vincular a um morador<br>ESTIMATIVA: 5 pontos                           | <HU004: <b>Manter Moradores</b> ><br>SENDO: O síndico<br>QUERO: Manter os dados dos moradores<br>PARA: Que seus dados fiquem atualizados no sistema do condomínio<br>ESTIMATIVA: 5 pontos             | <HU006: <b>Manter Pagamentos</b> ><br>SENDO: O síndico<br>QUERO: Registrar o pagamento dos condomínios<br>PARA: Estarem regularizados perante o condomínio<br>ESTIMATIVA: 5 pontos         | <HU008: <b>Manter Manutenção</b> ><br>SENDO: O síndico<br>QUERO: Manter os dados das manutenções atualizados<br>PARA: Melhorar o controle interno e facilitar na gestão dos gastos com despesas do condomínio<br>ESTIMATIVA: 5 pontos                                |

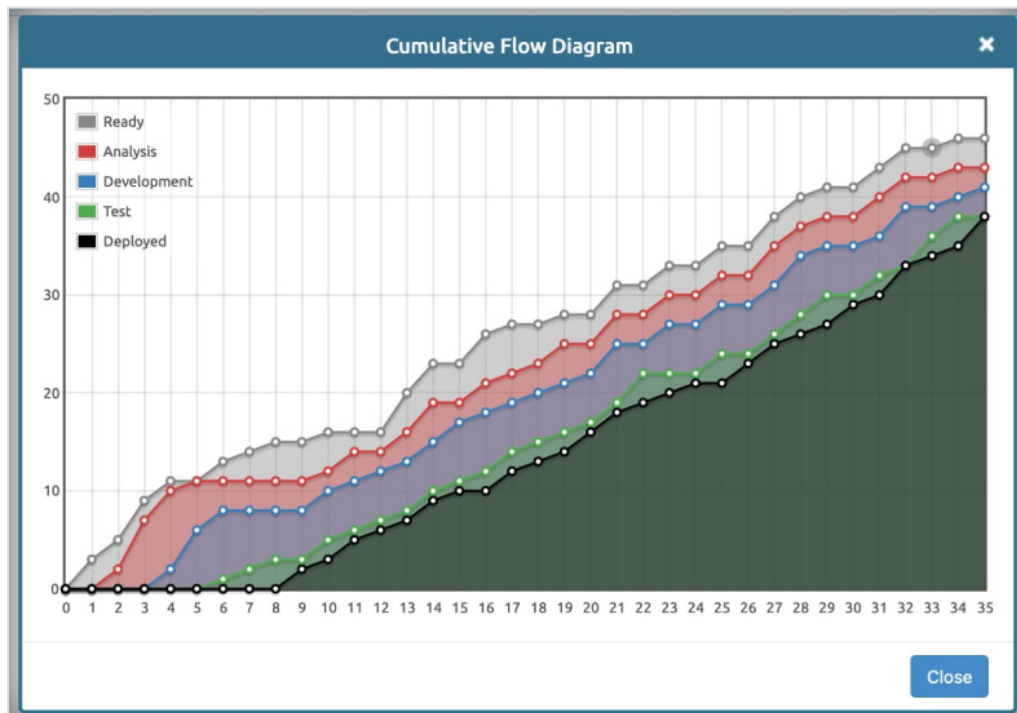
FONTE: Autora (2025)

FIGURA 11 – GERENCIAMENTO ÁGIL DE PROJETOS II - EXECUÇÃO DE UM CICLO NO KANBAN



FONTE: Autora (2025)

FIGURA 12 – GERENCIAMENTO ÁGIL DE PROJETOS II - CUMULATIVE FLOW DIAGRAM (CFD)



FONTE: Autora (2025)

## 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

O projeto prático da disciplina foi concebido como a principal ferramenta de consolidação do conhecimento adquirido. O desenvolvimento ágil, nesse momento, focou-se na construção de software funcional.

A disciplina introduz a linguagem de programação Java e a prática de Programação Orientada a Testes (TDD) (Matos, 2020) . Durante as atividades os *testes eram escritos antes da implementação do código*, orientando o desenvolvedor a pensar na especificação e nos resultados esperados e criando uma rede de segurança que permite refatorar e adicionar novas funcionalidades com confiança, sem quebrar o que já existe.

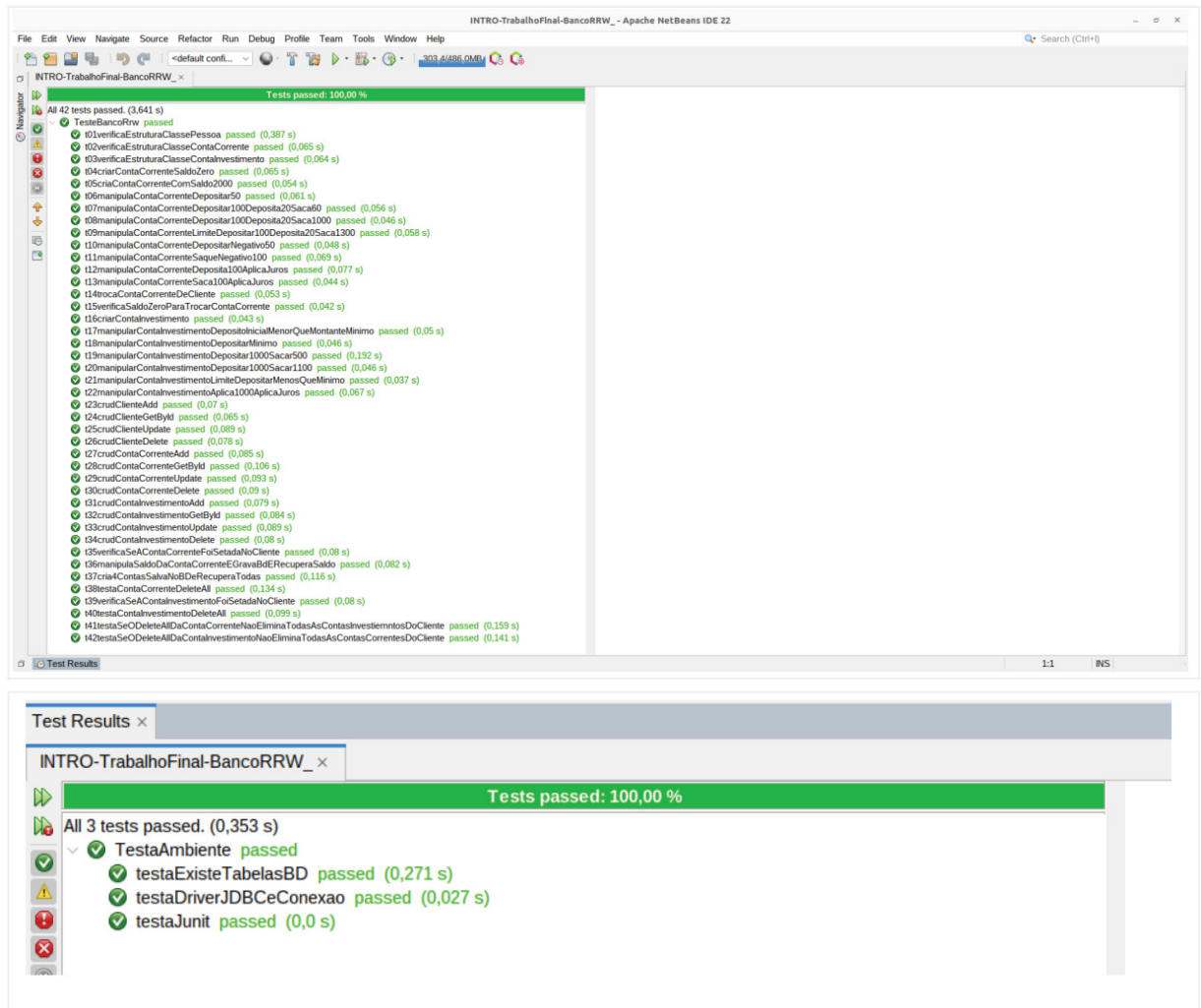
A disciplina, dividida em unidades denominadas sintaxe, orientação a objetos e banco de dados, permitiu que o projeto fosse construído de forma incremental. Destacando a abordagem relacionada aos *sprints* de Metodologias Ágil, onde o software evolui em ciclos curtos, agregando complexidade e funcionalidades a cada etapa. Logo, o projeto final da disciplina trata-se de um estudo de caso prático para a implementação do *backend* de um *Sistema Bancário Simplificado*, cujo escopo abrange o cadastro de clientes e a gestão de contas correntes e de investimento.

Além disso, para guiar o desenvolvimento e simular um ambiente profissional, foram fornecidos artefatos essenciais, como um diagrama de classes detalhado e um *script DDL* para a estrutura do banco de dados em *MySQL* (Heffelfinger, 2008, p. 117). Também foi disponibilizado um projeto NetBeans, que incluía uma suíte completa com 42 testes unitários e de integração. Dessa forma, desenvolveu-se a lógica de negócio e as camadas de acesso a dados, escrevendo o código Java necessário para que a suíte de testes pré-existente fosse executada com sucesso.

Nessa abordagem, os testes unitários foram criados antes da funcionalidade em si. Após a escrita do teste, desenvolveu-se a quantidade mínima de código de produção necessária para que ele fosse aprovado, seguida por uma etapa de refatoração, se preciso. Assim, a avaliação é diretamente atrelada à qualidade da implementação, medida pela quantidade de testes que passam simultaneamente, solidificando a compreensão de que software funcional e testado é o principal indicador de progresso (Fucci; Turhan, 2013).

## 5.1 ARTEFATOS DO PROJETO

FIGURA 13 – INTRODUÇÃO A PROGRAMAÇÃO - RELATÓRIO BANCORRW



FONTE: Autora (2025)

## 6 DISCIPLINA: BD – BANCO DE DADOS

A atividade prática da disciplina simulou um projeto de banco de dados completo, desde a fundamentação conceitual, passando pela modelagem lógica e relacional até a implementação e manipulação com a linguagem SQL . No projeto, cada *sprint* ou iteração demandou novas funcionalidades que, por sua vez, exigiram alterações no esquema do banco de dados.

O objetivo foi desenvolver um *Sistema de Controle de Biblioteca e de Estoque* utilizando técnicas de normalização e modelagem para criar um banco de dados que fosse ao mesmo tempo consistente e flexível, capaz de acomodar novas regras de negócio sem a necessidade de refatorações complexas e custosas.

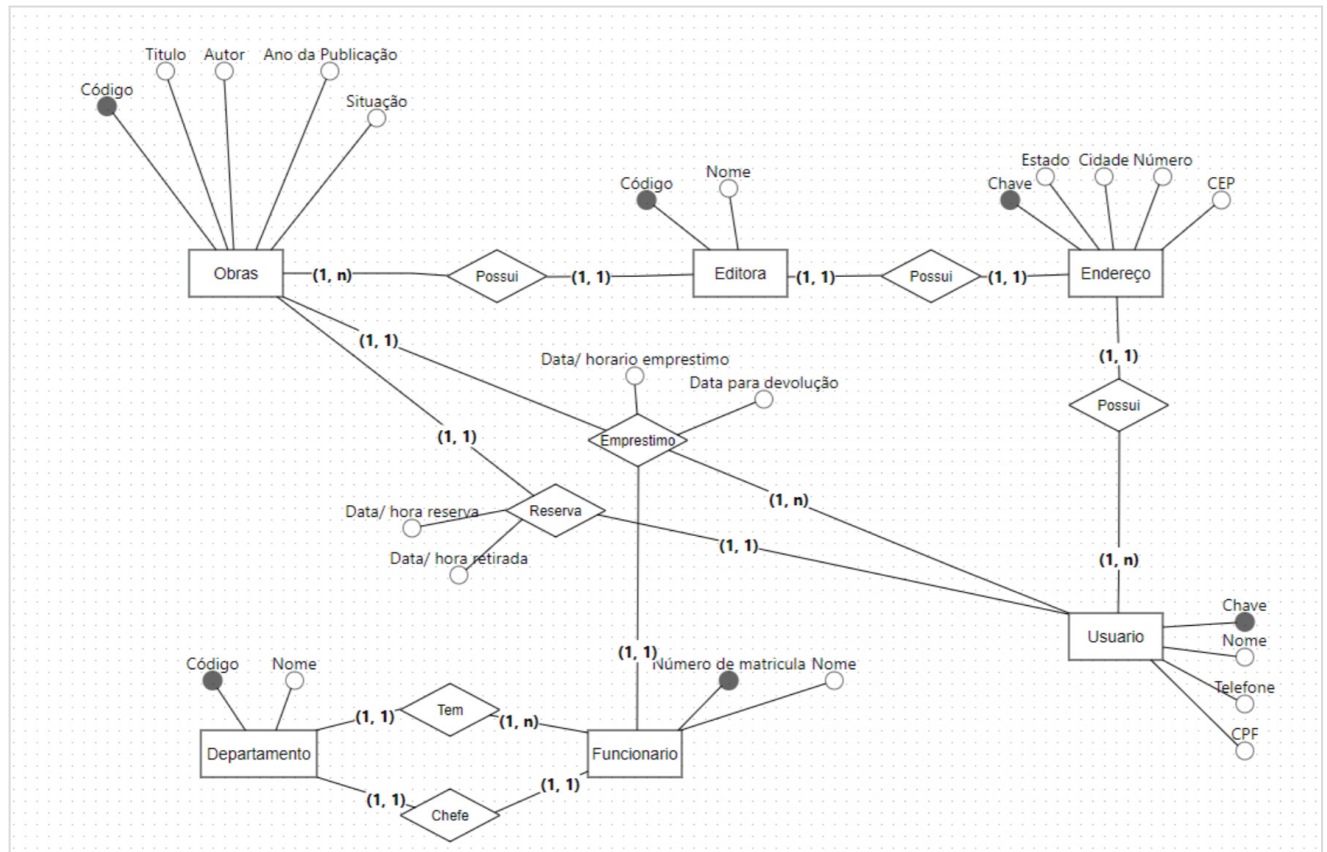
Para isso, a aplicação correta de restrições, chaves e o entendimento do modelo transacional (propriedades ACID) foram utilizadas para manter a integridade dos dados, um dos pilares da qualidade do software. Em um ambiente de entregas rápidas, entende-se que a automação de testes de banco de dados torna-se uma prática indispensável (Coutinho, 2025).

O modelo de dados serviu como uma linguagem comum entre desenvolvedores, analistas de negócio e *product owners*. Para isso, a modelagem foi documentada, como prática desenvolvida no projeto, facilitando a comunicação e garantindo que todos os envolvidos tenham uma compreensão compartilhada das regras de negócio.

Por fim, os modelos gerados, Entidade-Relacionamento Conceitual e o Lógico, funcionam como a planta baixa da arquitetura de dados da aplicação, pois eles são artefatos de comunicação. Antes de se desenvolver o código SQL, estes diagramas permitem que desenvolvedores, arquitetos e analistas de negócio visualizem e validem a estrutura, identificando falhas de lógica ou omissões nos requisitos.

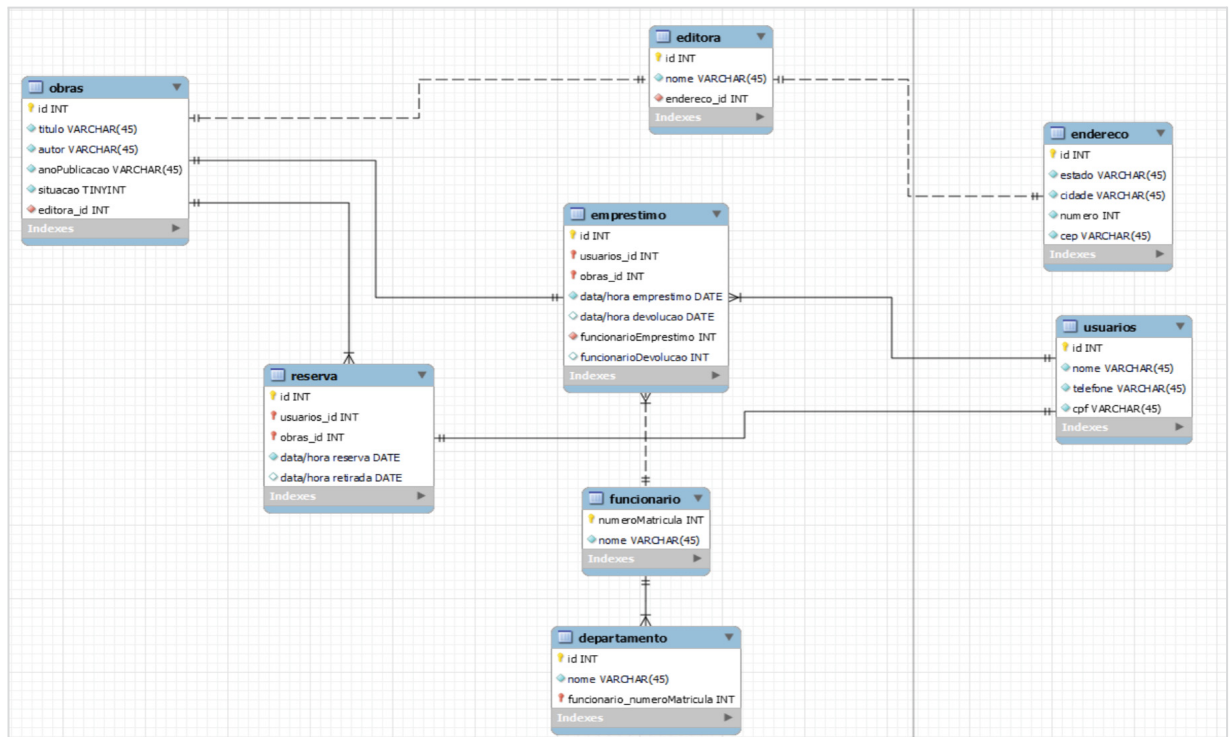
## 6.1 ARTEFATOS DO PROJETO

FIGURA 14 – MODELO ENTIDADE-RELACIONAMENTO CONCEITUAL FEITO NO BRMODELO (MODELAGEM DE DADOS)



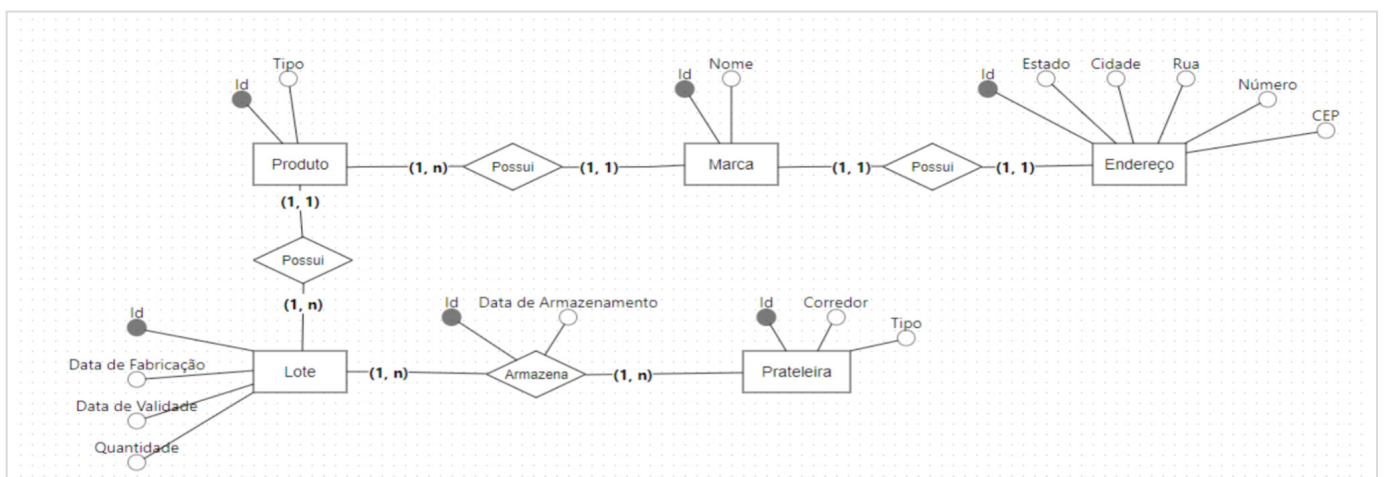
FONTE: Autora (2025)

FIGURA 15 – MODELO LÓGICO (MODELAGEM DE DADOS)



FONTE: Autora (2025)

FIGURA 16 – BANCO DE DADOS (IMPLEMENTAÇÃO EM SQL)



FONTE: Autora (2025)

FIGURA 17 – BANCO DE DADOS (SCRIPT GERADO PARA POPULAR O BANCO DE DADOS I)

| id   | nome            | endereco_id |
|------|-----------------|-------------|
| 1    | Eletrofast      | 3           |
| 2    | Móveis Sul      | 4           |
| 3    | Estamparia José | 2           |
| 4    | Cosméticos BR   | 5           |
| 5    | Roupas Legais   | 1           |
| NULL | NULL            | NULL        |

```

INSERT INTO `estoque`.`marca` (`nome`, `endereco_id`) VALUES ('Eletrofast', '3');
INSERT INTO `estoque`.`marca` (`nome`, `endereco_id`) VALUES ('Móveis Sul', '4');
INSERT INTO `estoque`.`marca` (`nome`, `endereco_id`) VALUES ('Estamparia José', '2');
INSERT INTO `estoque`.`marca` (`nome`, `endereco_id`) VALUES ('Cosméticos BR', '5');
INSERT INTO `estoque`.`marca` (`nome`, `endereco_id`) VALUES ('Roupas Legais', '1');

```

FONTE: Autora (2025)

FIGURA 18 – BANCO DE DADOS (SCRIPT GERADO PARA POPULAR O BANCO DE DADOS II)

| id | dataFabricacao | dataValidade | quantidade | produto_id |
|----|----------------|--------------|------------|------------|
| 1  | 2024-06-30     | 2024-12-31   | 100        | 1          |
| 2  | 2024-06-28     | 2024-12-31   | 50         | 2          |
| 3  | 2024-06-25     | 2024-12-31   | 75         | 3          |
| 4  | 2024-06-29     | 2024-12-31   | 120        | 1          |
| 5  | 2024-06-27     | 2024-12-31   | 80         | 2          |

```

-- Dados para lote
INSERT INTO estoque.lote (id, dataFabricacao, dataValidade, quantidade, produto_id)
VALUES
(1, '2024-06-30', '2024-12-31', 100, 1),
(2, '2024-06-28', '2024-12-31', 50, 2),
(3, '2024-06-25', '2024-12-31', 75, 3),
(4, '2024-06-29', '2024-12-31', 120, 1),
(5, '2024-06-27', '2024-12-31', 80, 2);

```

FONTE: Autora (2025)

## 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

A importância desta disciplina é central para a filosofia ágil, cujo objetivo é entregar software funcional e de alta qualidade de forma contínua. Conforme visto anteriormente, práticas ágeis não se resumem a cerimônias como *sprints* e *daily meetings*, pois elas dependem fundamentalmente de uma base de código saudável que permita mudanças rápidas e seguras.

Técnicas como Código Limpo e Refatoração garantem que o software não acumule débito técnico, tornando-o mais fácil de entender ou modificar, pois um código limpo e bem estruturado reduz o tempo gasto em depuração e permite que novos desenvolvedores se tornem produtivos mais rapidamente. Segundo Magalhães (2019, p. 199):

Um código limpo nada mais é que a aplicação de uma simples técnica para tornar fácil a compreensão de um código. A filosofia do código limpo tem como principal objetivo tornar as tarefas de leitura e escrita de códigos mais fáceis e intuitivas, mostrando sua real intenção.

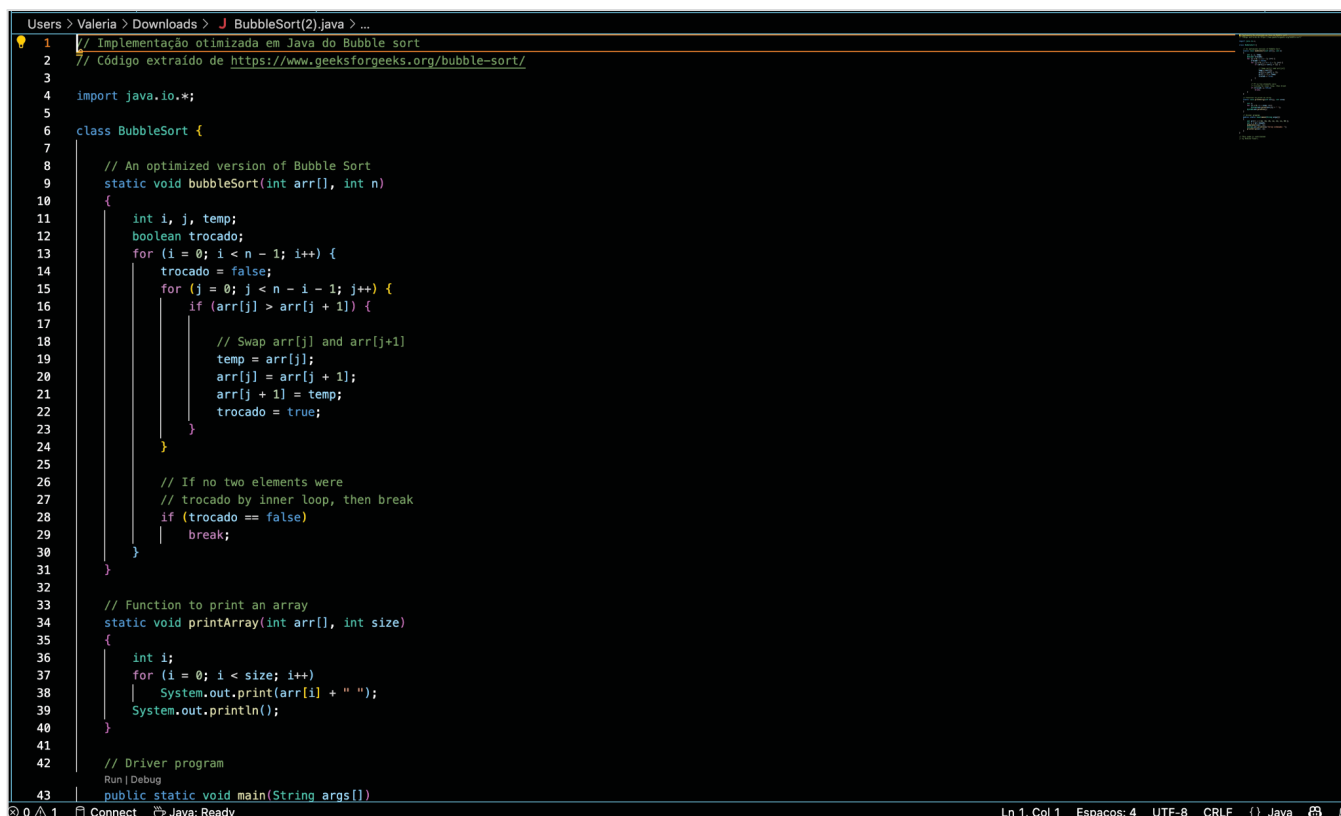
Nesse sentido, a arquitetura, assegura que o sistema seja robusto, escalável e resiliente a mudanças tecnológicas, desacoplando a lógica de negócio de detalhes de implementação. Em conjunto, essas práticas permitem que as equipes de desenvolvimento mantenham um ritmo de entrega constante e sustentável, que é a promessa central do desenvolvimento ágil.

Para garantir a qualidade e a longevidade de um software, é fundamental adotar boas práticas de programação que visam a manutenção e eficiência do código, como: a definição de nomes precisos, evitar a repetição de código, formatação consistente, comentários devem ser usados com moderação e a prática contínua da refatoração é essencial para aprimorar a estrutura interna do código sem alterar seu comportamento externo (Magalhães, 2019, p. 200).

Por fim, o trabalho final da disciplina consistiu em uma tarefa prática de refatoração, onde era necessário aplicar os princípios de Código Limpo para melhorar um algoritmo existente. Dessa forma, por meio de um código-fonte fornecido, foi possível aplicar um processo de refatoração para torná-lo um código limpo e para isso foi necessário a identificação de *Code Smells*, Aplicação de Melhorias, Testes e Relatório Final.

## 7.1 ARTEFATOS DO PROJETO

FIGURA 19 – ASPECTOS ÁGEIS DE PROGRAMAÇÃO (CÓDIGO BUBBLE SORT)



```
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
8     // An optimized version of Bubble Sort
9     static void bubbleSort(int arr[], int n)
10    {
11        int i, j, temp;
12        boolean trocado;
13        for (i = 0; i < n - 1; i++) {
14            trocado = false;
15            for (j = 0; j < n - i - 1; j++) {
16                if (arr[j] > arr[j + 1]) {
17                    // Swap arr[j] and arr[j+1]
18                    temp = arr[j];
19                    arr[j] = arr[j + 1];
20                    arr[j + 1] = temp;
21                    trocado = true;
22                }
23            }
24
25            // If no two elements were
26            // trocado by inner loop, then break
27            if (trocado == false)
28                break;
29        }
30    }
31
32    // Function to print an array
33    static void printArray(int arr[], int size)
34    {
35        int i;
36        for (i = 0; i < size; i++)
37            System.out.print(arr[i] + " ");
38        System.out.println();
39    }
40
41    // Driver program
42    public static void main(String args[])
43    {
44    }
```

FONTE: Autora (2025)

FIGURA 20 – APLICANDO CLEAN CODE NO ALGORITMO BUBBLE SORT

### Nomes Significativos

- Alterado **arr**, **n**, **temp**, **trocado** para **listaDeInteiros**, **total**, **valorTemporario** **trocouPosicao**.
- Alterada a função **bubbleSort** para **ordenar**.
- Alterada a função **printArray** para **imprimirLista**.

### Funções Longas

- A lógica da função **bubbleSort** (**ordenar**) foi modificada e dividida em subfunções, reduzindo a complexidade ciclomática e facilitando a leitura.

### Comentários Redundantes

- Comentários redundantes foram removidos. Muitos comentários redundantes indicam que o código não é suficientemente autoexplicativo.

### Testes

- Adicionado testes de unidade

FONTE: Autora (2025)

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

O projeto central das disciplinas de Desenvolvimento Web I e II foram concebidos de forma incremental que simula o ciclo de vida completo de um produto de software em um ambiente ágil. O objetivo foi a construção de uma aplicação web *full-stack*, desde sua concepção inicial no *front-end* até a completa funcionalidade com um *back-end*.

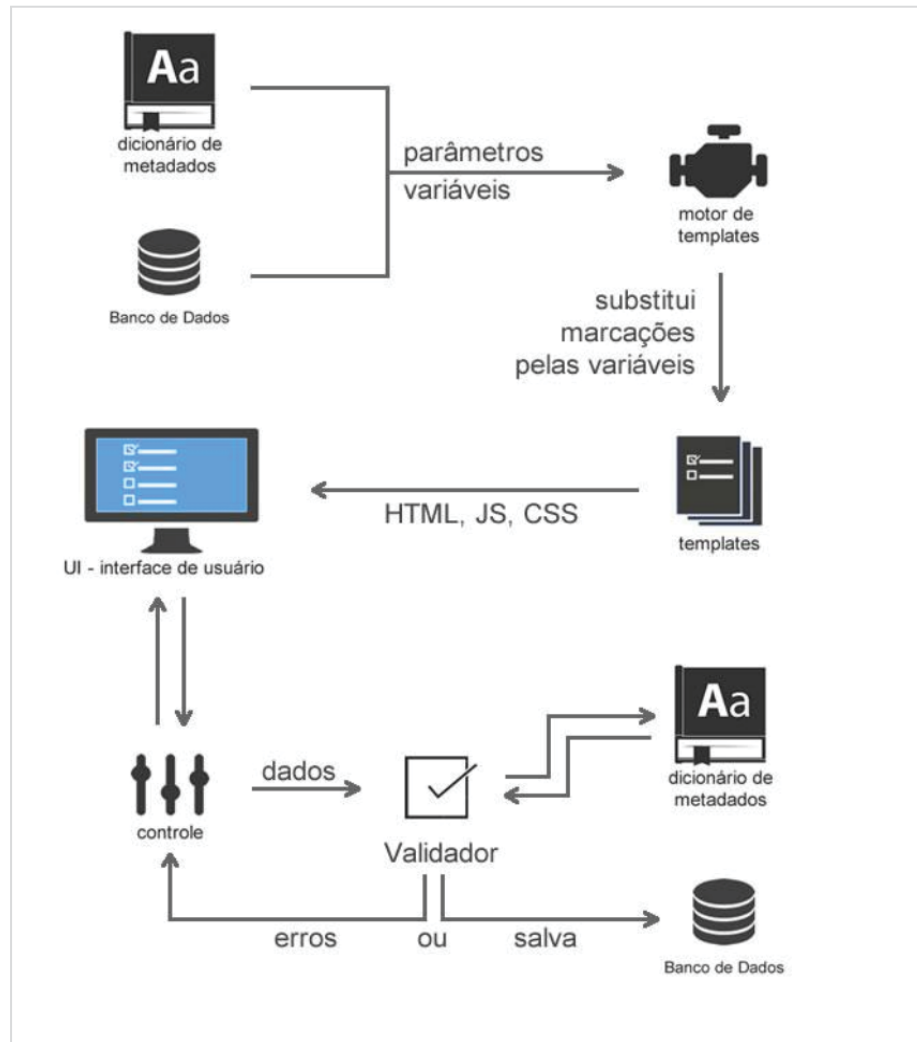
Inicialmente, em Desenvolvimento Web I, o foco é a interface e a experiência do usuário, construindo a interface cliente da aplicação. Partindo dos fundamentos de *HTML*, *TypeScript* e orientação a objetos, o projeto evolui para a criação de uma aplicação com funcionalidades de CRUD (*Create, Read, Update, Delete*).

Posteriormente, em Desenvolvimento Web II, o objetivo foi a construção do lado servidor utilizando tecnologias Java e o *framework* Spring (Zhang *et al*, 2021). Nesta etapa, desenvolveu-se construir uma API RESTful, que serviu como ponte de comunicação com a interface. Tópicos como programação reativa, autenticação e a criação de endpoints para gerenciar os dados foram abordados, resultando na integração é o total: a interface passa a consumir a API, e os dados são persistidos em um banco de dados, completando a arquitetura cliente-servidor.

Por fim, a metodologia do projeto foi intrinsecamente ligada aos princípios do desenvolvimento ágil, pois sua natureza iterativa e incremental espelha o conceito de *sprints*, onde a cada etapa uma nova porção de valor é adicionada ao produto. Isso reforça o foco na entrega de software em funcionamento ao permitir avaliações constantes. Além disso, a separação entre a interface do usuário e a lógica do servidor, por meio de uma API definida, reflete a arquitetura e a forma de colaboração de equipes ágeis modernas.

## 8.1 ARTEFATOS DO PROJETO

FIGURA 21 – FLUXO DE GERAÇÃO DA UI PARA OPERAÇÕES CRUD



FONTE: Bauer (2018, p. 25)

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

O projeto central da disciplina UX no Desenvolvimento Ágil de Software foi desenvolvido para imersão prática nas etapas fundamentais do design de experiência do usuário (UX). Inicialmente realizou-se a concepção, prototipagem e validação de uma solução digital, simulando um ciclo de desenvolvimento de produto em escala reduzida.

Essa abordagem metodológica, que transita da abstração à concretude, espelha modelos consolidados para o desenvolvimento de experiência do usuário, como os cinco planos propostos por Garrett (2011): Estratégia, Escopo, Estrutura, Esqueleto e Superfície. Dessa forma, a equipe refina desde a arquitetura da informação até o design sensorial final, garantindo que cada decisão seja validada antes de se tornar um custo de desenvolvimento (Souza, 2016).

Diante disso, a importância da disciplina manifesta-se no foco no usuário como princípio ágil, estabelecendo o alinhamento com as necessidades reais desde o início e garantindo que o desenvolvimento seja guiado por validação, o que evita o desperdício de recursos.

Compreende-se que a prototipagem como ferramenta de iteração rápida, permite que a equipe visualize e interaja com a solução muito antes de o código ser escrito, facilitando a identificação de problemas de forma precoce e eficiente. Além disso, a validação contínua para mitigação de riscos, onde a coleta de avaliações com um usuário real se torna uma estratégia para orientar as próximas iterações e reduzir significativamente os riscos de rejeição do produto pelo mercado.

Como proposta de projeto na disciplina, desenvolveu-se no Figma (Mustika, 2024), um protótipo de *Adoção de Animais*, sendo desenvolvidas 08 (oito) telas e uma pesquisa com o usuário final, com aplicação de testes e análises comportamentais durante o uso do aplicativo.

Por fim, o projeto da disciplina é uma simulação estratégica que demonstra, na prática, que o UX não é apenas uma fase, mas uma mentalidade contínua que deve estar presente durante todo o ciclo de vida do produto.

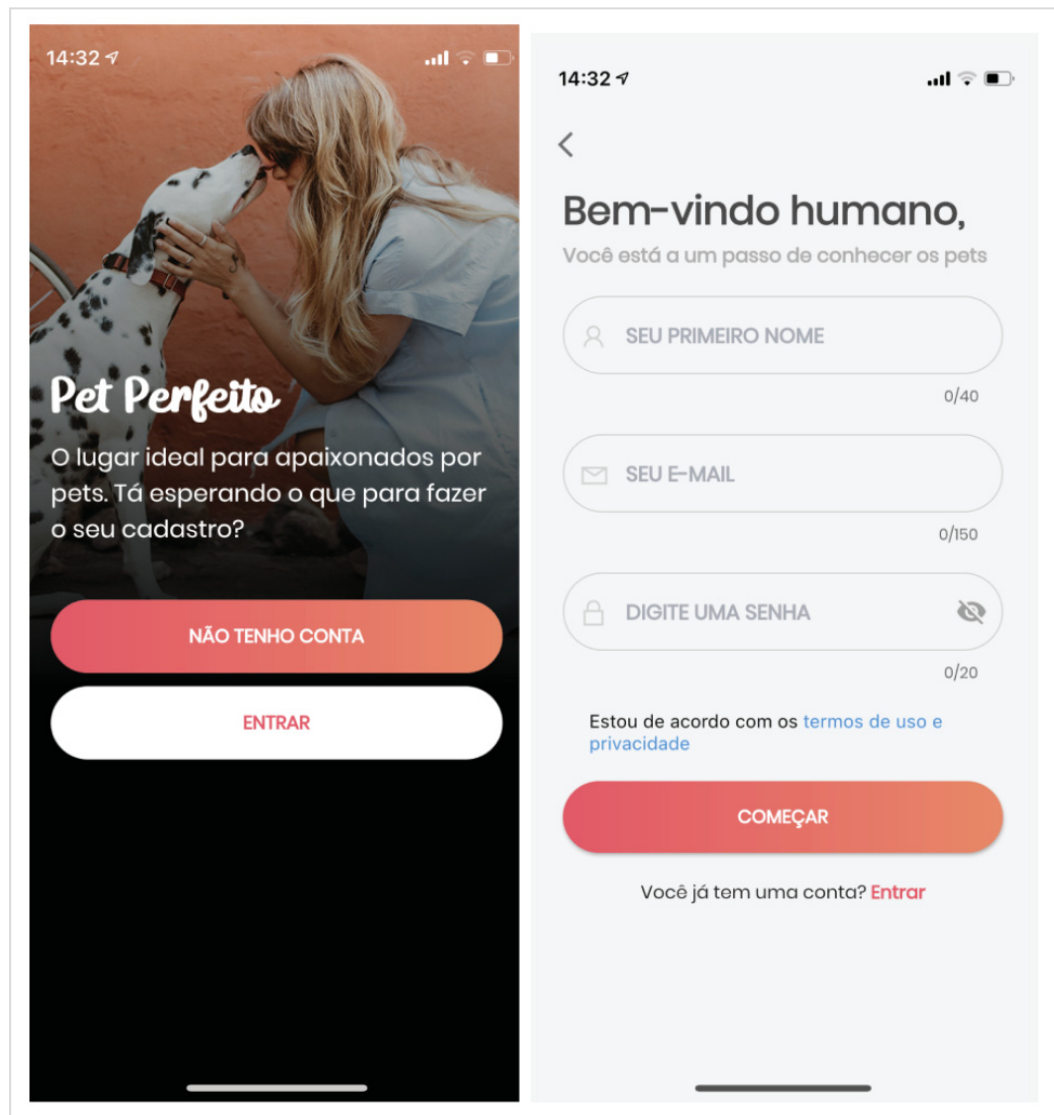
## 9.1 ARTEFATOS DO PROJETO

FIGURA 22 – UX DESIGN (TELAS DE BOAS VINDAS)



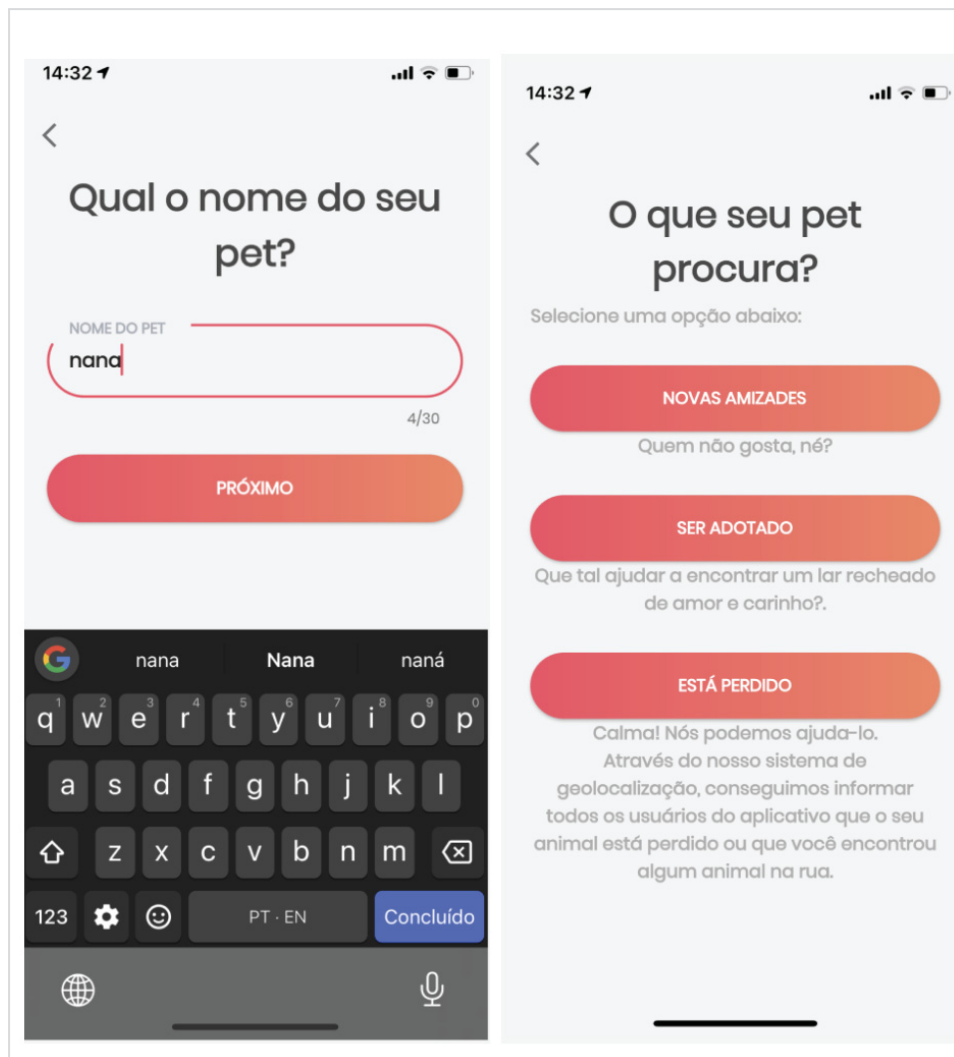
FONTE: Autora (2025)

FIGURA 23 – UX DESIGN (TELAS DE REGISTRO)



FONTE: Autora (2025)

FIGURA 24 – UX DESIGN (TELAS DE PESQUISA E FILTROS)



FONTE: Autora (2025)

FIGURA 25 – UX DESIGN (APLICAÇÃO DE TESTE DE USUÁRIO)

**Teste de 01 Usuário****Feedback:****Tela 1: Telas de Boas-Vindas**

- Na primeira tela, enquanto esperava pelo carregamento, o usuário gostaria que a cabecinha do cachorrinho balanceasse, porque se não tivesse animação ficaria clicando sem parar.
- O usuário demonstrou que faria a leitura de cada uma das telas iniciais, caso tivesse tempo disponível, pois achou as informações enxutas e por isso não foi cansativo. Além disso, compreendeu exatamente o número de telas que teria à frente com o recurso da barra de navegação. Caso contrário, utilizaria o recurso de pular para a tela de cadastro ao apertar o botão “vamos lá”.
- Após a quarta tela, o usuário espera ter acesso a tela de login.

**Tela 2: Tela de Login/Registro**

- O usuário demonstrou não ler a mensagem inicial da tela, concentrando-se em clicar no botão “não tenho conta”.
- O usuário preencheu com facilidade a tela de cadastro, mas indicou que gostaria de ter a opção de login com o uso da conta do Google para ter mais agilidade.

**Tela 3: Tela de Pesquisa e Filtros**

- O usuário indicou que o terceiro botão, “está perdido”, contém muito texto, o que leva a não ler completamente o conteúdo abaixo do botão.

**Tela 4: Tela de Perfil do Pet**

- O usuário navegou nas telas com facilidade demonstrando compreender as funcionalidades e botões disponibilizados.
- O usuário apresentou interesse em interagir com perfil do pet “Boss”, respondendo a mensagem postada: “passeio no parque com o papai ❤️❤️”, com a seguinte interação: “Até que horas você vai ficar no parque ? Estou pensando em ir para nos encontrar ☺”.

FONTE: Autora (2025)

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

A importância do projeto no contexto do Desenvolvimento Mobile I e II refere-se ao desenvolvimento de um Produto Mínimo Viável (MVP), cujo ciclo contínuo de aprendizado e aplicação não apenas reforça o conhecimento, mas também ensina a gerenciar a complexidade de forma crescente.

Nessa perspectiva, o trabalho propicia a aplicação de metodologias de gestão ágil, como Scrum ou Kanban, em seu próprio fluxo de trabalho, criando um backlog de funcionalidades e organizando o desenvolvimento (Ferreira, 2023, p. 22). Bem como, a construção das telas e do fluxo de navegação considerando os princípios de experiência do usuário.

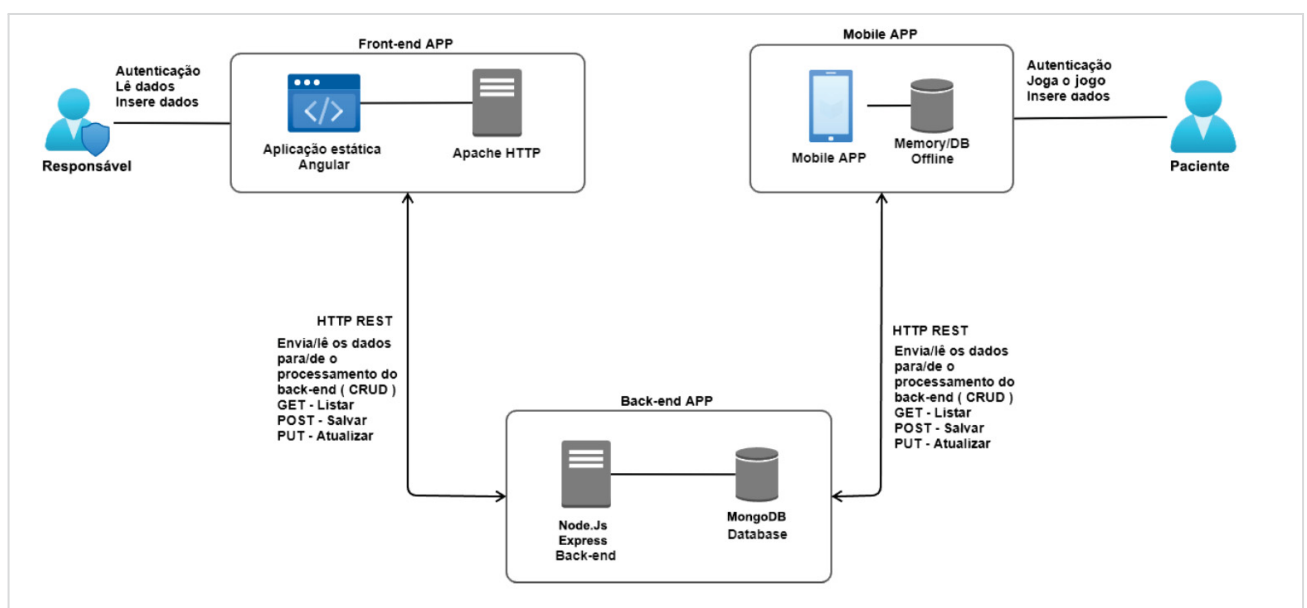
Para a disciplina de Desenvolvimento Mobile I, o projeto consistiu na criação de um aplicativo de finanças chamado FinApp, cujo objetivo foi desenvolver um aplicativo para uma startup fictícia que permitiria ao usuário registrar seus gastos (débitos) e ganhos (créditos) financeiros.

O projeto da disciplina de Desenvolvimento Mobile II foca na aplicação de conhecimentos sobre corrotinas e integração de sistemas através do consumo da HP-API, uma API temática do universo Harry Potter. O objetivo foi criar um aplicativo com uma tela principal que funcione como um dashboard e o código final precisa ser disponibilizado em um repositório do GitHub (Destro, 2029, p. 03).

A HP-API ([hp-api.onrender.com](https://hp-api.onrender.com)), utilizada como referência no segundo projeto, funciona como um serviço web público e gratuito que disponibiliza dados estruturados, em formato JSON (HP-API, 2025). A ideia central deste tipo de API é permitir que desenvolvedores se concentrem na prática de consumir dados de um *web service* sem a complexidade de gerenciar autenticação ou um banco de dados próprio, conforme exemplificado na Figura 26.

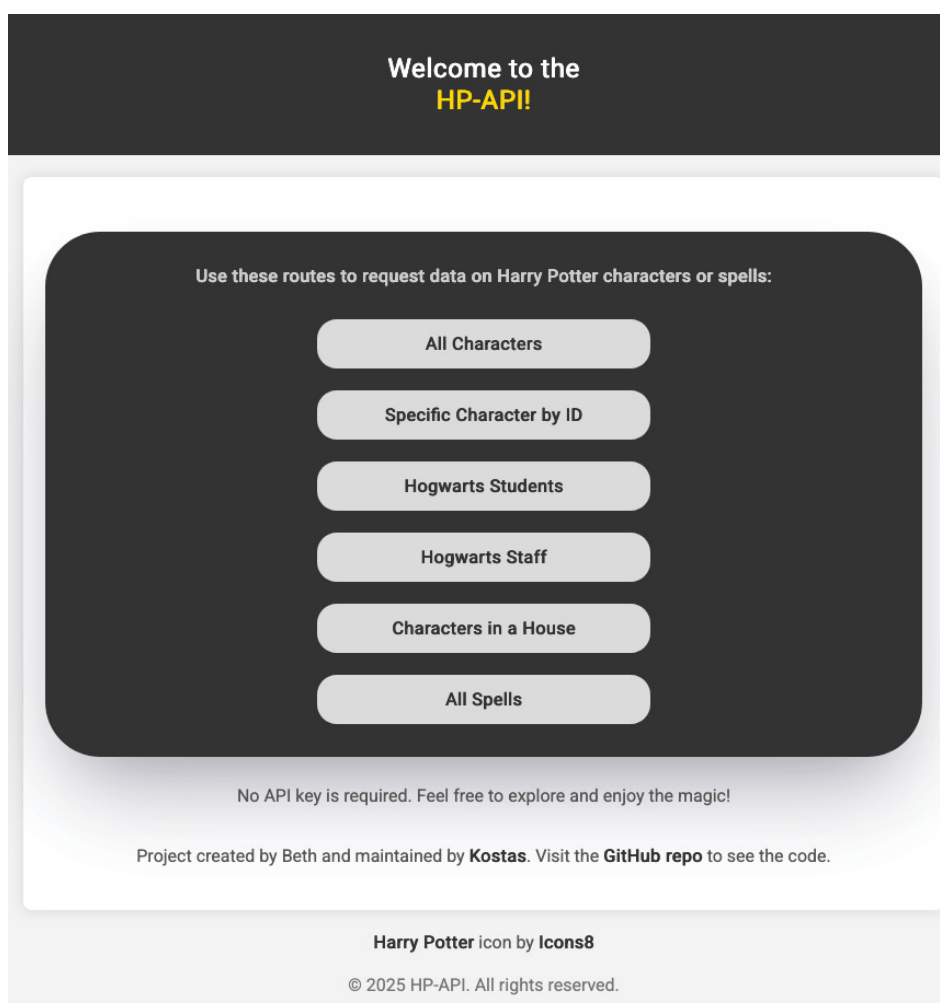
## 10.1 ARTEFATOS DO PROJETO

FIGURA 26 - ESTRUTURA DE COMUNICAÇÃO ENTRE APLICATIVO MÓVEL, API DE BACKEND E PÁGINA WEB



FONTE: Pavanelli (2023, p. 35)

FIGURA 27 – DESENVOLVIMENTO MOBILE (HP-API)



FONTE: <https://hp-api.onrender.com/>

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

O projeto da disciplina de Infraestrutura para Desenvolvimento e Implantação de Software (DevOps) representa a aplicação prática e consolidada de todos os conceitos teóricos abordados ao longo das unidades, como SDLC, Git, Docker, Pipelines de CI/CD, Kubernetes e Observabilidade (Maia, 2024, p. 26-29). Cujo objetivo foi simular um ambiente de desenvolvimento.

No contexto prático da disciplina, o projeto consistiu na configuração do ambiente local, utilizando Docker para baixar e executar uma imagem pré-configurada que contém GitLab e Jenkins (Smart, 2011, p. 9). A entrega do projeto é documentada através de *screenshots* que comprovam a execução bem-sucedida de cada um desses passos, desde o comando de inicialização do container até o log do Git confirmando as alterações.

A metodologia ágil entrega de valor de forma contínua, rápida e iterativa, e o projeto desta disciplina é fundamental para materializar esses princípios, pois aborda os pilares técnicos que viabilizam a agilidade em escala. Primeiramente, no que tange à velocidade e automação, o epicentro do projeto está na construção de *pipelines* de integração e entrega contínua (CI/CD).

Ao automatizar os processos de build, teste e implantação, eliminamos gargalos manuais e demorados, o que permite que as equipes de desenvolvimento entreguem novas funcionalidades e correções aos usuários de forma rápida e segura, alinhando-se diretamente ao princípio ágil de entregar software funcional com frequência.

Por fim, o projeto promove a colaboração e a cultura DevOps, pois as ferramentas como o Git são a base para a colaboração em código, mas o projeto vai além ao utilizar Docker para empacotar a aplicação e Kubernetes para orquestrá-la, quebrando a barreira entre desenvolvimento (Dev) e operações (Ops). Através da containerização com Docker, o projeto garante a consistência dos ambientes, eliminando o clássico problema "na minha máquina funciona", enquanto a automação de testes na *pipeline* assegura que cada alteração seja validada, aumentando a confiança nas entregas.

## 11.1 ARTEFATOS DO PROJETO

FIGURA 28 – RODANDO CONTAINER DOCKER



FONTE: Autora (2025)

FIGURA 29 – PÁGINA DO GITLAB (TESTE COMMIT)

Informações do container 40001016398E1

```
~/Development/src/valeria (0.022s)
docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
0178770bb36f   dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       37 minutes ago Up 37 minutes (healthy) 0.0.0.0:22->22/tcp, [::]:22->22/tcp, 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp, 0.0.0.0:9091->9091/tcp, [::]:9091->9091/tcp 40001016398E1
```

Gitlab

Página do gitlab

Menu

Monitoring

Project information

Repository

Issues

Merge requests

CI/CD

Security & Compliance

Deployments

Packages & Registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

You can't push or pull repositories using SSH until you add an SSH key to your profile.

The Auto DevOps pipeline has been enabled and will be used if no alternative CI configuration file is found. Container registry is not enabled on this GitLab instance. Ask an administrator to enable it in order for Auto DevOps to work.

GitLab Instance > Monitoring

M Monitoring

Project ID: 1

1 Commit 1 Branch 0 Tags 72 KB Project Storage

This project is automatically generated and helps monitor this GitLab instance. Learn more.

main Monitoring / Find file Web IDE Clone

Teste commit Administrator authored 1 minute ago 4521c2a5

Auto DevOps enabled Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Configure Integrations

| Name     | Last commit  | Last update  |
|----------|--------------|--------------|
| test.txt | Teste commit | 1 minute ago |

FONTE: Autora (2025)

## 12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

O projeto da disciplina de Testes Automatizados, consiste na aplicação prática de técnicas e ferramentas para a criação de suítes de testes unitários e de interface de usuário.

O objetivo foi construir uma rede de segurança para garantir a qualidade e a estabilidade da aplicação. Para isso, foram utilizadas tecnologias como JUnit, para testes de unidade na lógica do servidor, e Playwright, para automação das interações na interface do usuário (Truong, 2024, p. 7-10).

Conforme reiterado, durante cada disciplina do curso, o desenvolvimento ágil de software preza por ciclos de entrega curtos, com avaliações rápidas e melhoria contínua. Contudo, sem uma estratégia de automação de testes, esses objetivos podem ser desafiadores. Nesse sentido, segundo Oliveira (2023, p.18):

As ferramentas de automação de testes têm um papel essencial na criação e execução automatizada de testes, reduzindo a intervenção manual repetitiva. Elas oferecem uma interface gráfica amigável e recursos poderosos para gravar, reproduzir ações do usuário, validar resultados esperados e reportar resultados de teste. Além disso, permitem a criação de planos de teste, gerenciamento de casos de teste, rastreamento de defeitos e geração de relatórios de status e cobertura de teste.

Dessa forma, os testes abrangentes oferecem segurança para refatoração e evolução, dando à equipe a possibilidade de realizar melhorias e adaptar a arquitetura do software.

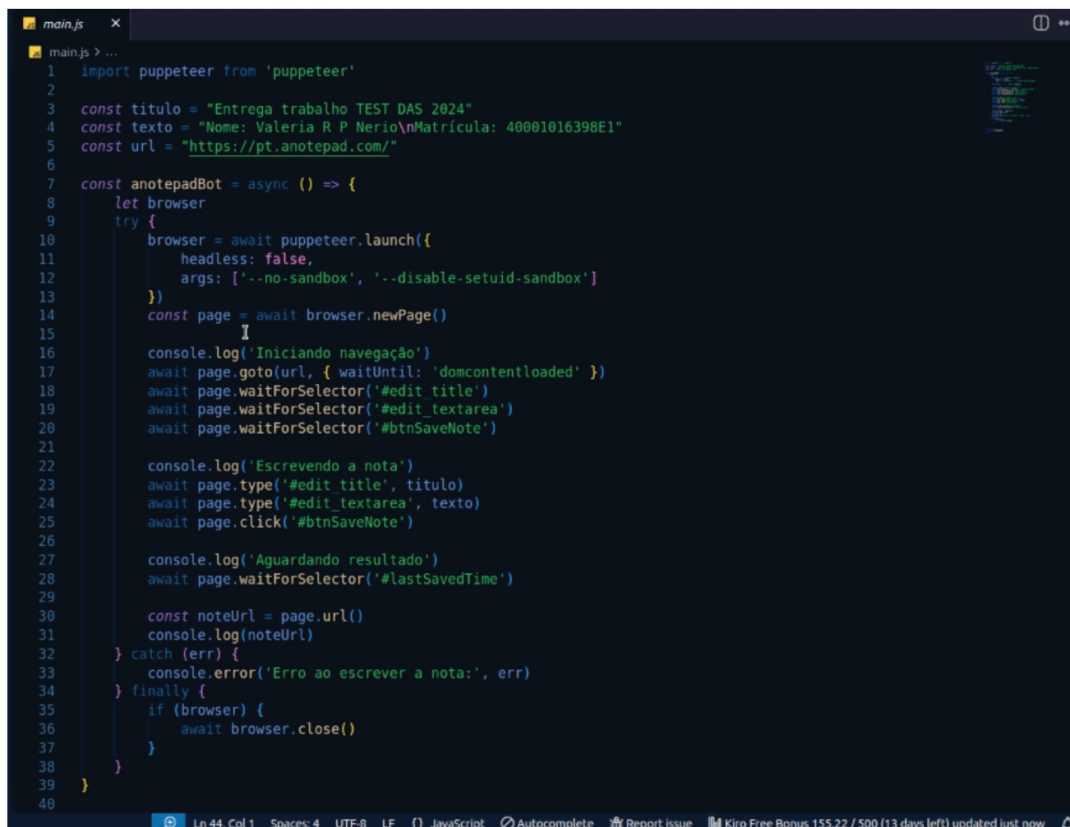
Portanto, destaca-se, ainda, que o propósito da *Garantia da Qualidade de Software* é atuar como um elo de visibilidade para a gerência, informando sobre a execução do processo de desenvolvimento e a situação dos produtos de software (Marczak, 2003).

## 12.1 ARTEFATOS DO PROJETO

FIGURA 30 – EXECUÇÃO DO CÓDIGO (NODEJS)

### Execução:

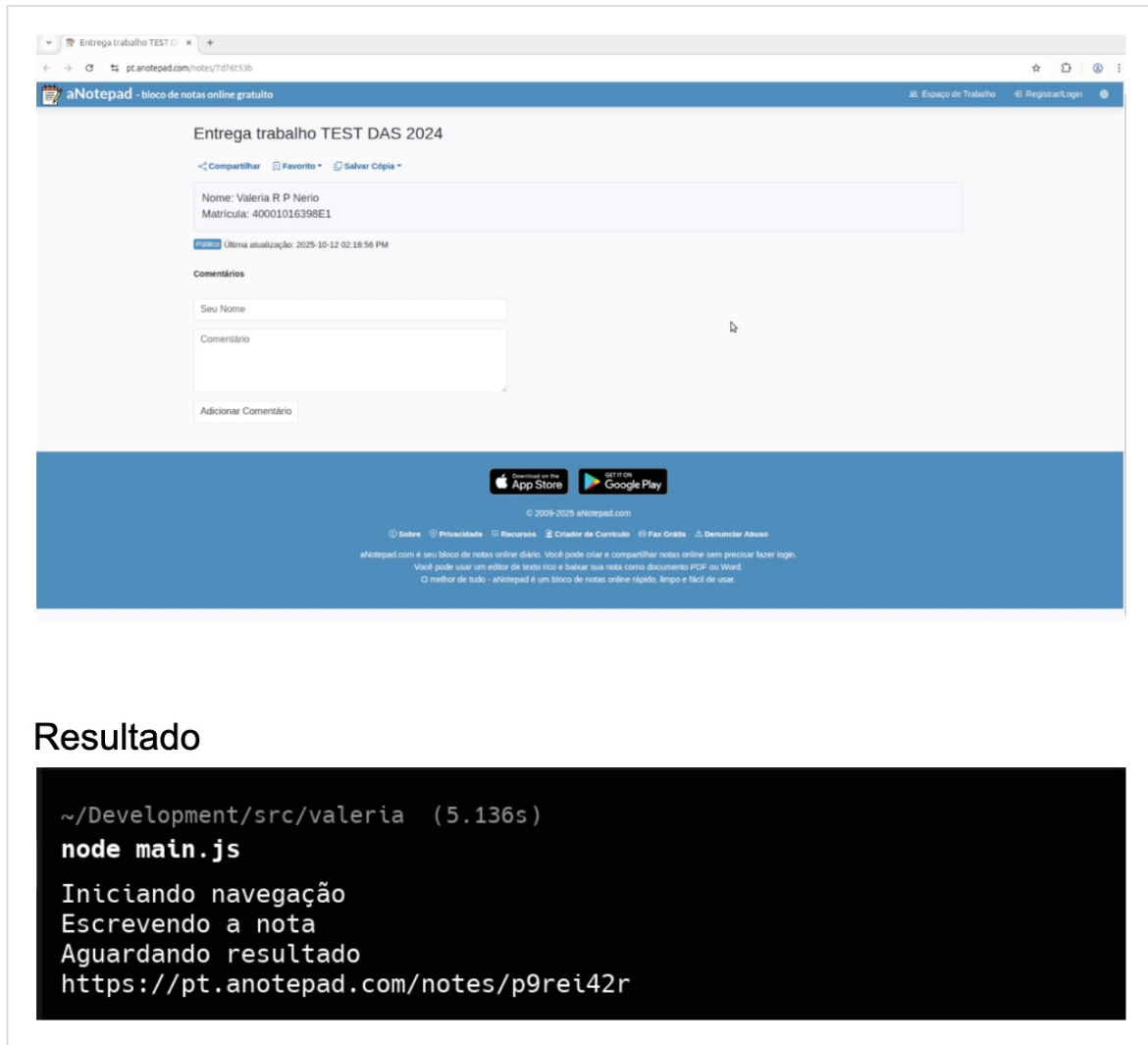
Para executar o código, foi necessário instalar o nodejs <https://nodejs.org/pt>



```
1 import puppeteer from 'puppeteer'
2
3 const titulo = "Entrega trabalho TEST DAS 2024"
4 const texto = "Nome: Valeria R P Merio\nMatricula: 40001016398E1"
5 const url = "https://pt.anotepad.com/"
6
7 const anotepadBot = async () => {
8   let browser
9   try {
10     browser = await puppeteer.launch({
11       headless: false,
12       args: ['--no-sandbox', '--disable-setuid-sandbox']
13     })
14     const page = await browser.newPage()
15     console.log('Iniciando navegação')
16     await page.goto(url, { waitUntil: 'domcontentloaded' })
17     await page.waitForSelector('#edit_title')
18     await page.waitForSelector('#edit_textarea')
19     await page.waitForSelector('#btnSaveNote')
20
21     console.log('Escrevendo a nota')
22     await page.type('#edit_title', titulo)
23     await page.type('#edit_textarea', texto)
24     await page.click('#btnSaveNote')
25
26     console.log('Aguardando resultado')
27     await page.waitForSelector('#lastSavedTime')
28
29     const noteUrl = page.url()
30     console.log(noteUrl)
31   } catch (err) {
32     console.error('Erro ao escrever a nota:', err)
33   } finally {
34     if (browser) {
35       await browser.close()
36     }
37   }
38 }
39
40
```

FONTE: Autora (2025)

FIGURA 31 – URL PREENCHIDA AUTOMATICAMENTE



FONTE: Autora (2025)

## 13 CONCLUSÃO

Este memorial técnico analisou os projetos desenvolvidos ao longo da Especialização em Desenvolvimento Ágil de Software, defendendo a tese de que a eficácia no desenvolvimento de software emerge de um gerenciamento ágil de projetos e uma profunda orientação à experiência do usuário (UX). A análise integrada dos trabalhos demonstrou que as metodologias ágeis, como Scrum e Kanban, fornecem a estrutura para a entrega iterativa e a otimização do fluxo, enquanto as práticas de UX favorecem que o produto desenvolvido seja relevante, útil e alinhado às necessidades reais do cliente.

Apesar da definição conceitual e dos benefícios evidentes, a adoção prática do desenvolvimento ágil de software enfrenta desafios significativos. O maior deles é a mudança cultural, exigindo a transição de uma cultura preditiva, baseada em comando e controle, para uma que valorize a colaboração, a autonomia e a adaptação, conforme analisado nos versionamentos do PMBOK.

Outro desafio crítico é a integração contínua de UX no fluxo ágil, para que o design não se torne um gargalo ou uma atividade superficial realizada apenas no início do projeto. Soma-se a isso a complexa gestão do débito técnico, pois a pressão por entregas rápidas pode levar equipes a negligenciar a qualidade do código e potencial dificuldade em estabelecer métricas focadas em valor real para o cliente, em vez de métricas de produção.

Portanto, conclui-se que a verdadeira agilidade não reside unicamente na aplicação de um *framework*, porém na internalização de seus princípios, sobretudo do aprendizado contínuo, entregando softwares que não apenas funcionam, mas que encantam e resolvem problemas reais.

## REFERÊNCIAS

BAUER, E. J. **Geração de interfaces de usuário para operações CRUD com base em metadados**. 2018. Trabalho de Conclusão de Curso do Bacharelado em Ciência da Computação. Universidade do Extremo Sul Catarinense, Criciúma, 2018. Disponível em: <http://repositorio.unesc.net/handle/1/8143>. Acesso em: 08 ago. 2025.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Campus, 2006. Disponível em: <https://books.google.com.br/books?id=hzi2tmT8QkUC>. Acesso em: 18 out. 2025.

COHN, M. Desenvolvimento de software com Scrum: aplicando métodos ágeis com sucesso. Porto Alegre: Bookman, 2011.

COUTINHO, N. de M.; NASCIMENTO, E. L. B. do. Desafios e benefícios da implementação de testes automatizados em empresas de software. **Cuadernos de Educación y Desarrollo - QUALIS A4**, v. 17, n. 4, p. e8221, 2025. DOI: 10.55905/cuadv17n4-176. Disponível em: <https://ojs.cuadernoseducacion.com/ojs/index.php/ced/article/view/8221>. Acesso em: 19 ago. 2025.

DA SILVA, R. O.; MARTINS, B. R.; DINIZ, W. G. A complexibilidade da UML e seus diagramas. **TECNOLOGIAS EM PROJEÇÃO**, [S. l.], v. 8, n. 1, p. 86–99, 2017. Disponível em: <https://www.projecaociencia.com.br/index.php/Projecao4/article/view/825>. Acesso em: 19 out. 2025.

DESTRO, G. A.; FRANÇA, B. B. N. de. **Avaliação do Nível da Aplicação de Práticas de Entrega e Integração Contínua em Repositórios de Código Aberto**. 2019. Projeto de Final de Graduação. Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2019. Disponível em: <https://www.ic.unicamp.br/~reltech/PFG/2019/PFG-19-31.pdf>. Acesso em: 20 out. 2025.

FERREIRA, A. S. **Aplicação de Metodologias Ágeis de gestão no desenvolvimento de projetos de tecnologia para aceleração de entregas, validação e melhoria na integração da equipe**. 2022. Trabalho de Conclusão de Curso. Graduação em Engenharia Mecatrônica. Universidade Federal de Uberlândia, Uberlândia, 2023. 52 f.

FUCCI, D.; TURHAN, B. A Replicated Experiment on the Effectiveness of Test-First Development. *In*: ACM / IEEE INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING AND MEASUREMENT, 2013, Baltimore. **Anais [...]**. Baltimore, MD: IEEE, 2013. p. 103-112. DOI: 10.1109/ESEM.2013.15.

GONZALEZ, L. F. **Análise comparativa sobre duas metodologias de gestão ágil: Extreme Programming e Scrum**. Trabalho de Conclusão de Curso. Graduação em Engenharia de Controle e Automação. Universidade Federal de Uberlândia, Uberlândia, 2021. 76 f.

HEFFELFINGER, D. R. **NetBeans: a plataforma do desenvolvedor Java**. Rio de Janeiro: Alta Books, 2008. Disponível em: <https://books.google.com.br/books?id=JCMBWozfcJUC>. Acesso em: 19 out. 2025.

HERNÁNDEZ-REVELES, J. G.; FERNANDEZ, A. G.; ROMERO MOYANO, A. A. **Serious games to understand the Kanban method: recommendations, and options for an effective simulation**. In: INTERNATIONAL CONFERENCE ON SOFTWARE PROCESS IMPROVEMENT (CIMPS), 13., 2024, Mérida. **Anais...** Mérida: IEEE, 2024. p. 1-5. Disponível em: <https://doi.org/10.1109/CIMPS65195.2024.11095941>.

HP-API: **The Harry Potter API**. 2025. Disponível em: <https://hp-api.onrender.com/>. Acesso em: 20 out. 2025.

KANBAN Simulator. **Software development**. Disponível em: <http://www.kanbanboardgame.com/>. Acesso em: 10 jun. 2025.

LOPES, L.; SOUZA, G. J. R. de; ALVES, A. F.; NUNES, R. B. Metodologias ágeis: explorando o impacto do Scrum e do Kanban na qualidade e produtividade do software. **Revista Multidisciplinar do Nordeste Mineiro**, Timóteo, v. 12, n. 2, 2024. Disponível em: <https://doi.org/10.61164/rnm.v12i2.3060>. Acesso em: 12 set. 2025.

MAGALHÃES, P. A.; TIOSSO, F. Código Limpo: padrões e técnicas no desenvolvimento de software. **Revista Interface Tecnológica**, Taquaritinga, SP, v. 16, n. 1, p. 197–207, 2019. DOI: [10.31510/infa.v16i1.597](https://doi.org/10.31510/infa.v16i1.597). Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/597>. Acesso em: 13 ago. 2025.

MAIA, J. T. P. **Aprimorando a entrega e disponibilidade de um sistema de informação com CI/CD e estratégias de implantação**. 2024. Trabalho de Conclusão de Curso. Graduação em Sistemas de Informação). Centro de Ensino Superior do Seridó, Universidade Federal do Rio Grande do Norte, Natal, 2024. Disponível em: <https://repositorio.ufrn.br/handle/123456789/59948>. Acesso em: 16 out. 2025.

MARCZAK, S.; SÁ, L.; CECCATO, I.; AUDY, J.; ANTUNES, D. **Uma proposta de organização e funcionamento da função de Garantia da Qualidade de Software em um contexto de implantação do SW-CMMi**. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE (SBQS), 2., 2003, Fortaleza. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2003. p. 1-16. DOI: <https://doi.org/10.5753/sbqs.2003.15688>.

MATOS, A. de. **Efeitos do uso do desenvolvimento orientado por testes em conjunto com critérios de teste na indústria de software ágil**. 2020. Dissertação de Mestrado em Informática. Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná, 2020. 73 f. Disponível em: <https://magsilva.pro.br/publications/Matos-MscThesis-2020.pdf>. Acesso em: 19 out. 2025.

MUSTIKA, N.; MIRFAN, M.; ABDUL KADIR PAREWE, A. M.; LAYUK, T. T. Designing the User Interface for the Virtual Tour Selayar System using Figma. **Brilliance: Research of Artificial Intelligence**, [S. l.], v. 4, n. 2, p. 478–483, 2024. DOI: 10.47709/brilliance.v4i2.4587. Disponível em: <https://itscience-indexing.com/jurnal/index.php/brilliance/article/view/4587>. Acesso em: 20 oct. 2025.

OLIVEIRA, E. R. de. **A importância dos testes de software**: O que testar, como testar e o porquê testar. Castanhal: Faculdade de Computação. Universidade Federal do Pará, 2023.

PAVANELI, T. H. **Desenvolvimento de uma API e uma Aplicação Web para auxiliar na análise de dados providos por Aplicação Móvel**. Trabalho de Conclusão de Curso. Bacharelado em Sistemas de Informação. Faculdade de Computação, Universidade Federal de Uberlândia, Uberlândia, 2023. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/37886/3/DesenvolvimentoAPIAplicacao.pdf>. Acesso em: 25 mai. 2025.

PEREIRA, P.; TORREÃO, P.; MARÇAL, A. S. Entendendo *Scrum* para Gerenciar Projetos de Forma Ágil. **MundoPM**, p. 2-11, 2007.

PROJECT MANAGEMENT INSTITUTE (PMI). **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK®)**. 6. ed. Newtown Square: PMI, 2017.

PROJECT MANAGEMENT INSTITUTE (PMI). **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK®)**. 7. ed. Newtown Square: PMI, 2021.

SATO, D. T. **Uso Eficaz de Métricas em Métodos Ágeis de Desenvolvimento de Software**. 2007. Dissertação de Mestrado em Ciências. Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2007. Disponível em: <https://pdfs.semanticscholar.org/1a8d/dad8b1698548f8e131a33d2829d98ec9cbf9.pdf>. Acesso em: 8 set. 2025.

SMART, J. F. **Jenkins: the definitive guide: continuous integration for the masses**. Sebastopol: O'Reilly, 2011. Disponível em: [https://books.google.com.br/books?id=m0N\\_yhblGysC](https://books.google.com.br/books?id=m0N_yhblGysC). Acesso em: 18 out. 2025.

SOUSA, M. R. de; BERTOMEU, J. V. C. UX Design na criação e desenvolvimento de aplicativos digitais. **Informática na educação: teoria & prática**, Porto Alegre, v. 18,

n. 2, 2016. DOI: 10.22456/1982-1654.54897. Disponível em: <https://seer.ufrgs.br/index.php/InfEducTeoriaPratica/article/view/54897>. Acesso em: 13 mai. 2025.

SPIES, E. H. **Um plano de métricas para monitoramento de projetos scrum**. Dissertação de Mestrado em Ciência da Computação. Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2013. 86 f.

TRUONG, D. **Automated web-based testing framework for the interface of a power plant control**. 2024. Dissertação de Mestrado em Engenharia Elétrica e Automação. Aalto University, 2024. Disponível em: <https://aaltodoc.aalto.fi/handle/123456789/134293>. Acesso em: 20 out. 2025.

ZHANG, F.; SUN, G.; ZHENG, B.; DONG, L. Design and Implementation of Energy Management System Based on Spring Boot Framework. **Information**, v. 12, n. 11, art. 457, nov. 2021. Disponível em: <https://doi.org/10.3390/info12110457>. Acesso em: 20 out. 2025.