

UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO LUCAS DELLA COLETTA FRANGELLA

MEMORIAL DE PROJETOS: UMA JORNADA COMPLETA PELO
DESENVOLVIMENTO ÁGIL DE SOFTWARE

CURITIBA

2025

JOÃO LUCAS DELLA COLETTA FRANGELLA

MEMORIAL DE PROJETOS: UMA JORNADA COMPLETA PELO
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: Profa. Dra. Rafaela Mantovani Fontana

CURITIBA

2025



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

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação Desenvolvimento Ágil de Software da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **JOÃO LUCAS DELLA COLETTA FRANGELLA**, intitulada: **MEMORIAL DE PROJETOS: UMA JORNADA COMPLETA PELO 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, 29 de Agosto de 2025.



RAFAELA MANTOVAN FONTANA
Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

O memorial de projetos tem como objetivo mostrar o que foi elaborado ao durante a especialização em Desenvolvimento Ágil de Software, mostrando a integração entre teoria e prática nas disciplinas do curso. Neste documento são mostrados artefatos como diagramas, protótipos, testes, tabelas e bancos de dados, imagens e trechos de código, que representam as etapas do processo de desenvolvimento, desde a ideia inicial até a entrega do sistema. Cada disciplina trabalhou práticas específicas, como modelagem, gestão, programação, testes, UX, infraestrutura e desenvolvimento mobile. A organização cronológica dos projetos ajudou entender a evolução dos aprendizados, destacando a importância do trabalho em equipe, adaptação a mudanças, entregas contínuas e com qualidade. As disciplinas de Introdução à Programação e Banco de Dados foram o ponto de partida da base técnica, ajudando a entender a lógica e a modelagem de dados no *MySQL*. Em seguida, em Modelagem Ágil de Software e Aspectos Ágeis de Programação, ficou evidente a importância de escrever um código limpo, organizado e reutilizável. Com essa base, foi possível seguir com o desenvolvimento de aplicações *Web* e *Mobile*, aprendeu-se, na prática o uso de *frameworks* modernos. Nas disciplinas de UX e Testes Automatizados, o aprendizado se voltou para a qualidade e experiência do cliente final com a elaboração de protótipos e a aplicação de testes *end-to-end* que mostraram como garantir o valor e a efetividade da aplicação.

Palavras-chave: Agilidade, Integração, Artefatos, Colaboração.

ABSTRACT

The project portfolio aims to present the work carried out during the specialization in Agile Software Development, highlighting the integration between theory and practice across the course subjects. This document includes artifacts such as diagrams, prototypes, tests, tables and databases, images, and code snippets that represent the stages of the development process, from the initial idea to the final delivery of the system. Each subject covered specific practices such as modeling, management, programming, testing, UX, infrastructure, and mobile development. The chronological organization of the projects helped to show the evolution of learning, emphasizing the importance of teamwork, adaptation to changes, and continuous, high-quality deliveries. The courses Introduction to Programming and Databases formed the initial technical foundation, supporting the understanding of logic and relational modeling in MySQL. Subsequently, Agile Software Modeling and Agile Programming Aspects reinforced the importance of writing clean, organized, and reusable code. Based on this foundation, it was possible to advance to Web and Mobile application development, where the practical use of modern frameworks was applied. In UX and Automated Testing, the focus was on quality and the end-user experience, with the design of prototypes and the execution of end-to-end tests that demonstrated how to ensure both the value and effectiveness of the applications.

Keywords: Agility, Integration, Artifacts, Collaboration.

SUMÁRIO

1 PARECER TÉCNICO.....	7
2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....	9
2.1 ARTEFATOS DO PROJETO.....	10
3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2.....	11
3.1 ARTEFATOS DO PROJETO.....	11
4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2	20
4.1 ARTEFATOS DO PROJETO.....	20
5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....	23
5.1 ARTEFATOS DO PROJETO.....	24
6 DISCIPLINA: BD – BANCO DE DADOS.....	25
6.1 ARTEFATOS DO PROJETO.....	26
7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO	29
7.1 ARTEFATOS DO PROJETO.....	30
8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2	31
8.1 ARTEFATOS DO PROJETO.....	32
9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE.....	33
9.1 ARTEFATOS DO PROJETO.....	34
10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....	35
11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)	36
11.1 ARTEFATOS DO PROJETO.....	37
12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS	41
12.1 ARTEFATOS DO PROJETO.....	41
13 CONCLUSÃO	43

1 PARECER TÉCNICO

O memorial reúne os projetos desenvolvidos ao longo da especialização em Desenvolvimento Ágil de *Software*, permitindo a verificação de como cada disciplina contribuiu para a formação de competências técnicas e relacionadas a metodologias. O compartilhamento de conceitos entre as disciplinas foi de extrema importância para entender como o desenvolvimento ágil se satisfaz, através de requisitos, modelagem, implementação, testes, integração contínua e entregas de qualidade (Prikladnicki et al., 2014). Além disso, o Gerenciamento de Projetos no contexto ágil segue as orientações e diretrizes do PMI (2021).

Inicialmente, as disciplinas de introdução a programação e banco de dados ajudaram com conceitos de lógica e modelagem de informações, o projeto do sistema com cadastramentos em banco de dados *SQL* deu consistência da estrutura e desenho da arquitetura de um software alinhado com lógicas básicas de programação em *Java* (Eckel, 2006).

A partir do primeiro desenvolvimento realizado em *Java*, os conceitos e metodologias vistos em aspectos e modelagem ágil ampliaram a importância de um *design* limpo, código reutilizável e boas práticas para um sistema em constante evolução (Martin, 2009). O projeto do *Bubble Sort* por exemplo, aliado ao uso de *UML* e outras técnicas demonstrou que a simplicidade e a organização impactam diretamente na melhoria e manutenção de aplicações.

Com foco em *front-end*, foi possível aplicar metodologias ágeis em exemplos de projetos reais no mercado. Com as disciplinas de Desenvolvimento Web (WEB) e Desenvolvimento *Mobile* (MOB) foi utilizado *Angular* e *Spring Boot* para desenvolvimento *web* e aplicativos *mobile Android* usando *Kotlin*, esses CRUDs que significa *create*, *read*, *update* e *delete* ou em português criar, ler, atualizar e apagar viabilizaram a utilização de *frameworks* modernos e já consolidados no mercado.

Outra parte fundamental foi o aprendizado nas disciplinas de UX no desenvolvimento ágil de *software* e testes automatizados, apoiando na qualidade e experiência do usuário como elementos principais no processo ágil, sendo UX em português a experiência do usuário (Levy, 2021). O protótipo elaborado em UX mostrou como o sucesso de um aplicativo depende do *design* centrado na usabilidade do cliente final e em testes foi usado o *framework Playwright* para testes fim a fim

direto no *front-end* garantindo a confiabilidade das funcionalidades e agilizando as entregas em produção.

Na disciplina de Infra-Estrutura - *DevOps* foi marcado como um concentrador de todas as demais disciplinas, significando em português o desenvolvimento e operações, pois sem uma estrutura baseada em código como o *Docker* a complexidade seria um desafio. Além disso o versionamento de código usando o *GitLab* é amplamente utilizado em todas as demais disciplinas com versionamento de código em sistemas de alta complexidade alinhando os testes com *pipeline* em *Jenkins*.

Os projetos desenvolvidos durante o curso, concluíram que o desenvolvimento ágil não é somente nos processos e organização como o *Kanban*, mas também as práticas que garantem qualidade, segurança e evolução permanente do produto (Anderson, 2011). A integração entre elas ajudou a entender o processo de fim a fim, desde a concepção de uma ideia até a melhoria e soluções de *bugs*.

2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

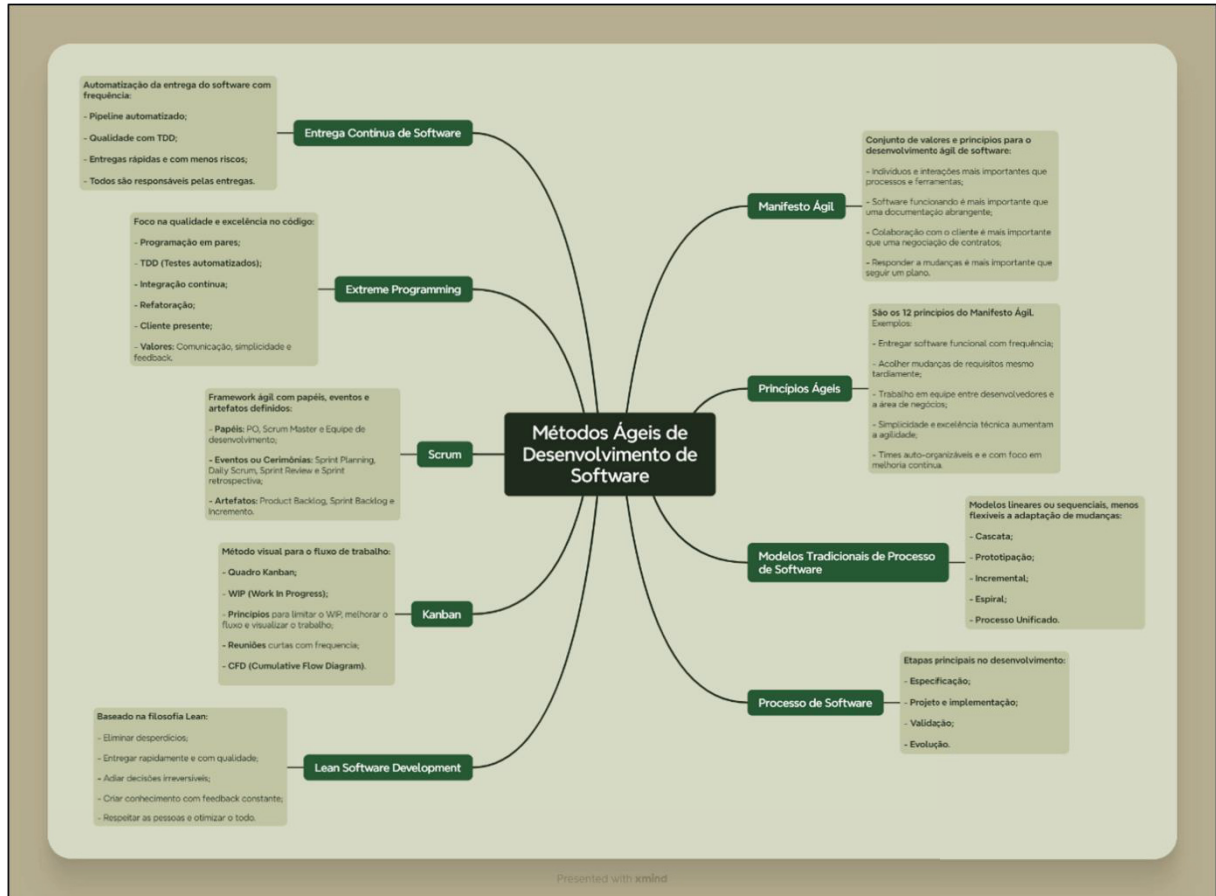
O projeto da disciplina métodos ágeis para desenvolvimento de *software* (MADS) tem como objetivo a experiência prática sobre os valores, princípios e práticas essenciais das abordagens ágeis. Foi utilizado *frameworks* como *Scrum*, *Kanban* e *Extreme Programming (XP)*, além de práticas como entrega contínua para promover o alinhamento entre a teoria e a prática na visão de desenvolvimento de software.

A importância deste projeto é o foco na entrega com incrementos funcionais e adaptação contínua a mudanças solicitadas, priorizando a colaboração e o *feedback* frequente com o cliente final. Destacando a auto-organização da equipe, comunicação intensa, priorização baseada no valor e foco na simplicidade e excelência técnica, os quais são pilares do Manifesto Ágil.

Além disso, a disciplina exigiu a integração com conhecimentos de outras áreas, como engenharia de requisitos, testes, qualidade, práticas *Lean* e o gerenciamento de configurações. A abordagem multidisciplinar ajudou em uma visão ampla do ciclo de desenvolvimento, estimulando a preparação para atuar em contextos reais.

2.1 ARTEFATOS DO PROJETO

FIGURA 1 – MAPA MENTAL RESUMIDO NO PROJETO FINAL



FONTE: O autor (2025)

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

Os projetos desenvolvidos nas disciplinas modelagem ágil de software, MAG1 e MAG2 proporcionaram uma vivência prática dos principais conceitos de desenvolvimento ágil aplicado à construção de sistemas completos. Através de um estudo de caso envolvendo um sistema de gestão condominial, foi possível garantir aprendizados em todas as etapas do processo ágil, desde o levantamento de requisitos a modelagem funcional e estrutural da aplicação.

Durante o desenvolvimento do projeto, foram utilizados artefatos como histórias de usuário, critérios de aceitação, diagramas de casos de uso, diagramas de sequência e de classes, seguindo uma abordagem iterativa e incremental. Essas práticas alinhadas aos princípios descritos por autores como Cohn (2004), ao reforçar a importância das histórias de usuário para aproximar cliente e equipe de desenvolvimento. A modelagem baseada em objetos também esteve presente, conforme fundamentos propostos por Rumbaugh (1991) e Guilleanes (2013), sendo de grande valia para representar visualmente os elementos da aplicação e seus relacionamentos de forma clara e objetiva.

3.1 ARTEFATOS DO PROJETO

A seguir, estão os principais artefatos desenvolvidos no trabalho, organizados conforme as etapas do processo ágil:

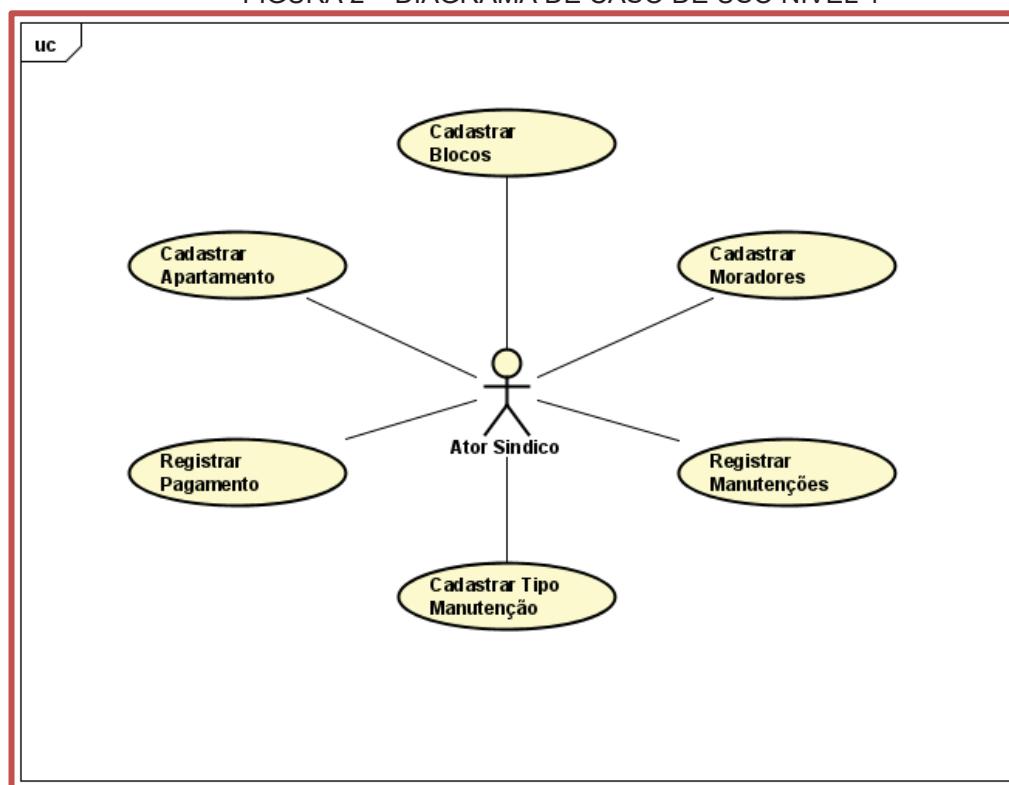
QUADRO 1 – LISTA DE REQUISITOS

Requisito	Descrição
Manter Cadastro dos Apartamentos	Permitir o cadastro completo dos apartamentos, incluindo número do apartamento e bloco
Manter Cadastro dos Moradores	Permitir o cadastro completo dos moradores, incluindo CPF, nome, telefone, apartamento, se é responsável ou se é proprietário. Deve verificar se o CPF já existe e se o número do apartamento já existe no cadastro. Deve guardar os dados dos veículos dos moradores, bem como suas vagas de garagem.

Registrar Pagamento do Condomínio	Registrar o pagamento do condomínio, incluindo apartamento, mês/ano de referência, data de vencimento, data de pagamento e valor do condomínio. Ao digitar o número do apartamento, o sistema deve apresentar o nome do morador responsável e o valor do condomínio. Um pagamento só pode ser registrado se não houver valores anteriores vencidos.
Registrar Manutenções	Registrar as manutenções no prédio, que podem ser de pintura, limpeza de caixa d'água, jardinagem etc.

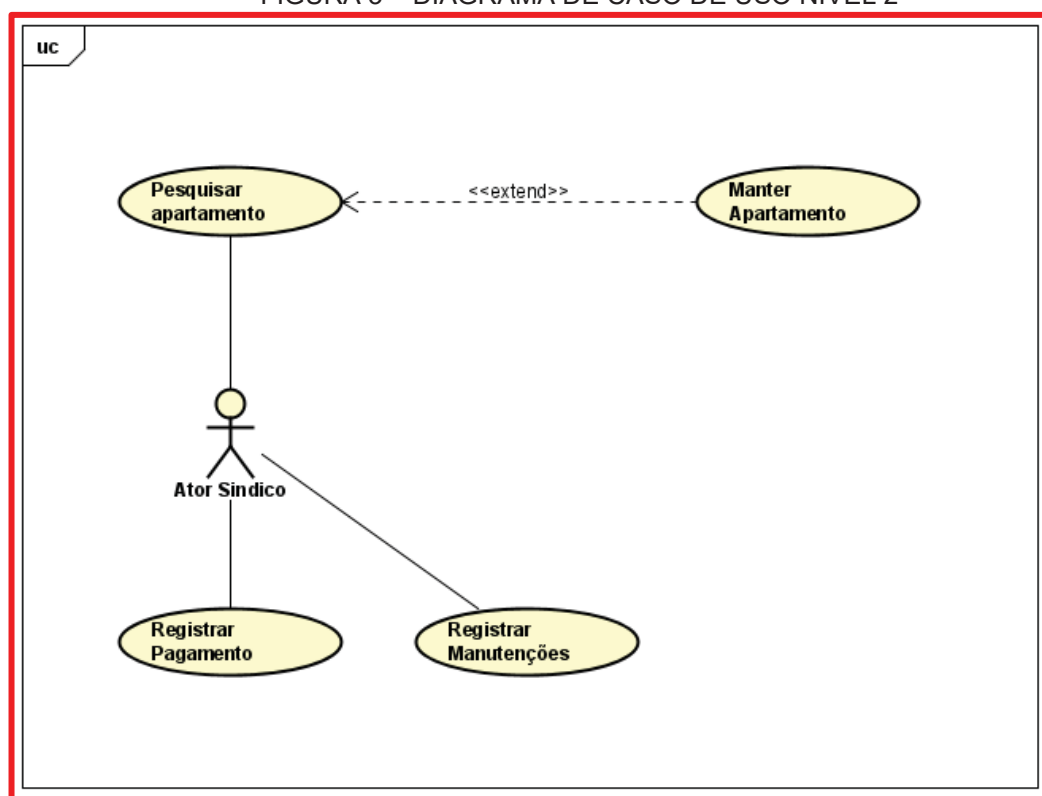
FONTE: O autor (2025)

FIGURA 2 – DIAGRAMA DE CASO DE USO NÍVEL 1



FONTE: O autor (2025)

FIGURA 3 – DIAGRAMA DE CASO DE USO NÍVEL 2



FONTE: O autor (2025)

- **Histórias de Usuário**

Nome:

HU001 – Registrar manutenções

Sendo/quero/para:

Sendo um síndico

Quero registrar manutenções do condomínio

Para formalizar a manutenção e prestar contas aos condôminos

FIGURA 4 – DESENHO DE TELA

Cadastro De Nova Manutenção

Informações:

Título

Descrição

Responsável ou prestador

Documentos (Orçamentos, recibos, etc.):

Browse...

Status:

Concluído

Tipo:

Preventiva
Corretiva
Emergencial

Prioridade:

Alta
Média
Baixa

Data início:

03/01/2024

Data conclusão:

03/31/2024

Cadastrar

FONTE: O autor (2025)

QUADRO 2 - CRITÉRIOS DE ACEITAÇÃO DETALHADOS

1) Deve apresentar título, descrição e responsável pela manutenção

Dado que	O síndico entrou no formulário de cadastro
Quando	O formulário é apresentado
Então	As informações devem ser preenchidas

2) Deve apresentar o tipo e prioridade da manutenção

Dado que	O síndico preencheu as informações
Quando	O formulário é apresentado
Então	A seleção de tipo e prioridade deve ser preenchido

R1 – Seleção de tipo e prioridade

Inconsistência	Mensagem
Tipo diferente de Preventiva, Corretiva ou Emergencial	“Tipo de manutenção inválido”
Prioridade diferente de Alta, média ou Baixa	“Prioridade inválida”

3) Deve efetivar uma nova manutenção

Dado que	Todos os campos foram preenchidos
Quando	O síndico clica no botão cadastrar
Então	O Sistema salva as informações da nova manutenção e apresenta a mensagem "Cadastro realizado com sucesso"

4) Deve apresentar a data início e fim previsto

Dado que	O síndico preencheu o tipo e prioridade
Quando	O síndico seleciona a data início e fim
Então	O Sistema define a data prevista finalização

R2 – Data fora do formato

Inconsistência	Mensagem
Data fora da formatação	"Insira a data no formato correto MM/DD/AAAA"

5) Deve armazenar os documentos necessários para prestação de contas

Dado que	É necessário armazenar os documentos
Quando	O síndico estiver cadastrando, precisa anexar a documentação
Então	O Sistema permite o upload e armazenamento de arquivos

R3 – Formato do arquivo não permitido

Inconsistência	Mensagem
Formato do documento não permitido	"São permitidos apenas arquivos PDF, jpeg e .xlsx"

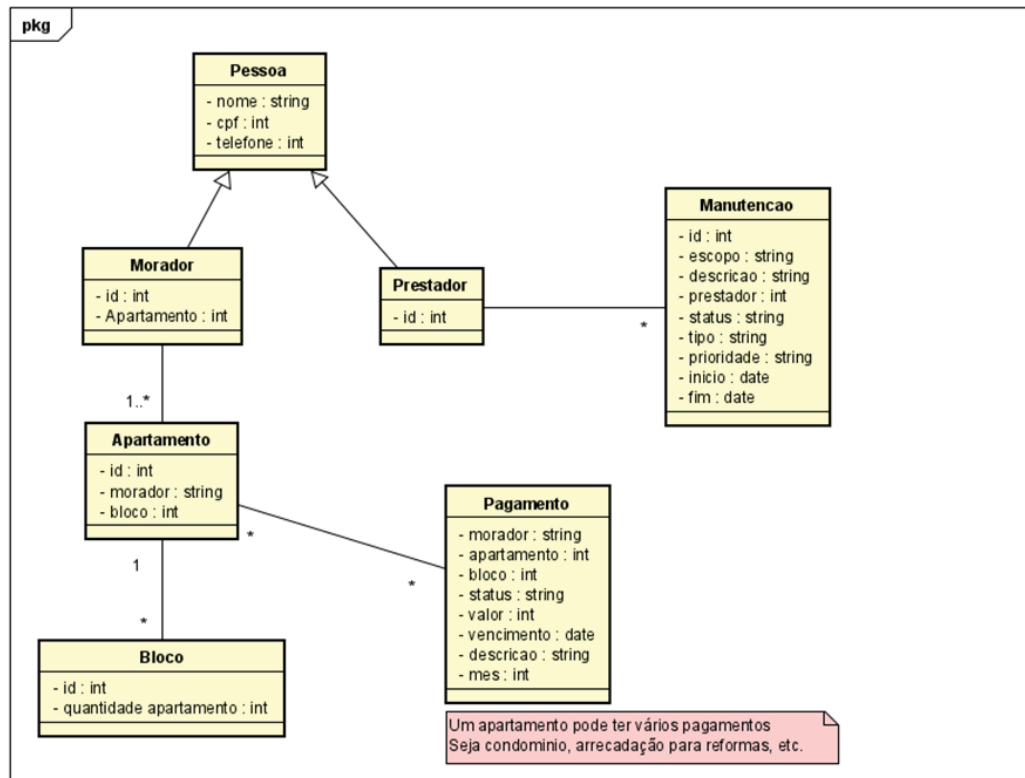
6) Deve apresentar o status da manutenção

Dado que	É necessário apresentar o status da manutenção
----------	--

Quando	O síndico visualize a página de detalhes da nova manutenção
Então	O Sistema deve exibir o status atual da manutenção

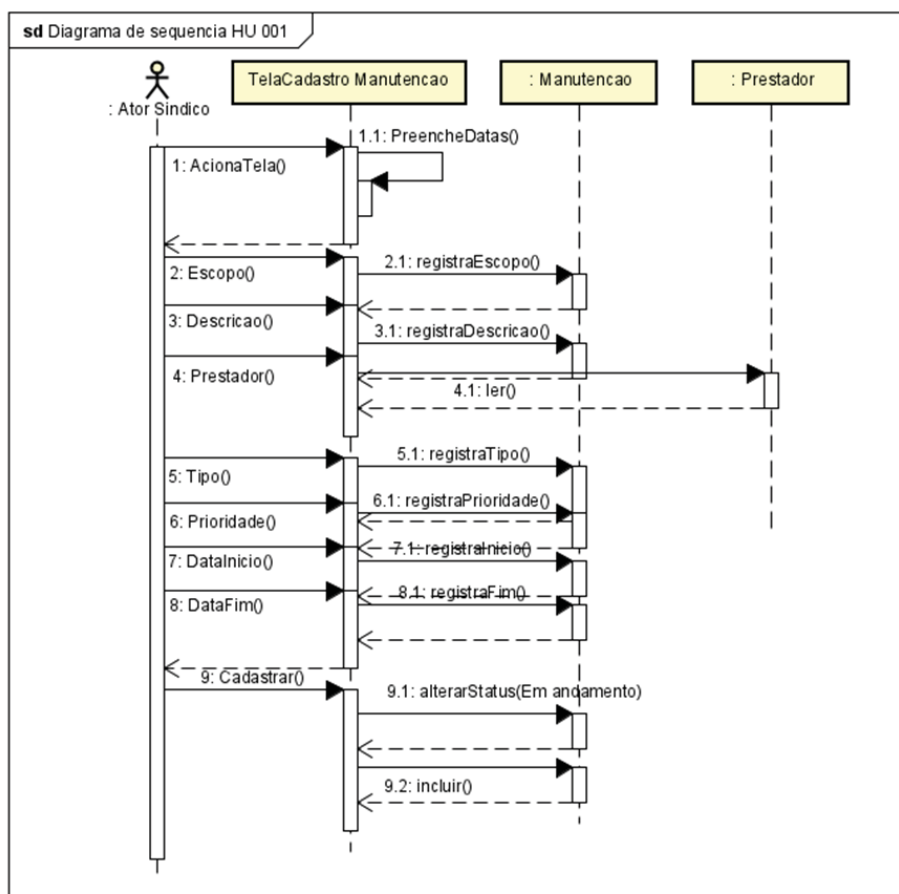
FONTE: O autor (2025)

FIGURA 5 – DIAGRAMA DE CLASSES



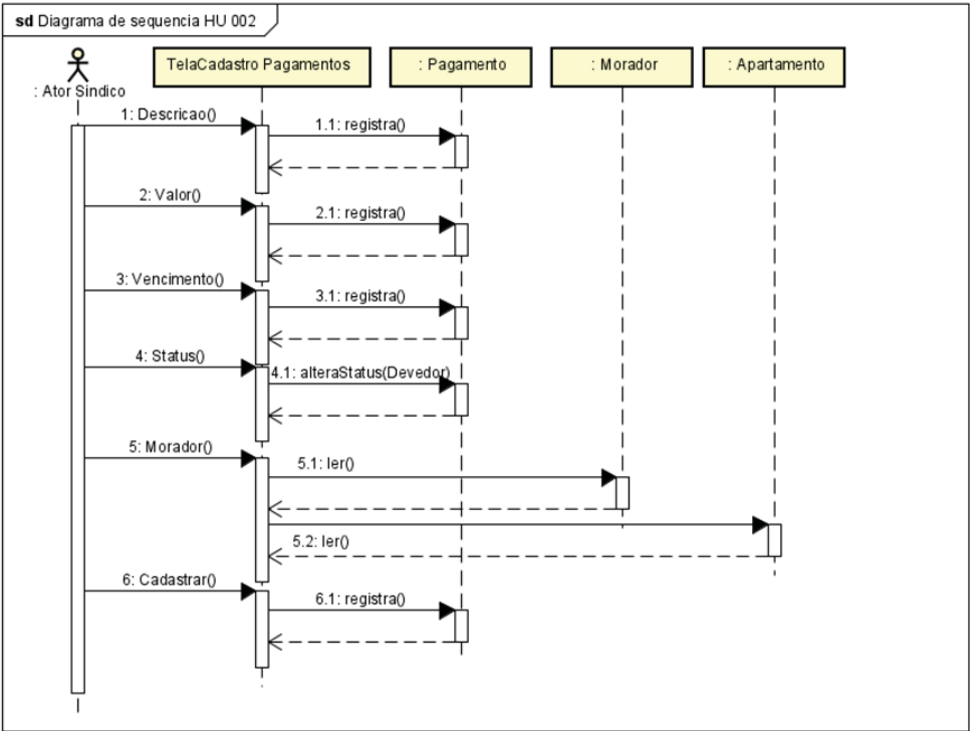
FONTE: O autor (2025)

FIGURA 6 – DIAGRAMA DE SEQUÊNCIA



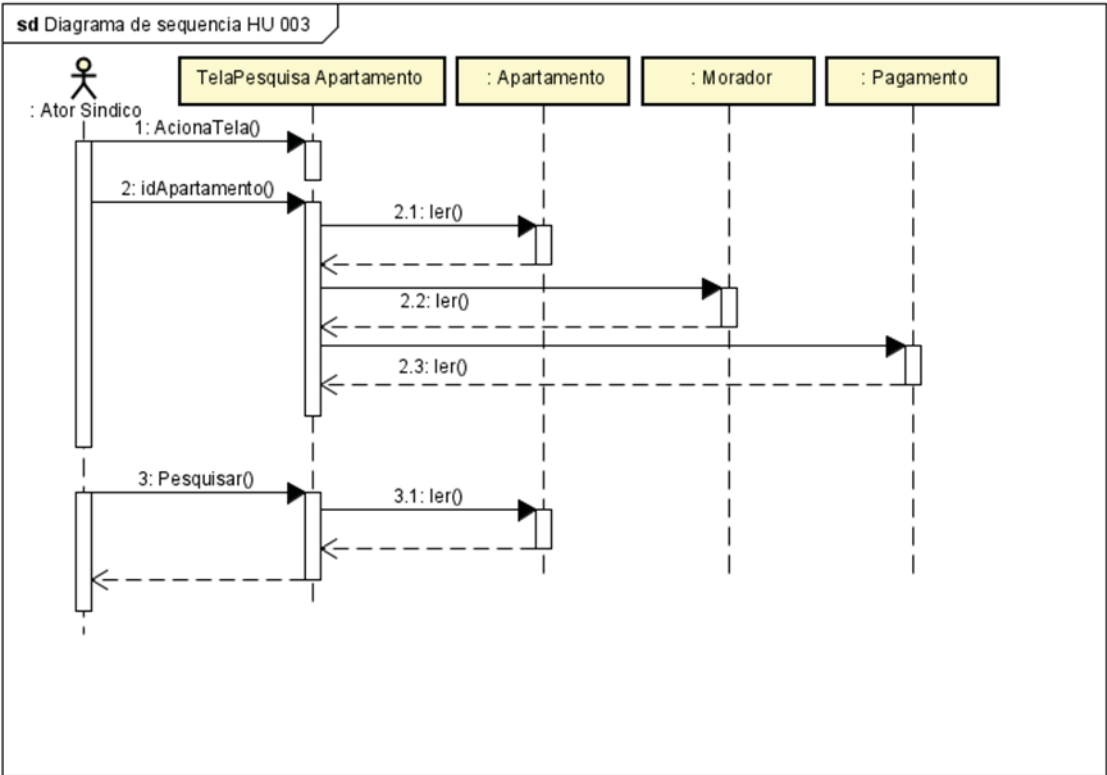
FONTE: O autor (2025)

FIGURA 7 – DIAGRAMA DE SEQUÊNCIA



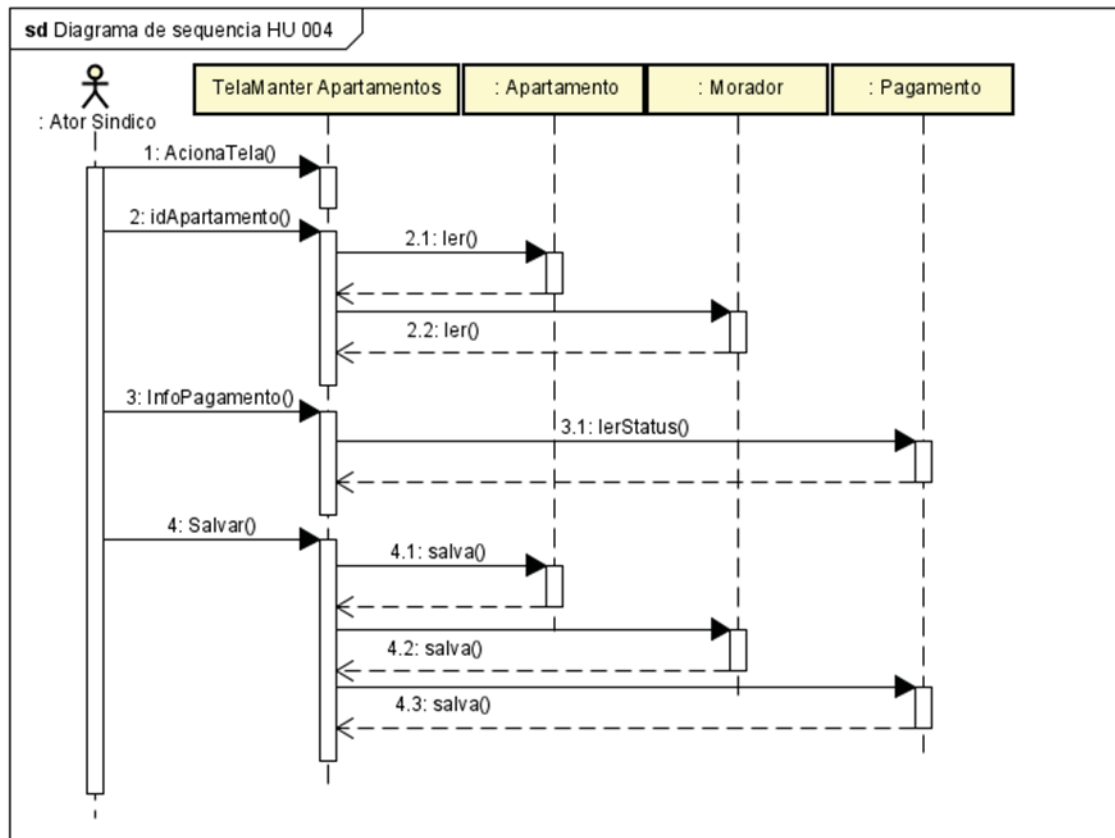
FONTE: O autor (2025)

FIGURA 8 – DIAGRAMA DE SEQUÊNCIA



FONTE: O autor (2025)

FIGURA 9 – DIAGRAMA DE SEQUÊNCIA



FONTE: O autor (2025)

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

As disciplinas gerenciamento ágil de projetos de *software*, GAP1 e GAP2 proporcionaram um aprendizado prático e aprofundado dos princípios e ferramentas de gerenciamento ágil de projetos. Por meio de abordagens como *Scrum*, *Kanban* e fundamentos do PMBOK 7ª edição, onde foi possível planejar, executar e acompanhar projetos com foco nos valores da entrega, colaboração e adaptação contínua. Esse aprendizado é essencial para promover ambientes complexos e incertos, muito frequente em desenvolvimentos moderno de software.

O projeto desenvolvido viabilizou integrar conceitos como ciclos iterativos, planejamento incremental e uma gestão visual. Com isso, permitiu a simulação real da gestão de projetos, usando técnicas como *Desing Thinking*, *Lean Inception* e métricas ágeis como o gráfico CFD (*Cumulative Flow Diagram*), em português diagrama de fluxo cumulativo e o cálculo de velocidade da equipe.

A estruturação do plano de *releases* criou a possibilidade de dividir as funcionalidades em diferentes *sprints*, proporcionando entregas constantes e previsíveis. Os artefatos desenvolvidos, como histórias de usuário, plano de release, acompanhamento via *Kanban* e o gráfico de fluxo cumulativo consolidaram o aprendizado e assim na contribuição na formação de profissionais aptos a trabalharem em times ágeis e multidisciplinares.

4.1 ARTEFATOS DO PROJETO

- **Cálculo da velocidade**

A equipe estimou 6 horas disponíveis por dia útil, totalizando em 60 horas por *sprint* (10 dias). Considerando que cada ponto de história corresponde a 8 horas, a velocidade média estimada foi de 7,5 pontos por *sprint*, assim orientando a divisão das funcionalidades.

• **Plano de Release**

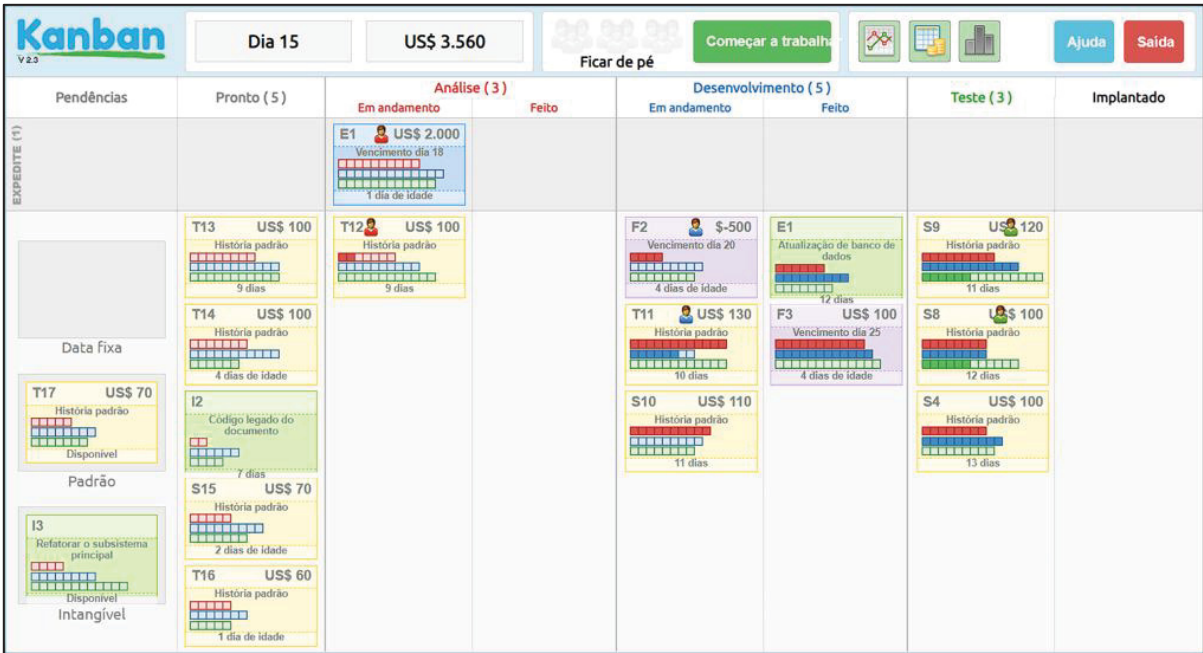
O planejamento das sprints foi distribuído em três, com início em 06/05/2024. As histórias de usuário foram priorizadas com base no valor de negócio e complexidade técnica.

QUADRO 3 – PLANO DE RELEASE

Iteração/ Sprint 1 (total 7 pontos)	Iteração/ Sprint 2 (total 7 pontos)	Iteração/ Sprint 3 (total 5 pontos)
Data Início: 06/05/2024	Data Início: 20/05/2024	Data Início: 03/06/2024
Data Fim: 17/05/2024	Data Fim: 31/05/2024	Data Fim: 14/06/2024
<HU001 - Registrar manutenções> SEUDO um síndico QUERO registrar manutenções do condomínio PARA formalizar a manutenção e prestar contas aos condôminos ESTIMATIVA (3 pontos)	<HU004 – Manter apartamento> SEUDO um síndico QUERO manter dados dos apartamentos, incluindo número, nome do morador e status de pagamento de taxas condominiais PARA garantir o registro e precisão dos dados para a gestão do condomínio ESTIMATIVA (7 pontos)	<HU003 – Pesquisar apartamento> SEUDO um síndico QUERO pesquisar apartamentos PARA verificar informações e detalhes de cada condômino ESTIMATIVA (5 pontos)
<HU002 – Registrar pagamento> SEUDO um síndico QUERO registrar pagamento de taxas condominiais PARA manter um registro preciso e atualizado dos pagamentos realizados ESTIMATIVA (4 pontos)		

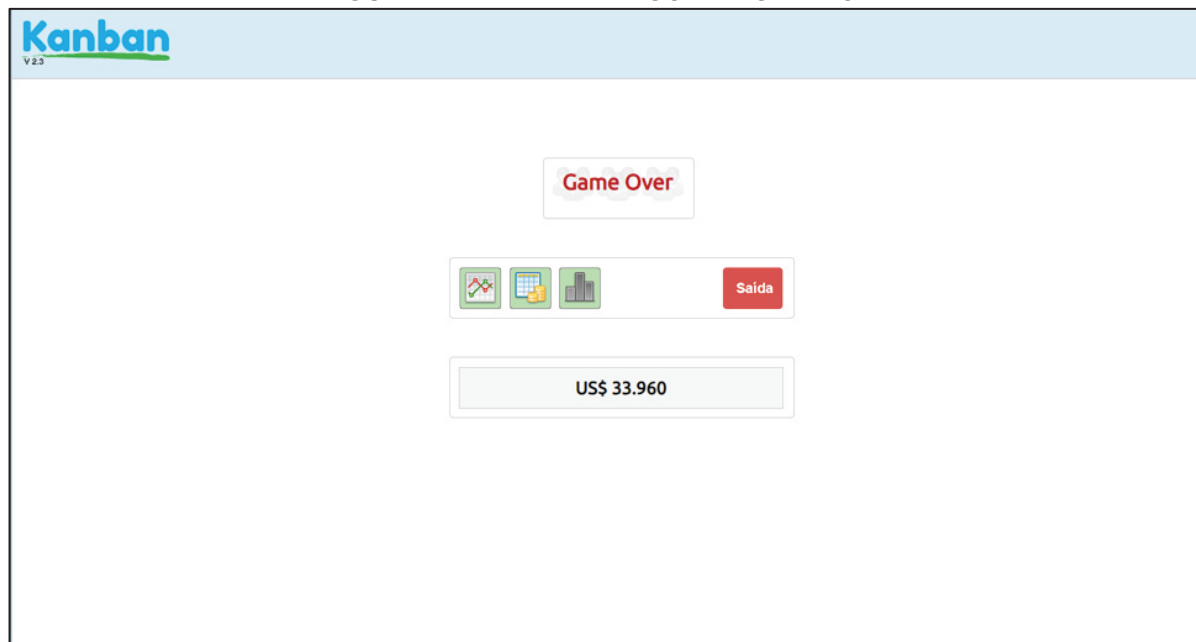
FONTE: O autor (2025)

FIGURA 10 - ANDAMENTO DO JOGO DIA 15



FONTE: Adaptada de Kanban Board Game (2025)

FIGURA 11 – TELA FINAL COM RECEITA OBTIDA

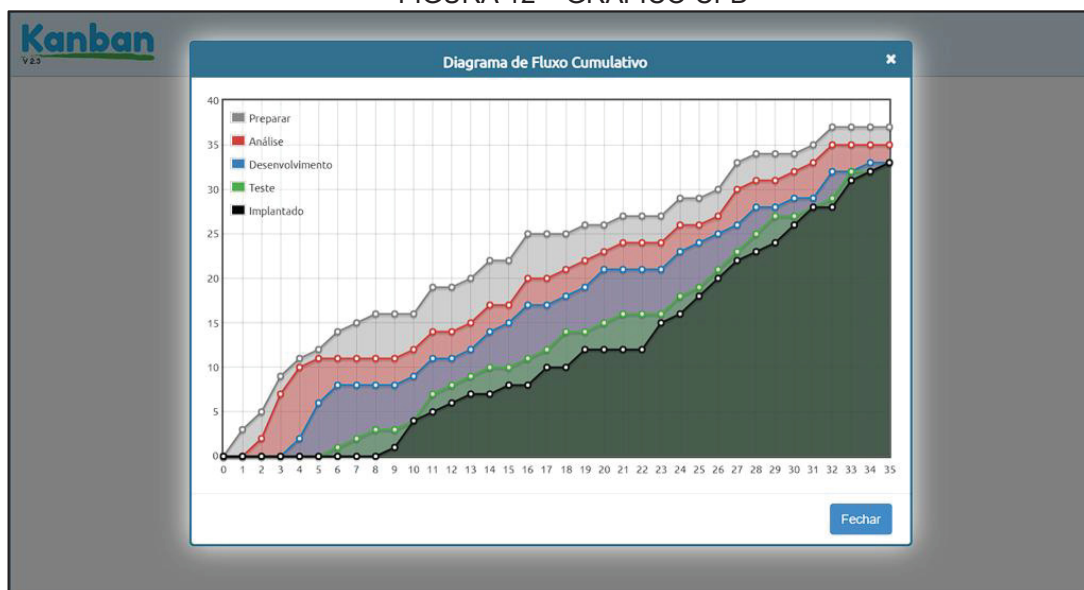


FONTE: Adaptada de Kanban Board Game (2025)

- **Gráfico CFD (*Cumulative Flow Diagram*)**

Esse artefato permitiu monitorar o andamento do projeto ao longo do tempo, visualizando gargalos e a distribuição das tarefas em andamento, concluídas ou pendentes alinhando o uso do *Kanban* para limitar o *WIP (Work in Progress)*.

FIGURA 12 – GRÁFICO CFD



FONTE: Adaptada de Kanban Board Game (2025)

5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

O trabalho final da disciplina introdução à programação teve como objetivo final a construção completa de um *backend* de um sistema bancário simplificado, com foco na programação orientada a objetos e na aplicação de testes automatizados. O sistema contempla o cadastro de novos clientes, contas correntes e contas de investimentos, persistindo os dados em um banco relacional *MySQL* (Date, 2004).

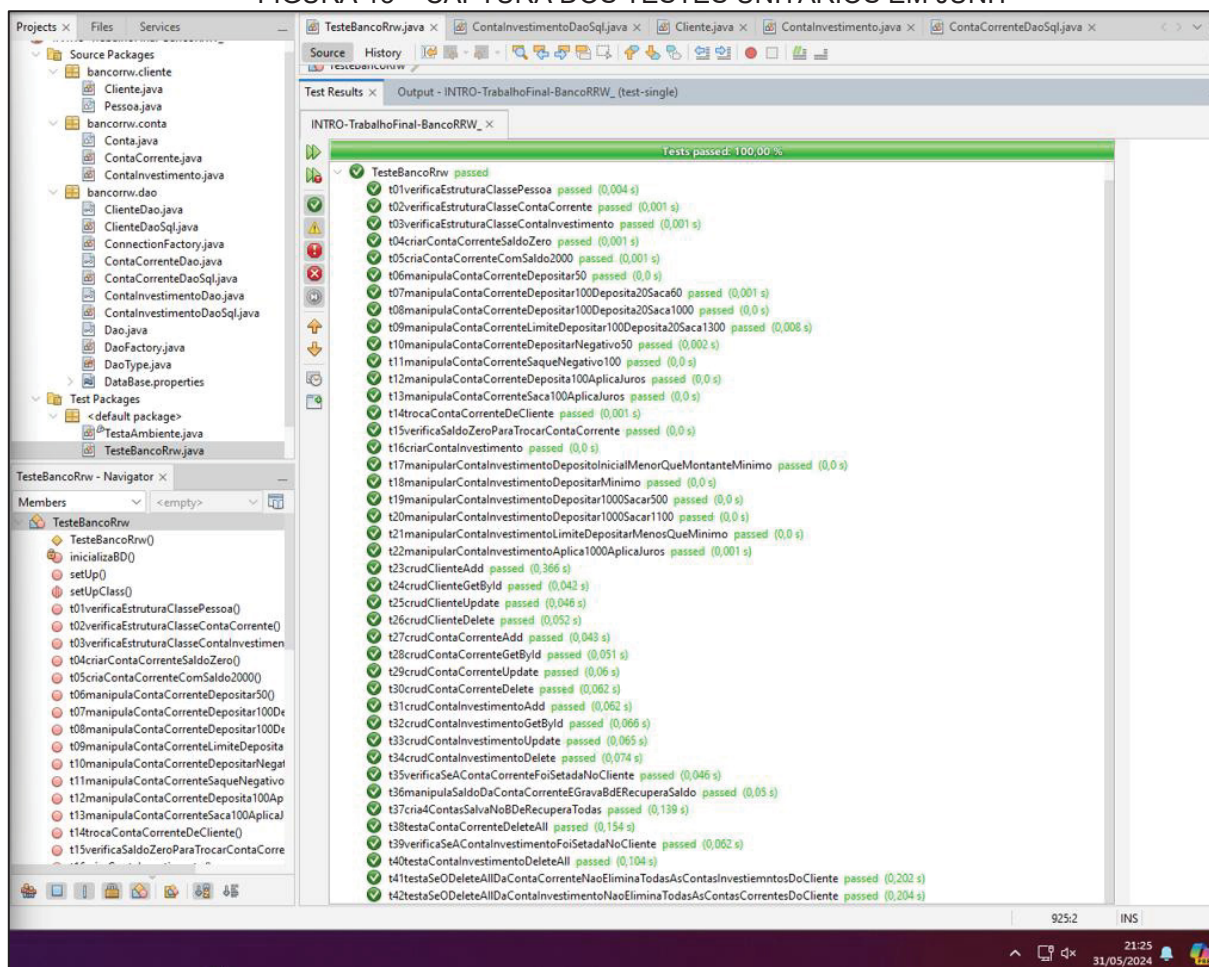
Foi estruturada para introduzir aprendizados em programação *Java* com foco prático, usando conceitos de modelagem, encapsulamento, herança e polimorfismo. O diferencial foi a abordagem de desenvolvimento orientado a testes, onde todas as funcionalidades deveriam ser desenvolvidas a partir de 42 casos de teste previamente definidos em *JUnit*.

A entrega do projeto se deu pela implementação das classes de domínio e acesso a dados com base no diagrama de classes fornecido, assim garantiu a execução correta de todos os testes automatizados. A execução de com sucesso de 100% dos testes evidencia que os requisitos funcionais do sistema foram devidamente atendidos conforme a especificação inicial.

O aprendizado da linguagem *Java* foi essencial para consolidar o conhecimento sobre testes automatizados, integração com um banco de dados, organização do código e as boas práticas de programação. Além disso reforçou a importância de garantir a qualidade desde o início do ciclo de desenvolvimento, fazendo entregas confiáveis, testáveis e alinhadas com a agilidade e engenharia de *software*.

5.1 ARTEFATOS DO PROJETO

FIGURA 13 – CAPTURA DOS TESTES UNITÁRIOS EM JUNIT



FONTE: O autor (2025)

6 DISCIPLINA: BD – BANCO DE DADOS

A disciplina de Banco de Dados visou a introdução para desenvolver, modelar e implementar soluções relacionais que satisfaçam as demandas reais de sistemas de informações completos. No andamento da disciplina foram discutidos os aspectos teóricos e práticos essenciais da modelagem conceitual, técnicas de normalização, construção de esquemas lógicos e manipulação de dados, utilizando a linguagem *SQL* (Date, 2004).

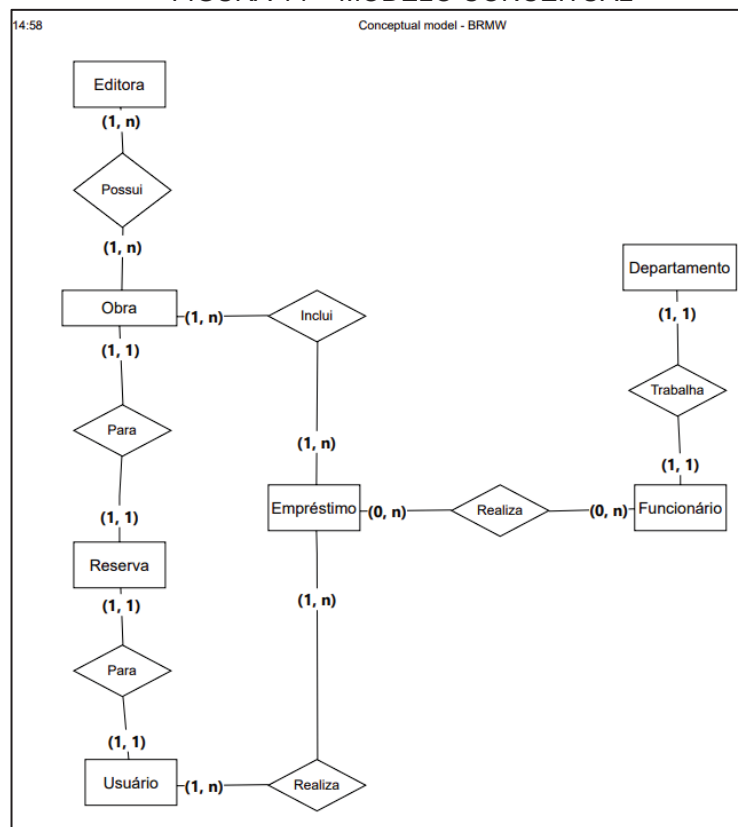
O projeto final desempenhou um papel fundamental, solicitando a projeção dois cenários distintos: o primeiro voltado a um controle de uma biblioteca, abordando a modelagem e lógica com base em requisitos fictícios, o segundo focado em um sistema de controle de atividades de uma empresa de telecomunicações, com maior complexidade de entidades, relacionamentos e regras de negócio.

O segundo cenário exigiu a criação de tabelas relacionando chamados, analistas, armários com tecnologia *GPON* (*Gigabit-capable Passive Optical Network*) e *frame relay*, com integrações 1x1, 1xN e NxN. A modelagem foi seguida da implementação em *SQL* (*Structured Query Language*), com criação dos *scripts* para a estrutura, inserção e consultas. Foram executados usando o *DBeaver* vinculado a um banco de dados *MySQL Community*.

A disciplina permitiu integrar os conhecimentos de estrutura de dados, lógica de programação e requisitos de sistemas em um contexto prático. Além disso ajudou na compreensão da importância das referências, normalização e otimização de consultas em bancos relacionais.

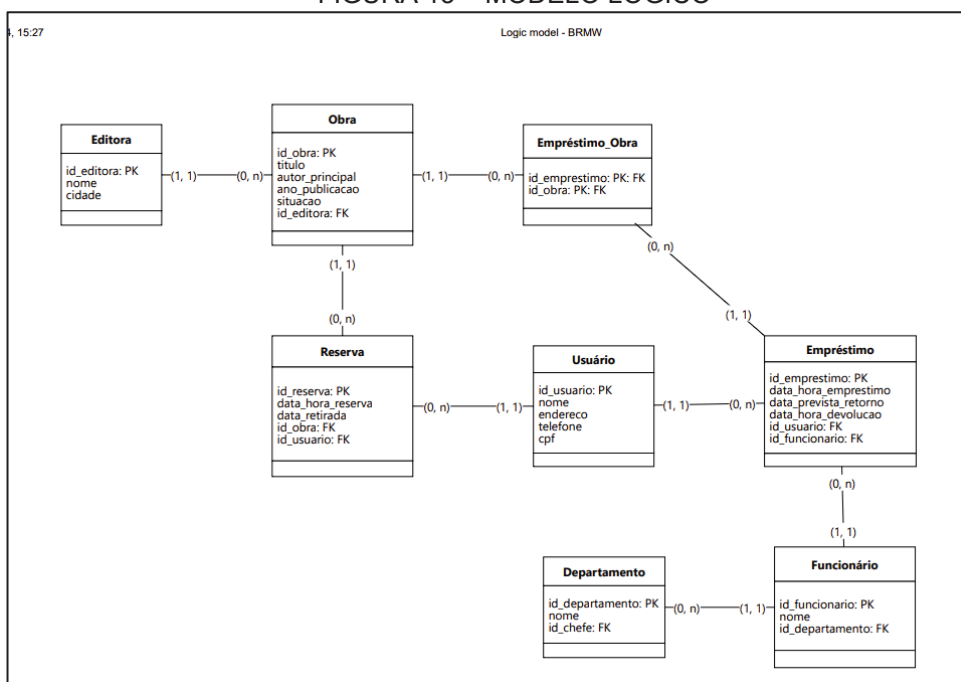
6.1 ARTEFATOS DO PROJETO

FIGURA 14 – MODELO CONCEITUAL



FONTE: O autor (2025)

FIGURA 15 – MODELO LÓGICO



FONTE: O autor (2025)

FIGURA 16 – DDL DAS TABELAS

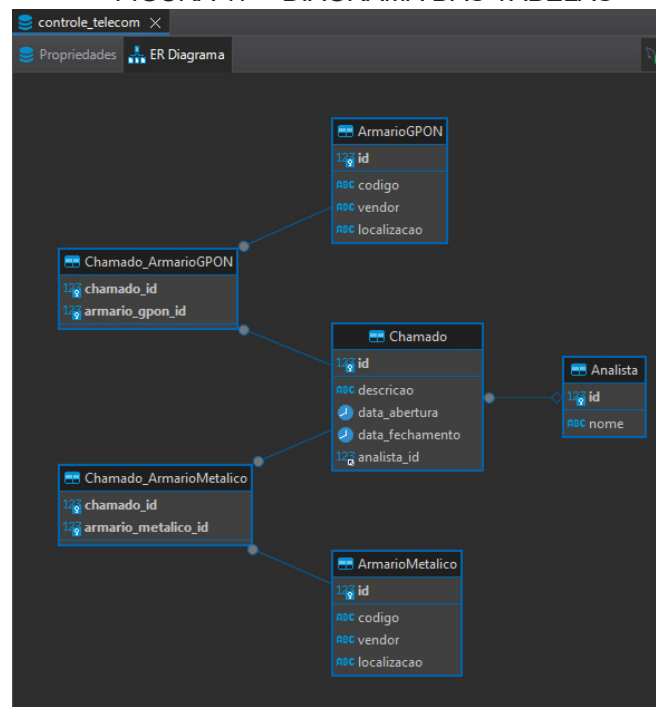
```

1 CREATE TABLE Analista (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     nome VARCHAR(100) NOT NULL
4 );
5 CREATE TABLE Chamado (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     descricao TEXT NOT NULL,
8     data_abertura DATE NOT NULL,
9     data_fechamento DATE,
10    analista_id INT,
11    FOREIGN KEY (analista_id) REFERENCES Analista(id)
12 );
13 CREATE TABLE ArmarioGPON (
14     id INT AUTO_INCREMENT PRIMARY KEY,
15     codigo VARCHAR(50) NOT NULL,
16     vendor VARCHAR(50) NOT NULL,
17     localizacao VARCHAR(100) NOT NULL
18 );
19 CREATE TABLE ArmarioMetalico (
20     id INT AUTO_INCREMENT PRIMARY KEY,
21     codigo VARCHAR(50) NOT NULL,
22     vendor VARCHAR(50) NOT NULL,
23     localizacao VARCHAR(100) NOT NULL
24 );
25 CREATE TABLE Chamado_ArmarioGPON (
26     chamado_id INT,
27     armario_gpon_id INT,
28     PRIMARY KEY (chamado_id, armario_gpon_id),
29     FOREIGN KEY (chamado_id) REFERENCES Chamado(id),
30     FOREIGN KEY (armario_gpon_id) REFERENCES ArmarioGPON(id)
31 );
32 CREATE TABLE Chamado_ArmarioMetalico (
33     chamado_id INT,
34     armario_metalico_id INT,
35     PRIMARY KEY (chamado_id, armario_metalico_id),
36     FOREIGN KEY (chamado_id) REFERENCES Chamado(id),
37     FOREIGN KEY (armario_metalico_id) REFERENCES ArmarioMetalico(id)
38 );

```

FONTE: O autor (2025)

FIGURA 17 – DIAGRAMA DAS TABELAS



FONTE: O autor (2025)

FIGURA 18 – CÓDIGO SQL PARA TESTES DA ESTRUTURA

```
TRABALHO > Exemplo_insertsql
1  INSERT INTO Analista (nome) VALUES ('João Lucas'), ('Felipe Suares'), ('Patrik Camargo'), ('Ana Claudia'), ('Denilson Santiago');
2
3  INSERT INTO Chamado (descricao, data_abertura, analista_id) VALUES
4  ('Instalação de armário GPON', '2024-06-01', 1),
5  ('Desinstalação de armário metálico', '2024-06-05', 2),
6  ('Manutenção de armário GPON', '2024-06-10', 3),
7  ('Desinstalação de armário metálico', '2024-06-15', 4),
8  ('Atualização de software no armário GPON', '2024-06-20', 5);
9
10 INSERT INTO ArmarioGPON (codigo, localizacao, vendedor) VALUES
11 ('CTA_I87', 'Centro', 'Nokia'),
12 ('CTA_I85', 'Centro', 'Nokia'),
13 ('CTA_M01', 'Zona Sul', 'Nokia'),
14 ('CTA_M04', 'Zona Sul', 'Nokia'),
15 ('AUC_A01', 'Zona Oeste', 'Nokia');
16
17 INSERT INTO ArmarioMetalico (codigo, localizacao, vendedor) VALUES
18 ('CTA_A01', 'Centro', 'Keymile'),
19 ('CTA_A85', 'Zona Norte', 'Keymile'),
20 ('CTA_B01', 'Zona Sul', 'Keymile'),
21 ('CTA_A08', 'Zona Leste', 'Keymile'),
22 ('CTA_A49', 'Zona Oeste', 'Keymile');
23
24 INSERT INTO Chamado_ArmarioGPON (chamado_id, armario_gpon_id) VALUES
25 (1, 1),
26 (1, 2),
27 (3, 3),
28 (5, 4),
29 (5, 5);
30
31 INSERT INTO Chamado_ArmarioMetalico (chamado_id, armario_metalico_id) VALUES
32 (2, 1),
33 (2, 2),
34 (4, 3),
35 (4, 4),
36 (4, 5);
```

FONTE: O autor (2025)

7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

A disciplina de Aspectos Ágeis de Programação teve como proposta aplicar práticas de desenvolvimento voltadas à qualidade e adoção de princípios ágeis no processo da implementação de um sistema. Teve como objetivo informar a importância de desenvolver um *software* funcional, mas ao mesmo tempo sustentável, de fácil leitura, manutenção, reduzindo riscos de falhas e facilitando sua melhoria e evolução a longo prazo.

Dentro desse contexto ágil, a clareza e a organização do código se tornam fundamentais para garantir entregas incrementais e rápidas. Um dos principais conceitos apresentados foi o *DRY (Don't Repeat Yourself)*, além disso também foram contextualizados a simplicidade, testabilidade e modularidade em um desenvolvimento, sendo estes elementos que ajudam a aumentar a produtividade de todo o time.

O projeto final desenvolvido consistiu na implementação de um algoritmo *Bubble Sort* em *Java*, com foco em demonstrar na prática como os princípios de código limpo devem ser efetuados desde a criação até os testes finais. Isso permitiu a integração dos conceitos teóricos adquiridos em outras disciplinas do curso.

7.1 ARTEFATOS DO PROJETO

FIGURA 19 – CÓDIGO JAVA

```
import java.io.*;

class OrdenaVetor {
    static void ordenaVetor(int[] vetor, int tamanho) {
        boolean foiTrocado;
        for (int i = 0; i < tamanho - 1; i++) {
            foiTrocado = false;
            for (int j = 0; j < tamanho - i - 1; j++) {
                if (vetor[j] > vetor[j + 1]) {
                    exchangeValues(vetor, j);
                    foiTrocado = true;
                }
            }
            if (!foiTrocado)
                break;
        }
    }

    private static void exchangeValues(int[] vetor, int j) {
        int temporario = vetor[j];
        vetor[j] = vetor[j + 1];
        vetor[j + 1] = temporario;
    }

    static void imprimeVetor(int[] vetor, int tamanho) {
        for (int indice = 0; indice < tamanho; indice++)
            System.out.print(vetor[indice] + " ");
        System.out.println();
    }

    public static void main(String args[])
    {
        int vetor[] = { 64, 34, 25, 12, 22, 11, 90, 111, 234, 455 };
        int tamanho = vetor.length;
        ordenaVetor(vetor, tamanho);
        System.out.println("Vetor ordenado: ");
        imprimeVetor(vetor, tamanho);
    }
}
```

FONTE: O autor (2025)

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

O projeto final das disciplinas de desenvolvimento web, WEB1 e WEB2 teve como objetivo consolidar, de forma prática, os conceitos de desenvolvimento ágil de *software*, aplicando metodologias e ferramentas amplamente usadas no mercado. Na parte de WEB1, foi elaborado dois *CRUDs*: Aluno e Curso, implementados com o *framework Angular*, que é uma ferramenta moderna para o desenvolvimento de aplicações web e armazenamento dos dados no *Local Storage* (Angular, 2025). Já em WEB2, o projeto foi ampliado com a criação de um terceiro *CRUD*, referente à Matrícula, desta vez integrando o *back-end* em *Spring Boot* conectado a um banco de dados *PostgreSQL* (Spring, 2025).

O trabalho teve como relevância a aproximação com situações reais de desenvolvimento, exigindo organização do código, utilização de *frameworks* modernos e integração eficiente entre *back-end* e *front-end*. Foram empregadas metodologias ágeis, com entregas incrementais dos *CRUDs*, refletindo as práticas de mercado e preparando para a atuação em equipes multidisciplinares.

O sistema também se relacionou com outras disciplinas do curso, como MAG1 e MAG2 aplicando a definição das entidades e dos relacionamentos do sistema. Já em MADS, os conceitos de planejamento de *sprints* e a priorização de requisitos, demonstrando a importância destes elementos para a organização e a entrega completa e funcional do sistema.

8.1 ARTEFATOS DO PROJETO

FIGURA 20 – TELAS ELABORADAS

The figure displays five screenshots of a web application interface, all titled 'Trabalho Final - WEB1'.

Screenshot 1: Alunos
 This screen shows a table with columns: Nome, CPF, E-mail, and Data de Nascimento. A green '+ Novo' button is in the top right. A blue 'Cursos' button is on the left. Below the table, it says 'Nenhum aluno cadastrado.'

Screenshot 2: Novo Aluno
 This is a form for adding a new student. It includes input fields for 'Nome:', 'CPF:', 'E-mail:', and 'Data de Nascimento:'. At the bottom are 'Salvar' and 'Voltar' buttons.

Screenshot 3: Alunos
 This screen shows the same table as Screenshot 1, but now it contains two rows of data. Each row has 'Editar' and 'Remover' buttons. A blue 'Cursos' button is on the left.

Screenshot 4: Editar Curso
 This is a form for editing a course. It includes input fields for 'Nome:' (containing 'Eng. Eletrica') and 'Link:' (containing 'www.ufpr.br/eng-el'). At the bottom are 'Atualizar' and 'Voltar' buttons.

Screenshot 5: Cursos
 This screen shows a table with columns: Nome and Link. A green '+ Novo' button is in the top right. Each row has 'Editar' and 'Remover' buttons. A blue 'Alunos' button is on the left.

FONTE: O autor (2025)

9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

O trabalho final da disciplina de *UX* foi o desenvolvimento de um protótipo visual de uma aplicação móvel para gerenciamento de medicamentos, com objetivo de auxiliar pacientes no controle de seus tratamentos de forma simples, eficiente e organizada. O objetivo foi para suprir as necessidades de quem precisa manter a rotina correta de um determinado tratamento, prevenindo esquecimentos e garantindo a sua eficácia.

Houve a aplicação prática de conceitos de *UX design*, como a facilidade de uso com interface clara e direta, a acessibilidade para pessoas de qualquer idade e um *design* sob medida para o usuário final, usando cores suaves e botões bem visíveis. A adoção de técnicas para criar soluções digitais e inovadoras, como as propostas por Levy (2021), foi fundamental para o projeto. Uma das etapas mais relevantes foi o trabalho conjunto com possíveis usuários finais, essa interação possibilitou o levantamento de melhorias, como incluir uma geração de relatórios PDF.

Foram elaboradas as telas de usuário, sendo o *login* por e-mail e senha com uma interface limpa e de fácil entendimento e usando cores que remetem à saúde e bem-estar. A tela principal com a lista dos medicamentos, com atalhos para novos itens e acesso rápido a um calendário. Clicando no atalho de novo medicamento, um formulário para o registro de novo medicamento, incluindo dosagem, nome, horários e o período inicial e final do tratamento. Uma tela adicional com um calendário exibindo as datas e horários das doses, ajudando na organização da rotina. E por fim uma tela de lembretes para mostrar as próximas notificações com horas exatas no telefone celular do usuário.

9.1 ARTEFATOS DO PROJETO

FIGURA 21 – TELAS ELABORADAS



FONTE: O autor (2025)

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

O desenvolvimento de aplicativos para a plataforma *Android* foi o principal aprendizado dessas disciplinas. A proposta previa desenvolver um aplicativo completo, desde a concepção da ideia, desenho das telas, implementação de funcionalidades até a realização de testes em dispositivos reais. Assim permitindo se beneficiar de práticas ágeis como lançamento de protótipos, versões parciais e ajustes conforme novas necessidades.

Esse campo de aplicativos se beneficia diretamente de abordagens ágeis, permite prototipar, lançar versões incrementais e coletar *feedback* dos usuários de forma rápida, assim ajustando ele de acordo com as novas demandas. Isso reforça o Manifesto Ágil, interagindo com a adaptação a mudanças, entrega contínua de valor e colaboração com o cliente (Prikladnicki et al., 2014).

Embora a execução final do aplicativo não tenha ocorrido, por se tratar de um desenvolvimento opcional, os conceitos essenciais foram amplamente explorados por meio de aulas síncronas, exercícios e atividades avaliativas. O conteúdo abordou fundamento da linguagem *Kotlin* e o uso do *Android Studio* até as boas práticas de desenvolvimento e publicação de aplicativos.

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

No projeto de infraestrutura para desenvolvimento e implantação de *software* foi proposto a configuração de um ambiente de integração contínua, usando ferramentas como *Docker*, *GitLab* e *Jenkins*. A ideia foi montar, na prática, uma estrutura para equipes de desenvolvimento para viabilizar a entrega de *software* rápida, confiável e com menor risco de falhas.

Enquanto *Scrum* e *Kanban* ajudam a organizar as entregas e determinar suas prioridades, o *DevOps* garante que tudo isso ocorra de maneira automatizada e consistente. Isso reduz o tempo gasto entre criar uma nova funcionalidade e a sua disponibilização ao cliente final.

Durante as aulas e a elaboração do projeto final, aplicamos na prática conceitos chamados de “Três caminhos”, descritos por Kim et al. (2018):

- **Fluxo:** O provisionamento do *Docker* e *GitLab* auxilia na criação de um fluxo de trabalho simplificado e padronizado desde o versionamento do código até a sua centralização em um único local.
- **Feedback:** A integração com o *Jenkins* permite acionar *builds* e testes automatizados a cada *commit* dos desenvolvedores, assim possibilitando identificar problemas e corrigi-los antes do envio para a produção.
- **Aprendizado contínuo:** O ambiente é seguro e automatizado, logo estimula melhorias e experimentação pelos desenvolvedores e time de garantia de qualidade (QA) de *software*, diminuindo o custo de erros em produção e o tempo gasto em processos e por consequência garantindo melhoria no produto final.

11.1 ARTEFATOS DO PROJETO

FIGURA 22 – INSTALAÇÃO DO GITLAB

```

31/01/2025 23:39:50 /home/mobaxterm docker run -d --name 40001016398E1 -p 22:22 -p 80:80 -p 443:443 -p 9091:9091 dfwandarti/gitlab_jenki
ns:3
Unable to find image 'dfwandarti/gitlab_jenkins:3' locally
3: Pulling from dfwandarti/gitlab_jenkins
d7bfe07ed847: Pull complete
25bb9590d9c9: Pull complete
d46df4aa614d: Pull complete
c695fd85ed38: Pull complete
3f36ee87421e: Pull complete
c0436bfc19a7: Pull complete
061a5147aada: Pull complete
5d4e92472abe: Downloading [=====>] 417.2MB/1.259GB
Scf33ab37f3b: Downloading [=====>] 209MB/864MB
b42364568e18: Downloading [=====>] 61.78MB/62.85MB
[
]

31/01/2025 23:39:50 /home/mobaxterm docker run -d --name 40001016398E1 -p 22:22 -p 80:80 -p 443:443 -p 9091:9091 dfwandarti/gitlab_jenki
ns:3
Unable to find image 'dfwandarti/gitlab_jenkins:3' locally
3: Pulling from dfwandarti/gitlab_jenkins
d7bfe07ed847: Pull complete
25bb9590d9c9: Pull complete
d46df4aa614d: Pull complete
c695fd85ed38: Pull complete
3f36ee87421e: Pull complete
c0436bfc19a7: Pull complete
061a5147aada: Pull complete
5d4e92472abe: Pull complete
Scf33ab37f3b: Pull complete
b42364568e18: Pull complete
Digest: sha256:c8f21731114d5e795315534e827d13a09a2c63d8956eedac53cca475c6e3780
Status: Downloaded newer image for dfwandarti/gitlab_jenkins:3
32b90363a103dc6f8756f4f7b6428d4606ae269f59d6260ab8db05f227e2c64

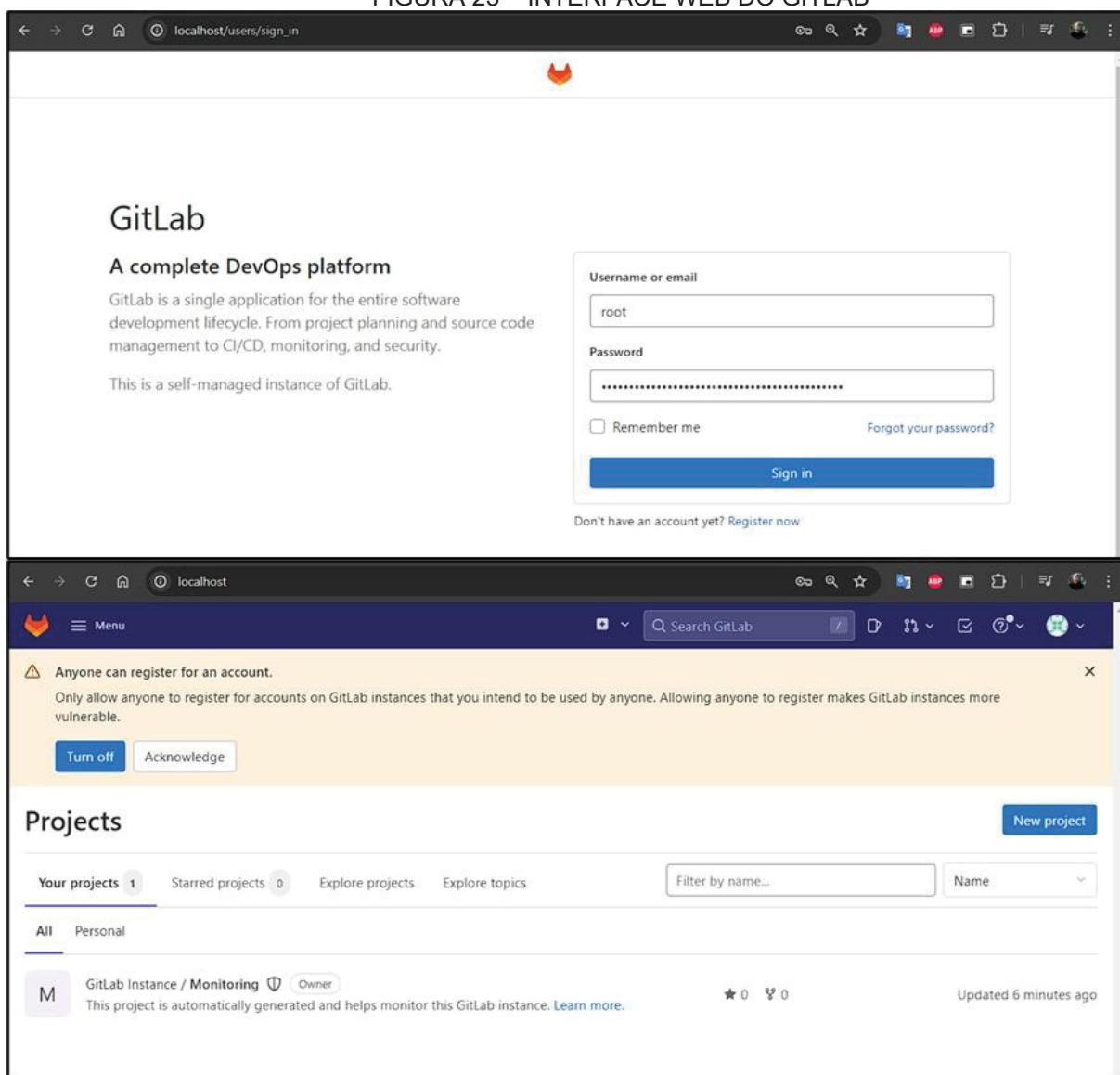
31/01/2025 23:43:19 /home/mobaxterm docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS                    PORTS
32b90363a103   dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       2 minutes ago Up About a minute (health: starting) 0.0.0.0:22→22/tcp, 0.0.0.0:80→80/tcp, 0.0.0.0:443→443/tcp, 0.0.0.0:9091→9091/tcp

31/01/2025 23:44:19 /home/mobaxterm
31/01/2025 23:46:27 /home/mobaxterm docker exec -it 32b90363a103 /bin/bash
root@32b90363a103:/#
root@32b90363a103:/#
root@32b90363a103:/#
root@32b90363a103:/# ls
RELEASE  assets  bin      boot  dev  etc  home  jenkins  lib  lib32  lib64  libx32  linuxrc  media  mnt  opt  proc  root  run /sbin  srv  sys  usr  var
root@32b90363a103:/# cd /etc/gitlab/
gitlab-secrets.json      initial_root_password  ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub
gitlab.rb                ssh_host_ecdsa_key    ssh_host_ed25519_key    ssh_host_rsa_key          trusted-certs/
root@32b90363a103:/# cat /etc/gitlab/initial_root_password
# WARNING: This value is valid only in the following conditions
# 1. If provided manually (either via 'GITLAB_ROOT_PASSWORD' environment variable or via 'gitlab_rails['initial_root_password']' setting in 'gitlab.rb'
# , it was provided before database was seeded for the first time (usually, the first reconfigure run).
# 2. Password hasn't been changed manually, either via UI or via command line.
#
# If the password shown here doesn't work, you must reset the admin password following https://docs.gitlab.com/ee/security/reset_user_password.html#res
et-your-root-password.
Password: gM0nIuqB9yngj/1e+1V+ENZYKNIp/LPZrjdMikjfoI=
# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
root@32b90363a103:/#

```

FONTE: O autor (2025)

FIGURA 23 – INTERFACE WEB DO GITLAB



FONTE: O autor (2025)

FIGURA 24 – INICIANDO UM NOVO MERGE

```

31/01/2025 23:56:13 /home/trabalho_final_devops ls -la
total 1
drwxr-xr-x 1 197609 197609 0 Jan 31 23:56 .
drwxr-xr-x 1 197609 197609 0 Jan 31 23:54 ..
-rw-r--r-- 1 197609 197609 178 Jan 31 23:56 README.md

31/01/2025 23:56:17 /home/trabalho_final_devops git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/trabalho_final_devops/.git/

31/01/2025 23:56:30 /home/trabalho_final_devops master git remote add origin http://localhost/gitlab-instance-eaf6614e/Monitoring.git

31/01/2025 23:57:25 /home/trabalho_final_devops master git add README.md

31/01/2025 23:58:18 /home/trabalho_final_devops master git config --global user.email "joao_lucas1612@hotmail.com"

31/01/2025 23:58:38 /home/trabalho_final_devops master git config --global user.name "Joao Lucas"

31/01/2025 23:58:47 /home/trabalho_final_devops master git commit -m "Adicionando README para o trabalho final de INFRA"
[master (root-commit) 6dea97e] Adicionando README para o trabalho final de INFRA
1 file changed, 3 insertions(+)
create mode 100644 README.md

31/01/2025 23:58:50 /home/trabalho_final_devops master git push -u origin master
Username for 'http://localhost':
Password for 'http://root@localhost':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 401 bytes | 401.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://localhost/gitlab-instance-eaf6614e/Monitoring.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.

```

FONTE: O autor (2025)

FIGURA 25 – VERIFICAÇÃO DO MERGE VIA INTERFACE WEB

The figure consists of two screenshots of the GitLab web interface, illustrating the process of verifying a merge via the web interface.

Top Screenshot: The browser address bar shows the URL `localhost/gitlab-instance-eaf6614e/Monitoring`. The page displays the project overview for the `Monitoring` project. The commit `6dea97ef` is highlighted, authored by `Joao Lucas` 3 minutes ago. The commit message is "Adicionando README para o trabalho final de INFRA". The commit details show the file `README.md` with the content:

```
Hello World

Trabalho final da materia DAS2024 - INFRA - Infraestrutura para desenvolvimento e implantação de Software (DevOps)

Aluno: João Lucas Della Coletta Frangella
```

Bottom Screenshot: The browser address bar shows the URL `localhost/gitlab-instance-eaf6614e/Monitoring/-/commit/6dea97ef58d0943ebe93f6b11ebc5979bbdc77`. The page displays the commit details for `6dea97ef`, authored by `Joao Lucas` 4 minutes ago. The commit message is "Adicionando README para o trabalho final de INFRA". The commit details show the file `README.md` with the content:

```
1 + # Hello World
2 + # Trabalho final da materia DAS2024 - INFRA - Infraestrutura para desenvolvimento e implantação de Software (DevOps)
3 + # Aluno: João Lucas Della Coletta Frangella
```

FONTE: O autor (2025)

12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

A disciplina de testes automatizados teve foco na criação de um teste *end-to-end* usando o *framework Playwright*, aplicado a um site *web* chamado Anotepad. O objeto foi validar o processo de criação de uma nova nota, automatizando e usando os conceitos das aulas em um cenário real de verificação em uma ferramenta funcional.

Os testes automatizados, sejam no *front-end* ou diretamente em funções e classes garante a identificação de falhas mais cedo, aumenta a confiabilidade dos sistema e são indispensáveis para equipes que querem trabalhar com entregas rápidas e incrementais. A ferramenta, um *framework* de testes *end-to-end* para aplicativos *web* modernos (Microsoft, 2025), foi essencial para a validação das funcionalidades do projeto.

Esse tipo de teste abordado no trabalho final pode ser classificado com grande valia, apesar de demandar mais tempo em comparação com testes unitários a abordagem *end-to-end* é essencial para garantir que todas as integrações entre o *back-end* e *front-end* estão satisfeitas e garantindo a experiencia do usuário.

12.1 ARTEFATOS DO PROJETO

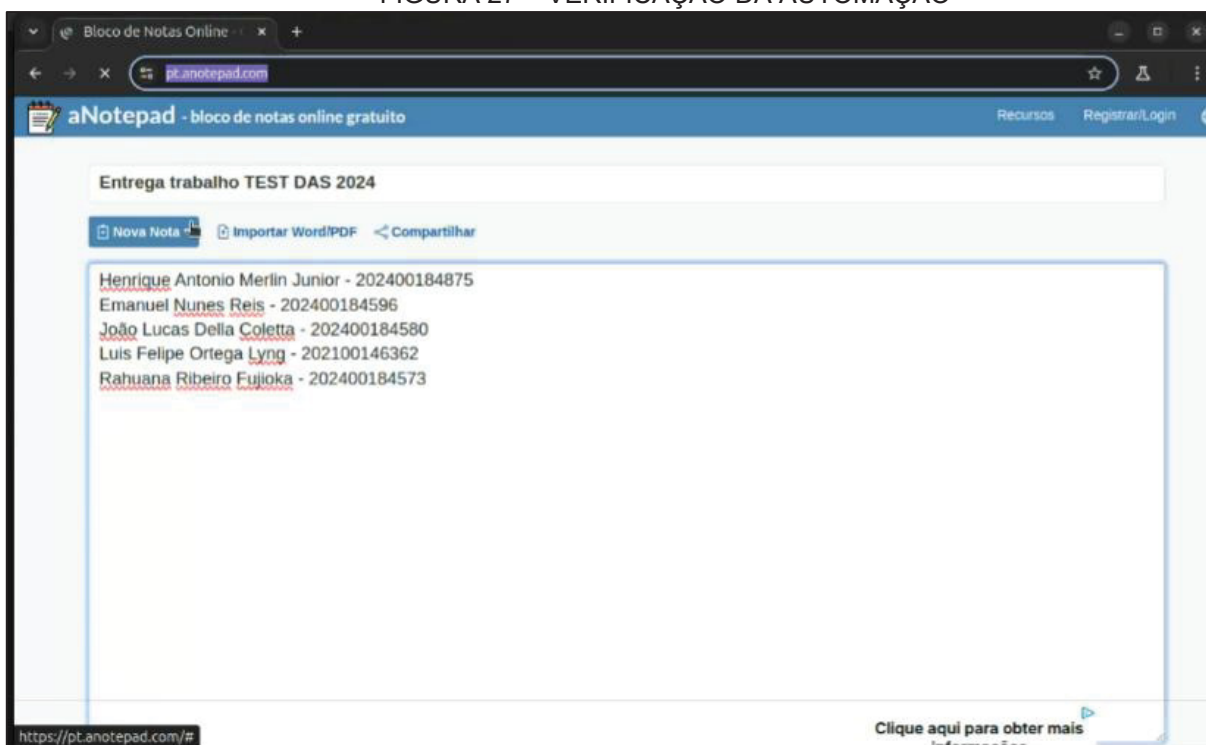
FIGURA 26 – CÓDIGO DE TESTE

```
import { test } from '@playwright/test';

test('test', async ({ page }) => {
  await page.goto('https://pt.anotepad.com/', {waitUntil:
'domcontentloaded'});
  await page.getByRole('textbox', { name: 'Título da Nota' }).fill('Entrega
trabalho TEST DAS 2024');
  await page.getByRole('textbox', { name: 'Conteúdo da Nota' }).fill('Henrique
Antonio Merlin Junior - 202400184875\nEmanuel Nunes Reis - 202400184596\nJoão
Lucas Della Coletta - 202400184580\nLuis Felipe Ortega Lyng -
202100146362\nRahuana Ribeiro Fujioka - 202400184573\n');
  await page.getByRole('button', { name: 'Salvar' }).click();
});
```

FONTE: O autor (2025)

FIGURA 27 – VERIFICAÇÃO DA AUTOMAÇÃO



FONTE: O autor (2025)

13 CONCLUSÃO

Os projetos realizados ao longo das disciplinas apresentadas neste memorial reforçam a importância da integração entre a parte teórica e a prática no desenvolvimento ágil de *software*. Cada projeto ajudou a consolidar uma visão completa do processo, como a modelagem, programação, banco de dados, qualidade do código, testes automatizados, experiência do usuário e a integração contínua para um único objetivo, a entrega final com valor, qualidade e eficiência.

O memorial demonstra a aplicação prática de conceitos de desenvolvimento ágil de *software*, como modelagem, programação, banco de dados, testes automatizados, experiência do usuário e integração contínua. As disciplinas iniciais de programação e banco de dados serviram como base, apresentando conceitos de lógica e modelagem de informações que deram consistência à arquitetura de *software*. Já as disciplinas de modelagem ágil e de aspectos ágeis de programação aprofundaram a importância de um código limpo e reutilizável para a evolução do sistema. As disciplinas de desenvolvimento *web* e *mobile* utilizaram *frameworks* modernos como *Angular* e *Spring Boot* para o desenvolvimento de aplicações focadas no usuário final. As disciplinas de UX e testes foram o ponto alto para a experiência do usuário e a qualidade do nosso trabalho. Aprendeu-se, na prática, a usar protótipos para que o *design* conversasse diretamente com a usabilidade do cliente. E, com o *framework Playwright*, conseguimos ter a certeza de que cada funcionalidade era confiável. No final, a disciplina de infraestrutura (*DevOps*) foi a que juntou todas as peças do quebra-cabeça, mostrando como ferramentas como *Docker*, *GitLab* e *Jenkins* simplificam de verdade o nosso trabalho, do primeiro código à entrega final.

Houve alguns desafios durante os desenvolvimentos, como alinhar os requisitos técnicos diante de cenários com mudanças rápidas e a dificuldade em manter qualidade nos códigos sem a aplicação de testes. Apesar disso, a experiência adquirida mostrou que quando se tem os conceitos bem aplicados, os princípios ágeis trazem ganhos significativos tanto para a equipe de desenvolvimento quanto para o usuário final da aplicação.

REFERÊNCIAS

ANDERSON, David J. **Kanban: mudança evolucionária de sucesso para seu negócio de tecnologia**. São Paulo: Blue Hole Press, 2011.

ANGULAR. **Deliver web apps with confidence**. Disponível em: <http://angular.io>. Acesso em: 10 ago. 2025.

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

DATE, C. J. **Introdução a sistemas de banco de dados**. 8. ed. Rio de Janeiro: LTC, 2004.

ECKEL, Bruce. **Thinking in Java**. 4. ed. São Paulo: Prentice Hall, 2006.

GUILLEANES, T. A. **UML: uma abordagem prática**. São Paulo: Novatec, 2013.

KANBAN **Board Game**. In: [site]. [S. l.]: [s. n.]. Disponível em: <http://www.kanbanboardgame.com/>. Acesso em: 18 maio 2024.

KIM, Gene; HUMBLE, Jez; DEBOIS, Patrick; WILLIS, John. **O Projeto Fênix: Um Romance Sobre TI, DevOps e a Transformação Digital que Está Revolucionando o Mundo**. São Paulo: Alta Books, 2018.

LEVY, Jaime. **Estratégia de UX: Técnicas de Estratégia de Produto Para Criar Soluções Digitais Inovadoras**. São Paulo: Novatec Editora, 2021.

MARTIN, Robert C. **Clean code: a handbook of agile software craftsmanship**. Upper Saddle River: Prentice Hall, 2009.

MICROSOFT. **Playwright: Fast and reliable end-to-end testing for modern web apps**. Disponível em: <https://playwright.dev/>. Acesso em: 14 ago. 2025.

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

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.

RUMBAUGH, James. **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1991.

SPRING. **Spring Boot Reference Documentation**. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 10 ago. 2025.