

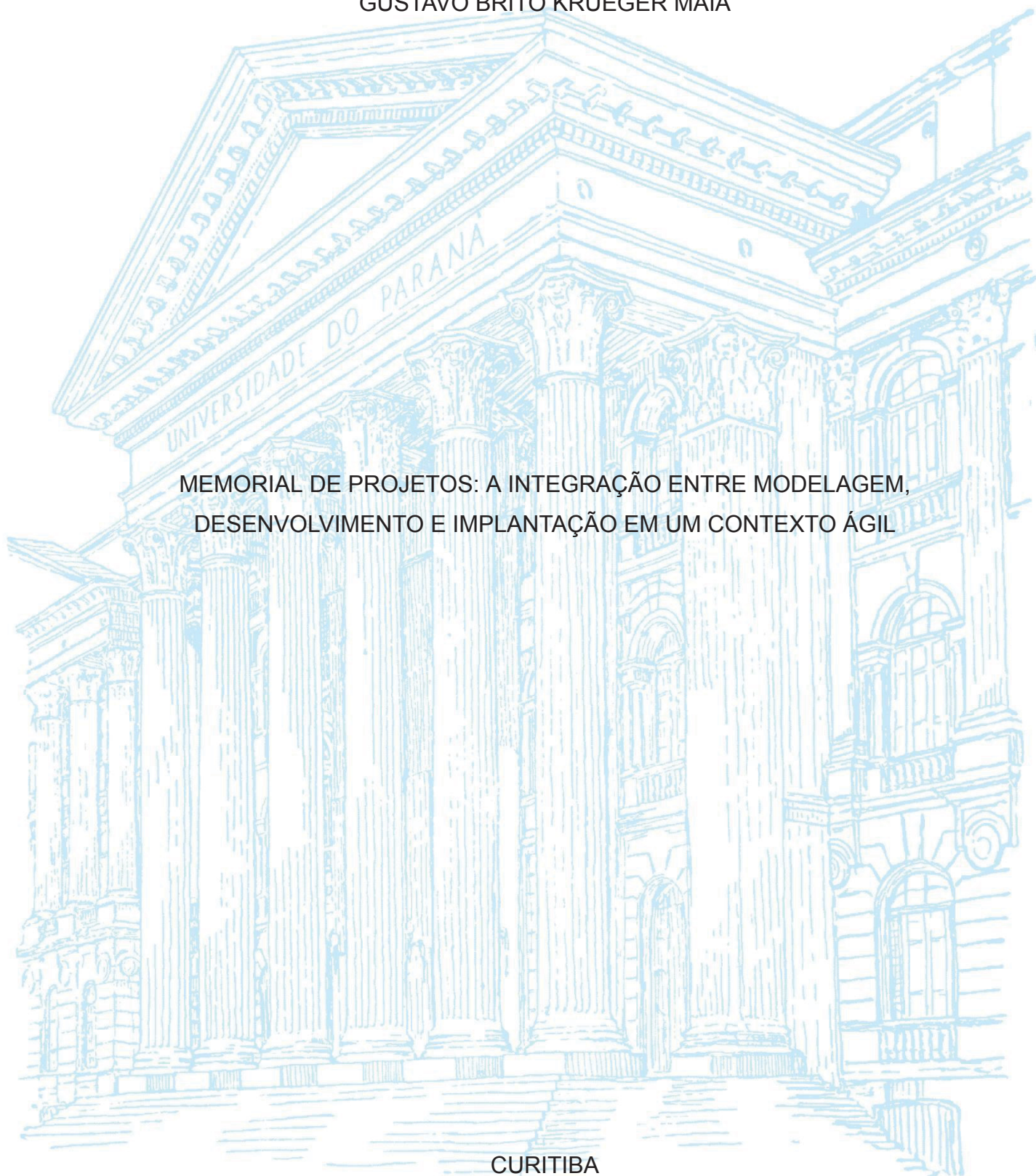
UNIVERSIDADE FEDERAL DO PARANÁ

GUSTAVO BRITO KRUEGER MAIA

MEMORIAL DE PROJETOS: A INTEGRAÇÃO ENTRE MODELAGEM,
DESENVOLVIMENTO E IMPLANTAÇÃO EM UM CONTEXTO ÁGIL

CURITIBA

2025



GUSTAVO BRITO KRUEGER MAIA

MEMORIAL DE PROJETOS: A INTEGRAÇÃO ENTRE MODELAGEM,
DESENVOLVIMENTO E IMPLANTAÇÃO EM UM CONTEXTO ÁGIL

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

Orientador: Prof. 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 **GUSTAVO BRITO KRUEGER MAIA**, intitulada: **MEMORIAL DE PROJETOS: A INTEGRAÇÃO ENTRE MODELAGEM, DESENVOLVIMENTO E IMPLANTAÇÃO EM UM CONTEXTO ÁGIL**, 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, 19 de Março de 2025.

RAFAELA MANTOVANI FONTANA
Presidente da Banca Examinadora

JAIME WOJCIECHOWSKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Este trabalho enfatiza a importância da formação em Desenvolvimento Ágil de Software para profissionais do mercado de tecnologia, destacando a integração entre disciplinas e projetos que se complementam. A abordagem engloba desde as etapas iniciais do desenvolvimento, como planejamento e modelagem, até aspectos avançados, incluindo infraestrutura, implementação e monitoramento de software, de forma prática, concisa e progressiva, preparando os alunos para atuar em um ambiente dinâmico e alinhado às demandas do setor.

Palavras-chave: desenvolvimento de software, metodologias ágeis, tecnologia da informação, planejamento.

ABSTRACT

This document emphasizes the importance of training in Agile Software Development for professionals in the technology market, highlighting the integration of complementary disciplines and projects. The approach covers everything from the initial stages of development, such as planning and modeling, to advanced aspects, including infrastructure, implementation, and software monitoring. All of this is explored in a practical, concise, and progressive manner, preparing students to operate in a dynamic environment aligned with industry demands.

Keywords: software development, agile methodology, information technology, planning.

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	13
3.1 ARTEFATOS DO PROJETO	14
4 DISCIPLINA: GAP1 E GAP2– GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2	18
4.1 ARTEFATOS DO PROJETO	19
5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO	21
5.1 ARTEFATOS DO PROJETO	22
6 DISCIPLINA: BD – BANCO DE DADOS	23
6.1 ARTEFATOS DO PROJETO	24
7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO	27
7.1 ARTEFATOS DO PROJETO	28
8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2	29
8.1 artefatos do projeto	30
9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE	32
9.1 ARTEFATOS DO PROJETO	33
10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2	35
10.1 ARTEFATOS DO PROJETO	36
11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)	39
11.1 ARTEFATOS DO PROJETO	40
12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS	42
12.1 ARTEFATOS DO PROJETO	43
13 CONCLUSÃO	44

1 PARECER TÉCNICO

A análise dos projetos desenvolvidos ao longo da pós-graduação em Desenvolvimento Ágil de Software evidencia a forte integração entre as disciplinas, proporcionando uma visão abrangente e aplicada do desenvolvimento ágil. Desde a modelagem inicial até a implementação, testes e implantação, o curso enfatizou abordagens iterativas, adaptáveis e colaborativas, essenciais para a entrega contínua de valor no desenvolvimento de software.

Conforme o PMBOK 7 (PMI, 2021), a entrega de valor está diretamente ligada à adoção de sistemas dinâmicos e adaptáveis, garantindo que produtos e serviços atendam às necessidades dos stakeholders de maneira eficiente e escalável. Essa perspectiva se alinha aos princípios ágeis aplicados nos projetos desenvolvidos ao longo do curso, nos quais flexibilidade, eficiência e colaboração foram fatores determinantes para o sucesso das implementações.

Os projetos desenvolvidos em Introdução a Programação (Capítulo 5) e Desenvolvimento Web 1 e 2 (Capítulo 8) demonstraram, na prática, a importância da escalabilidade e flexibilidade no desenvolvimento de software, utilizando conceitos fundamentais de arquitetura e integração de sistemas. Complementando-os, as estratégias de organização, métricas e gerenciamento de software foram fundamentadas pelos conceitos abordados em Gestão de Projetos 1 e 2 (Capítulo 4).

Esses projetos tiveram como base as modelagens elaboradas em Modelagem de Software 1 e 2 (Capítulo 3), utilizando como base conceitos da UML (Unified Modeling Language), consolidando práticas vistas nas disciplinas de Banco de Dados (Capítulo 6). Além de estruturar a arquitetura do sistema, a utilização da UML também auxilia na definição e organização dos dados, garantindo um alinhamento eficiente entre o modelo conceitual e a estrutura do banco de dados, o que, de acordo com Guedes (2018), permite dar enfoque a organização estrutural do sistema ou até mesmo as necessidades físicas desse sistema, o que acaba contribuindo diretamente com a questão da agilidade no desenvolvimento de software. Dessa forma, reforça-se a integração entre as disciplinas do curso, demonstrando a contribuição da modelagem e planejamento para o desenvolvimento de soluções escaláveis, de maneira ágil.

Da mesma forma, os projetos de Desenvolvimento Mobile 1 e 2 (Capítulo 10) conectaram a importância da experiência do usuário (*User Experience* - Capítulo 9) no design de interfaces, garantindo a usabilidade e acessibilidade das aplicações. Além disso, disciplinas como Métodos Ágeis para Desenvolvimento de Software, Aspectos Ágeis de Programação, Infraestrutura para Desenvolvimento e Implantação de Software e Testes Automatizados (Capítulos 2, 7, 11 e 12, respectivamente) interligam-se para apresentar não apenas conceitos, mas também estratégias práticas para a construção de sistemas robustos, desde a definição da arquitetura até a testabilidade e manutenção do código.

Esse alinhamento com as disciplinas reflete a ideia defendida por Willis et al. (2016), que destaca a importância de entregar software de forma contínua e eficiente, sem interrupções no serviço ou impactos negativos na experiência do usuário. Como os autores apontam, o valor real só é criado quando os serviços estão em produção e operando de maneira confiável, mas que devem estar em um ambiente estável, em um fluxo rápido porém não caótico.

A experiência adquirida ao longo do curso evidencia a importância da integração entre modelagem, desenvolvimento e implantação no contexto do desenvolvimento ágil de software. Esse conhecimento consolidado prepara o profissional para atuar de forma estratégica no mercado de trabalho, garantindo que os sistemas desenvolvidos sejam eficientes, escaláveis e adaptáveis às mudanças, características fundamentais para o sucesso em ambientes ágeis.

2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

O projeto de Métodos Ágeis para Desenvolvimento de Software foi fundamental para a aplicação prática dos conceitos do desenvolvimento ágil. Durante a disciplina, exploramos diversos aspectos da Engenharia de Software, desde modelos tradicionais até abordagens modernas amplamente utilizadas em projetos reais.

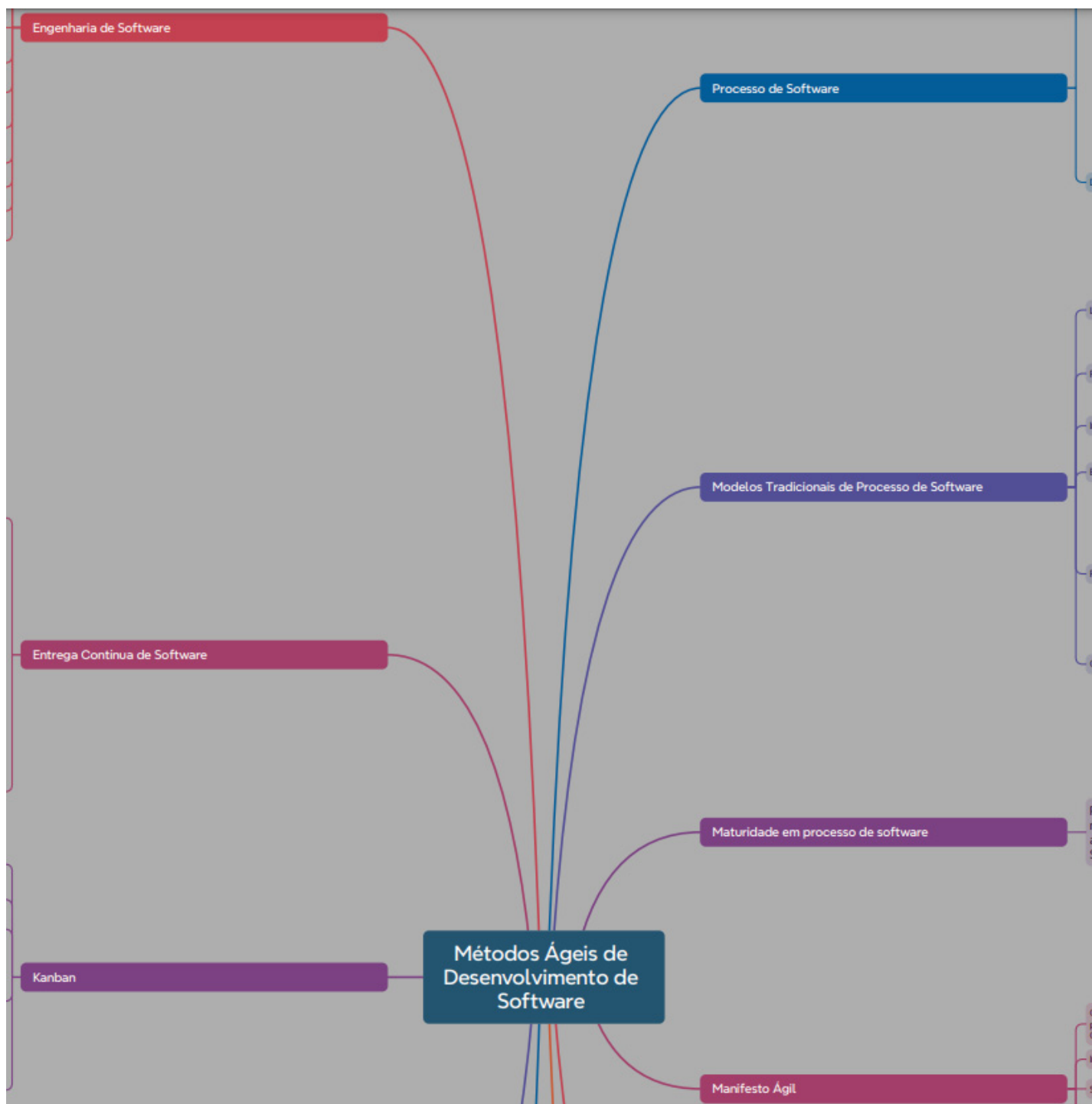
Foram apresentadas e aplicadas técnicas e metodologias como *Scrum*, *Kanban*, *Lean* e *Extreme Programming*, permitindo estruturar e gerenciar atividades de forma eficiente. O foco na entrega de valor iterativa, por meio de ciclos curtos de desenvolvimento, proporcionou uma compreensão clara sobre a importância da adaptação e melhoria contínua.

Além disso, o projeto aprofundou os princípios do Manifesto Ágil, detalhando seus valores e etapas necessárias para atingir maior maturidade no desenvolvimento de software. Esse conhecimento teórico aliado à prática fortaleceu a capacidade de planejamento e execução de projetos com eficiência.

A integração desses conceitos com outras disciplinas é evidente, pois oferece uma base essencial para o planejamento e desenvolvimento de software de forma ágil. Dessa forma, a disciplina não apenas amplia o entendimento sobre métodos modernos, mas também contribui para a formação de profissionais preparados para os desafios reais da área de tecnologia.

2.1 ARTEFATOS DO PROJETO

FIGURA 1 - ARTEFATO MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.



FONTE: O autor (2025).

FIGURA 2 - ARTEFATO MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.



FONTE: O autor (2025).

FIGURA 3 - ARTEFATO MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.



FONTE: O autor (2025).

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

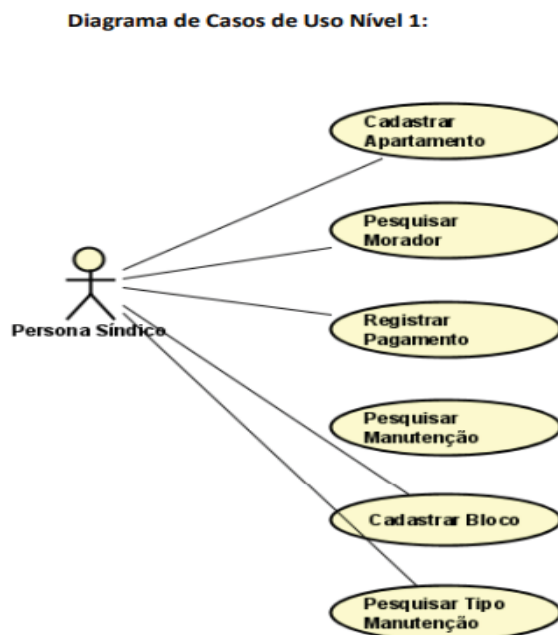
Os projetos de Modelagem Ágil de Software 1 e 2 proporcionaram ao aluno um conhecimento aprofundado sobre as principais formas de documentação e modelagem de software desde sua concepção inicial. Utilizando como base a UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada), os projetos abordaram desde o gerenciamento de requisitos até a criação de Histórias de Usuário, desenvolvimento de tarefas essenciais para o cumprimento do projeto e documentação através de diversos diagramas, como casos de uso níveis 1 e 2, diagrama de classes e diagrama de sequência. Além disso, contemplaram a prototipação de telas e a definição de critérios de aceitação para testes.

O projeto utilizou como estudo de caso, um sistema de gestão de condomínios, sendo um síndico o usuário principal. Utilizando os conceitos apresentados na disciplina, foram modeladas e documentadas funcionalidades essenciais, incluindo o cadastro e pesquisa de moradores, cadastro e pesquisa de apartamentos, gerenciamento de blocos do condomínio, registro de pagamentos e controle de manutenções.

Essa abordagem permitiu a aplicação prática dos conceitos de modelagem ágil, tornando o aprendizado mais dinâmico e alinhado às necessidades do mercado de desenvolvimento de software.

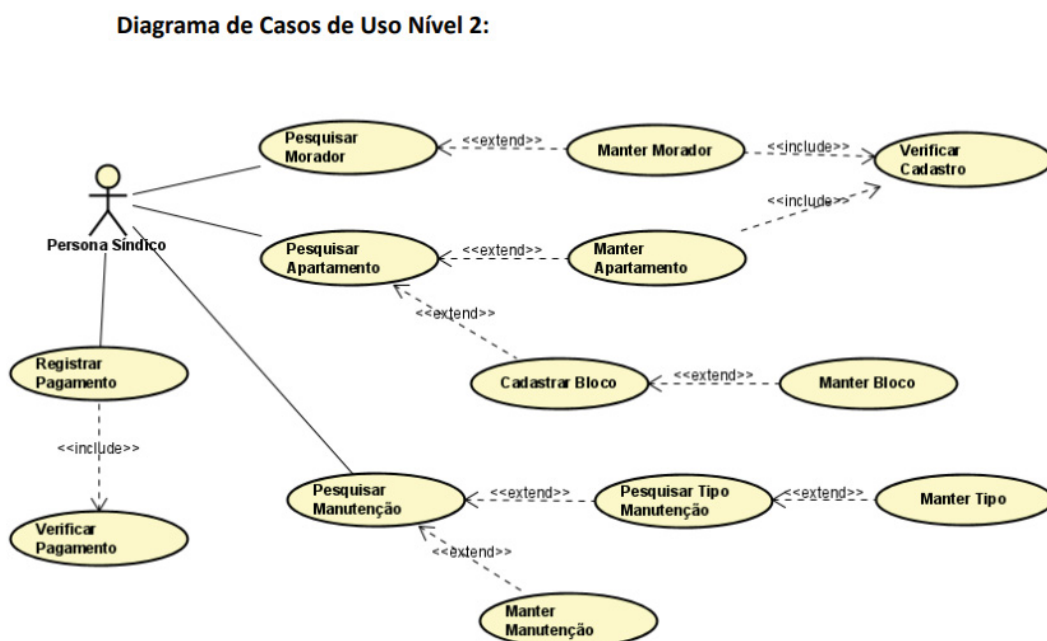
3.1 ARTEFATOS DO PROJETO

FIGURA 4 - ARTEFATO MODELAGEM ÁGIL DE SOFTWARE.



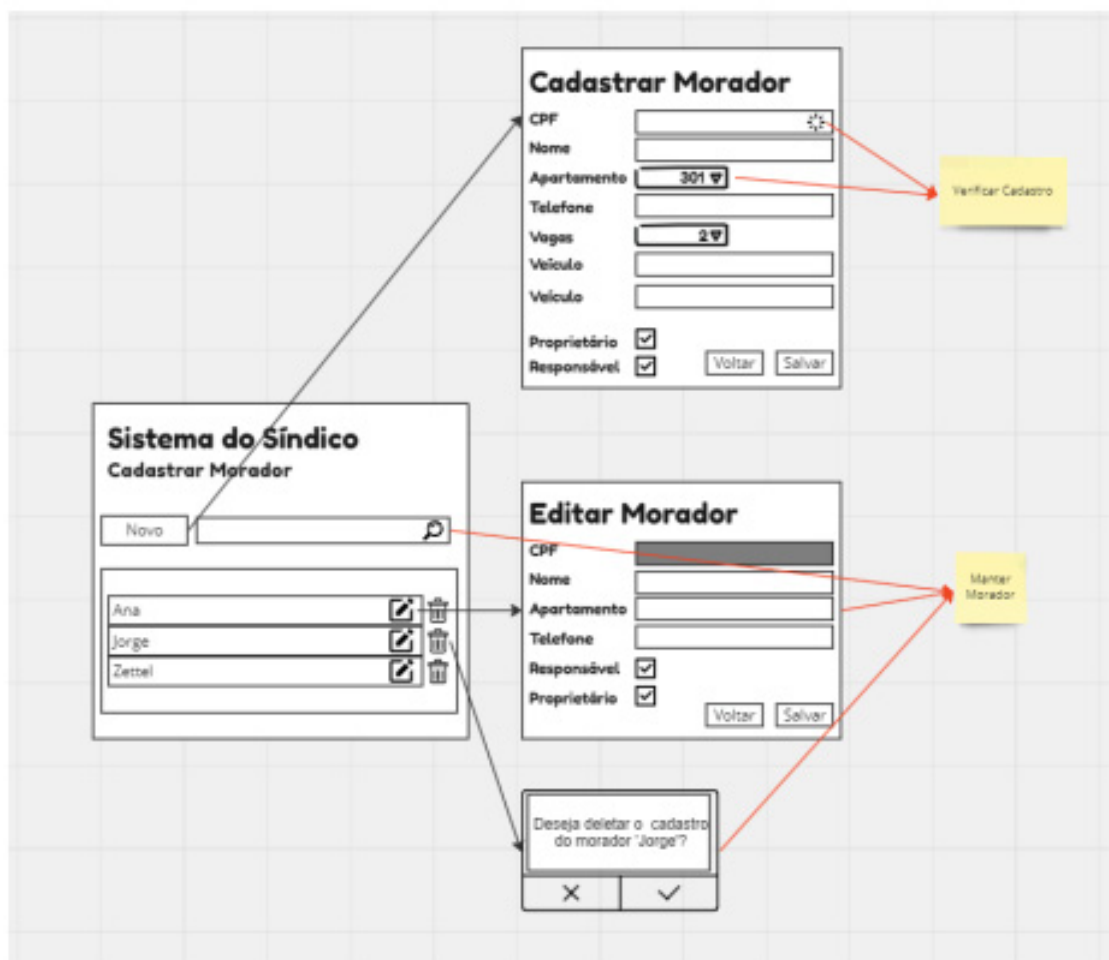
FONTE: O autor (2025).

FIGURA 5 - ARTEFATO MODELAGEM ÁGIL DE SOFTWARE.



FONTE: O autor (2025).

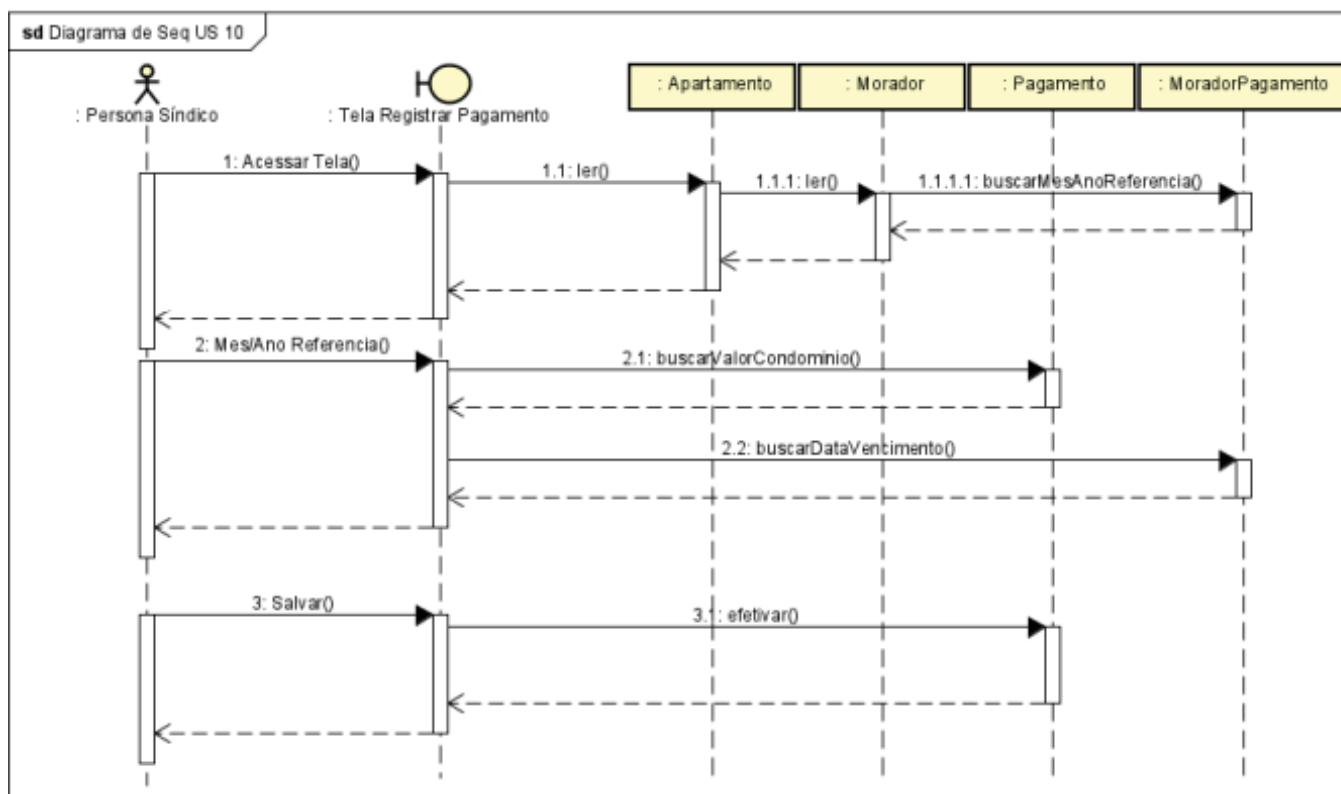
FIGURA 6 - ARTEFATO MODELAGEM ÁGIL DE SOFTWARE.

**US 001 - PESQUISAR MORADOR****SENDO** um Síndico,**QUERO** Pesquisar os dados dos moradores**PARA** fazer manutenção nos seus dados**Critérios de aceitação:**

1. Deve preencher a tabela com os dados cadastrados
2. Deve permitir incluir um novo morador
3. Deve permitir alterar os dados de um morador cadastrado
4. Deve permitir excluir um morador cadastrado
5. Deve permitir consultar os dados de um morador cadastrado
6. Deve pesquisar os candidatos na tabela da tela

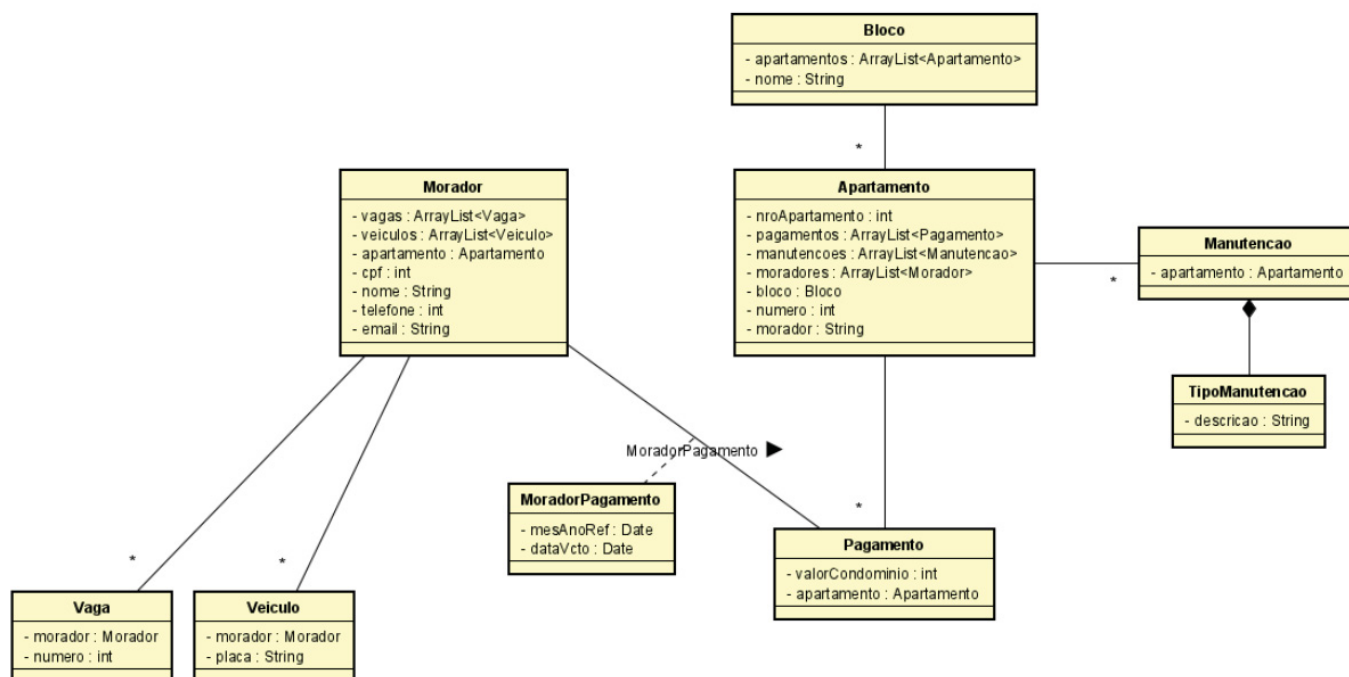
FONTE: O autor (2025).

FIGURA 7 - ARTEFATO MODELAGEM ÁGIL DE SOFTWARE.

Diagrama de Sequência:

FONTE: O autor (2025).

FIGURA 8 - ARTEFATO MODELAGEM ÁGIL DE SOFTWARE.

Diagrama de Classes:

FONTE: O autor (2025).

FIGURA 9 - ARTEFATO MODELAGEM ÁGIL DE SOFTWARE.

Critérios de aceitação para Testes:**Critério 1 - Deve carregar o combo mês/ano referência antes de apresentar a tela**

DADO QUE	O usuário
QUANDO	a tela é acionada
ENTÃO	O sistema carrega o combo Mês/Ano Referência conforme (R1)

Critério 2 - Não deve permitir apartamento não cadastrado

DADO QUE	Síndico informa o apartamento N°1000
QUANDO	preenche o campo número do apartamento
ENTÃO	O sistema emite mensagem "apartamento não cadastrado"

Critério 3 - Deve preencher o CPF, Nome e telefone do morador do apartamento informado

DADO QUE	foi informado o apartamento 3 no campo número do apartamento
QUANDO	quando o foco sai do campo apartamento
ENTÃO	O sistema busca os dados do apartamento e preenche conforme (R2)

Critério 4 - Deve apresentar o valor e vencimento do condomínio do mês/ano de referência selecionado no combo

DADO QUE	o combo Mês/Ano foi preenchido
QUANDO	o síndico selecionar o Mês/Ano 02/2023
ENTÃO	O sistema preenche os campos valor '1000R\$' e vencimento do condomínio '10/02/2023'

Critério 5 - Deve efetivar o registro do pagamento do condomínio

DADO QUE	Os campos foram preenchidos corretamente
QUANDO	O botão 'pagar' for pressionado
ENTÃO	O sistema inclui o registro do pagamento e emite a mensagem "pagamento realizado com sucesso"

FONTE: O autor (2025).

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

Nas disciplinas de Gerenciamento Ágil de Projetos 1 e 2, são apresentados conceitos fundamentais sobre como planejar projetos com agilidade e medir essa agilidade, comparando modelos tradicionais de gerenciamento com abordagens contemporâneas. São exemplificados os benefícios e a implementação dessas metodologias. Tópicos como introdução ao Scrum, Kanban, mapeamento de funcionalidades e histórias de usuário, estimativas, planejamento de releases e análise de pontos são amplamente aprofundados. Além disso, as disciplinas abordam qualidade de software, incluindo testes unitários e de integração, além da identificação de débitos técnicos e estratégias para evitar inconsistências no código. Esses conceitos são fundamentados, principalmente, no PMBOK (Project Management Body of Knowledge) – sétima edição.

Os projetos desenvolvidos nessas disciplinas são complementares. O primeiro consiste em uma simulação de planejamento de *release*, aplicada a um estudo de caso de um sistema de gestão de condomínios. Esse projeto, com duração média de duas semanas, inclui histórias de usuário previamente estimadas. Com base nelas, são calculadas a soma dos pontos das funcionalidades, a velocidade por sprint e as horas disponíveis para trabalho.

Já o segundo projeto envolve uma simulação em tempo real utilizando o software Kanban Board Game. Nele, os alunos enfrentam uma situação simulada de uma empresa que utiliza Kanban. O objetivo é organizar o quadro de tarefas de forma eficiente ao longo de 30 dias simulados, resultando, ao final, na geração de um gráfico CFD (Cumulative Flow Diagram).

Esses projetos representam exemplos práticos da aplicação de metodologias ágeis no ambiente corporativo, proporcionando aos alunos uma experiência enriquecedora e preparando-os para lidar com desafios reais no gerenciamento de projetos ágeis.

4.1 ARTEFATOS DO PROJETO

FIGURA 10 - ARTEFATO GERENCIAMENTO ÁGIL DE PROJETOS.

Considerações:

- 1 ponto equivale a 8 horas.
- A Sprint tem 2 semanas, o que equivale a 10 dias úteis (5 dias por semana).
- Tamanho da Equipe: 1 pessoa.
- Features somam 41 pontos / 10 velocidade da Equipe = 5 iterações.
- Tamanho do Projeto: 2 meses e 8 dias aproximadamente, calculado de acordo com velocidade da equipe e pontos totais das Histórias de Usuário.
- Software escolhido: Sistema de Gestão de Condomínios

Cálculo da Velocidade:

Horas disponíveis por dia:	8 horas	Tamanho da Sprint:	2 Semanas
Horas disponíveis por Sprint:	80 horas	Velocidade:	10 pontos / sprint

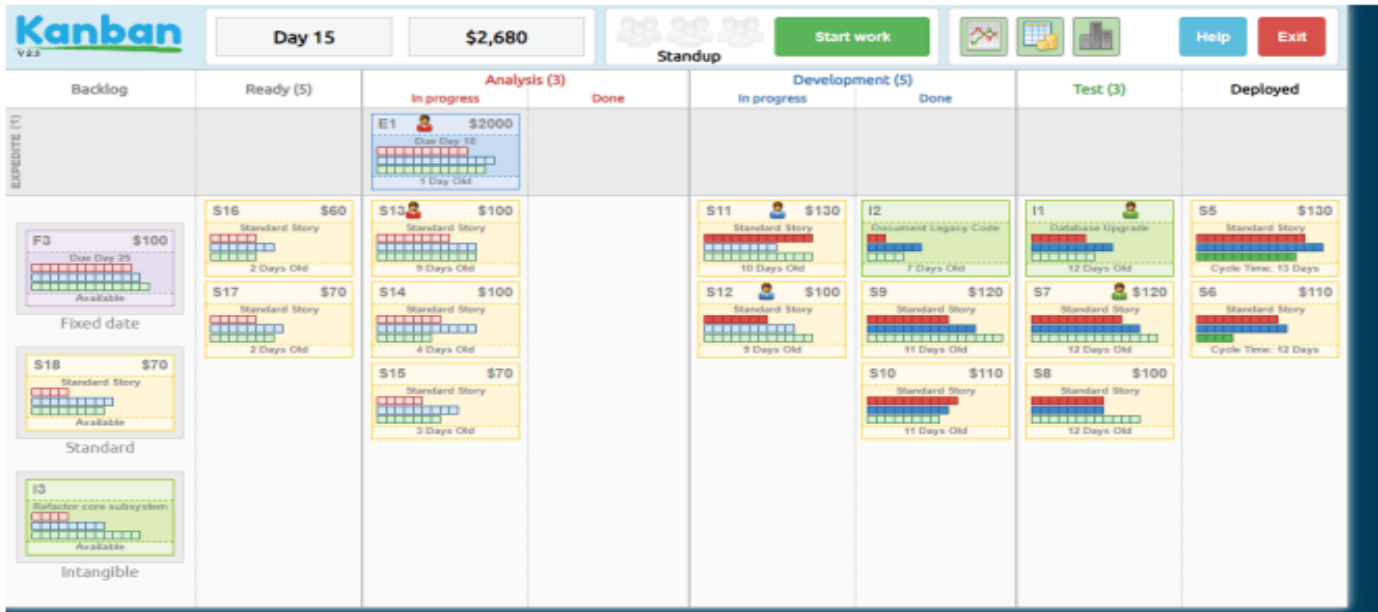
Plano de Release:

Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint 4	Iteração/Sprint 5
Data Início: 01/05/2024	Data Início:15/05/2024	Data Início:29/05/2024	Data Início:12/06/2024	Data Início:26/06/2024
Data Fim: 14/05/2024	Data Fim:28/05/2024	Data Fim:11/06/2024	Data Fim:25/06/2024	Data Fim:09/07/2024
<US 001 - Pesquisar Apartamento> SENDO o Síndico QUERO Pesquisar apartamentos PARA poder visualizar os apartamentos cadastrados, bem como poder filtrá-los ESTIMATIVA (4 PTS)	<US 002 Manter Apartamento PT 2> SENDO o Síndico QUERO Manter os dados do apartamento PARA possibilitar Editar, Excluir, Consultar e Inserir novos dados de Apartamentos ESTIMATIVA (2 PTS)	<US 008 - Pesquisar Morador> SENDO o Síndico, QUERO Pesquisar os dados dos moradores PARA fazer manutenção nos seus dados ESTIMATIVA (4 PTS)	<US 009 Manter Morador PT 2> SENDO o Síndico, QUERO Manter os dados dos moradores PARA fazer manutenção nos seus dados ESTIMATIVA (2 PTS)	<US 010 Registrar Pagamento PT 2> SENDO um Síndico, QUERO visualizar os dados do morador PARA realizar o registro do pagamento do condomínio ESTIMATIVA (2 PTS)
<US 002 - Manter Apartamento PT 1> SENDO o Síndico QUERO Manter os dados do apartamento PARA possibilitar Editar, Excluir, Consultar e Inserir novos dados de Apartamentos ESTIMATIVA (6 PTS)	<US 003 - Cadastrar Bloco> SENDO o Síndico QUERO Cadastrar novos Blocos de apartamento PARA fazer a edição, exclusão, pesquisa, listagem, consulta e cadastro de novo bloco ESTIMATIVA (1 PT)	<US 009 -Manter Morador PT 1> SENDO o Síndico, QUERO Manter os dados dos moradores PARA fazer manutenção nos seus dados ESTIMATIVA (6 PTS)	<US 010 Registrar Pagamento PT 1> SENDO um Síndico, QUERO visualizar os dados do morador PARA realizar o registro do pagamento do condomínio ESTIMATIVA (8 PTS)	
	<US 004 - Pesquisar Manutenção> SENDO o Síndico, QUERO Pesquisar os dados de manutenção nos apartamentos PARA pode registrar ou validar as informações ESTIMATIVA (1 PT)			

FONTE: O autor (2025).

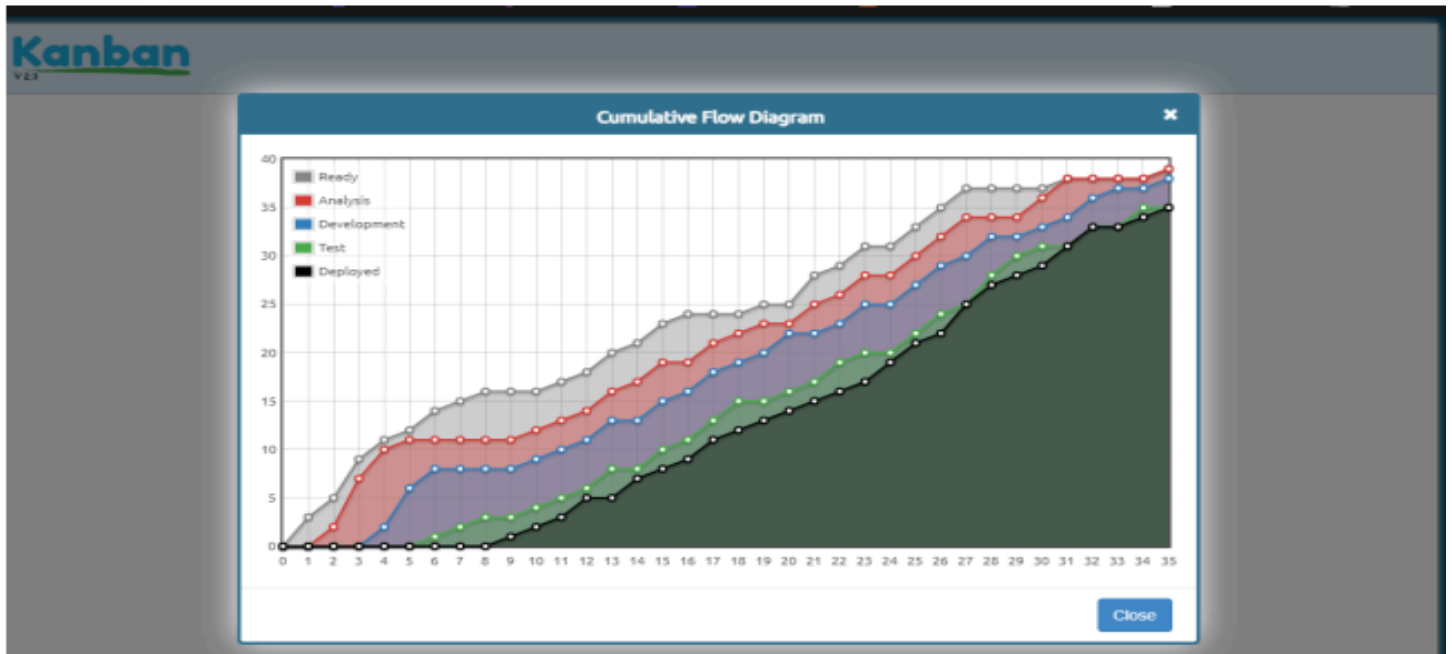
FIGURA 11 - ARTEFATO GERENCIAMENTO ÁGIL DE PROJETOS.

Primeiro Print: Dia 15 no Simulator



FONTE: O autor (2025).

FIGURA 12 - ARTEFATO GERENCIAMENTO ÁGIL DE PROJETOS.

Terceiro Print: CFD

FONTE: O autor (2025).

5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

Utilizando a linguagem de programação Java, esta disciplina tem como objetivo introduzir e reforçar conceitos fundamentais de programação, além de demonstrar como programar de maneira ágil por meio de abordagens como o *Test-Driven Development* (TDD). O foco está na escrita de um código limpo, testável e de qualidade.

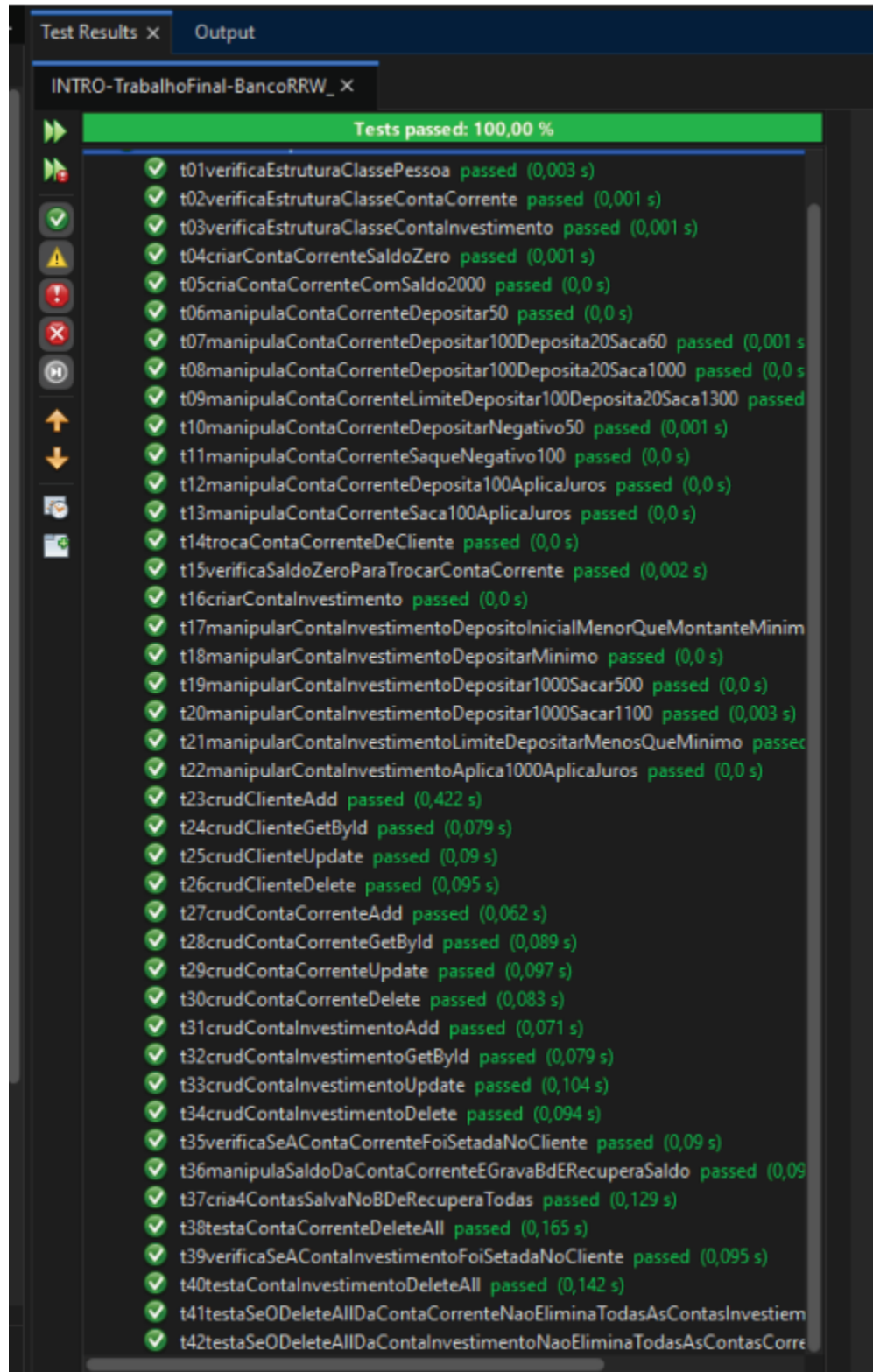
Ao longo do curso, foram abordados temas essenciais, como os pilares da programação orientada a objetos, sintaxe da linguagem Java, integração com bancos de dados e a aplicação de padrões de projeto.

O principal desafio do trabalho final foi a implementação de testes unitários pré-estabelecidos para um projeto, garantindo que todos os testes fossem verificados e aprovados, assegurando assim o funcionamento correto do sistema.

Ao concluir esta disciplina, é desenvolvida a capacidade de compreender e abstrair conceitos fundamentais de programação, aplicando-os a um projeto real por meio de uma abordagem ágil, melhorando sua experiência prática no desenvolvimento de software.

5.1 ARTEFATOS DO PROJETO

FIGURA 13 - ARTEFATO INTRODUÇÃO A PROGRAMAÇÃO



FONTE: O autor (2025).

6 DISCIPLINA: BD – BANCO DE DADOS

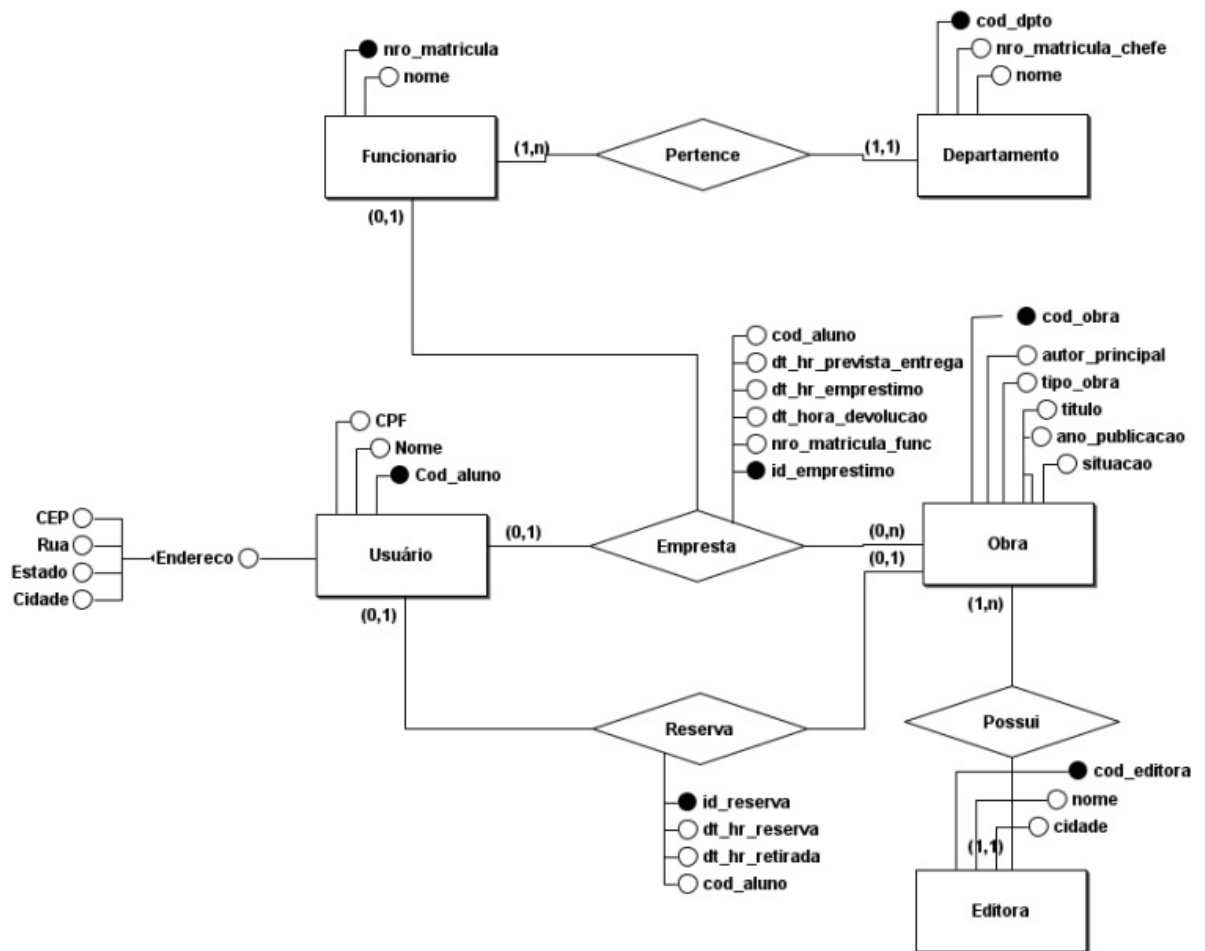
A disciplina de Banco de Dados introduziu e aprofundou conceitos sobre bancos relacionais, abrangendo desde a modelagem conceitual e lógica, normalização até a utilização da linguagem SQL e suas principais funcionalidades, sendo esses conceitos essenciais para o desenvolvimento ágil de software, tendo em vista que a modelagem correta de um banco de dados afeta diretamente a possibilidade de incrementos vindos de novas sprints, facilidade na manutenção e escalabilidade do sistema.

O projeto foi dividido em duas etapas. Na primeira, os alunos desenvolveram o modelo entidade-relacionamento e o modelo lógico para um sistema fictício de bibliotecas, com base nos requisitos levantados. Na segunda etapa, foi solicitado o desenvolvimento de um modelo relacional, seguido da implementação das tabelas e seus relacionamentos por meio de scripts SQL, além da população das tabelas com dados fictícios para um sistema de posto de vacinação.

Por meio da prática e da análise de estudos de caso, a disciplina capacita o aluno a criar e manter bancos de dados relacionais, preparando-o para atuar no desenvolvimento e administração de banco de dados.

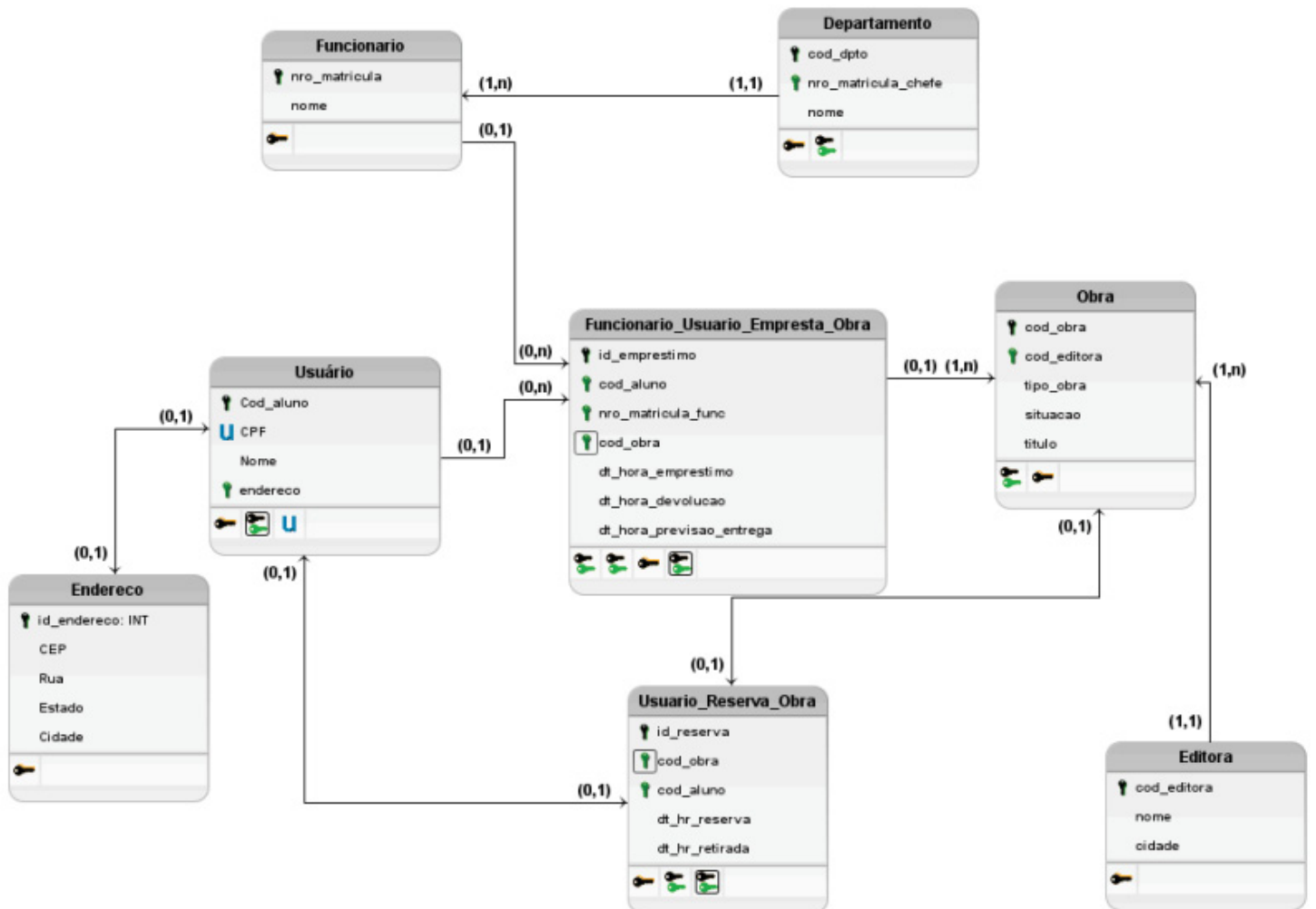
6.1 ARTEFATOS DO PROJETO

FIGURA 14 - ARTEFATO BANCO DE DADOS

Questão 1 – modelagem de dados**1) Modelo Entidade-relacionamento:**

FONTE: O autor (2025).

FIGURA 15 - ARTEFATO BANCO DE DADOS

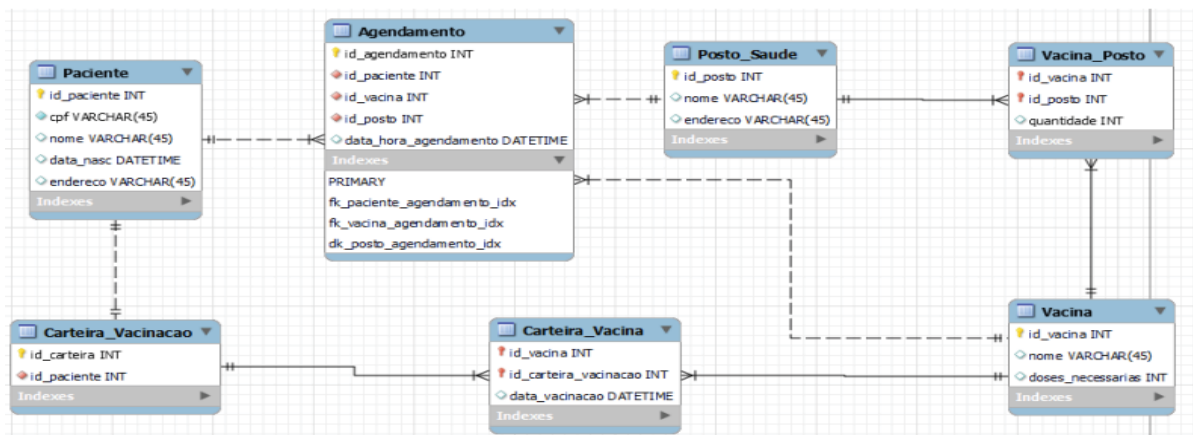
2) Modelo Lógico - Diagrama de Entidade Relacionamento (tabelas):

FONTE: O autor (2025).

FIGURA 16 - ARTEFATO BANCO DE DADOS

Descrição:

Em um posto de vacinação, o paciente pode agendar sua vacinação. Esse agendamento deve conter, obrigatoriamente, um posto associado, uma vacina e um paciente associado. Esse posto de saúde possui vários tipos de vacina, enquanto os tipos de vacina estão presentes em vários postos diferentes. Cada paciente possui uma carteira de vacinação. Essa carteira pode conter muitas vacinas. A vacina pode estar presente em várias carteiras de vacinação diferentes.



FONTE: O autor (2025).

FIGURA 17 - ARTEFATO BANCO DE DADOS

1) Criação das tabelas:

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Paciente` (
  `id_paciente` INT NOT NULL,
  `cpf` VARCHAR(45) NOT NULL,
  `nome` VARCHAR(45) NULL,
  `data_nasc` DATETIME NULL,
  `endereco` VARCHAR(45) NULL,
  PRIMARY KEY (`id_paciente`),
  UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) VISIBLE)
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Vacina` (
  `id_vacina` INT NOT NULL,
  `nome` VARCHAR(45) NULL,
  `doses_necessarias` INT NULL,
  PRIMARY KEY (`id_vacina`))
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Posto_Saude` (
  `id_posto` INT NOT NULL,
  `nome` VARCHAR(45) NULL,
  `endereco` VARCHAR(45) NULL,
  PRIMARY KEY (`id_posto`))
ENGINE = InnoDB;
```

FONTE: O autor (2025).

7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

A disciplina de Aspectos Ágeis de Programação apresenta práticas para escrever códigos sustentáveis, de fácil manutenção e escaláveis, seguindo abordagens ágeis. Para isso, são utilizadas metodologias como *Clean Code*, TDD (*Test-Driven Development*), BDD (*Behavior-Driven Development*), garantindo o alinhamento dos requisitos da sprint durante a implementação, além de conceitos como *Clean Architecture* e *Pair Programming*.

Ao longo da disciplina, são analisados exemplos reais de código, comparando implementações com e sem boas práticas, destacando suas vantagens e ensinando como aplicá-las corretamente em novos projetos ou na refatoração de sistemas legados.

No projeto final, o aluno é desafiado a refatorar o código do algoritmo de ordenação *Bubble Sort*, aplicando os princípios de *Clean Code* para torná-lo mais eficiente, legível e sustentável, consolidando os aprendizados da disciplina.

7.1 ARTEFATOS DO PROJETO

FIGURA 18 - ARTEFATO ASPECTOS ÁGEIS DE PROGRAMAÇÃO

```
class BubbleSort {  
  
    public static void main(String args[]) {  
        int array[] = {64, -1, 22222, 123, 1, 34, 65, 25, 12, 22, 11, 90};  
        bubbleSort(array);  
        printArray(array);  
    }  
  
    private static void bubbleSort(int arr[]) {  
        for (int i = 0; i < arr.length - 1; i++) {  
            if (arrayOrdenado(arr, i)) {  
                break;  
            }  
        }  
    }  
  
    private static boolean arrayOrdenado(int arr[], int i) {  
        boolean arrayOrdenado = true;  
  
        for (int j = 0; j < arr.length - i - 1; j++) {  
            if (arr[j] > arr[j + 1]) {  
                realizaTroca(j, j + 1, arr);  
                arrayOrdenado = false;  
            }  
        }  
  
        return arrayOrdenado;  
    }  
  
    private static void realizaTroca(int primeiroIndex, int segundoIndex, int[] array) {  
        int varAuxiliar = array[primeiroIndex];  
        array[primeiroIndex] = array[segundoIndex];  
        array[segundoIndex] = varAuxiliar;  
    }  
  
    private static void printArray(int arrayOrdenado[]) {  
        System.out.println("Array ordenado: ");  
  
        for (int numero : arrayOrdenado) {  
            System.out.print(numero + " ");  
        }  
  
        System.out.println();  
    }  
}
```

FONTE: O autor (2025).

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

Nas disciplinas de Desenvolvimento Web 1 e 2, foram introduzidos e aprofundados conceitos essenciais do desenvolvimento *full stack*, abrangendo tanto o lado do cliente, com experiência do usuário, telas dinâmicas e responsivas, quanto o lado do servidor, garantindo ambientes robustos e escaláveis.

Desde a introdução a conceitos básicos de TypeScript e Java até o uso de frameworks robustos como Angular e Spring Boot, as disciplinas orientam os alunos desde os fundamentos até as práticas mais modernas para a criação de sistemas eficientes. O projeto desenvolvido exigiu a criação de uma tela para cadastro e manutenção de alunos, cursos e matrículas, garantindo o funcionamento das operações de inserção, remoção, atualização e consulta, além da integração entre cliente e servidor.

Para o desenvolvimento do projeto, utilizou-se o *framework* Angular com TypeScript e a linguagem de programação Java com Spring Boot, assegurando a comunicação entre ambos por meio do conceito arquitetural REST API (*Representational State Transfer Application Programming Interface*), amplamente requisitado no mercado.

As disciplinas proporcionam uma abordagem prática e concreta para a construção de sistemas robustos e escaláveis, permitindo aos alunos aplicar conceitos ágeis aprendidos em outras disciplinas no desenvolvimento dos projetos.

8.1 ARTEFATOS DO PROJETO

FIGURA 18 - ARTEFATO DESENVOLVIMENTO WEB 1 E 2

Alunos

Cursos

Matriculas

Nome	CPF	Email	Data de Nascimento	
Joao 4	09849080990	teste@teste.com	2000-10-23	+ Novo Ver Editar Remover
teste	09849080990	teste@teste.com	2000-10-23	Ver Editar Remover

FONTE: O autor (2025).

FIGURA 19 - ARTEFATO DESENVOLVIMENTO WEB 1 E 2

Alunos

Cursos

Matriculas

Nome	Link	
curso4	link.com	+ Novo Ver Editar Remover
teste curso	curso.com	Ver Editar Remover

FONTE: O autor (2025).

FIGURA 20 - DESENVOLVIMENTO WEB 1 E 2

Alunos

Cursos

Matriculas

Numero de matricula	Aluno	Curso	Data de Matricula	Nota	
3	Joao5	curso4	2024-09-29	10	+ Novo Edita Remover

FONTE: O autor (2025).

FIGURA 21 - DESENVOLVIMENTO WEB 1 E 2

The image shows a web application interface for creating a new student record. At the top, there is a dark navigation bar with three items: 'Alunos' (with a person icon), 'Cursos' (with a document icon), and 'Matriculas' (with a graduation cap icon). Below the navigation bar, the main heading is 'Novo Aluno'. The form contains several input fields: 'Nome:' with the value 'teste 6', 'CPF:' with the value '09849080990', 'Email:' with the value 'teste@teste.com', and 'Data de Nascimento:' with the value '23/10/0020'. At the bottom of the form, there are two buttons: a blue 'Salvar' button with a save icon and a grey 'Voltar' button with a back arrow icon.

Alunos Cursos Matriculas

Novo Aluno

Nome: teste 6

CPF: 09849080990

Email: teste@teste.com

Data de Nascimento: 23/10/0020

Salvar Voltar

FONTE: O autor (2025).

9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

A disciplina de UX no Desenvolvimento Ágil de Software tem como objetivo orientar o aluno no planejamento e na construção de telas e protótipos de forma responsiva e inclusiva, garantindo que atendam às necessidades da maior parte do público-alvo e promovam uma interação eficiente com o sistema.

Foram abordados tópicos como princípios e história do UX (*User Experience*), sua evolução ao longo do tempo e sua importância no desenvolvimento de software. Além disso, a disciplina explorou conceitos de usabilidade, acessibilidade, design centrado no usuário, organização da informação na tela e técnicas essenciais para o desenvolvimento de interfaces, como o uso de cores, tipografia, ícones, grids e layouts.

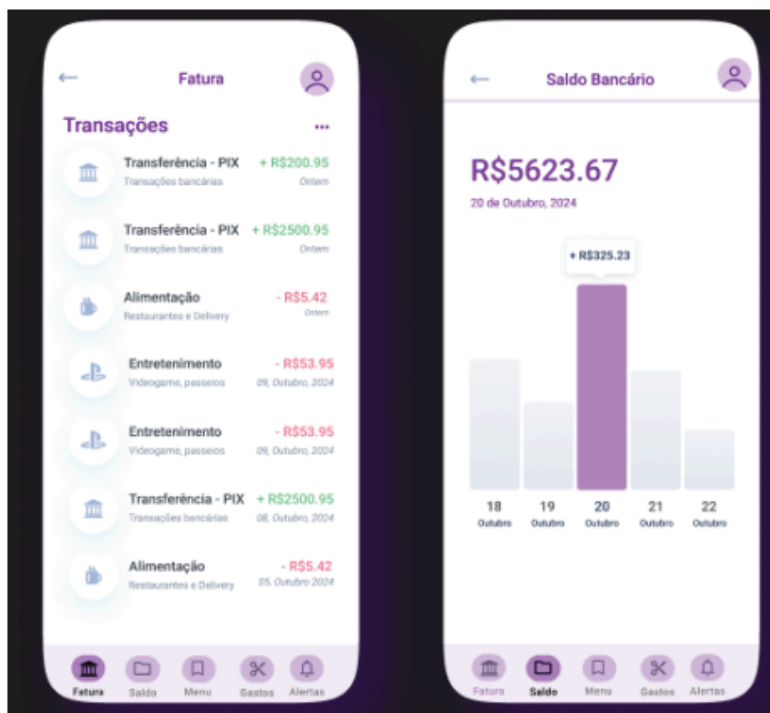
No projeto, foi solicitada a prototipação de uma tela para um sistema bancário, aplicando os conceitos aprendidos durante a disciplina. Além disso, os alunos justificaram suas escolhas e realizaram uma análise com feedback de um usuário sobre a interface criada.

Com os conhecimentos adquiridos, foi possível ter uma experiência prática na aplicação dos conceitos de UX, tanto no projeto desenvolvido quanto em outras disciplinas que envolvem o design e desenvolvimento de interfaces.

9.1 ARTEFATOS DO PROJETO

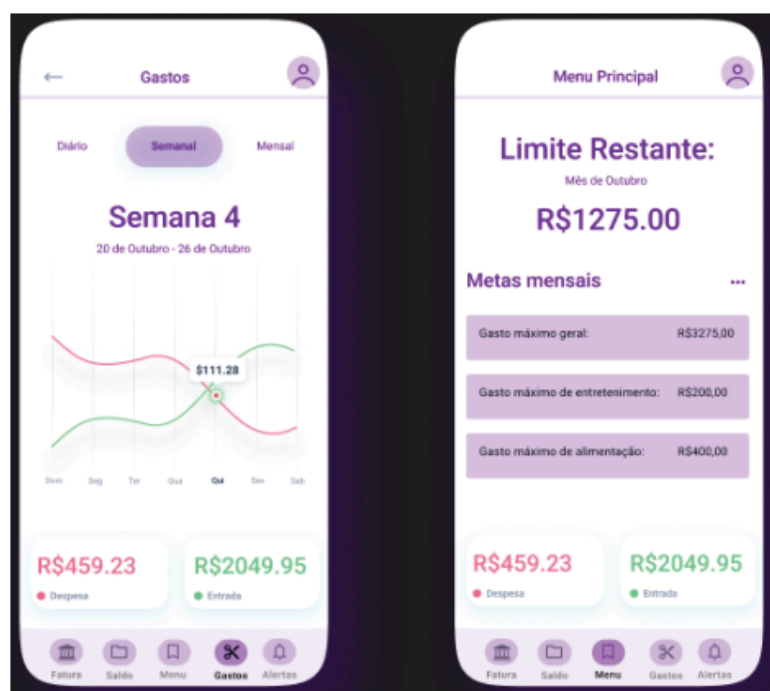
FIGURA 22 - ARTEFATO UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

Telas:



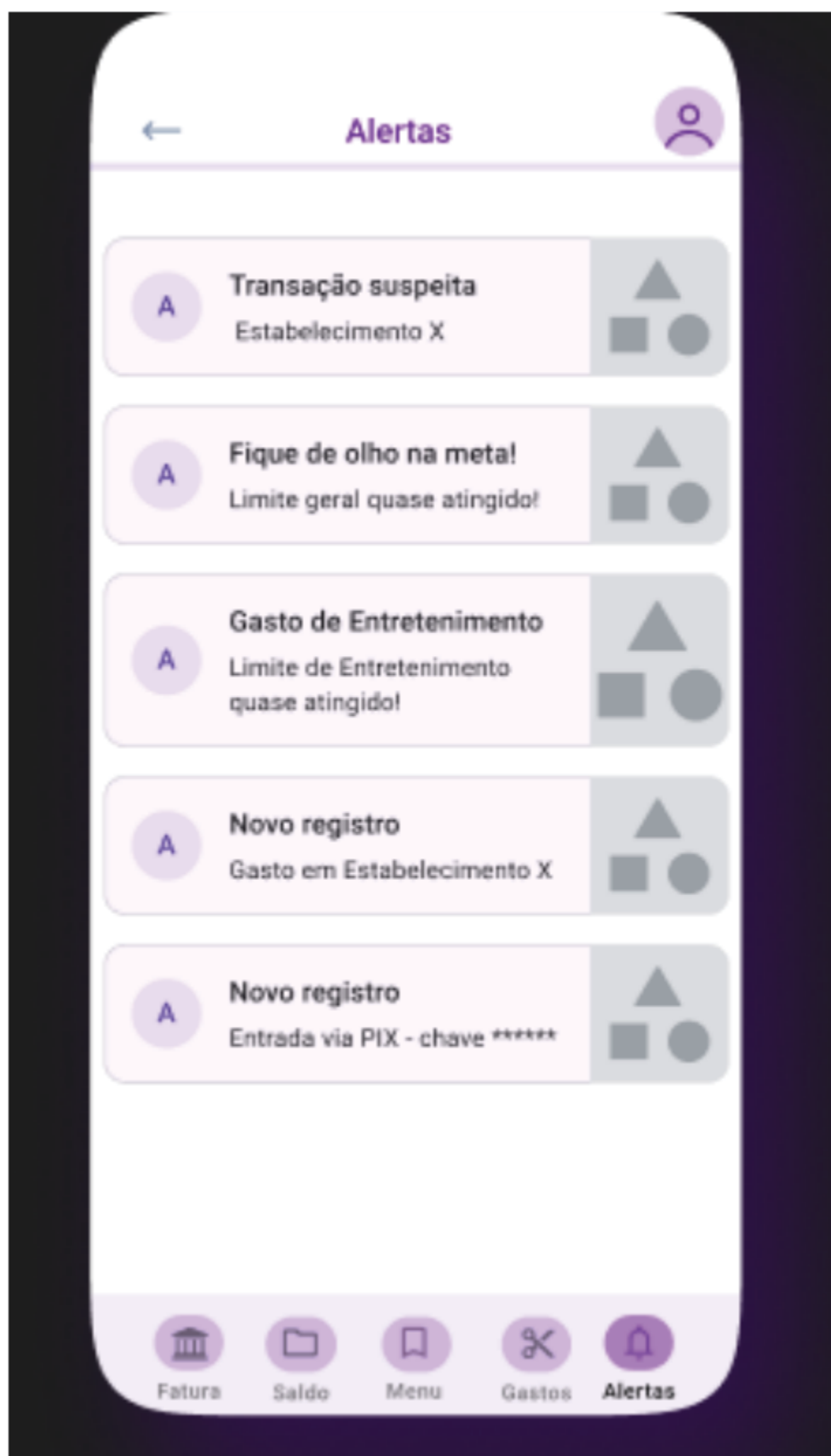
FONTE: O autor (2025).

FIGURA 23 - ARTEFATO UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE



FONTE: O autor (2025).

FIGURA 24 - ARTEFATO UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE



FONTE: O autor (2025).

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

As disciplinas de Desenvolvimento Mobile 1 e 2 capacitaram os alunos a desenvolver aplicativos para dispositivos móveis de forma ágil, utilizando as linguagens de programação Java e Kotlin na plataforma Android. Foram aplicados conceitos essenciais e atuais dessas tecnologias por meio de desafios práticos, abordando a criação de Activities, diferentes tipos de *layout*, personalização de elementos da interface, listagens, corrotinas e consumo de APIs.

O projeto desenvolvido reuniu esses conceitos em um aplicativo que consome dados do *endpoint* HP-API, uma API que disponibiliza informações sobre diversos personagens, casas e professores do universo de *Harry Potter*. No aplicativo, foram criadas telas para busca e listagem de personagens específicos, listagem de professores e alunos, além de um menu dashboard.

As disciplinas não apenas aprofundaram o conhecimento adquirido anteriormente, mas também introduziram conceitos fundamentais para o desenvolvimento mobile, uma área em constante crescimento e de grande relevância no mercado de tecnologia.

10.1 ARTEFATOS DO PROJETO

FIGURA 25 - ARTEFATO DESENVOLVIMENTO MOBILE 1 E 2



FONTE: O autor (2025).

FIGURA 26 - ARTEFATO DESENVOLVIMENTO MOBILE 1 E 2

	Harry Potter Gryffindor
	Hermione Granger Gryffindor
	Ron Weasley Gryffindor
	Draco Malfoy Slytherin
	Minerva McGonagall Gryffindor
	Cedric Diggory Hufflepuff
	Cho Chang Ravenclaw

FONTE: O autor (2025).

FIGURA 27 - ARTEFATO DESENVOLVIMENTO MOBILE 1 E 2

☐ Lufa Lufa

☒ Corvinal

☐ Sonserina

☐ Grifinória



Cho Chang
Ravenclaw



Luna Lovegood
Ravenclaw



Terry Boot
Ravenclaw



Mandy Brocklehurst
Ravenclaw



Padma Patil
Ravenclaw

FONTE: O autor (2025).

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

A disciplina de Infraestrutura para Desenvolvimento e Implantação de Software orientou os alunos na criação e organização de ambientes estruturados para a entrega de sistemas ágeis. Foram abordados conceitos como Dockerização, containerização, ciclo de vida do desenvolvimento de software, implementação de DevOps, utilização de Git, pipelines para entrega contínua e observabilidade de sistemas por meio de ferramentas de logs e métricas. Além disso, o uso do Kubernetes foi explorado para a organização e estruturação de containers.

Também foram reforçados conceitos de *Infrastructure as Code* (IaC), permitindo a automação da criação e configuração de infraestruturas dentro de uma aplicação, alinhando-se aos princípios do desenvolvimento ágil. Esse conhecimento é essencial para garantir a escalabilidade, a eficiência e a confiabilidade dos sistemas modernos.

O projeto da disciplina consistiu na configuração da Dockerização de uma aplicação e na realização de commits em uma pipeline de um projeto existente via Git. Essa atividade proporcionou uma experiência prática e recorrente, essencial para o dia a dia de desenvolvedores que atuam com integração e implantação contínua de software.

11.1 ARTEFATOS DO PROJETO

FIGURA 28 - ARTEFATO INFRA

Container rodando:

```
C:\Users\stark>docker run --name gitlab --detach --hostname localhost --publish 22:22 --publish 443:443 --publish 80:80 --publish 9091:9091 --shm-size 256m dfwandarti/gitlab_jenkins:3
Unable to find image 'dfwandarti/gitlab_jenkins:3' locally
Pulling from dfwandarti/gitlab_jenkins
7bfe87ed847: Pull complete
5bb9599d9c9: Pull complete
4d6df4aa614d: Pull complete
c695fd85ed38: Pull complete
f36ee87421e: Pull complete
8436bfc19a7: Pull complete
61a5147aada: Pull complete
d4e92472abe: Pull complete
cfc33ab37f3b: Pull complete
42364568e18: Pull complete
Digest: sha256:c8f21731114d5e795315534e827d13a09a2c63d8956eedac53cca475c6e3780
Status: Downloaded newer image for dfwandarti/gitlab_jenkins:3
9324bcb7d62f2b2368fba3640554417090d436228a76f9ac845cf20fe0b51c7

C:\Users\stark>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
9324bcb7d62f	dfwandarti/gitlab_jenkins:3	"/assets/wrapper"	About a minute 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

FONTE: O autor (2025).

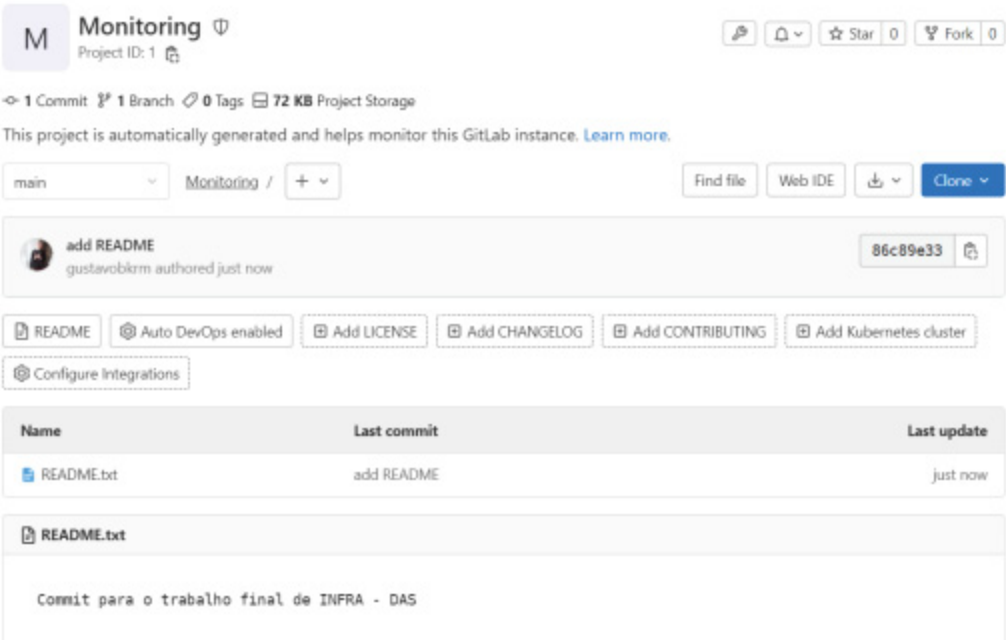
FIGURA 29 - ARTEFATO INFRA

Iniciando Jenkins

```
C:\Users\stark>docker exec -it gitlab bash
root@localhost:/# cd jenkins
root@localhost:/jenkins# java -jar jenkins.war --httpPort=9091
Running from: /jenkins/jenkins.war
webroot: /root/.jenkins/war
2024-12-17 12:58:15.807+0000 [id=1] INFO winstone.Logger$logInternal: Beginning extraction from war file
2024-12-17 12:58:15.832+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2024-12-17 12:58:15.879+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-10.0.12; built: 2022-09-14T01:54:40.076Z; git: 400d01
39887e27a57b54ed52e2d92a36731a7e88; jvm 11.0.17+8-post-Ubuntu-lubuntu220.04
2024-12-17 12:58:16.048+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor$visitServlet: NO JSP Support for /, did not find org.eclipse.jet
ty.jsp.JettyJspServlet
2024-12-17 12:58:16.080+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
2024-12-17 12:58:16.317+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /root/.jenkins found at: $user.home/.j
enkins
2024-12-17 12:58:16.423+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.024be2d9c{Jenkins v2.383.1, file:///root/.jenkins/w
ar/AVAILABLE}[/root/.jenkins/war]
2024-12-17 12:58:16.434+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector@45afc369{HTTP/1.1, (http/1.1)}{0.0.0
.0:9091}
2024-12-17 12:58:16.446+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started Server@5a4aa2f2{STARTING}[10.0.12, sto=0] @921ms
2024-12-17 12:58:16.447+0000 [id=35] INFO winstone.Logger$logInternal: Winstone Servlet Engine running: controlPort=disabled
2024-12-17 12:58:16.591+0000 [id=41] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2024-12-17 12:58:16.676+0000 [id=51] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2024-12-17 12:58:18.833+0000 [id=41] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java7$1 (file:/root/.jenkins/war/WEB-INF/lib/groovy-all-2.4.21.jar) to constr
uctor java.lang.invoke.MethodHandles$Lookup(java.lang.Class,int)
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java7$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2024-12-17 12:58:19.148+0000 [id=66] INFO h.p.b.g.GlobalTimeoutConfigurationLoad: global timeout not set
2024-12-17 12:58:19.439+0000 [id=58] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2024-12-17 12:58:19.444+0000 [id=58] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2024-12-17 12:58:19.575+0000 [id=60] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2024-12-17 12:58:19.575+0000 [id=69] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2024-12-17 12:58:19.626+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2024-12-17 12:58:19.628+0000 [id=40] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2024-12-17 12:58:19.655+0000 [id=84] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2024-12-17 12:58:19.669+0000 [id=46] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2024-12-17 12:58:19.717+0000 [id=32] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
```

FONTE: O autor (2025).

FIGURA 30 - ARTEFATO INFRA



FONTE: O autor (2025).

12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

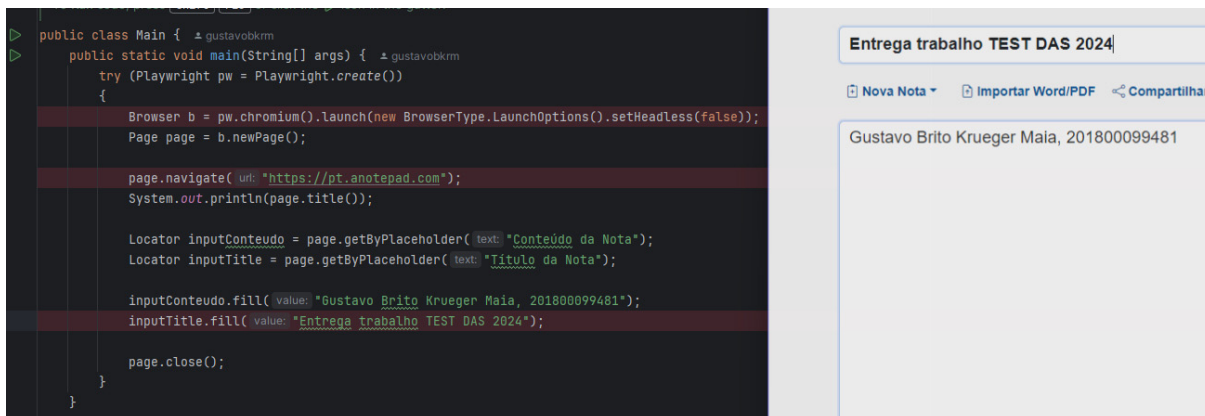
A disciplina de Testes Automatizados tem como objetivo ensinar aos alunos a criação correta de testes unitários e testes de integração, além de orientá-los na construção de testes eficazes para interfaces. Para isso, foram utilizadas ferramentas como JUnit, Mocks, Codegen e PlayWright, proporcionando uma abordagem prática para garantir a qualidade do software.

O projeto desenvolvido consistiu na utilização do PlayWright e Codegen em um ambiente Java para simular um teste de interface. A atividade envolveu o acesso ao site *Anotepad*, onde os alunos precisaram automatizar a inserção do número de matrícula e do nome completo em uma caixa de texto, validando o funcionamento da aplicação de forma automatizada.

Ferramentas como as abordadas na disciplina são fundamentais para o desenvolvimento ágil de software, garantindo que as aplicações sejam robustas, escaláveis e atendam aos requisitos definidos. Além disso, a prática de testes automatizados complementa os conhecimentos adquiridos em outras disciplinas, reforçando a importância da qualidade no ciclo de desenvolvimento.

12.1 ARTEFATOS DO PROJETO

FIGURA 31 - ARTEFATO TESTES AUTOMATIZADOS



FONTE: O autor (2025).

13 CONCLUSÃO

Com base nos projetos desenvolvidos e nos conhecimentos adquiridos ao longo da pós-graduação em Desenvolvimento Ágil de Software, fica evidente a sinergia entre as disciplinas, que se complementam para formar um profissional altamente capacitado para o mercado de trabalho.

Este documento apresenta objetivos, conceitos e resultados dos projetos realizados ao longo do curso, destacando suas correlações e a forma como as diferentes áreas de aprendizado se integram e evoluem ao longo da jornada acadêmica. O foco é a capacitação para atuar com excelência em ambientes ágeis, reforçando a importância da adaptação contínua às práticas mais modernas de desenvolvimento de software.

O conteúdo abordado em cada etapa do curso se conecta de maneira progressiva, promovendo uma evolução contínua no aprendizado e incentivando os alunos a se aprofundarem à medida que avançam no programa. Com foco no desenvolvimento ágil de software, os alunos são incentivados a aplicar metodologias ágeis, integração contínua e automação de processos, preparando-se para atuar em ambientes reais.

A estrutura da pós-graduação equilibra teoria e prática, priorizando a aplicação real dos conceitos apresentados. O curso não apenas transmite conhecimento, mas também estimula a vivência de desafios por meio de projetos alinhados à ementa de cada disciplina. Dessa forma, os alunos desenvolvem habilidades essenciais para enfrentar cenários do mundo real, consolidando uma formação sólida e relevante para o mercado de tecnologia.

14 REFERÊNCIAS BIBLIOGRÁFICAS

GUEDES, Gilleanes T. A. UML 2: *uma abordagem prática*. 3. ed. São Paulo: Novatec, 2018.

KIM, Gene; HUMBLE, Jez; DEBOIS, Patrick; WILLIS, John. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. 1. ed. Portland: IT Revolution Press, 2016.

PROJECT MANAGEMENT INSTITUTE. *Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)*. 7. ed. Pennsylvania: PMI, 2021.