

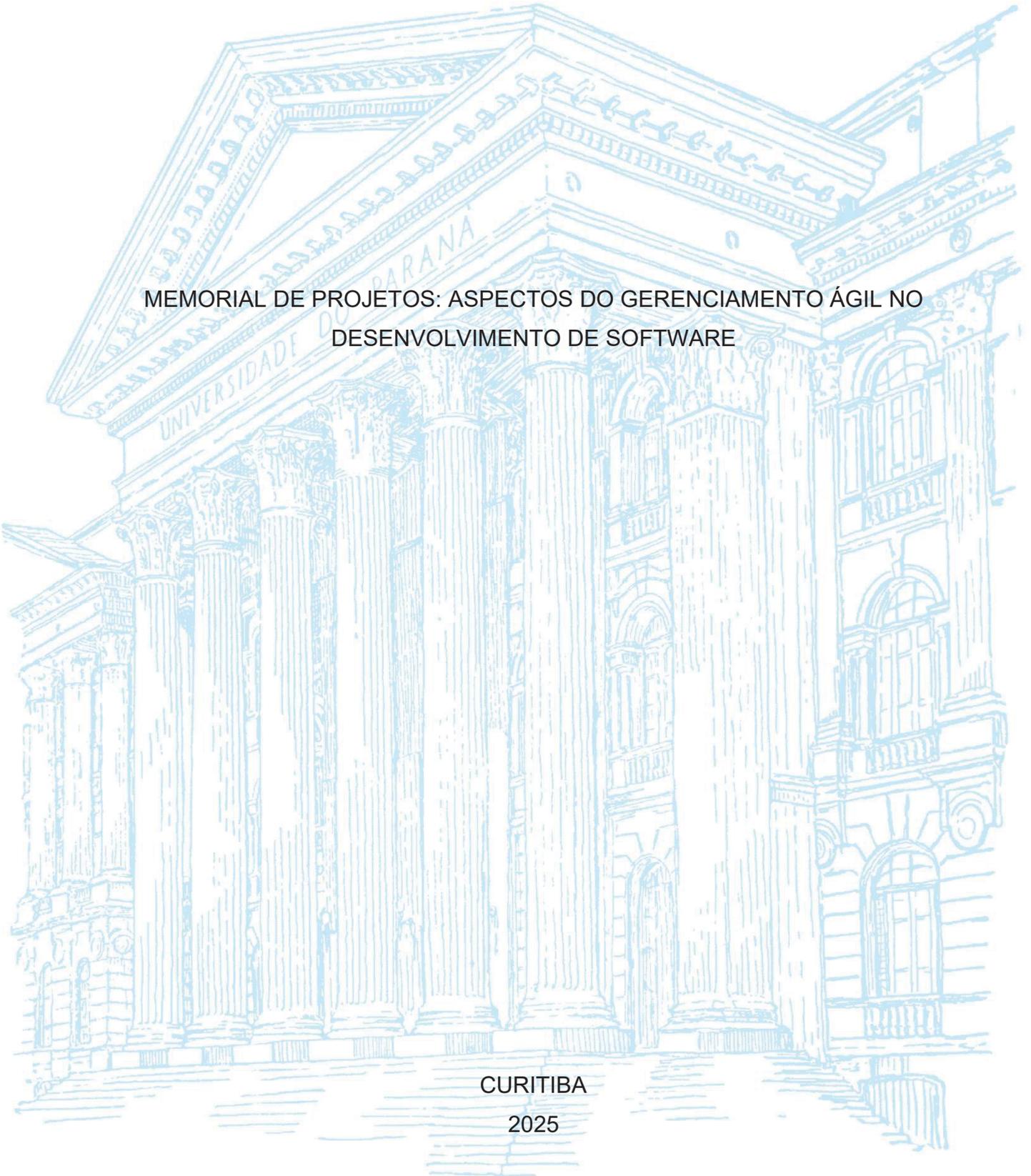
UNIVERSIDADE FEDERAL DO PARANÁ

LUIZ EMANOEL DE PONCE DA SILVA

MEMORIAL DE PROJETOS: ASPECTOS DO GERENCIAMENTO ÁGIL NO  
DESENVOLVIMENTO DE SOFTWARE

CURITIBA

2025



LUIZ EMANOEL DE PONCE DA SILVA

MEMORIAL DE PROJETOS: ASPECTOS DO GERENCIAMENTO ÁGIL NO  
DESENVOLVIMENTO DE SOFTWARE

Memorial de Projetos apresentado ao curso de Especialização em Desenvolvimento Ágil de Software do Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Desenvolvimento Ágil de Software.

Orientador: Prof. Dr. Razer Anthom Nizer Rojas Montaña

CURITIBA

2025

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação Desenvolvimento Ágil de Software da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **LUIZ EMANOEL DE PONCE DA SILVA**, intitulada: **MEMORIAL DE PROJETOS: ASPECTOS DO GERENCIAMENTO ÁGIL NO DESENVOLVIMENTO 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, 02 de Dezembro de 2025.



RAZER ANTHOM NIZER ROJAS MONTAÑO

Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

## RESUMO

Este Memorial de Projetos apresenta um parecer técnico sobre os aspectos do gerenciamento ágil no desenvolvimento de software, argumentando que a gestão ágil é um sistema sociotécnico indispensável para a entrega de software com qualidade e valor. A análise demonstra que a eficácia de *frameworks* de planejamento iterativo, como o Scrum, ou de gestão de fluxo visual, como o Kanban, é indissociável das práticas de engenharia que garantem a qualidade interna do produto. A modelagem ágil define a estrutura, o desenvolvimento orientado a testes (TDD) valida o comportamento, a refatoração contínua gerencia a dívida técnica e os pipelines de Integração e Entrega Contínua (CI/CD) automatizam a entrega. O parecer conclui que os principais desafios na adoção prática do desenvolvimento ágil residem na interseção entre a gestão do projeto e a execução técnica, como a otimização do *work in progress* (WIP), a alocação de tempo para melhorias técnicas, a adoção efetiva do TDD como prática de design e a manutenção dos pipelines de automação. A superação desses desafios exige uma mudança cultural em direção a um sistema puxado, focado na auto-organização e na qualidade contínua.

**Palavras-chave:** gerenciamento ágil de projetos; Scrum; Kanban; desenvolvimento ágil de software; devops.

## ABSTRACT

This Project Memorial presents a technical opinion on the aspects of agile management in software development, arguing that agile management is an indispensable sociotechnical system for delivering software with quality and value. The analysis demonstrates that the effectiveness of iterative planning frameworks, such as Scrum, or visual flow management, such as Kanban, is inseparable from the engineering practices that ensure the product's internal quality. Agile modeling defines the structure, test-driven development (TDD) validates behavior, continuous refactoring manages technical debt, and Continuous Integration and Delivery (CI/CD) pipelines automate delivery. The opinion concludes that the main challenges in the practical adoption of agile development lie at the intersection of project management and technical execution, such as optimizing work in progress (WIP), allocating time for technical improvements, the effective adoption of TDD as a design practice, and maintaining automation pipelines. Overcoming these challenges requires a cultural shift toward a pull system, focused on self-organization and continuous quality.

**Keywords:** agile project management; Scrum; Kanban; agile software development; devops.

## SUMÁRIO

<b>1</b>	<b>PARECER TÉCNICO.....</b>	<b>7</b>
<b>2</b>	<b>DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....</b>	<b>9</b>
2.1	ARTEFATOS DO PROJETO .....	10
<b>3</b>	<b>DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2 .....</b>	<b>14</b>
3.1	ARTEFATOS DO PROJETO .....	14
<b>4</b>	<b>DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2 .....</b>	<b>19</b>
4.1	ARTEFATOS DO PROJETO .....	20
<b>5</b>	<b>DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....</b>	<b>22</b>
5.1	ARTEFATOS DO PROJETO .....	23
<b>6</b>	<b>DISCIPLINA: BD – BANCO DE DADOS.....</b>	<b>25</b>
6.1	ARTEFATOS DO PROJETO .....	25
<b>7</b>	<b>DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO .....</b>	<b>30</b>
7.1	ARTEFATOS DO PROJETO .....	30
<b>8</b>	<b>DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE....</b>	<b>33</b>
8.1	ARTEFATOS DO PROJETO .....	34
<b>9</b>	<b>DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS) .....</b>	<b>35</b>
9.1	ARTEFATOS DO PROJETO .....	35
<b>10</b>	<b>DISCIPLINA: TEST – TESTES AUTOMATIZADOS .....</b>	<b>37</b>
10.1	ARTEFATOS DO PROJETO .....	38
<b>11</b>	<b>CONCLUSÃO .....</b>	<b>40</b>
	<b>REFERÊNCIAS .....</b>	<b>41</b>

## 1 PARECER TÉCNICO

Este memorial tem como objetivo consolidar os projetos desenvolvidos nas disciplinas da Especialização em Desenvolvimento Ágil de Software e apresentar um parecer técnico que analisa a integração desses projetos sob a ótica dos aspectos do gerenciamento ágil no desenvolvimento de *software*. A tese central deste parecer é que o gerenciamento ágil de projetos, representado por *frameworks* como Scrum (Sommerville, 2011) e Kanban (Anderson, 2011), não é apenas um conjunto de papéis e cerimônias, mas um sistema sociotécnico que depende fundamentalmente da integração com as práticas de modelagem, engenharia de *software* e automação para entregar valor de forma sustentável.

A jornada do curso demonstrou como o gerenciamento ágil permeia todo o ciclo de vida do desenvolvimento, começando pela concepção. A disciplina de Gerenciamento Ágil de Projetos 1 (GAP1) introduziu técnicas colaborativas de concepção, como *Lean Inception* e *Design Sprints* (Camargo; Ribas, 2019), que são cruciais para ambientes complexos e inovadores. Essa abordagem foi complementada pela Modelagem Ágil 1 (MAG1), onde a definição de histórias de usuário e pela disciplina de UX no Desenvolvimento de Software (UX), onde a prototipagem validou o escopo funcional antes do desenvolvimento. O projeto prático de GAP1 consolidou essa fase de planejamento através da criação de um plano de *release* para uma aplicação, aplicando conceitos do Scrum como estimativas em pontos (Cohn, 2011) e o cálculo de velocidade para distribuir as histórias de usuário em *sprints*.

Estabelecido o planejamento, o foco se voltou para a execução e a estrutura técnica. As disciplinas de Modelagem Ágil 2 (MAG2) e Banco de Dados (BD) forneceram a visão estrutural através dos Diagrama de Classes (Booch; Rumbaugh; Jacobson, 2005) e do Modelo Entidade-Relacionamento (Elmasri; Navathe, 2011).

Esses modelos forneceram a base para a implementação do *backend* em Java (Deitel, 2010) na disciplina de Introdução à Programação (INTRO). O aspecto de gerenciamento mais relevante dessa fase foi a introdução do Desenvolvimento Orientado a Testes (TDD), onde o progresso foi medido pela passagem dos testes unitários, garantindo que a qualidade fosse construída desde o início, e não apenas verificada ao final.

A gestão do fluxo de trabalho foi o foco de Gerenciamento Ágil de Projetos 2 (GAP2), que introduziu o Método Kanban. O projeto prático na plataforma Kanban Board Game (Kanban, 2025) demonstrou o poder dos Limites de trabalho em progresso como mecanismo de gestão (Anderson, 2011). Ao limitar o trabalho em progresso, o sistema força a equipe a focar nas tarefas em andamento, expondo gargalos que são analisados através do diagrama de fluxo cumulativo. Contudo, esse fluxo só é sustentável se a qualidade técnica for mantida. Aqui, a disciplina de Aspectos Ágeis de Programação (AAP) foi crucial, introduzindo a refatoração como ferramenta para combater a dívida técnica (Fowler, 2004), um dos maiores impedimentos ao fluxo ágil.

Finalmente, as práticas de gerenciamento de fluxo (GAP2) e qualidade técnica (AAP, TEST) são habilitadas pela automação, o pilar do DevOps, visto na disciplina de INFRA. O projeto de configuração de um *pipeline* de Integração Contínua (CI) com GitLab (GitLab, 2024) e Jenkins (Jenkins, 2024) demonstrou como automatizar a execução dos testes a cada *commit* (UFPR, 2024d). O *pipeline* de CI/CD é a expressão técnica do quadro Kanban: ele automatiza o movimento das tarefas pelas colunas, garantindo que o *software* em funcionamento seja a medida real de progresso.

Portanto, os projetos do curso demonstram que os aspectos do gerenciamento ágil não residem apenas em *frameworks* de planejamento (Scrum) ou fluxo (Kanban), mas na integração sistêmica destes com as práticas de engenharia (TDD, Refatoração) e automação (CI/CD, DevOps).

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

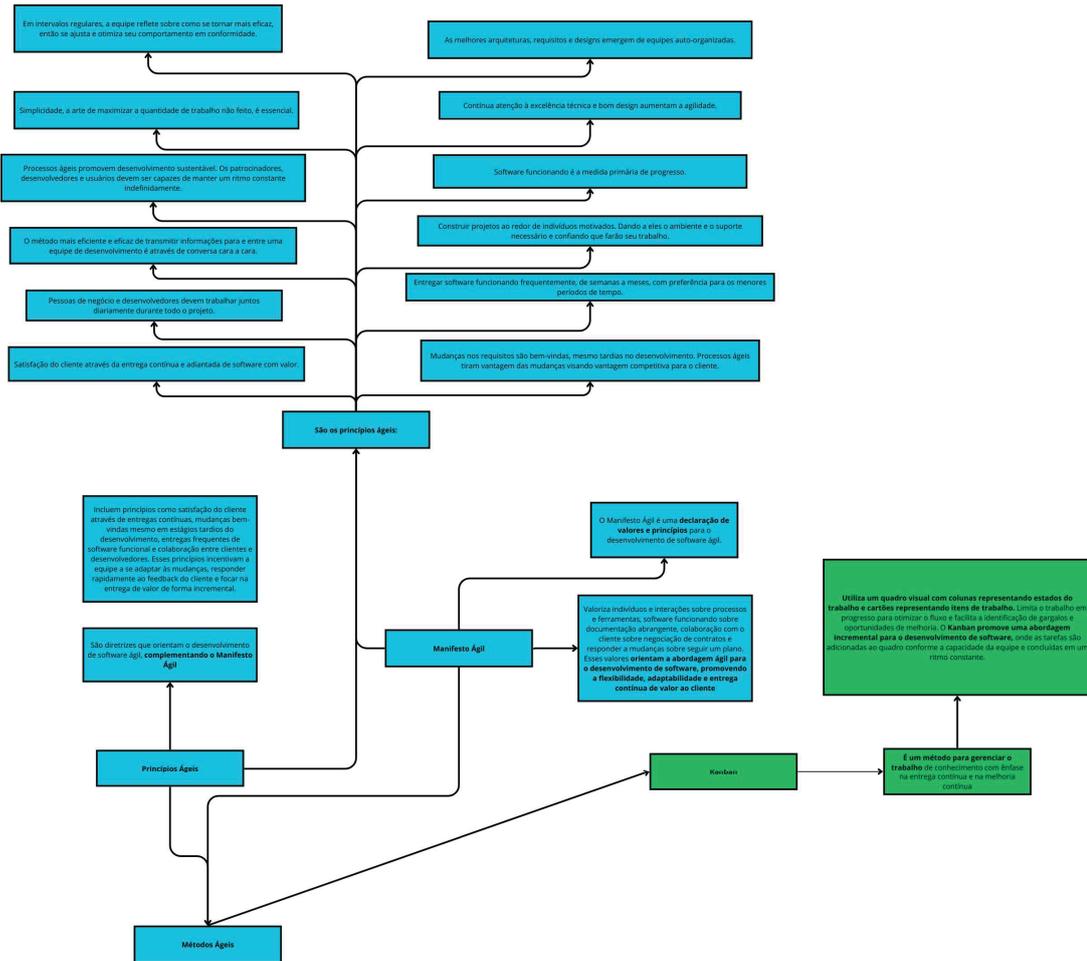
O projeto desenvolvido na disciplina consistiu no desenvolvimento de um mapa conceitual. Este mapa de conceito foi elaborado para trazer a representação visual dos diversos processos existentes, além de trazer a consolidação dos conceitos sobre cada processo.

Essa elaboração, com seu detalhamento foi fundamental para formar a base de conhecimentos sobre processo de software ao longo das demais disciplinas, conceituando e possibilitando a compreensão dos processos tradicionais e ágeis e seus aspectos. Tal desenvolvimento possibilita a comparação e compreensão sobre os divergentes processos, capacitando a escolha e entendimento das metodologias mais adequadas a cada cenário e a motivação da elaboração de cada metodologia.

Foram trabalhados o Processo de software, os modelos tradicionais de processo de *software* (Sommerville, 2011), o manifesto e princípios ágeis (Beck *et al.*, 2001), *Lean Software Development* (Poppendieck; Poppendieck, 2003), Scrum (Cohn, 2011), *Extreme Programming* (Beck, 2004), Kanban (Anderson, 2011) e entrega contínua de Software (Sommerville, 2011). Com a metodologia proposta para o projeto final, levando ao desenvolvimento dos principais aspectos de cada processo existente, estruturando e consolidando visualmente esses conceitos.

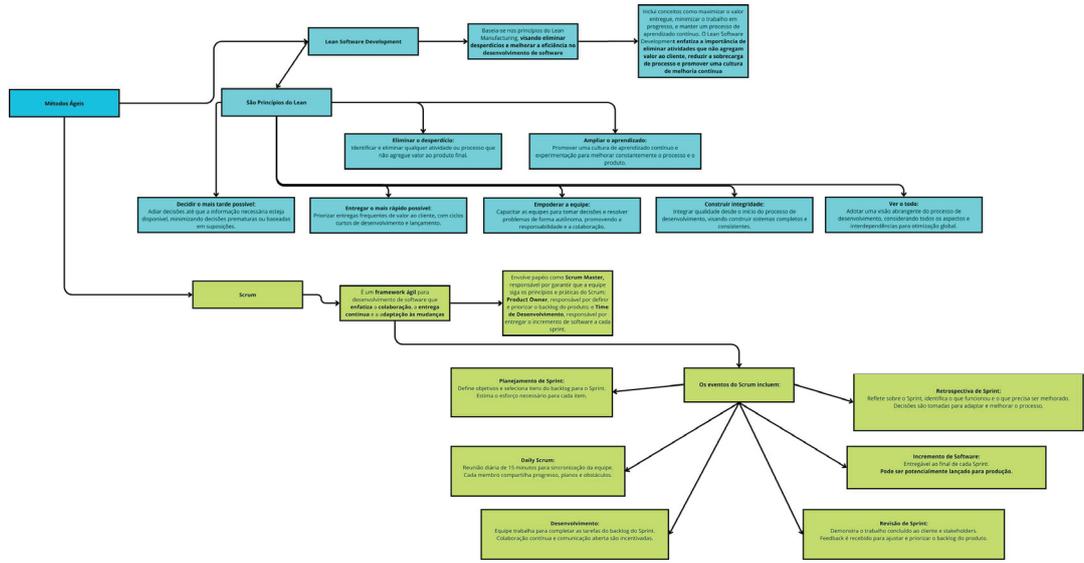
## 2.1 ARTEFATOS DO PROJETO

### FIGURA 1 – MAPA CONCEITUAL MADS, PINCÍPIOS ÁGEIS, MANIFESTO ÁGIL E KANBAN



FONTE: O Autor (2025)

FIGURA 2 – MAPA CONCEITUAL MADS, LEAN E SCRUM



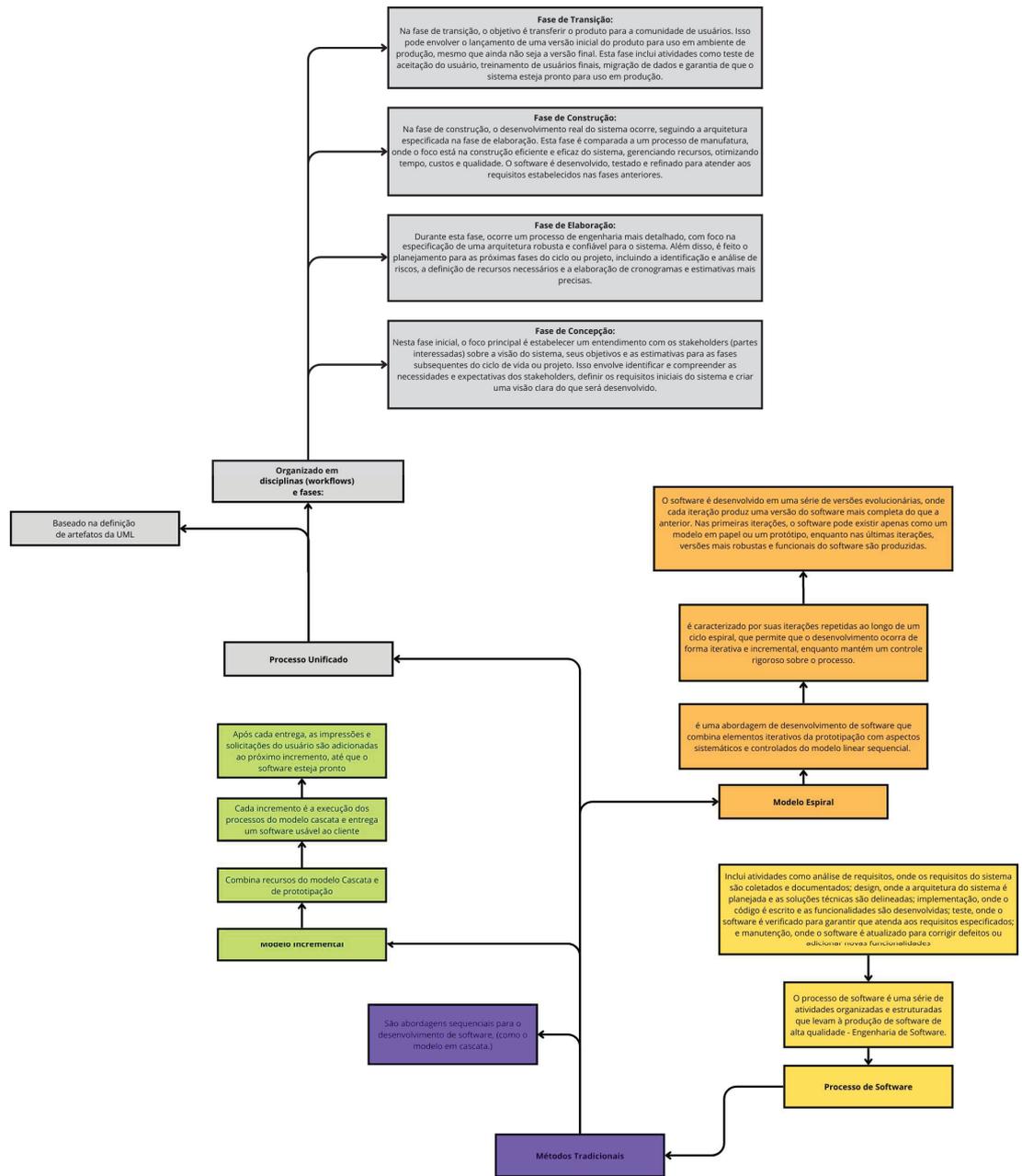
FONTE: O Autor (2025)

FIGURA 3 – MAPA CONCEITUAL MADS, EXTREME PROGRAMMING



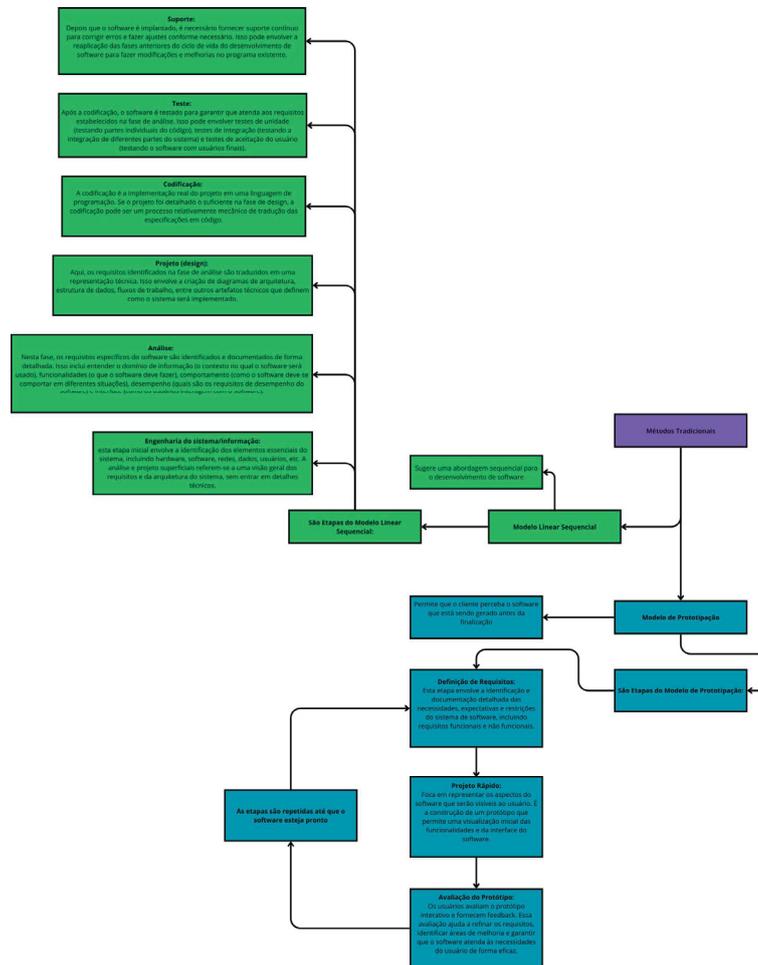
FONTE: O Autor (2025)

FIGURA 4 – MAPA CONCEITUAL MADS, PROCESSO DE SOFTWARE, METODOS TRADICIONAIS PARTE 1



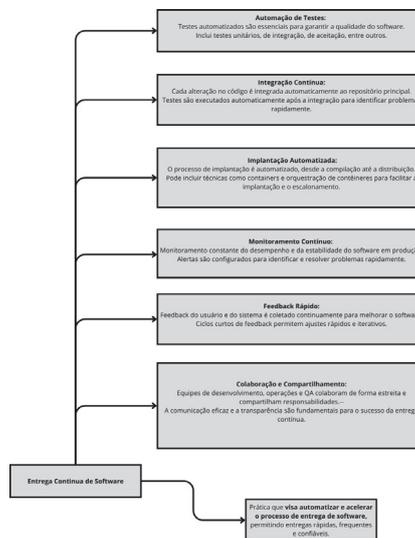
FONTE: O Autor (2025)

FIGURA 5 – MAPA CONCEITUAL MADS, PROCESSO DE SOFTWARE, METODOS TRADICIONAIS PARTE 2



FONTE: O Autor (2025)

FIGURA 6 – MAPA CONCEITUAL MADS, ENTREGA CONTÍNUA



FONTE: O Autor (2025)

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

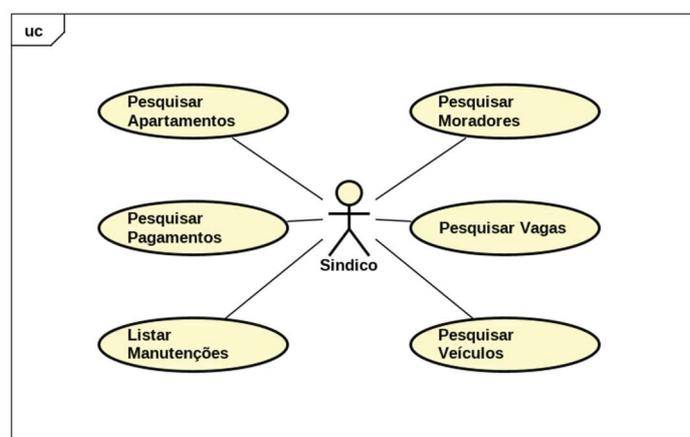
O projeto desenvolvido na disciplina de modelagem ágil de software foi a construção da modelagem de um sistema de gestão condominial, com o projeto da primeira parte da disciplina sendo a construção de diagramas de caso de uso de primeiro e segundo nível. A segunda parte, continuando o projeto de MAG1, sendo a construção dos diagramas de classes, classes com atributos associados, tabelas do banco de dados e diagramas de sequência (Booch; Rumbaugh; Jacobson, 2005).

A disciplina evidência, principalmente na sua prática, como a construção de diagramas, quando necessário e em quantidade e detalhamento suficiente, pode ser uma aliada nos processos de desenvolvimento ágil, garantindo mais clareza entre as partes no processo de desenvolvimento e apoiando a documentação e qualidade do *software* desenvolvido.

As atividades desenvolvidas durante a disciplina deram a capacidade de construir uma visão geral do *software*, considerando as metodologias ágeis e se transformando em um importante ferramental para o desenvolvimento dos projetos das demais disciplinas e na compreensão da engenharia dos sistemas em desenvolvimento.

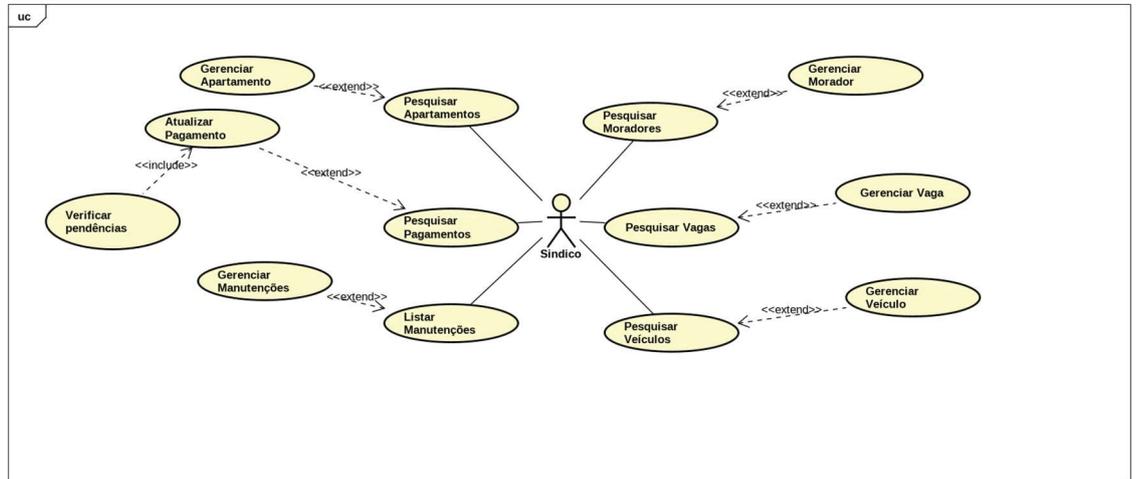
#### 3.1 ARTEFATOS DO PROJETO

FIGURA 7 – DIAGRAMA DE CASO DE USO NÍVEL 1



FONTE: O Autor (2025)

FIGURA 8 – DIAGRAMA DE CASO DE USO NÍVEL 2



FONTE: O Autor (2025)

FIGURA 9 – EXEMPLO DE HSTÓRIA DE USUÁRIO COM CRITÉRIOS DE ACEITAÇÃO

#### HU007 – Pesquisar Manutenções

<b>Sendo</b>	O síndico
<b>Quero</b>	Gerenciar as manutenções do condomínio
<b>Para</b>	Garantir o controle e registro das manutenções

**Pesquisar Manutenção** ←

NOVA MANUTENÇÃO

PESQUISAR MANUTENÇÃO

DATA	MANUTENÇÃO	CUSTO	EDITAR	APAGAR
01/04/2024	PINTURA	R\$ 3500,00	EDITAR	APAGAR
01/04/2024	CORTE DE GRAMA	R\$ 300,00	EDITAR	APAGAR

#### Críticos de Aceitação:

- 1) Deve Listar todas as manutenções registradas;
- 2) Deve Permitir, a partir da tela, incluir novas manutenções na lista;
- 3) Deve, a partir da tela, permitir o gerenciamento das manutenções existentes;
- 4) Deve permitir excluir o registro de uma manutenção;
- 5) Deve permitir pesquisar por uma manutenção específica;

#### Críticos de Aceitação – detalhamento:

**Crítico:** 1) Deve Listar todas as manutenções já cadastradas;

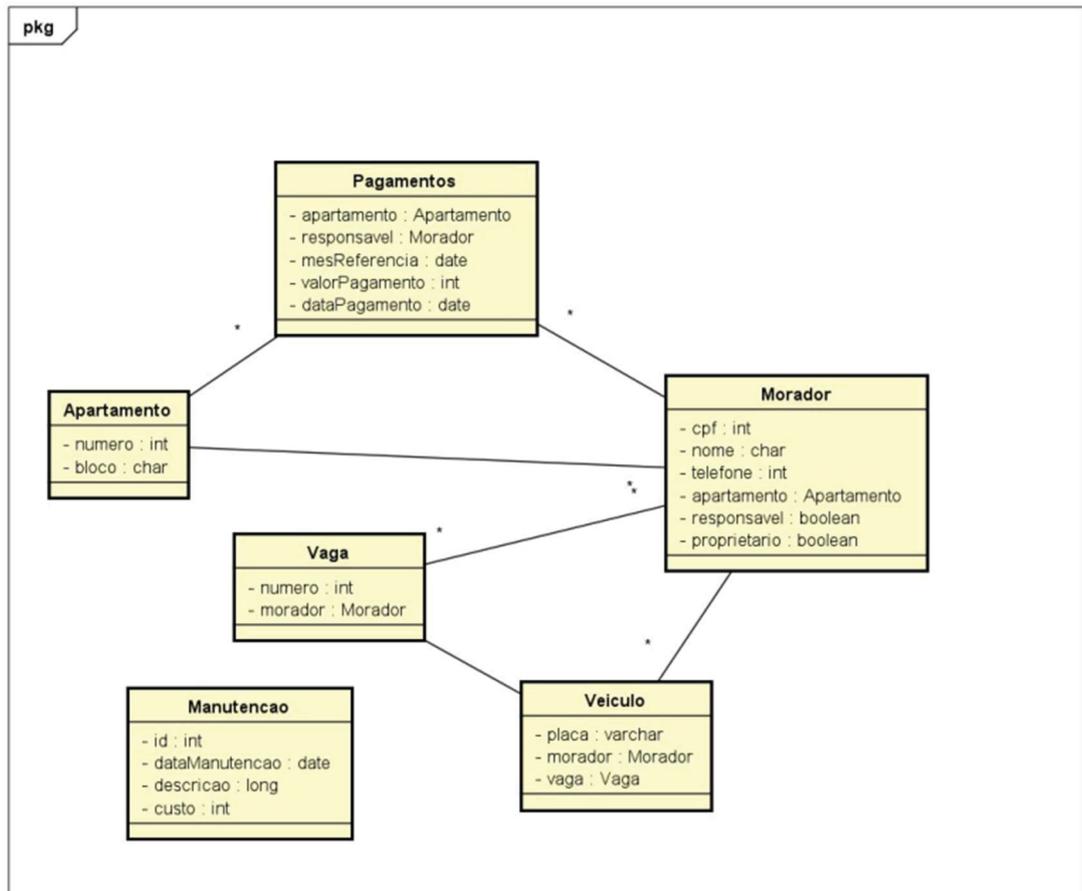
<b>Dado que</b>	O síndico acessou a tela
<b>Quando</b>	A tela for carregada
<b>Então</b>	Listar todas as manutenções já realizadas

**Crítico:** 2) Deve Permitir a partir da tela incluir novas manutenções na lista;

<b>Dado que</b>	O síndico acessou a tela
<b>Quando</b>	A tela for carregada
<b>Então</b>	Mostrar o botão para a modal que permite inserir novas manutenções

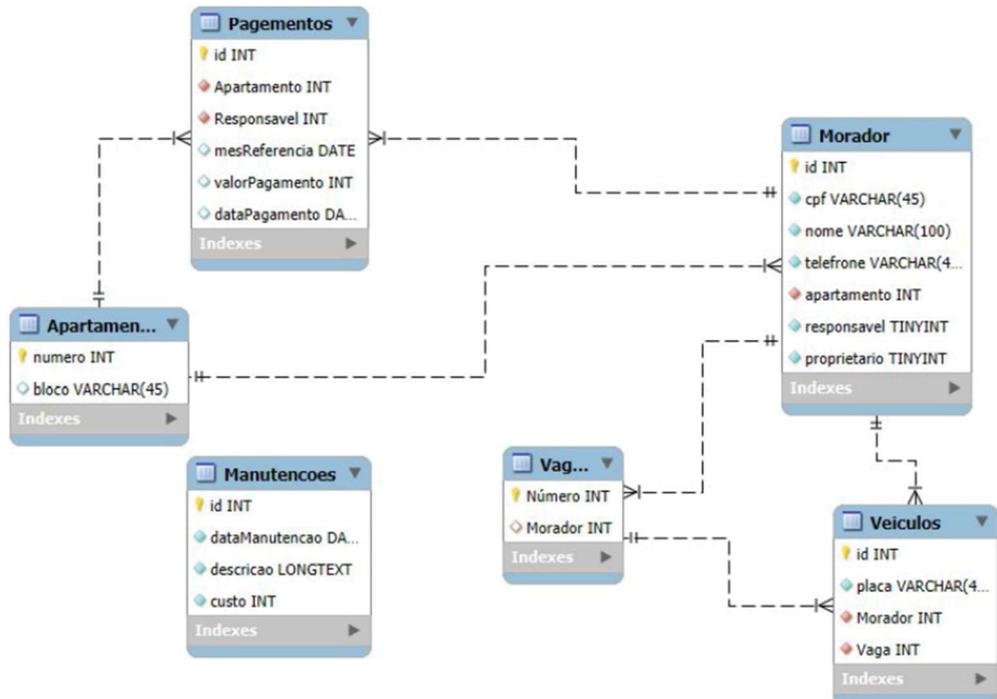
FONTE: O Autor (2025)

FIGURA 10 – DIAGRAMA DE CLASSES



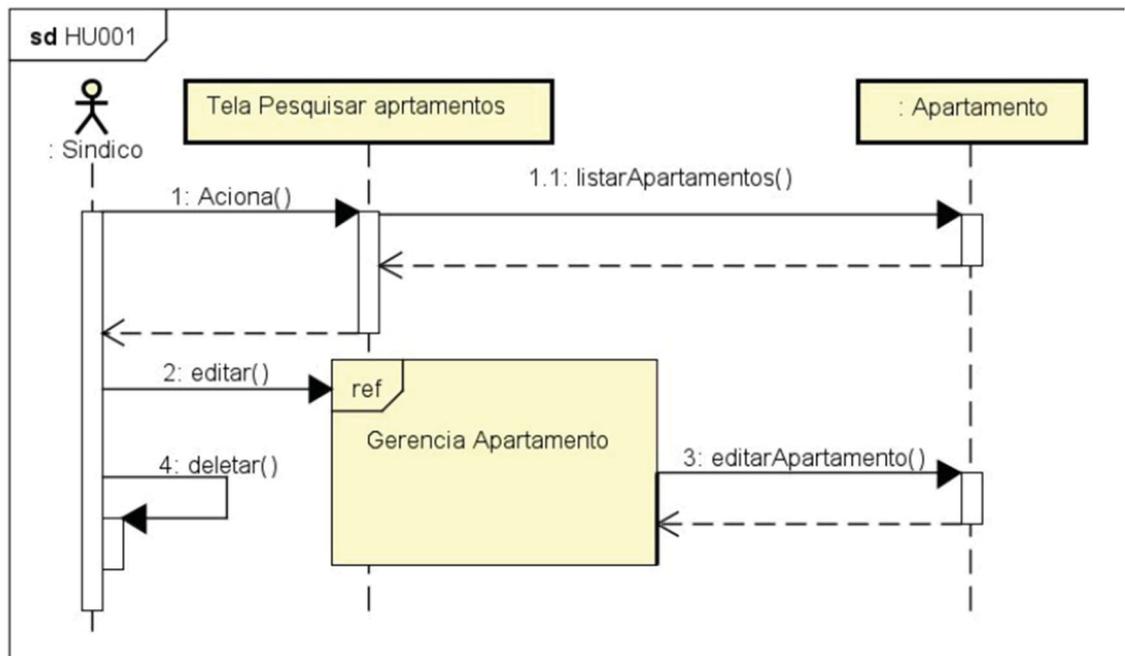
FONTE: O Autor (2025)

FIGURA 11 – TABELAS DO BANCO DE DADOS



FONTE: O Autor (2025)

FIGURA 12 – EXEMPLO DE DIAGRAMA DE SEQUÊNCIA DESENVOLVIDO



FONTE: O Autor (2025)

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

As disciplinas de Gerenciamento Ágil de Projetos 1 e 2, apresentaram projetos que apresentaram conceitos interdisciplinares entre as diferentes técnicas de gerenciamento de projetos.

O projeto da primeira apresentou a elaboração de um plano de *release* que consistiu na construção de um plano de desenvolvimento de um *software* imaginado pelo aluno, neste caso um *software* de transcrição de reuniões, elaborando, de acordo com o modelo disponibilizado, histórias de usuário por *sprints*, com previsões de datas e cálculo de pontos para a estimativa de tempo, aplicando conceitos relacionados aos métodos de planejamento e gerenciamento de *backlog* apresentado ao longo da disciplina (Camargo; Ribas, 2019).

Já a segunda, trouxe a abordagem da execução da simulação de gerenciamento em Kanban (Kanban, 2025). A Simulação consolidou a abordagem da metodologia do Kanban (Anderson, 2011) e a noção do gerenciamento visual aplicado a um projeto.

As disciplinas trouxeram aspectos fundamentais para o gerenciamento e planejamento de projetos ágeis, fundamentando as demais disciplinas construindo uma visão geral sobre os processos de gerenciamento, estabelecimento de métricas e *backlog* de *software* e a importância de práticas como a limitação de trabalho em andamento, que garante a qualidade no desenvolvimento evitando a sobrecarga das equipes envolvidas e o acompanhamento visual das tarefas em andamento.

### 4.1 ARTEFATOS DO PROJETO

FIGURA 13 – PLANO DE RELEASE GAP1

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

**Cálculo da Velocidade:**

Horas disponíveis por dia:	2	Tamanho da Sprint:	4 semanas
Horas disponíveis por Sprint:	40	Velocidade:	5

**Plano de Release:**

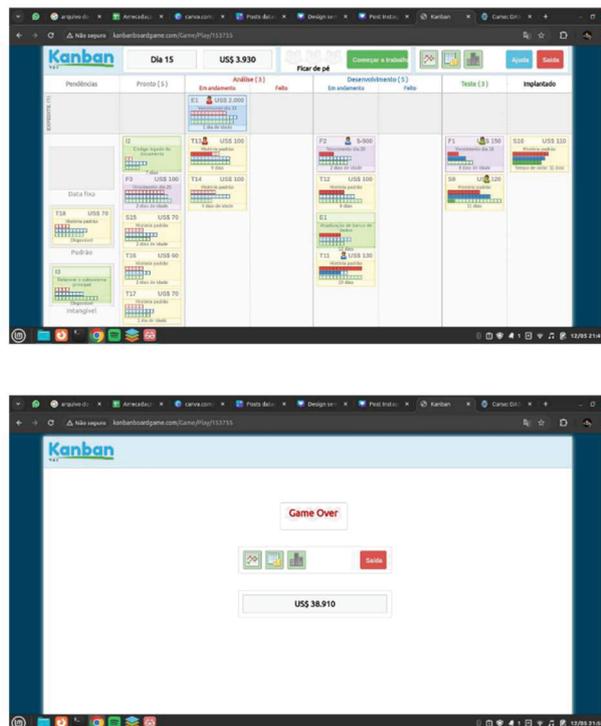
Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint N
Data Início: 01/08 Data Fim: 28/08	Data Início: 01/09 Data Fim: 26/09	Data Início: 30/09 Data Fim: 28/10	Data Início: 30/10 Data Fim: 20/11
H01 – Transcrever áudio SENDO um administrador QUERO transcrever o áudio de uma reunião PARA manter os registros e extrair insights ESTIMATIVA (5)	H02- Gerenciar transcrições – pt1 SENDO um administrador QUERO gerenciar as transcrições das minhas reuniões PARA poder consultar e editar os registros e insights gerados ESTIMATIVA (5)	H02- Gerenciar transcrições – pt2 SENDO um administrador QUERO gerenciar as transcrições das minhas reuniões PARA poder consultar e editar os registros e insights gerados ESTIMATIVA (5)	H03 – Interagir com IA SENDO um administrador QUERO Interagir com minhas transcrições com IA PARA interpretar, resumir e gerar valor com agilidade ESTIMATIVA (3)

FONTE: O Autor (2025)

FIGURA 14 – SIMULAÇÃO DE KANBAN GAP2 PÁGINA 1

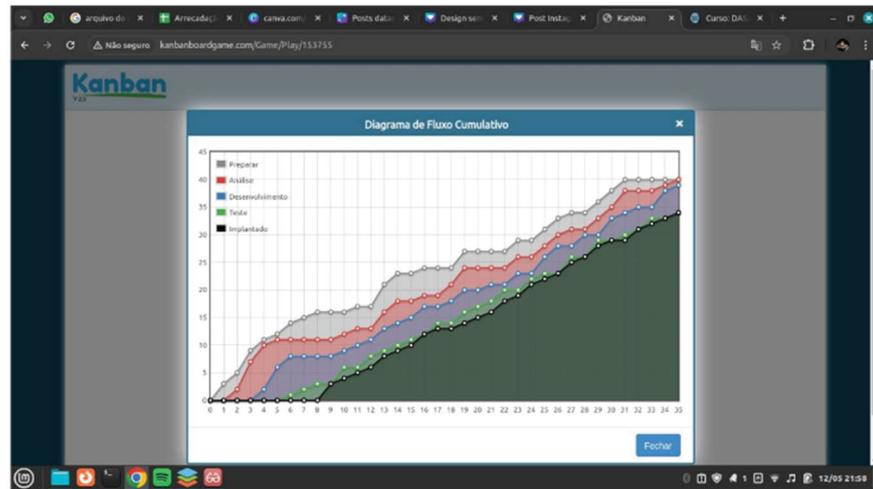
Nome: Luiz Emanuel de Ponce da Silva

Disc. GAP2



FONTE: O Autor (2025)

FIGURA 15 – SIMULAÇÃO DE KANBAN GAP2 PÁGINA 2



FONTE: O Autor (2025)

## 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

O projeto de Introdução à Programação consistiu no desenvolvimento de um *software* em Java (Deitel, 2010), orientado a objetos, com integração com banco e uma boa cobertura de testes. Foram desenvolvidas classes básicas, simulando operações bancárias. O entregável consistiu na entrega dos arquivos contendo as classes desenvolvidas, além de uma imagem contendo os testes unitários fornecidos passando em no mínimo 40 casos de testes simultaneamente.

Essa disciplina constituiu base técnica fundamental e prática no desenvolvimento de *software* com Java, não perdendo de vista os aspectos recomendáveis no desenvolvimento ágil, como a integração entre testes e desenvolvimento.

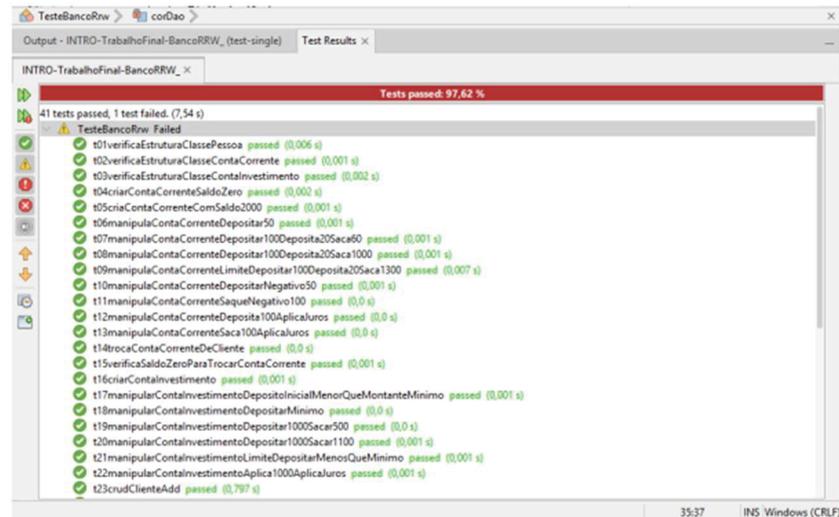
A integração e importância fica ainda mais evidente observado o relacionamento do projeto com as disciplinas de testes automatizados (cobertura de testes obrigatória no entregável), banco de dados, e modelagem de *software*, além de ser alicerce de outras disciplinas do programa que contaram com as noções desenvolvidas em desenvolvimento com Java.

## 5.1 ARTEFATOS DO PROJETO

FIGURA 16 – CASOS DE TESTES ENTREGUES

Relatório/ Casos de testes do Trabalho Final de Introdução à Programação

Luiz Emanuel de Ponce da Silva



FONTE: O Autor (2025)

QUADRO 1 – EXEMPLO DE CÓDIGO DE CLASSE DESENVOLVIDA

```
package bancorrw.conta;

import bancorrw.cliente.Cliente;

public abstract class Conta {
    private long id;
    protected double saldo;
    protected Cliente cliente;

    public Conta(long id, Cliente cliente, double saldoInicial) {
        this.id = id;
        this.cliente = cliente;
        this.saldo = saldoInicial;
    }

    public void deposita(double valor) throws Exception {
        if (valor <= 0) {
```

```
        throw new Exception("Valor do depósito não pode ser
negativo ou zero. Valor=" + valor);
    }
    this.saldo += valor;
}
public void saca(double valor) throws Exception {
    if (valor <= 0) {
        throw new Exception("Valor do saque não pode ser negativo
ou zero. Valor=" + valor);
    }
    this.saldo -= valor;
}
public abstract void aplicaJuros();
public long getId() {
    return id;
}
public void setId(long id){
    this.id = id;
}
public long getNumero() {
    return getId();
}
public double getSaldo() {
    return saldo;
}
public Cliente getCliente() {
    return cliente;
}
}
```

FONTE: O Autor (2025)

## 6 DISCIPLINA: BD – BANCO DE DADOS

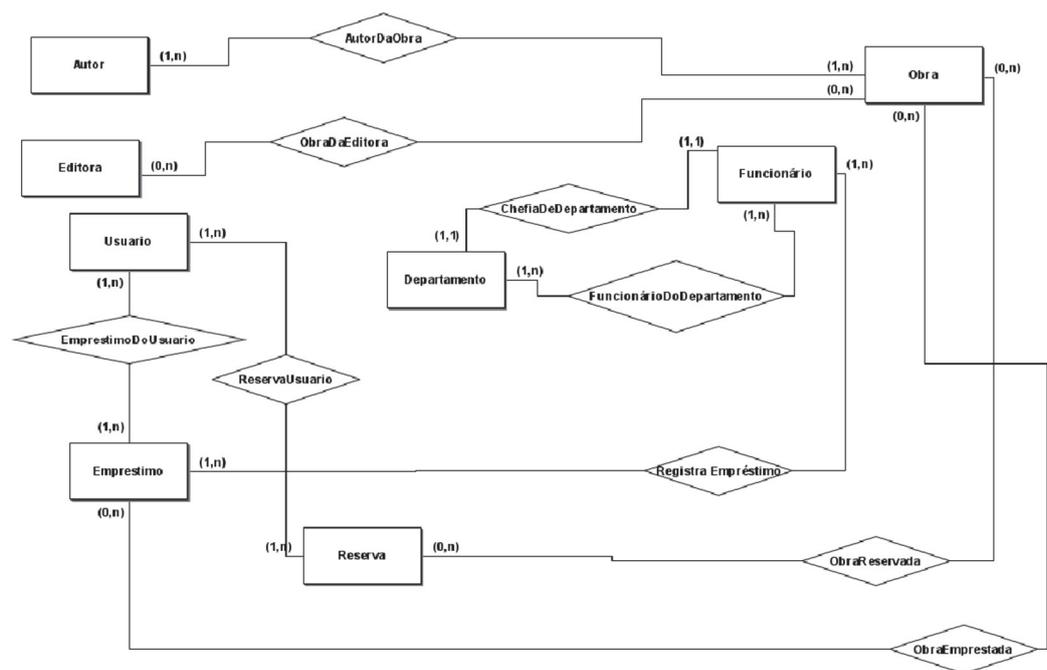
Na disciplina de Banco de Dados o projeto desenvolvido consistiu na elaboração de duas questões. A primeira, demandou a elaboração da modelagem de um sistema de biblioteca, considerando entidades e relacionamentos entre funcionários, leitores, livros, editoras e empréstimos (Elmasri; Navathe, 2011).

A segunda consistiu no desenvolvimento da modelagem de um sistema da escolha do estudante, no meu caso foi construída a modelagem de um sistema de estoque (Elmasri; Navathe, 2011), sendo orientada a modelagem, evidência da cardinalidade dos relacionamentos e geração do *script* SQL (MySQL, 2024) do banco desenvolvido.

O projeto consolidou as noções de modelagem e normalização de banco de dados, tornando base fundamental na construção de banco de dados com qualidade e padronização.

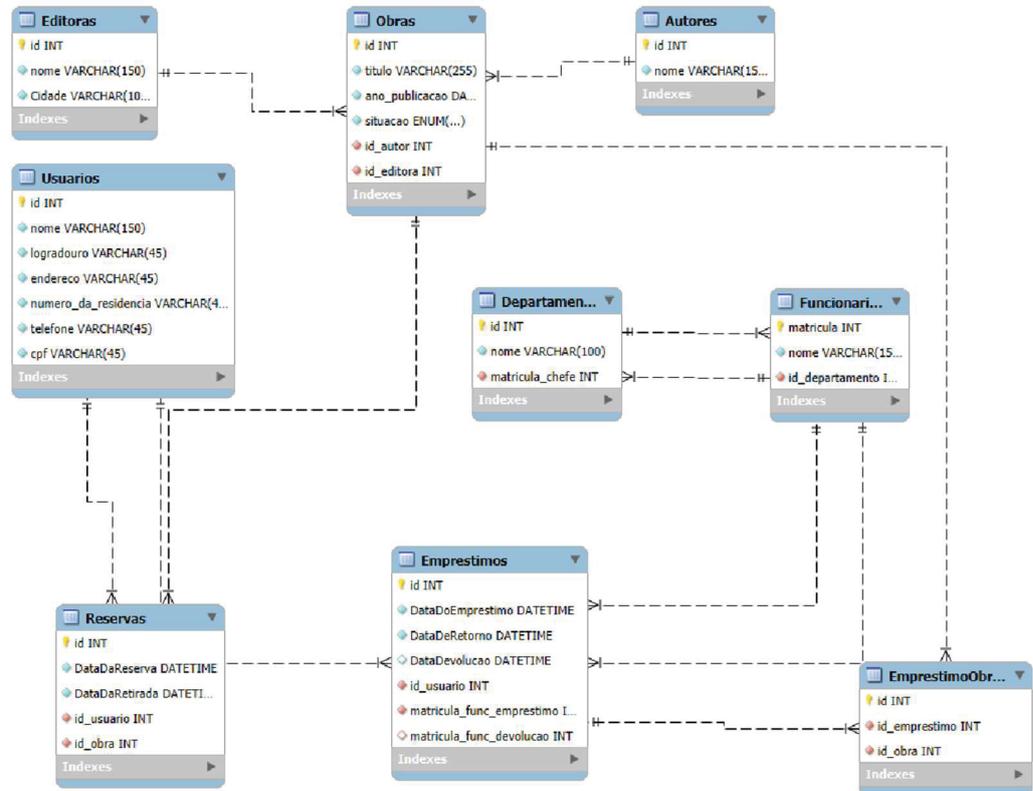
### 6.1 ARTEFATOS DO PROJETO

FIGURA 17 – ESQUEMA CONCEITUAL DO SISTEMA DE BIBLIOTECAS



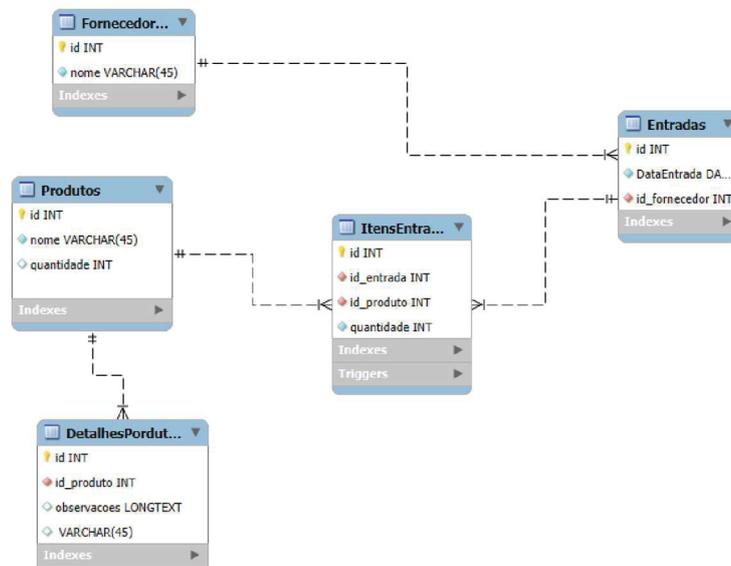
FONTE: O Autor (2025)

FIGURA 18 – ESQUEMA LÓGICO DO SISTEMA DE BIBLIOTECAS



FONTE: O Autor (2025)

FIGURA 19 – ESQUEMA LÓGICO DO SISTEMA DE ESTOQUE



FONTE: O Autor (2025)

## QUADRO 2 – SQL GERADO VIA MSQL WORKBENCH

```

-- MySQL Script generated by MySQL Workbench
-- Sun Aug 31 22:55:07 2025
-- Model: New Model Version: 1.0
-- MySQL Workbench Forward Engineering
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,
NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
-- -----
-- Schema mydb
-- -----
-- Schema mydb
-- -----
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
-- -----
-- Table `mydb`.`Fornecedores`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Fornecedores` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----
-- Table `mydb`.`Produtos`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Produtos` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `quantidade` INT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----
-- Table `mydb`.`Entradas`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Entradas` (

```

```

`id` INT NOT NULL AUTO_INCREMENT,
`DataEntrada` DATE NOT NULL,
`id_fornecedor` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `id_fornecedor_idx` (`id_fornecedor` ASC) VISIBLE,
CONSTRAINT `id_fornecedor`
FOREIGN KEY (`id_fornecedor`)
REFERENCES `mydb`.`Fornecedores` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `mydb`.`ItensEntrada`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`ItensEntrada` (
`id` INT NOT NULL AUTO_INCREMENT,
`id_entrada` INT NOT NULL,
`id_produto` INT NOT NULL,
`quantidade` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `id_entrada_idx` (`id_entrada` ASC) VISIBLE,
INDEX `id_produtos_idx` (`id_produto` ASC) VISIBLE,
CONSTRAINT `id_entrada`
FOREIGN KEY (`id_entrada`)
REFERENCES `mydb`.`Entradas` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `id_produtos`
FOREIGN KEY (`id_produto`)
REFERENCES `mydb`.`Produtos` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `mydb`.`DetalhesProdutos`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`DetalhesProdutos` (
`id` INT NOT NULL AUTO_INCREMENT,
`id_produto` INT NOT NULL,
`observacoes` LONGTEXT NULL,
`` VARCHAR(45) NULL,

```

```

PRIMARY KEY (`id`),
INDEX `id_produto_idx` (`id_produto` ASC) VISIBLE,
CONSTRAINT `id_produto`
FOREIGN KEY (`id_produto`)
REFERENCES `mydb`.`Produtos` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
USE `mydb` ;

-- -----
-- Placeholder table for view `mydb`.`view1`
-- -----

CREATE TABLE IF NOT EXISTS `mydb`.`view1` (`id` INT);
-- -----

-- View `mydb`.`view1`
-- -----

DROP TABLE IF EXISTS `mydb`.`view1`;
USE `mydb`;
USE `mydb`;
DELIMITER $$
USE `mydb`$$

CREATE DEFINER = CURRENT_USER TRIGGER
`mydb`.`ItensEntrada_AFTER_INSERT` AFTER INSERT ON `ItensEntrada`
FOR EACH
ROW
BEGIN
UPDATE Produtos
SET quantidade = quantidade + NEW.quantidade
WHERE id = NEW.id_produto;
END$$

DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

FONTE: O Autor (2025)

## 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

O projeto final da disciplina de Aspectos Ágeis de Programação (AAP) consistiu na aplicação prática de técnicas de refatoração sobre um algoritmo de ordenação *Bubble Sort*. O objetivo central foi transformar um código funcional, porém não otimizado, em um melhor qualidade, seguindo os preceitos de Código Limpo (Martin, 2009).

A atividade exigiu a identificação sistemática de *code smells*, localizando as más práticas presentes, e a aplicação de melhorias estruturais, como a renomeação de variáveis para aumentar a clareza, a remoção de comentários redundantes e a extração de métodos para garantir a coesão e o princípio da responsabilidade única.

Este exercício prático reforça a noção de que a trabalhar a dívida técnica não é uma etapa opcional, mas um processo contínuo de melhoria. A refatoração constante é o mecanismo que permite ao desenvolvimento ágil manter seu ritmo sustentável a longo prazo, garantindo que o código permaneça flexível a mudanças e com qualidade (Fowler, 2004).

### 7.1 ARTEFATOS DO PROJETO

#### QUADRO 3 – ALGORITIMO *BOUBLE SORT* APRESENTADO

```

/*
 * {
 *   estudante: "Luiz Emanuel de Ponce da Silva",
 *   materia: "Aspectos ageis de Programação",
 * }
 *
 * Foram Feitas alterações com relação a utilização de nomes mais
descriptivos,
 * que facilitam a leitura do método, Também houve alterações a fim de
reduzir
 * a assinatura do método (medindo o array dentro do próprio método)
 * e houve a extração de um método, relacionado a reordenação dos
elementos
 * na lista (trocaItens), além de ajustes com relação ao estilo da
formatação,

```

```
* ponto apenas de preferência
*/

class OrdenarLista {

    static void bubbleSort(int lista[]){
        int tamanhoDoArray = lista.length;
        int i, j;
        boolean teveTroca;
        for (i = 0; i < tamanhoDoArray - 1; i++){

            teveTroca = false;

            for (j = 0; j < tamanhoDoArray - i - 1; j++){

                if (lista[j] > lista[j + 1]) {
                    trocaItens(lista, j, j+1);
                    teveTroca = true;
                }

            }

            if (teveTroca == false){
                break;
            }

        }

        static void trocaItens(int lista[], int posicao1, int posicao2){

            int listaTemporaria = lista[posicao1];
            lista[posicao1] = lista[posicao2];
            lista[posicao2] = listaTemporaria;

        }

        static void printArray(int lista[]){
            int tamanhoDoArray = lista.length;
```

```
        for (int i = 0; i < tamanhoDoArray; i++){
            System.out.print(lista[i] + " ");
        }

        System.out.println();
    }

    public static void main(String args[]){

        int lista[] = { 64, 34, 25, 12, 22, 11, 90 };
        bubbleSort(lista);

        System.out.println("Array ordenado: ");
        printArray(lista);

    }
}
```

FONTE: O Autor (2025)

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

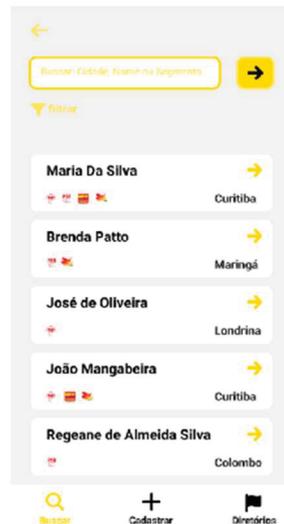
O projeto desenvolvido nesta disciplina consistiu na concepção e prototipagem de um aplicativo destinado ao gerenciamento de contatos em movimentos políticos e sociais. A abordagem adotada fundamentou-se nos preceitos do design centrado no usuário (Lowdermilk, 2013) e na metodologia *Lean UX* (Gothelf, 2021), priorizando a validação rápida de hipóteses de valor e a colaboração em detrimento de documentações extensas, alinhando o desenvolvimento às necessidades reais de engajamento do público-alvo desde a concepção.

As decisões de interface seguiram a estratégia *Mobile-First* (Grant, 2019), reconhecendo a onipresença dos dispositivos móveis no contexto de uso. A construção visual, pautada por tipografia Sans Serif e layout minimalista, buscou otimizar a usabilidade e a disponibilidade (Norman, 2006), reduzindo a carga cognitiva do usuário.

A etapa final do projeto consolidou a integração entre design e agilidade através da validação com uma usuária real do domínio. Este ciclo de *feedback* curto é essencial para mitigar riscos e ajustar a direção do produto de forma iterativa, assegurando que a interface seja validada antes do início do desenvolvimento (Gothelf, 2021).

## 8.1 ARTEFATOS DO PROJETO

FIGURA 20 – EXEMPLO DE TELA DESENVOLVIDA



Página simula o resultado de uma busca, aparecem:

1. nome da pessoa de acordo com a busca;
2. segmentos de interesse da pessoa;
3. botão de ação para acessar os dados da pessoa;
4. ao fim da lista paginação caso haja mais de X resultados;

FONTE: O Autor (2025)

FIGURA 21 – EXEMPLOS DAS ESCOLHAS TÉCNICAS

### 3. Explicar as escolhas feitas (cor, layout, fonte, etc.):

As escolhas foram feitas pensando, primeiro, em um dispositivo móvel, que possui uma tela com tamanho limitado.

As cores utilizadas representam a agremiação a qual fazia parte no momento da concepção.

Foi utilizada uma fonte Sans Serif, a fim de melhorar a legibilidade em telas, de tamanho grande e negrito, a fim de abranger o uso potencial por pessoas idosas

Foi pensada uma interface minimalista, a fim de tornar o uso mais intuitivo e evitar a sobrecarga de informações ao usuário, segmentando em funções globais e encadeadas - buscar, cadastrar e segmentações por movimento, localização etc, acessíveis facilmente e redundantes, a fim de reduzir as navegação entre telas para acesso às funções.

### 4. Mostrar as telas para, pelo menos, 1 (um) possível usuário explicando a função principal do produto, as opções de interação e navegação, e coletar informações de feedback do usuário:

#### Brenda Rompatto - Secretária de Juventude do PSB Paraná:

"Eu achei a parte da identidade visual dele bem clean, minimalista gosto, então me agradou bastante. Eu acho que o aplicativo ele tem que ser direto e rápido e ele aparentemente faz Isso né? É claro que a questão de processador dele ser rápido, dele achar as informações ou não vou conseguir dizer né? mas acredito que seja pra isso a função né? Do aplicativo... E eu acho que é isso. Aparenta ser bem direto."

- transcrição de áudio

FONTE: O Autor (2025)

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

A disciplina de Infraestrutura abordou as práticas, ferramentas e conceitos que unem o desenvolvimento e as operações para automatizar e otimizar o ciclo de vida da entrega de *software*. O conteúdo teórico abordou pilares essenciais como o gerenciamento de configuração, infraestrutura como código e os princípios de integração e entrega contínuas (CI/CD) (Humble; Farley, 2010).

O projeto prático consistiu na orquestração de um ambiente local de CI/CD utilizando a tecnologia de contêineres. Através do Docker (Docker, 2024), foi instanciada uma imagem personalizada `dfwandarti/gitlab_jenkins:3` (Wandarti, 2023) contendo os serviços do GitLab (GitLab, 2024), para controle de versão, e Jenkins (Jenkins, 2024), para automação de *pipelines*.

A atividade envolveu a configuração do ambiente via linha de comando, a execução de *commits* no repositório versionado e o disparo de *builds* automatizados.

Este laboratório proporcionou uma visão sistêmica de como a automação da infraestrutura suporta as práticas ágeis, reduzindo o tempo de espera e garantindo a confiabilidade do processo de *deploy* (Kim *et al.*, 2016).

### 9.1 ARTEFATOS DO PROJETO

FIGURA 22 – AMBIENTE LOCAL EM DOCKER

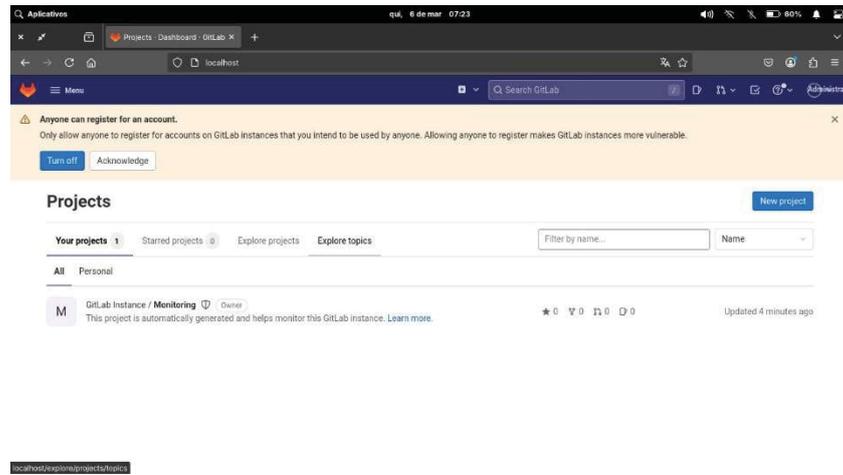
```

qu, 6 de mar 07:20
Downloads: docker
luizmanoelponce-matricula-202100154479 --memory=2g --detach --hostname localhost --publish 443:443 --publish 80:80
--publish 22:22 --publish 9091:9091 --shm-size 256m dfwandarti/gitlab_jenkins:3
a21b64ebde26 luizmanoelponce-matricula-202100154479 dfwandarti/gitlab_jenkins:3
luizmanoelponce-matricula-202100154479$ docker ps -l
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
a21b64ebde26  dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       14 seconds ago Up 13 seconds (health: starting)  0.0.0.0:22->22/tcp, [::]:22->22/tcp, 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp, 0.0.0.0:9091->9091/tcp, [::]:9091->9091/tcp  luizmanoelponce-matricula-202100154479
luizmanoelponce-matricula-202100154479$ docker ps -l
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
a21b64ebde26  dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       About a minute ago Up About 2 minutes (health: starting)  0.0.0.0:22->22/tcp, [::]:22->22/tcp, 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp, 0.0.0.0:9091->9091/tcp, [::]:9091->9091/tcp  luizmanoelponce-matricula-202100154479
luizmanoelponce-matricula-202100154479$ docker ps -l
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
a21b64ebde26  dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       3 minutes ago Up 3 minutes (health: starting)  0.0.0.0:22->22/tcp, [::]:22->22/tcp, 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp, 0.0.0.0:9091->9091/tcp, [::]:9091->9091/tcp  luizmanoelponce-matricula-202100154479
luizmanoelponce-matricula-202100154479$ docker ps -l
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
a21b64ebde26  dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       5 minutes ago Up 5 minutes (unhealthy)  0.0.0.0:22->22/tcp, [::]:22->22/tcp, 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp, 0.0.0.0:9091->9091/tcp, [::]:9091->9091/tcp  luizmanoelponce-matricula-202100154479
luizmanoelponce-matricula-202100154479$

```

FONTE: O Autor (2025)

FIGURA 23 – INTERFACE WEB DO GIT LAB



FONTE: O Autor (2025)

FIGURA 24 – COMMIT GIT EXECUTADO NO AMBIENTE LOCAL

```

luizponce@lenovo-luiz:~/Downloads/Monitoring$ git add *
luizponce@lenovo-luiz:~/Downloads/Monitoring$ git commit -m "commit do trabalho"
Author identity unknown

*** Please tell me who you are.

Run

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'luizponce@lenovo-luiz [none]')
luizponce@lenovo-luiz:~/Downloads/Monitoring$ git config --global user.email "luizmanoe1ponce@outlook.com.br"
luizponce@lenovo-luiz:~/Downloads/Monitoring$ git config --global user.name "Luiz Ponce"
luizponce@lenovo-luiz:~/Downloads/Monitoring$ git commit -m "commit do trabalho"
[main (root-commit) d218531] commit do trabalho
 1 file changed, 1 insertion(+)
 create mode 100644 Ola.txt
luizponce@lenovo-luiz:~/Downloads/Monitoring$ git push origin main
Username for 'http://localhost': root
Password for 'http://localhost':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 234 bytes | 234.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://localhost:gitlab-instance-1682980/Monitoring.git
 * [new branch]      main -> main
luizponce@lenovo-luiz:~/Downloads/Monitoring$ git log
commit d21853186070d80d9e15878e3f0b20e5d79eccc (HEAD -> main, origin/main)
Author: Luiz Ponce <luizmanoe1ponce@outlook.com.br>
Date: Thu Mar 6 07:29:12 2025 -0300

    commit do trabalho
luizponce@lenovo-luiz:~/Downloads/Monitoring$

```

FONTE: O Autor (2025)

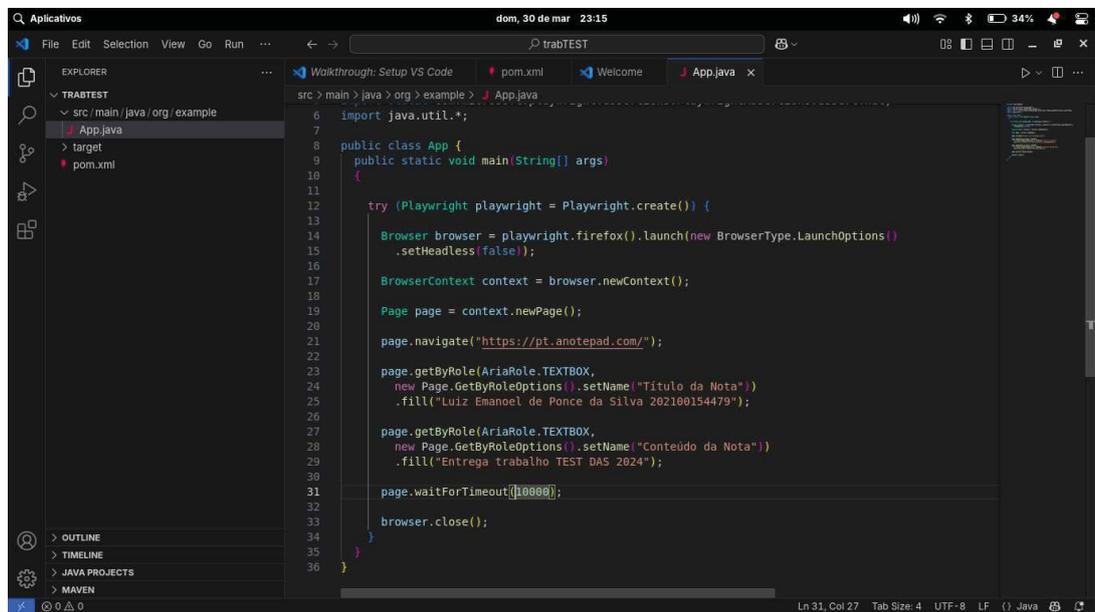
## 10 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

A disciplina de Testes Automatizados (TEST) abordou os fundamentos da garantia da qualidade, estabelecendo a distinção teórica entre verificação, garantir que o produto foi construído corretamente, e validação, garantir que o produto certo está sendo construído. O conteúdo enfatizou que a automação não é apenas uma ferramenta de execução, mas uma estratégia integrada ao ciclo de vida ágil para fornecer *feedback* rápido e contínuo sobre a integridade do sistema (Universidade Federal do Paraná, 2024).

O projeto prático consistiu na implementação de uma suíte de testes, simulando a jornada de um usuário real utilizando o *framework Playwright* (Microsoft, 2024), integrado à linguagem Java (Oracle, 2023), foi configurado um ambiente de execução que instância o navegador e automatiza a interação direta com o DOM (*Document Object Model*). O *script* desenvolvido realizou a navegação até a aplicação alvo (anotepad.com), efetuou a injeção de dados nos campos de entrada *título* e *conteúdo* e validou a persistência das informações na interface, assegurando a conformidade funcional do fluxo da aplicação.

## 10.1 ARTEFATOS DO PROJETO

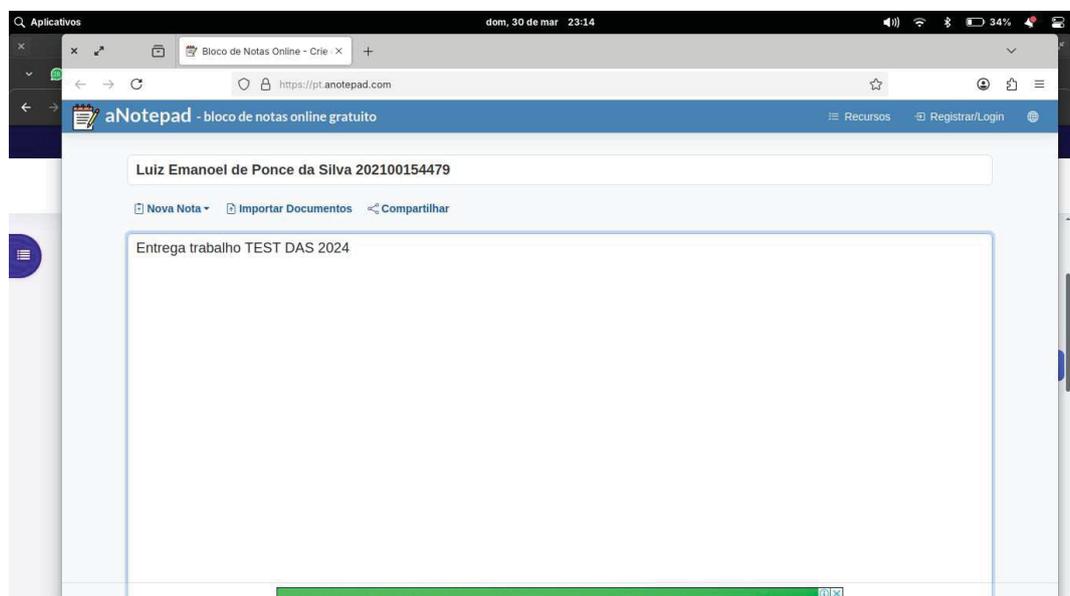
FIGURA 25 – CÓDIGO DO PLAYWEIGHT



```
src > main > java > org > example > App.java
6  import java.util.*;
7
8  public class App {
9      public static void main(String[] args)
10     {
11
12         try (Playwright playwright = Playwright.create()) {
13
14             Browser browser = playwright.firefox().launch(new BrowserType.LaunchOptions()
15                 .setHeadless(false));
16
17             BrowserContext context = browser.newContext();
18
19             Page page = context.newPage();
20
21             page.navigate("https://pt.anothepad.com/");
22
23             page.getByRole(AriaRole.TEXTBOX,
24                 new Page.GetByRoleOptions().setName("Título da Nota"))
25                 .fill("Luiz Emanuel de Ponce da Silva 202100154479");
26
27             page.getByRole(AriaRole.TEXTBOX,
28                 new Page.GetByRoleOptions().setName("Conteúdo da Nota"))
29                 .fill("Entrega trabalho TEST DAS 2024");
30
31             page.waitForTimeout(10000);
32
33             browser.close();
34         }
35     }
36 }
```

FONTE: O Autor (2025)

FIGURA 26 – NAVEGADOR DE INTERNET INVOCADO PELO SCRIPT



FONTE: O Autor (2025)



## 11 CONCLUSÃO

A análise dos projetos e conceitos explorados neste trabalho demonstra que os aspectos do gerenciamento ágil são indispensáveis para a entrega de *software* com qualidade e valor na atualidade. A eficácia de *frameworks* de planejamento iterativo, como o Scrum, ou de gestão de fluxo visual, como o Kanban, não existe de forma isolada. Essas abordagens de gerenciamento são indissociáveis das práticas de engenharia de *software* que garantem a qualidade interna do produto, formando, em conjunto, um sistema sociotécnico integrado.

A modelagem ágil define a estrutura, o desenvolvimento orientado a testes (TDD) valida o comportamento desde o início, e os *pipelines* de integração e entrega Contínua (CI/CD) automatizam a entrega. É essa integração sistêmica que permite responder às mudanças e entregar *software* funcional de forma contínua.

Com base na experiência prática, a adoção bem-sucedida desses aspectos gerenciais enfrenta desafios que são, em sua maioria, pontos de atrito entre a gestão do projeto e a execução técnica. A otimização do fluxo, por exemplo, exige a definição e o ajuste contínuo dos limites de *work in progress* (WIP). A gestão da dívida técnica requer que a refatoração seja priorizada no *backlog*, mesmo sob pressão por novas funcionalidades.

O sucesso prático depende da superação da curva de aprendizado do TDD, da manutenção da confiabilidade dos *pipelines* de automação e, acima de tudo, da mudança cultural de um sistema de *empurrar* tarefas para um sistema *puxado*, baseado na auto-organização da equipe de desenvolvimento *software*.

## REFERÊNCIAS

- ANDERSON, D. J. **Kanban: mudança evolucionária para seu negócio de tecnologia**. Seattle: Blue Hole Press, 2011.
- BECK, K. **Programação extrema explicada: acolha as mudanças**. Porto Alegre: Bookman, 2004.
- BECK, K. *et al.* **Manifesto para desenvolvimento ágil de software**. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 24 nov. 2025.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Campus, 2005.
- CAMARGO, R.; RIBAS, T. **Gestão ágil de projetos: as melhores soluções para suas necessidades**. São Paulo: Saraiva Educação, 2019.
- COHN, M. **Desenvolvimento de software com Scrum: aplicando métodos ágeis com sucesso**. Porto Alegre: Bookman, 2011.
- DEITEL, P.; DEITEL, H. **Java: como programar**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.
- DOCKER. **Docker Documentation**. 2024. Disponível em: <https://docs.docker.com/>. Acesso em: 25 nov. 2025.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.
- FOWLER, M. **Refatoração: aperfeiçoando o projeto de código existente**. Porto Alegre: Bookman, 2004.
- GITLAB. **GitLab Documentation**. 2024. Disponível em: <https://docs.gitlab.com/>. Acesso em: 25 nov. 2025.
- GOTHELF, J.; SEIDEN, J. **Lean UX: designing great products with agile teams**. 3. ed. Sebastopol: O'Reilly Media, 2021.
- GRANT, W. **UX Design: guia definitivo com as melhores práticas de UX**. São Paulo: Novatec Editora, 2019.
- HUMBLE, J.; FARLEY, D. **Continuous Delivery: reliable software releases through build, test, and deployment automation**. Upper Saddle River: Addison-Wesley, 2010.
- JENKINS. **Jenkins User Documentation**. 2024. Disponível em: <https://www.jenkins.io/doc/>. Acesso em: 25 nov. 2025.

KANBAN BOARD GAME. **Kanban Board Game**. 2025. Simulação online de método Kanban. Disponível em: <http://www.kanbanboardgame.com/>. Acesso em: 25 nov. 2025.

KIM, G. *et al.* **The DevOps Handbook**: how to create world-class agility, reliability, and security in technology organizations. Portland: IT Revolution Press, 2016.

LOWDERMILK, T. **Design centrado no usuário**: um guia para o desenvolvimento de aplicativos amigáveis. São Paulo: Novatec Editora, 2013.

MARTIN, R. C. **Código limpo**: habilidades práticas do agile software. Rio de Janeiro: Alta Books, 2009.

MICROSOFT. **Playwright for Java**: Documentation. 2024. Disponível em: <https://playwright.dev/java/>. Acesso em: 25 nov. 2025.

MYSQL. **MySQL 8.3 Reference Manual**. 2024. Disponível em: <https://dev.mysql.com/doc/refman/8.3/en/>. Acesso em: 25 nov. 2025.

NORMAN, D. A. **O design do dia-a-dia**. Rio de Janeiro: Rocco, 2006.

ORACLE. **Trail: Learning the Java Language**. The Java Tutorials. 2023. Disponível em: <https://docs.oracle.com/javase/tutorial/>. Acesso em: 25 nov. 2025.

POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development**: An Agile Toolkit. Boston: Addison-Wesley, 2003.

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

UNIVERSIDADE FEDERAL DO PARANÁ (UFPR). **INFRA - Infraestrutura para desenvolvimento e implantação de Software (DevOps)**: [Slides da disciplina]. Curitiba: UFPR/SEPT, 2024d. Material da Especialização em Desenvolvimento Ágil de Software.

UNIVERSIDADE FEDERAL DO PARANÁ (UFPR). **TEST - Testes Automatizados**: [Slides da disciplina]. Curitiba: UFPR/SEPT, 2024. Material da Especialização em Desenvolvimento Ágil de Software.

WANDARTI, D. **gitlab\_jenkins**. Versão 3. Docker Hub, 2023. Imagem de container. Disponível em: [https://hub.docker.com/r/dfwandarti/gitlab\\_jenkins](https://hub.docker.com/r/dfwandarti/gitlab_jenkins). Acesso em: 25 nov. 2025.