

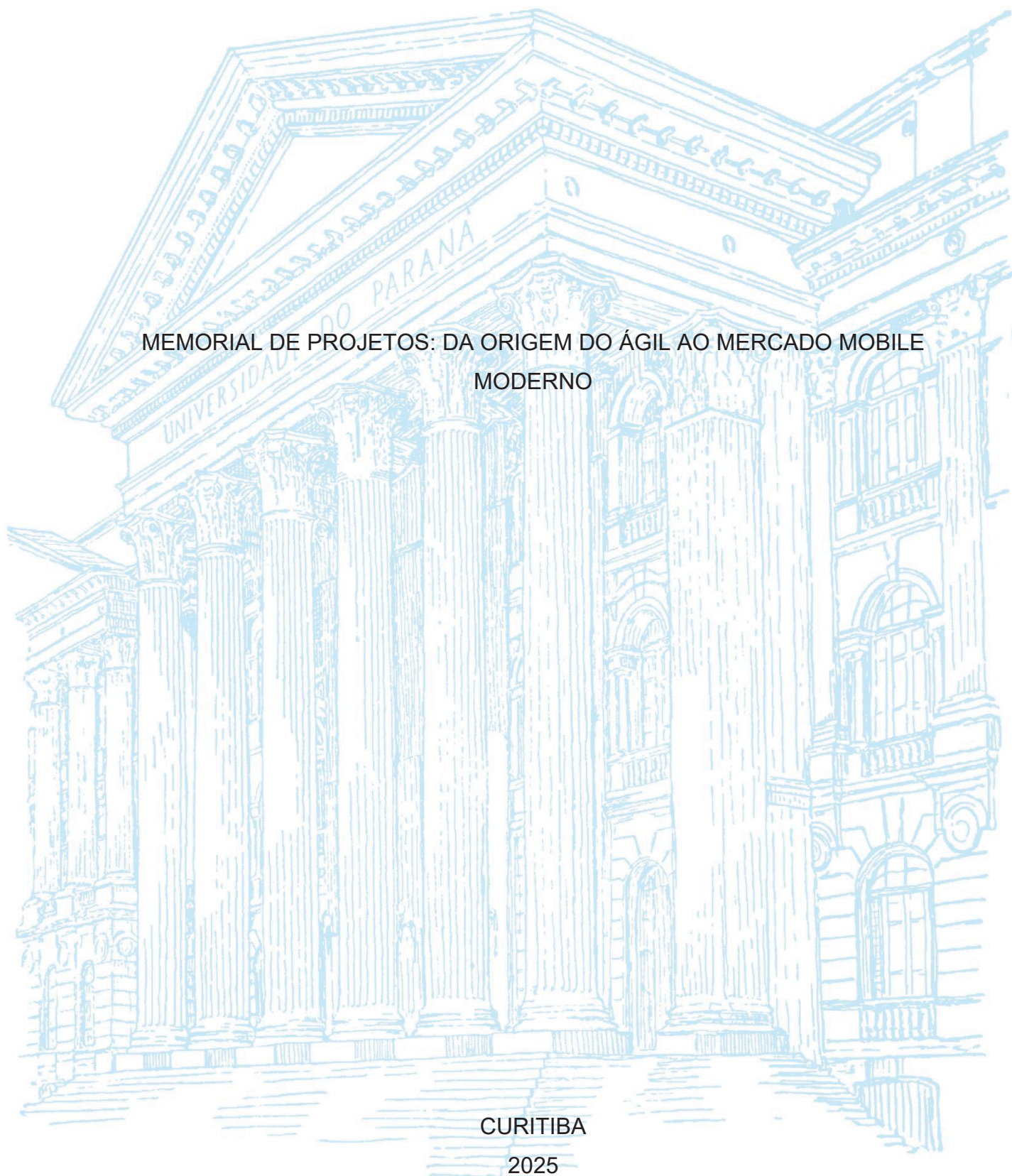
UNIVERSIDADE FEDERAL DO PARANÁ

MATHEUS YUKIO LOPES SHIMAZU

MEMORIAL DE PROJETOS: DA ORIGEM DO ÁGIL AO MERCADO MOBILE  
MODERNO

CURITIBA

2025



MATHEUS YUKIO LOPES SHIMAZU

MEMORIAL DE PROJETOS: DA ORIGEM DO ÁGIL AO MERCADO MOBILE  
MODERNO

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

Orientador: Prof. Dr. 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 **MATHEUS YUKIO LOPES SHIMAZU**, intitulada: **MEMORIAL DE PROJETOS: DA ORIGEM DO ÁGIL AO MERCADO MOBILE MODERNO**, 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, 01 de Dezembro de 2025.



RAZER ANTHOM NIZER ROJAS MONTAÑO

Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

## RESUMO

O trabalho Memorial de Projetos: Da Origem do Ágil ao Mercado *Mobile* Moderno é uma compilação dos projetos finais de cada disciplina do curso de Especialização em Desenvolvimento Ágil de Software de 2024 da UFPR. É apresentada a trajetória do aprendizado, desde a origem do Manifesto Ágil, acumulando teorias, princípios e valores como base para a aplicação prática dos preceitos ágeis em contextos Web e *Mobile*, assim como um breve resumo da história de aplicativos de celular e as necessidades do mercado de apps. Cada matéria demonstra seu projeto, seus objetivos e importância na área ágil assim como nas outras disciplinas. Em seguida são apresentados artefatos relevantes gerados ao longo de cada projeto, como diagramas, imagens de tela e trechos de códigos. O trabalho conclui com uma análise do curso, comentários sobre disciplinas especialmente relevantes e desafios no uso de metodologias ágeis, assim como discussões sobre a afinidade de tais no mercado *Mobile*.

**Palavras-chave:** desenvolvimento ágil; análise de mercado; aplicações móveis.

## **ABSTRACT**

The report Project Memorial: From the Origin of Agile to the Modern Mobile Market is a compilation of final projects of each subject of the UFPR 2024 Specialization in Agile Software Development. The learning path is presented, from the origin of Agile Manifesto, amassing theories, principles and values as basis for practical application of agile precepts in Web and Mobile contexts, as well as a brief summary of the history of cellphone applications and apps market needs. Each class demonstrates its project, its goals and its importance in the agile context as well as in other subjects. Subsequently presented are relevant artifacts generated throughout each project, like diagrams, screen images and code. The report concludes with an analysis of the course, remarks on especially relevant classes and challenges in the use of agile methodologies, as well as discussions about the affinity of such in the Mobile market.

**Keywords:** agile development; market analysis; mobile applications.

## 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>10</b>
2.1	ARTEFATOS DO PROJETO .....	11
<b>3</b>	<b>DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2.....</b>	<b>18</b>
3.1	ARTEFATOS DO PROJETO .....	19
<b>4</b>	<b>DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2 .....</b>	<b>23</b>
4.1	ARTEFATOS DO PROJETO .....	24
<b>5</b>	<b>DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....</b>	<b>27</b>
5.1	ARTEFATOS DO PROJETO .....	28
<b>6</b>	<b>DISCIPLINA: BD – BANCO DE DADOS.....</b>	<b>29</b>
6.1	ARTEFATOS DO PROJETO .....	30
<b>7</b>	<b>DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO .....</b>	<b>34</b>
7.1	ARTEFATOS DO PROJETO .....	35
<b>8</b>	<b>DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2 .....</b>	<b>37</b>
8.1	ARTEFATOS DO PROJETO .....	37
<b>9</b>	<b>DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE....</b>	<b>40</b>
9.1	ARTEFATOS DO PROJETO .....	41
<b>10</b>	<b>DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....</b>	<b>42</b>
<b>11</b>	<b>DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS) .....</b>	<b>43</b>
11.1	ARTEFATOS DO PROJETO .....	44
<b>12</b>	<b>DISCIPLINA: TEST – TESTES AUTOMATIZADOS .....</b>	<b>46</b>
12.1	ARTEFATOS DO PROJETO .....	47
<b>13</b>	<b>CONCLUSÃO .....</b>	<b>48</b>
	<b>REFERÊNCIAS .....</b>	<b>50</b>

## 1 PARECER TÉCNICO

Este parecer técnico apresenta os vários projetos elaborados ao longo do curso, evidenciando os diversos tópicos que juntos compõe o Desenvolvimento Ágil, desde sua origem, valores e princípios até sua aplicação no mercado *mobile* atual.

Cada disciplina da especialização de Desenvolvimento Ágil de Software fornece alicerces teóricos, práticos e ferramentas para a progressão das matérias subsequentes, de modo a permitir que o aluno forme um conhecimento abrangente das diversas etapas do processo de desenvolvimento ágil, a fim de que, na conclusão do curso, o aluno tenha fundação para construir aplicações completas em diversas plataformas aplicando os ideais do Manifesto Ágil de forma efetiva e eficiente.

A matéria inicial Métodos Ágeis de Desenvolvimento de Software (MADS) dá um panorama dos métodos de desenvolvimento chamados Tradicionais e seus problemas para explicar a motivação e origem do Manifesto Ágil (Beck *et al*, 2001), introduzindo alguns métodos e ferramentas ágeis que compartilham seus valores e princípios como o Lean (Poppendieck; Poppendieck, 2003), Scrum (Sutherland *et al*, 2004), *Extreme Programming* (Shrivastava *et al*, 2021), Kanban (Coleman; Vacanti, 2025) e Entrega Contínua (Chen, 2015).

Com essa base teórica, seguem as disciplinas de Modelagem Ágil de Software (MAG1 e MAG2), onde é visto o processo de modelagem e elaboração de requisitos, utilizando diagramas UML (OMG, 2017) e histórias de usuário, cujo propósito é facilitar o entendimento e mudanças posteriores por todas as partes do projeto.

As disciplinas GAP1 e GAP2 de Gerenciamento Ágil de Projetos aprofundam o aprendizado de Scrum (Sutherland *et al*, 2004) e Kanban (Coleman; Vacanti, 2025) introduzidos na matéria MADS, incluindo tópicos de planejamento de *release* e aplicando na prática a teoria, exemplificando e reforçando princípios como limitação de WIP (*Work in Progress*) e entrega contínua de valor.

As matérias Introdução a Programação (INTRO) e Aspectos Ágeis de Programação (APP) garantem uma familiaridade base com programação, utilizando o Desenvolvimento Movido a Testes ou TDD (Jeffries; Melnik, 2007) para garantir o comportamento esperado do código de forma rápida e independente de detalhes



internos, facilitando refatorações posteriores, cujas técnicas são vistas a fundo em APP.

Este conhecimento em programação é incrementado na disciplina de Banco de Dados (BD), fazendo possível a criação de aplicações mais relevantes, com a persistência de dados permitindo softwares mais complexos e o armazenamento, processamento e análise da utilização de usuários gerando alto valor para os desenvolvedores na melhoria da aplicação.

Todos esses fatores vistos nas disciplinas anteriores são então reunidos e aplicados nas disciplinas de Desenvolvimento WEB (WEB1, WEB2) e Mobile (MOB1, MOB2), onde são construídos sites e aplicações *mobile* completos e funcionais, introduzindo o uso de *frameworks* como o Angular (2025) e IDEs como o Android Studio (2025), que facilitam e agilizam o desenvolvimento de forma sistemática e automatizada. Com poucas linhas de comandos é possível criar várias partes do software que são burocráticas e comuns entre programas semelhantes. O Android Studio ainda permite programação gráfica, onde a tela é construída arrastando elementos visuais, e o uso de inteligência artificial para analisar, corrigir e autocompletar trechos grandes de código, de modo a criar funções inteiras com poucas letras digitadas.

Ambas as plataformas *web* e *mobile* se beneficiam muito do que é apresentado em UX (*User Experience* – experiência de usuário (Teixeira, 2014)) no Desenvolvimento Ágil de Software (UX). A prototipação e a coleta de *feedback* é fundamental para a redução do retrabalho e aceitação do usuário. Fatores aparentemente pequenos como fontes, cores e botões diferentes podem ser o bastante para que um usuário prefira um software concorrente, o que é exacerbado no contexto *mobile*.

Aplicativos *mobile* tendem a necessitar *feedback*, correções e melhorias constantes e mais rapidamente do que outras plataformas, principalmente devido ao fácil acesso por celulares e a ampla concorrência pelas lojas de aplicativos que tem alcance global. Neste contexto, a implantação de código constante amplia a chance de introduzir erros. Para sanar isso de maneira ágil e eficaz, é necessário aplicar o que é apresentado nas disciplinas Infraestrutura para Desenvolvimento e Implantação de Software - DevOps (INFRA) e Testes Automatizados (TEST) que finalizam o curso.



DevOps (Ebert *et al*, 2016) reduz gargalos burocráticos e erros de implantação, garantindo que o código entre em produção de forma rápida, eficaz e automatizada. Enquanto testes automatizados (Kong *et al*, 2019) complementam e se integram em DevOps, garantindo corretude do código e reduzindo erros humanos na testagem.

Nesta trajetória é visto como as ideias do Manifesto Ágil (Beck *et al*, 2001) seguiram evoluindo e se adequando a diferentes contextos de programação, para chegar à era das lojas de aplicativos *mobile* com o lançamento da App Store (2025) da Apple em 2008, que conecta desenvolvedores grandes e pequenos a usuários no mundo todo com as mais diversas necessidades e preferências, gerando uma explosão de aplicativos *mobile* e seus usuários.

Em 2014 aplicativos *mobile* ultrapassaram o uso de internet em computadores nos Estados Unidos (CNN, 2014), e em 2016, globalmente (Conte, 2025). Com tamanho alcance e concorrência, os aplicativos *mobile* precisam de diferenciais para ter sucesso no mercado, não basta funcionar, precisa ter UX melhor, *feedback* mais rápido e efetivo, desenvolvimento e implantação antes dos concorrentes, erros e falhas mínimas são um desastre neste contexto. Aplicativos de mensagens e redes sociais grandes, se ficarem sem serviço por um minuto, causam prejuízos globais financeiros e de reputação.

Este trabalho é então finalizado com uma análise da adequação dos conceitos teóricos e técnicas práticas do desenvolvimento ágil neste contexto atual de lojas de aplicativos móveis.

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

Nesta primeira disciplina do curso de Desenvolvimento Ágil de Software foram apresentados conceitos relevantes para o início do aprendizado de Métodos ágeis, assim como métodos de organização e síntese de materiais, como mapas mentais (Erdem, 2017), para facilitar o estudo.

O projeto final desta disciplina foi a elaboração de um mapa mental contendo conceitos e suas breves explicações, com os principais tópicos sendo Processo de Software (Leau *et al*, 2012), Modelos Tradicionais de Desenvolvimento (Leau *et al*, 2012), Manifesto Ágil (Beck *et al*, 2001), Princípios Ágeis (Beck *et al*, 2001), Lean (Poppendieck; Poppendieck, 2003), Scrum (Sutherland *et al*, 2004), *Extreme Programming* (Shrivastava *et al*, 2021), Kanban (Coleman; Vacanti, 2025) e Entrega Contínua (Chen, 2015).

Este projeto é importante porque agrega todo o conhecimento teórico básico sobre desenvolvimento ágil, mostrando as diferenças entre métodos tradicionais e ágeis, apresentando valores e princípios que permeiam tudo que será aprendido nas disciplinas subsequentes e descrevendo os principais métodos práticos de programação ágil.

Outro fator importante é que todos esses conceitos são apresentados de forma sucinta e de fácil entendimento através do mapa mental. Isso nos mostra o quanto importante um mapa mental é para o resumo de um conteúdo complexo, facilitando o aprendizado, o estudo e a posterior recordação da matéria. Tal ferramenta será muito útil nas próximas disciplinas, onde serão apresentados conceitos mais complexos que aprofundam os conceitos apresentados nesta matéria.

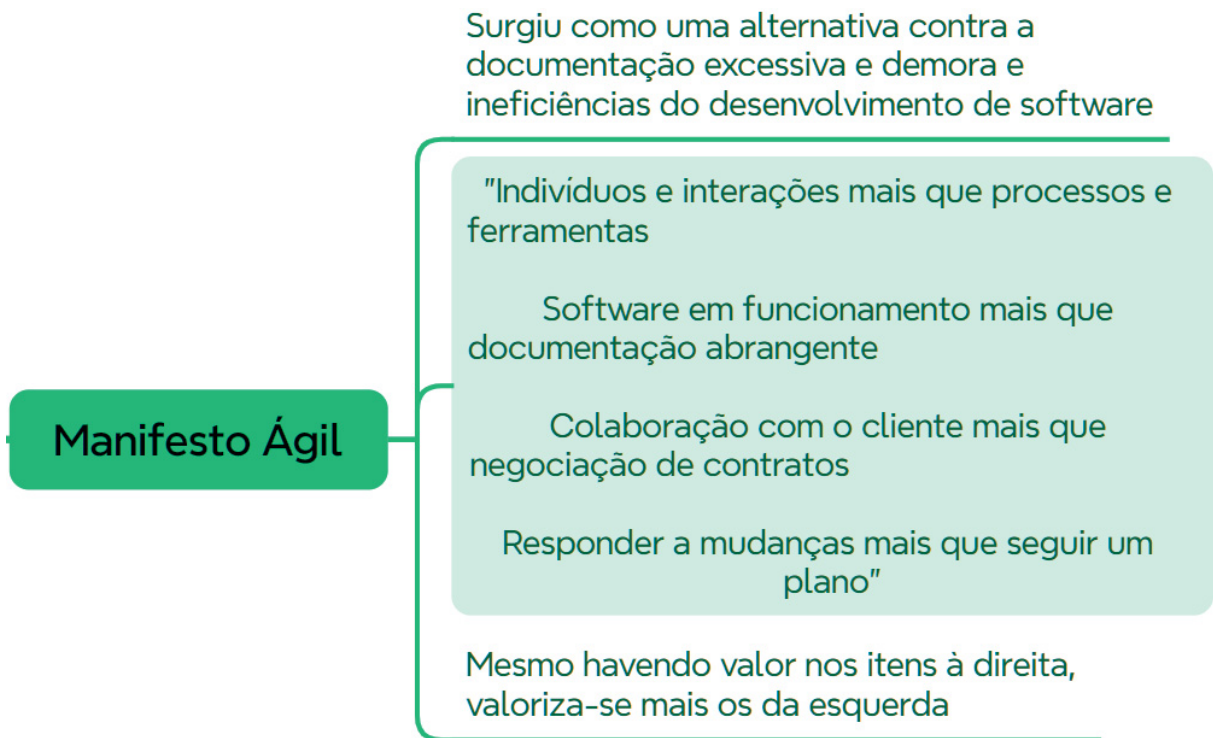
## 2.1 ARTEFATOS DO PROJETO

FIGURA 1 – MAPA MENTAL DE MÉTODOS ÁGEIS



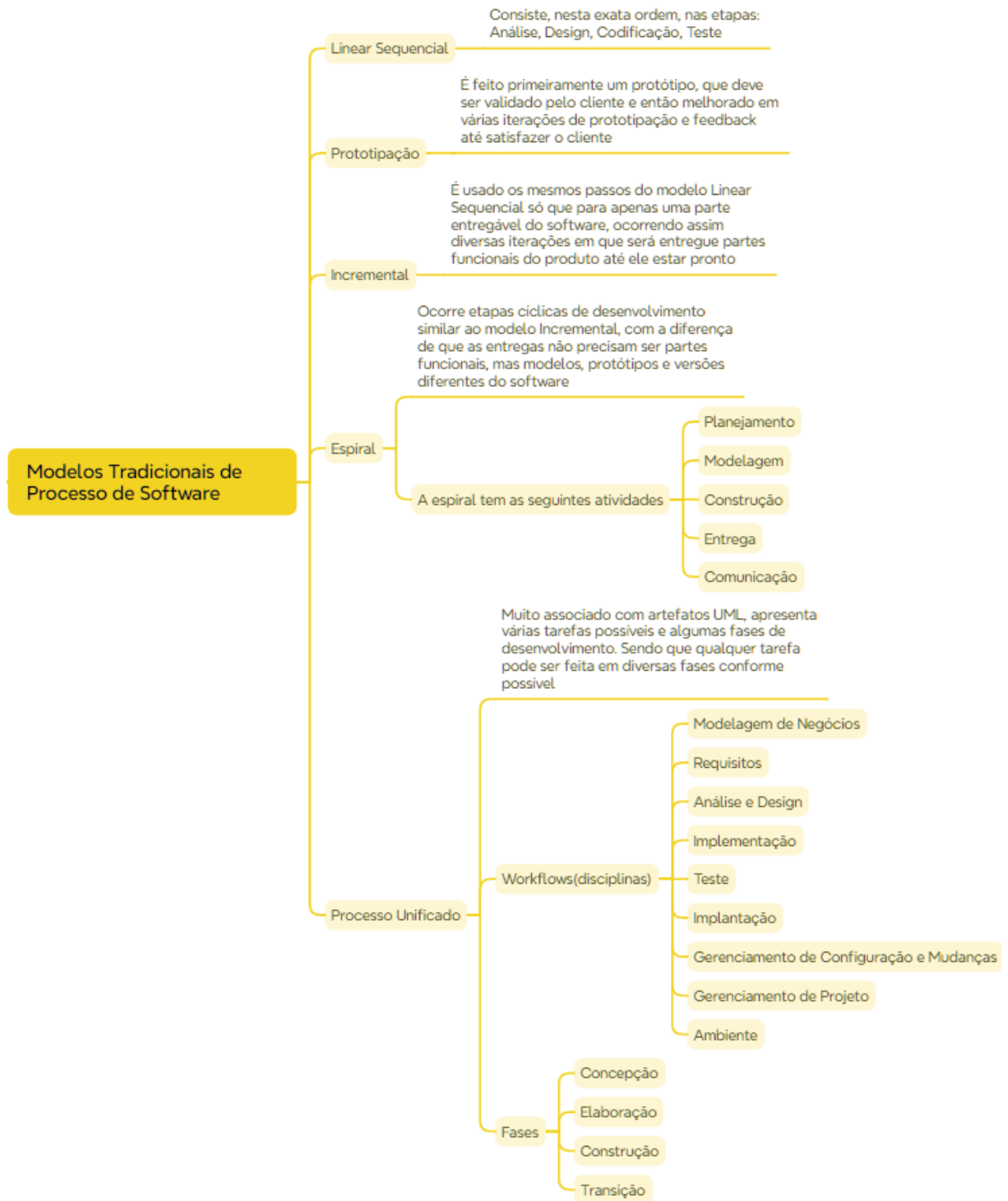
FONTE: O autor (2025)

FIGURA 2 – TÓPICO: MANIFESTO ÁGIL



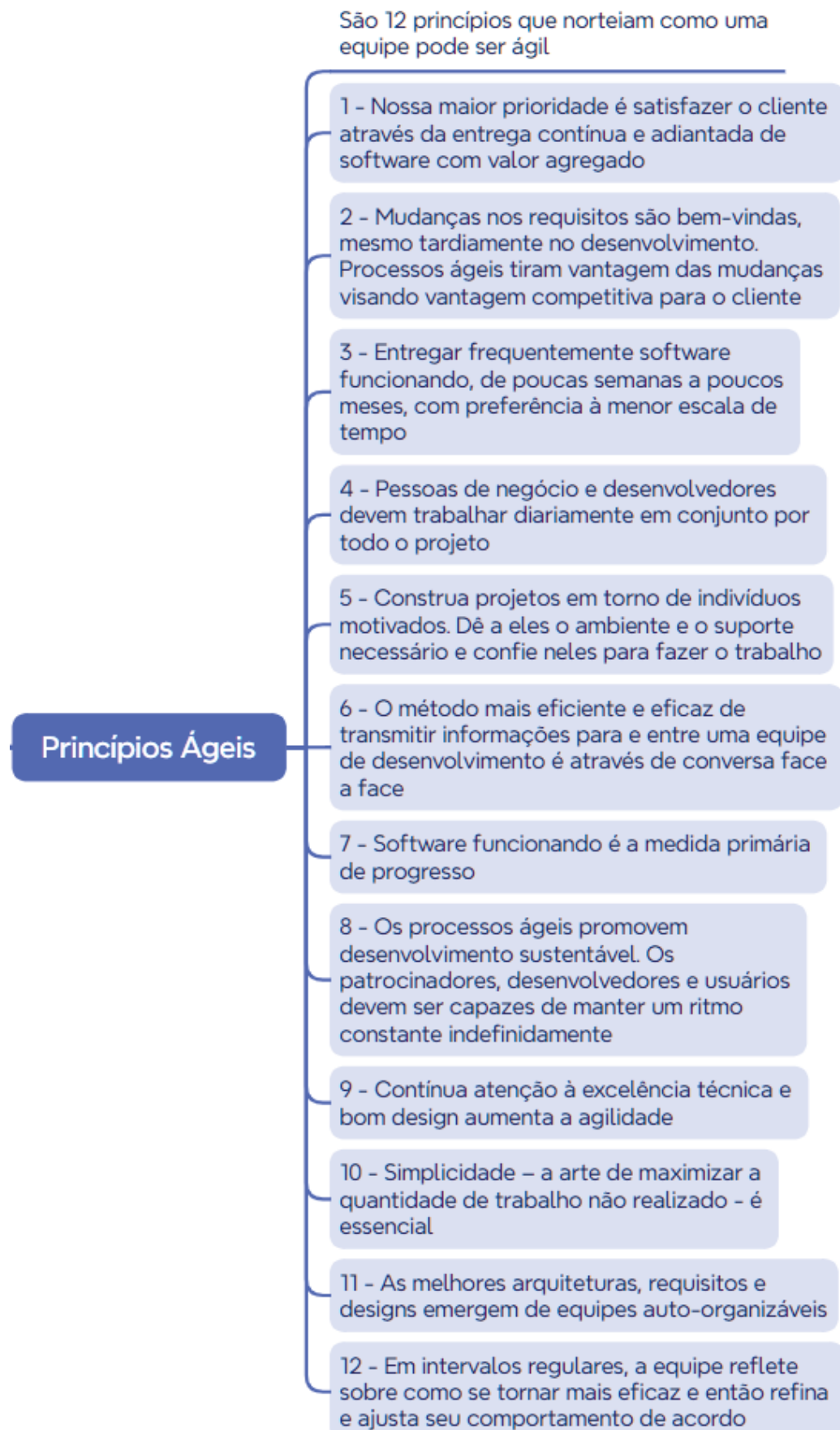
FONTE: O autor (2025)

FIGURA 3 – TÓPICO: MODELOS TRADICIONAIS



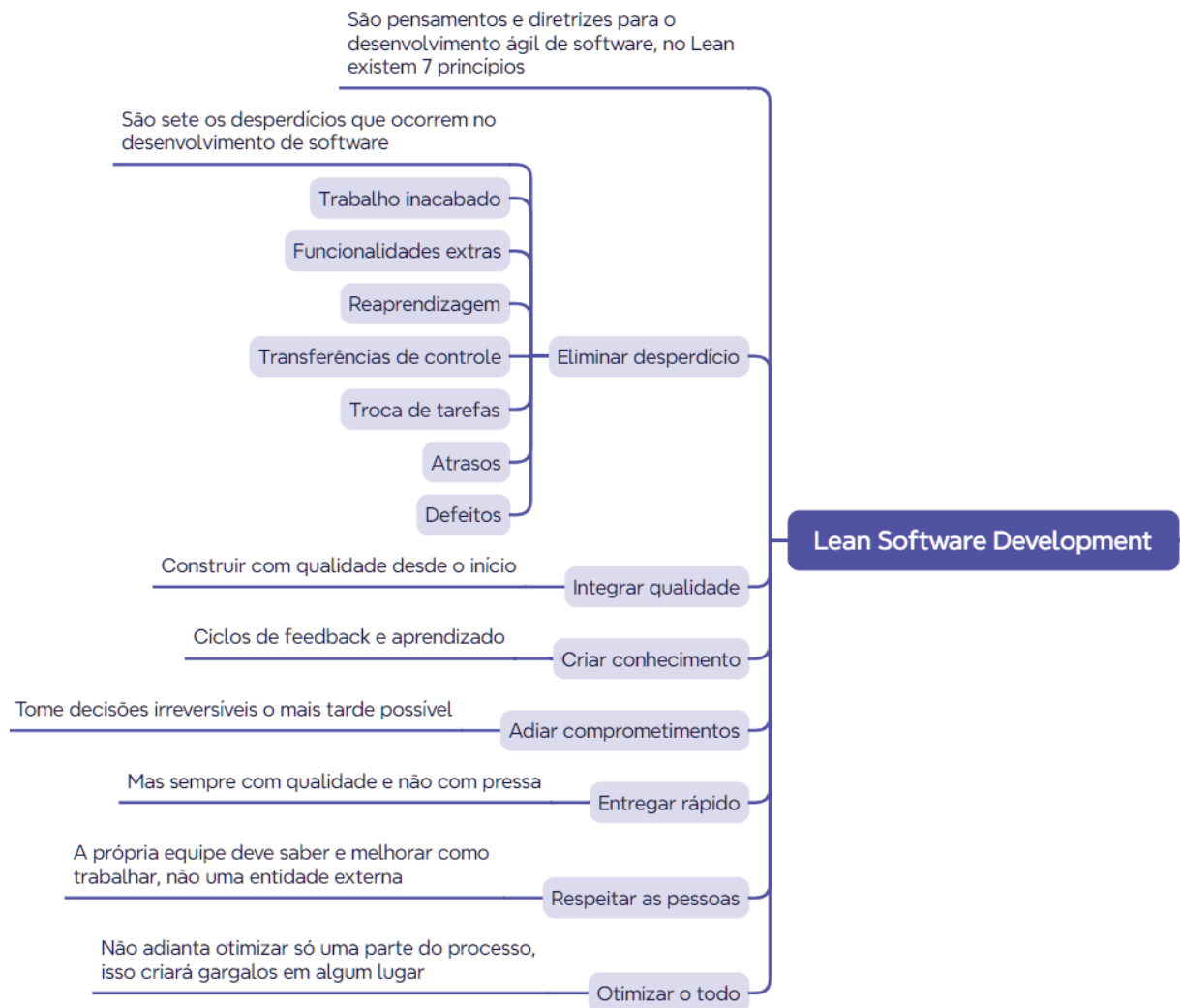
FONTE: O autor (2025)

FIGURA 4 – TÓPICO: PRINCÍPIOS ÁGEIS



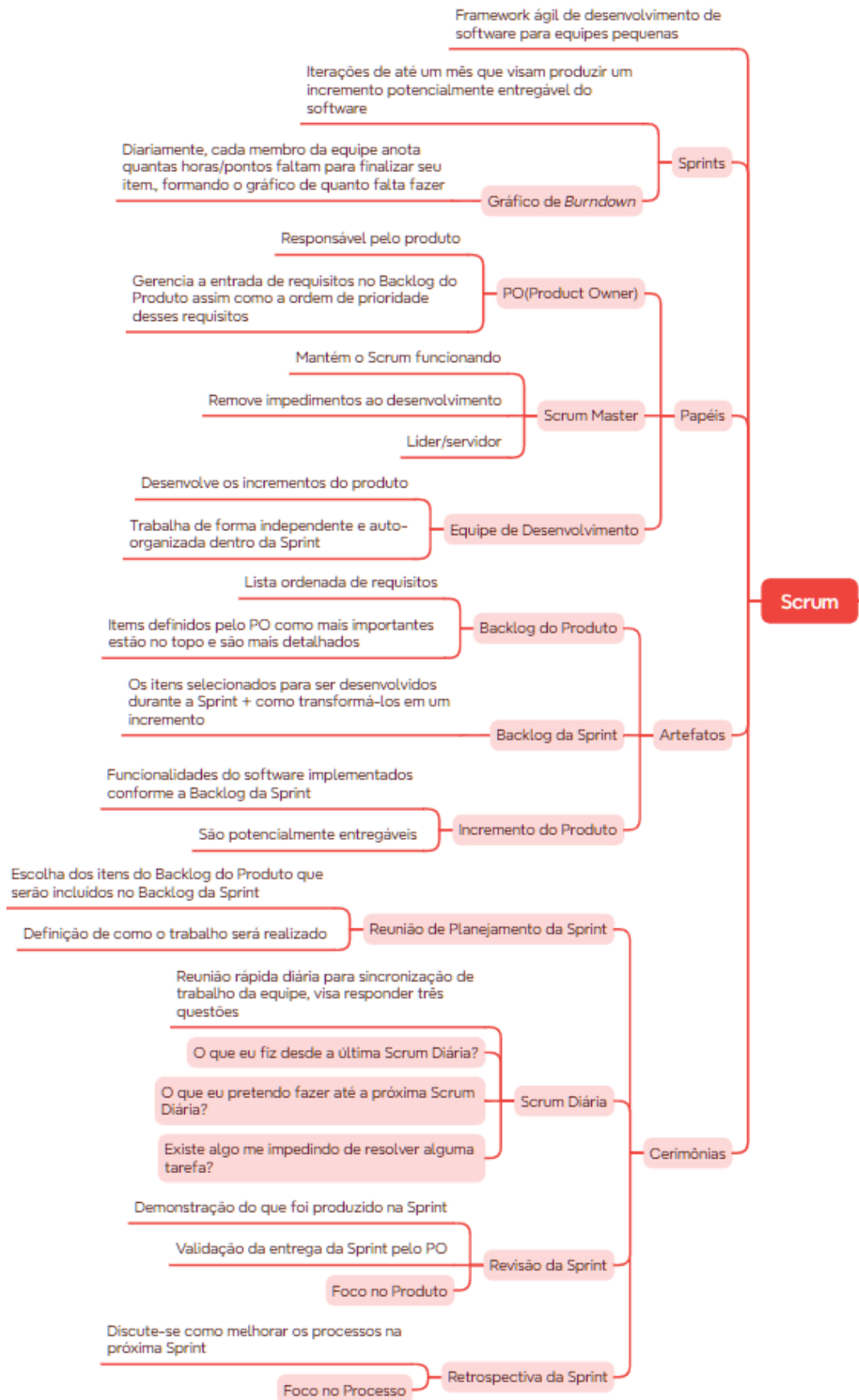
FONTE: O autor (2025)

FIGURA 5 – TÓPICO: LEAN



FONTE: O autor (2025)

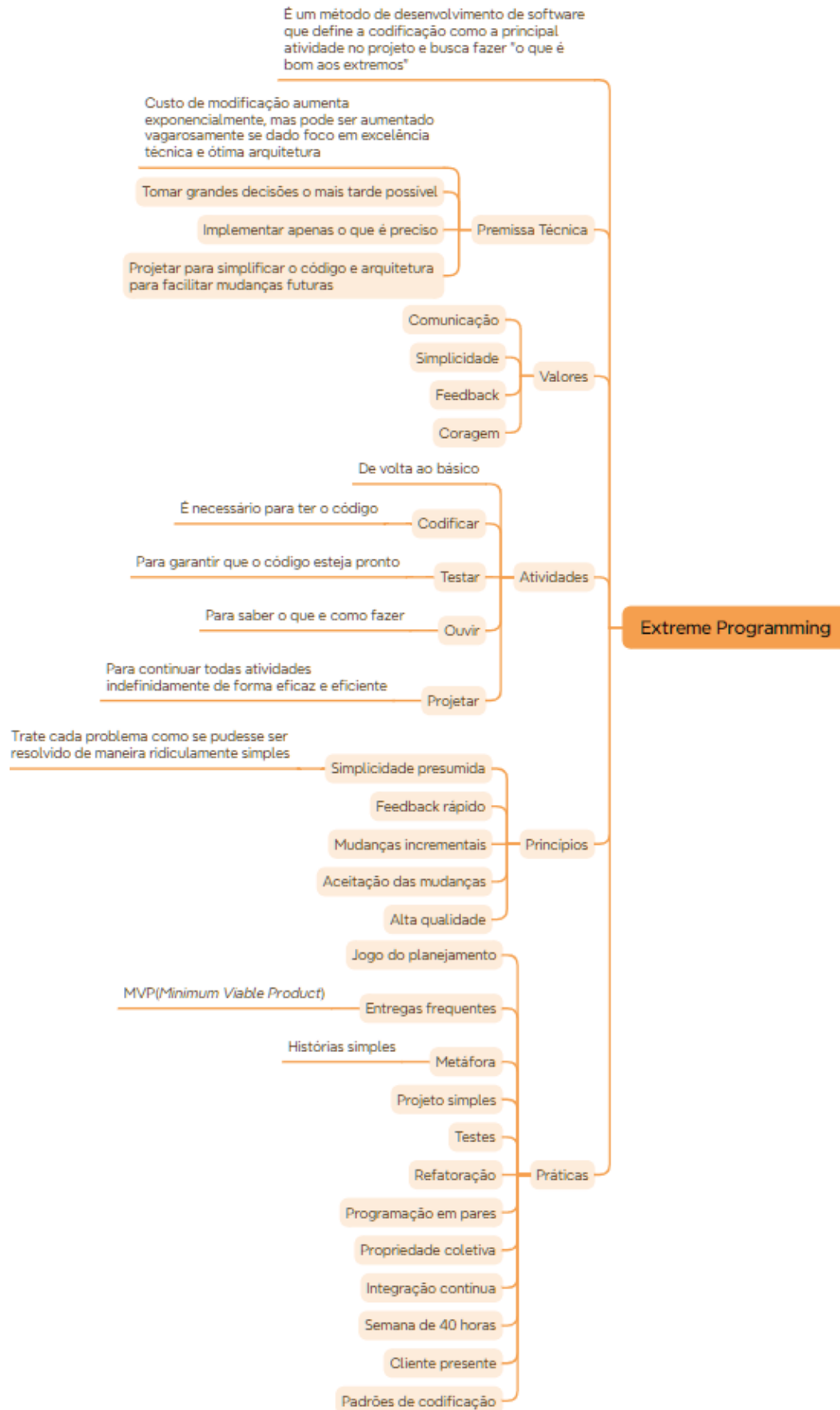
FIGURA 6 – TÓPICO: SCRUM



FONTE: O autor (2025)

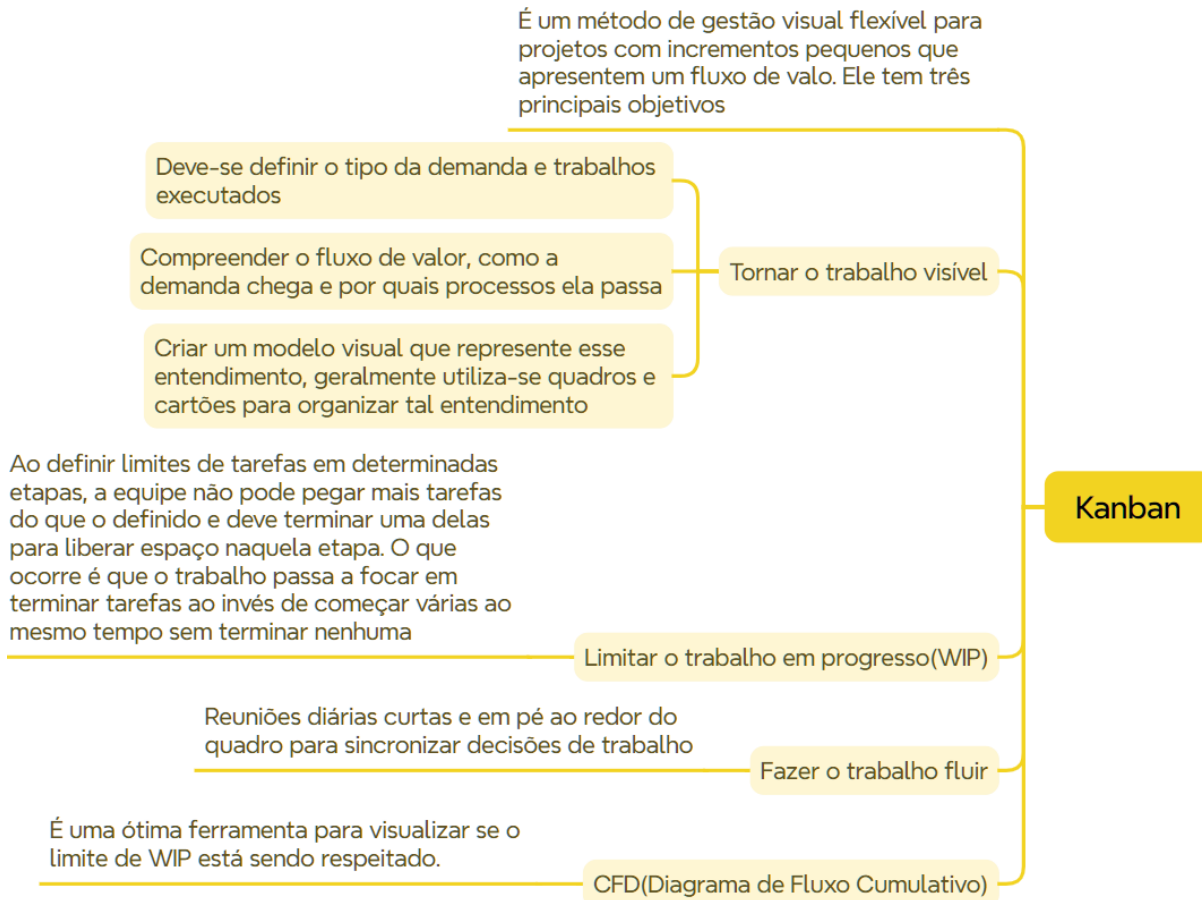


FIGURA 7 – TÓPICO: EXTREME PROGRAMMING



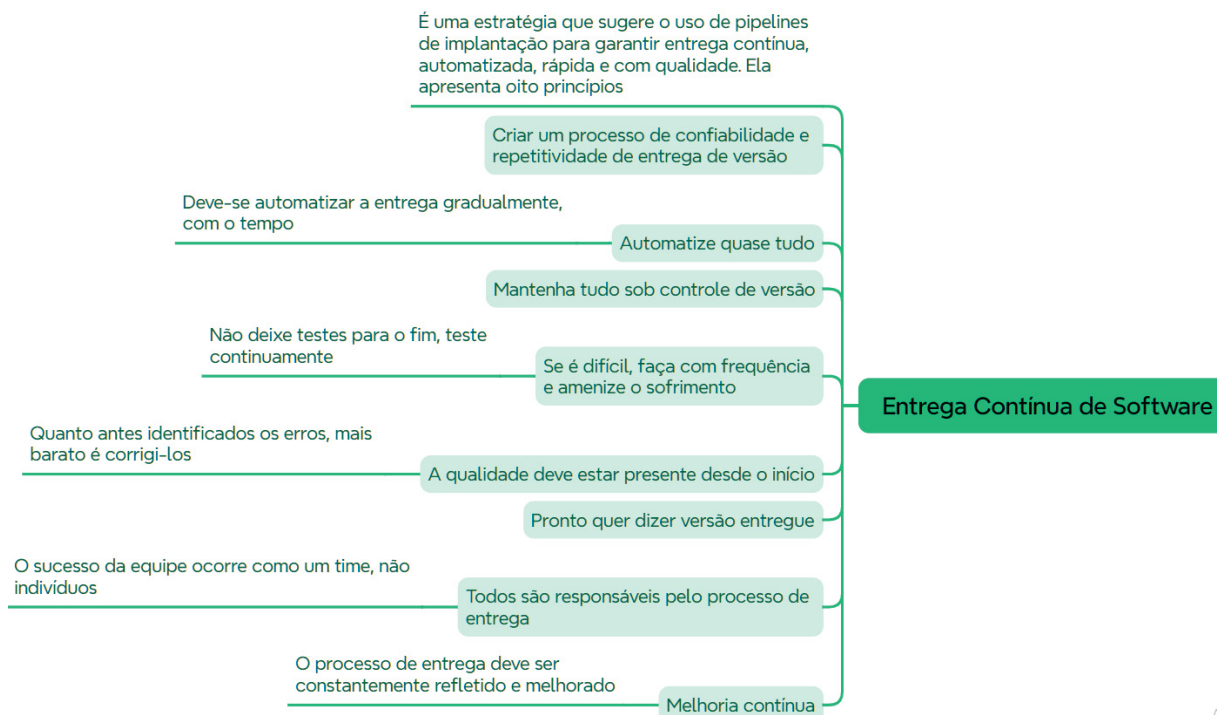
FONTE: O autor (2025)

FIGURA 8 – TÓPICO: KANBAN



FONTE: O autor (2025)

FIGURA 9 – TÓPICO: ENTREGA CONTÍNUA



FONTE: O autor (2025)

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

As disciplinas de Modelagem Ágil de Software 1 e 2 tiveram projetos finais incrementais, de modo a criar uma modelagem de um sistema de gestão de condomínio. Passando pela criação de diversos artefatos característicos do processo de modelagem de software, simulando o processo na prática.

Tais artefatos foram os diagramas de caso de uso de nível 1 e 2, diagramas de classes e histórias de usuário, esta última contendo desenhos de tela e diagramas de sequência (Rumbaugh *et al*, 2005).

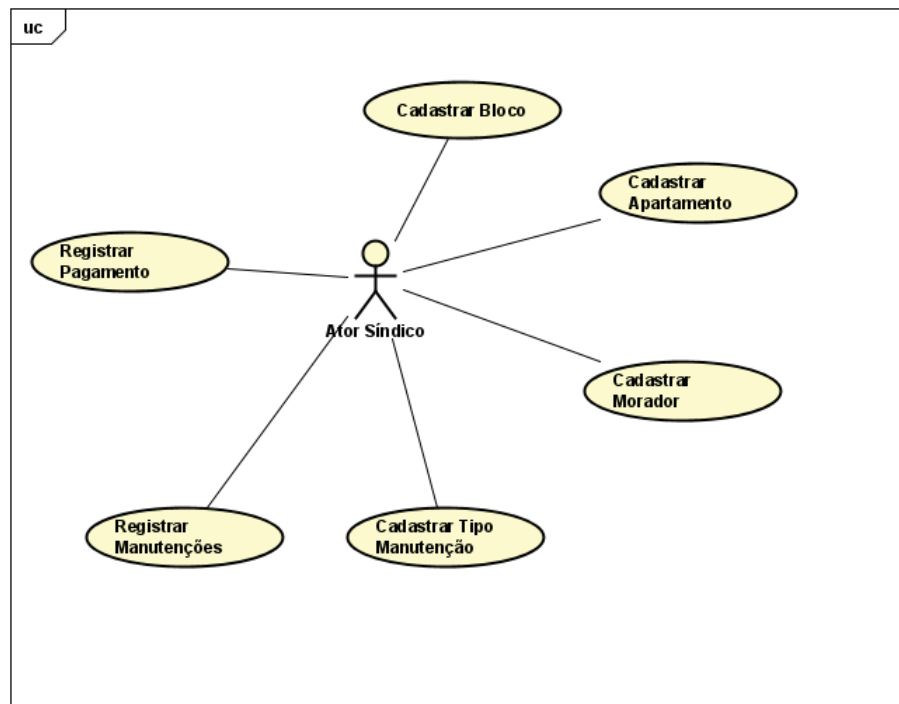
Este projeto foi muito importante, pois espelha o processo realizado na prática em desenvolvimento de software de mercado. Para a construção de um software robusto e bem planejado, de forma eficiente e reduzindo alterações posteriores, é fundamental a modelagem correta e bem feita.

Os diagramas UML organizam e estruturam o sistema de modo que enfatizam os diferentes componentes e o fluxo de interação entre eles de modo esclarecedor, mas ainda assim mantendo uma rigorosidade técnica.

Por outro lado as histórias de usuário são mais flexíveis e tem como objetivo facilitar a compreensão rápida dos requisitos por parte dos programadores. O uso de artefatos como desenho de telas, regras de negócio e critérios de aceitação permitem a implementação de requisitos de maneira muito ágil e mantendo a intenção do cliente, que é essencial no desenvolvimento ágil.

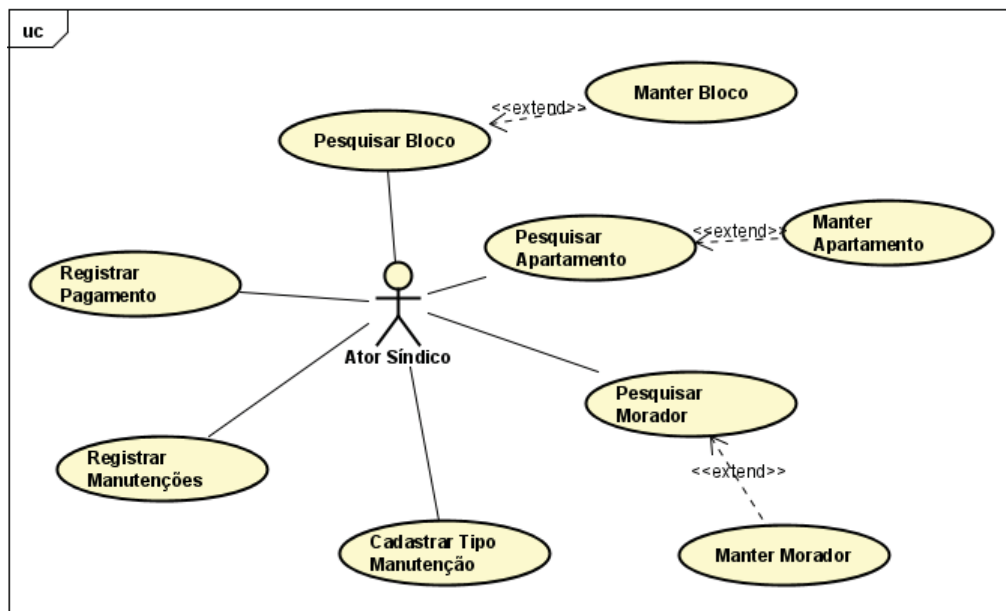
### 3.1 ARTEFATOS DO PROJETO

FIGURA 10 – DIAGRAMA DE CASOS DE USO (NÍVEL 1)



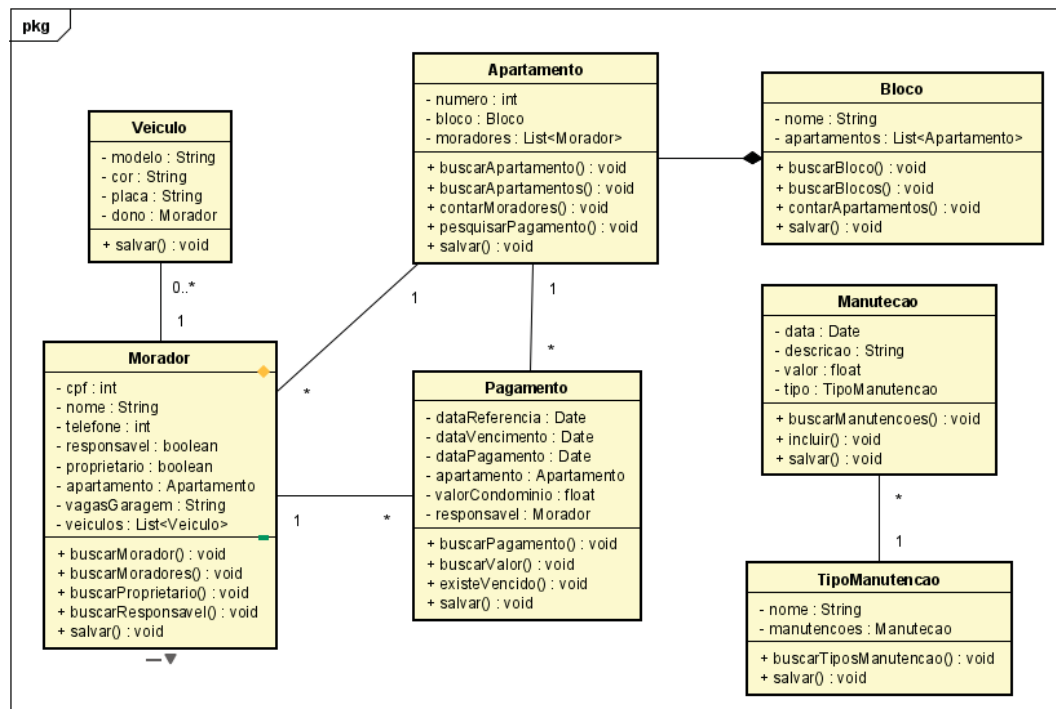
FONTE: O autor, em colaboração com von der Osten, Marcela (2025)

FIGURA 11 – DIAGRAMA DE CASOS DE USO (NÍVEL 2)



FONTE: O autor, em colaboração com von der Osten, Marcela (2025)

FIGURA 12 – DIAGRAMA DE CLASSES



FONTE: O autor, em colaboração com von der Osten, Marcela (2025)

## Histórias de Usuário

### HU001 - Pesquisar Bloco

**Sendo** síndico, **quero** pesquisar os blocos existentes **para** fazer manutenções nos dados do condomínio.

FIGURA 13 – DESENHO DA TELA



FONTE: O autor, em colaboração com von der Osten, Marcela (2025)

## LISTA DE CRITÉRIOS DE ACEITAÇÃO

1) Deve preencher a tabela com todos os blocos cadastrados do condomínio

- 2) Deve permitir incluir um novo bloco
- 3) Deve permitir alterar os dados de um bloco cadastrado
- 4) Deve permitir excluir um bloco cadastrado
- 5) Deve permitir consultar os dados de um bloco cadastrado
- 6) Deve pesquisar os bloco na tabela da tela
- 7) Deve voltar à tela anterior

### CRITÉRIOS DE ACEITAÇÃO - DETALHAMENTO

- 1) Deve preencher a tabela com todos os blocos cadastrados do condomínio

Dado que	
Quando	A tela é apresentada
Então	A tabela deve ser preenchida com todos os blocos do banco de dados

- 2) Deve permitir incluir um novo bloco

Dado que	Os blocos estão na tabela
Quando	O botão “Novo” é pressionado
Então	O sistema chama a HU002 – Manter Bloco passando o parâmetro “Novo”

- 3) Deve permitir alterar os dados de um bloco cadastrado

Dado que	Os blocos estão na tabela
Quando	O botão “Alterar” é pressionado em uma determinada linha
Então	O sistema chama a HU002 – Manter Bloco passando o parâmetro “Alterar” e os dados do bloco da linha

- 4) Deve permitir excluir um bloco

Dado que	Os blocos estão na tabela
Quando	O botão “Excluir” é pressionado em uma determinada linha
Então	O sistema pede uma confirmação, exclui o evento da base de dados e refaz a tabela da tela lendo novamente o banco de dados

- 5) Deve permitir consultar os dados de um bloco cadastrado

Dado que	Os blocos estão na tabela
Quando	O usuário clica em qualquer dado de um bloco em uma determinada linha
Então	O sistema chama a HU002 – Manter Bloco passando o parâmetro “Consultar” e os dados do bloco da linha

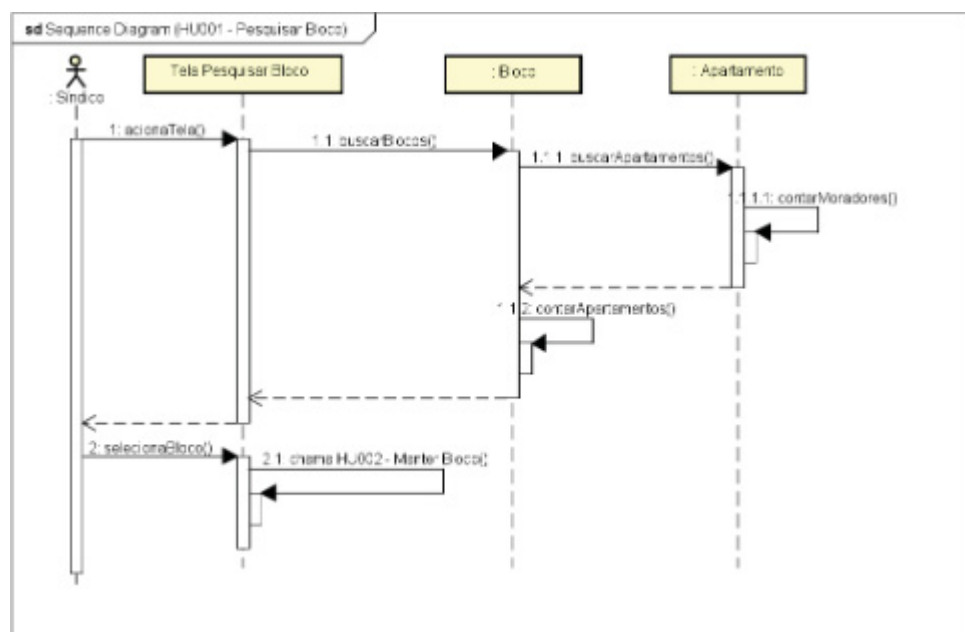
## 6) Deve pesquisar os blocos na tabela da tela

Dado que	Os blocos estão na tabela
Quando	For informado um argumento de pesquisa
Então	O sistema filtra e apresenta na tela somente os blocos que obedecem ao critério

## 7) Deve voltar à tela anterior

Dado que	Os blocos estão na tabela
Quando	O botão “Voltar” é pressionado
Então	O sistema volta para a tela anterior

FIGURA 14 – DIAGRAMA DE SEQUÊNCIA



FONTE: O autor, em colaboração com von der Osten, Marcela (2025)



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

O projeto final de Gerenciamento Ágil de Software 1 foi a criação de um plano de *release* para um software qualquer. Tendo escolhido o software, este deve ser separado em Histórias de Usuário, cada uma recebendo uma estimativa de tempo. Então montar um quadro de *release* contendo um cálculo da velocidade da equipe, as datas de cada *sprint* e quais Histórias serão feitas em cada *sprint* definida (Sutherland *et al*, 2004).

Esse plano de *release* é importante no planejamento de *sprints*, pois permite um balanceamento aproximado de tarefas ao longo do tempo, buscando manter um ritmo constante de trabalho e evitando procrastinação ou fadiga.

Em Gerenciamento Ágil de Software 2, o projeto final se resumiu em jogar o simulador de gestão de projetos Kanban Simulator (2025), usando Kanban (Coleman; Vacanti, 2025) e gerar o CFD ou Diagrama de Fluxo Cumulativo (Haidabrus *et al*, 2023) resultante da partida, demonstrando a eficiência e gargalos no fluxo nos 35 dias dentro do jogo.

Tal jogo é incrível em demonstrar de maneira lúdica e prática o funcionamento do Kanban, ajudando o aluno na compreensão da ferramenta, nas decisões tomadas para reduzir gargalos e como eventos imprevistos impactam um projeto.

Juntas essas disciplinas compõem aspectos teóricos e práticos da gestão ágil, calibrando o pensamento estratégico na tomada de decisões e enfatizando a importância de boas práticas e ferramentas para projetos ágeis eficientes e eficazes.

4.1 ARTEFATOS DO PROJETO

GAP1

FIGURA 15 – PLANO DE RELEASE

**Cálculo da Velocidade:**

Horas disponíveis por dia:	4h	Tamanho da Sprint:	2 semanas (10 dias)
Horas disponíveis por Sprint:	40h	Velocidade:	5 pontos

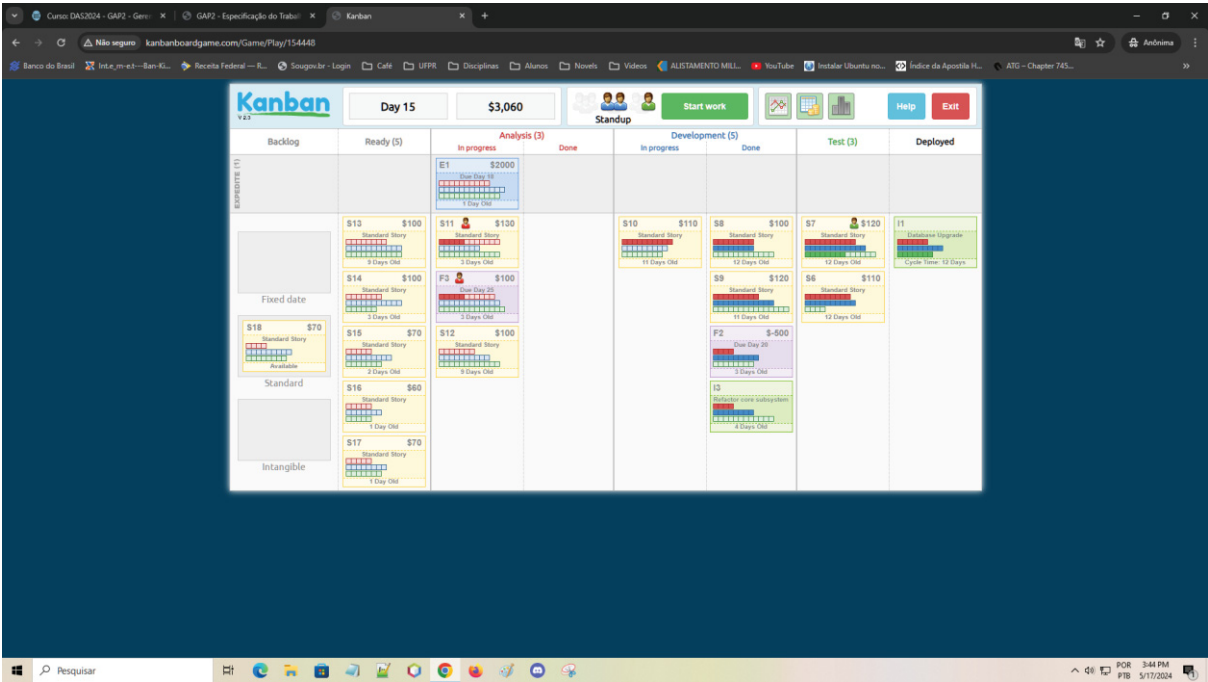
**Plano de Release: Fórum**

Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint N
Data Início:29/04 Data Fim:03/05	Data Início:06/05 Data Fim:10/05	Data Início:13/05 Data Fim:17/05	Data Início:20/05 Data Fim:24/05
<HU1-Cadastrar usuário> SEND O usuário QUERO me cadastrar PARA ter acesso ao fórum ESTIMATIVA (1)	<HU5-Ver tópico> SEND O usuário QUERO ver o tópico PARA ver os detalhes dele ESTIMATIVA (2)	<HU8-Editar Perfil> SEND O usuário QUERO editar meu perfil PARA personalizar meu perfil ESTIMATIVA (3)	<HU10-Conversar em Privado> SEND O usuário QUERO usar um chat com alguém PARA conversar sem estar em um tópico ESTIMATIVA (5)
<HU2-Login> SEND O usuário QUERO me logar no fórum PARA acessar meu perfil ESTIMATIVA (1)	<HU6-Comentar> SEND O usuário QUERO comentar em um tópico PARA interagir com o fórum ESTIMATIVA (2)	<HU9-Adicionar fotos> SEND O usuário QUERO adicionar fotos PARA usar imagens ESTIMATIVA (2)	
<HU3-Listar Tópicos> SEND O usuário QUERO ver todos os tópicos PARA escolher qual quero ver ESTIMATIVA (1)	<HU7-Like> SEND O usuário QUERO dar Like PARA deixar popular o tópico ESTIMATIVA (1)		
<HU4-Criar tópico> SEND O usuário QUERO criar um tópico PARA comentar e receber comentários ESTIMATIVA (2)			

FONTE: O autor (2025)

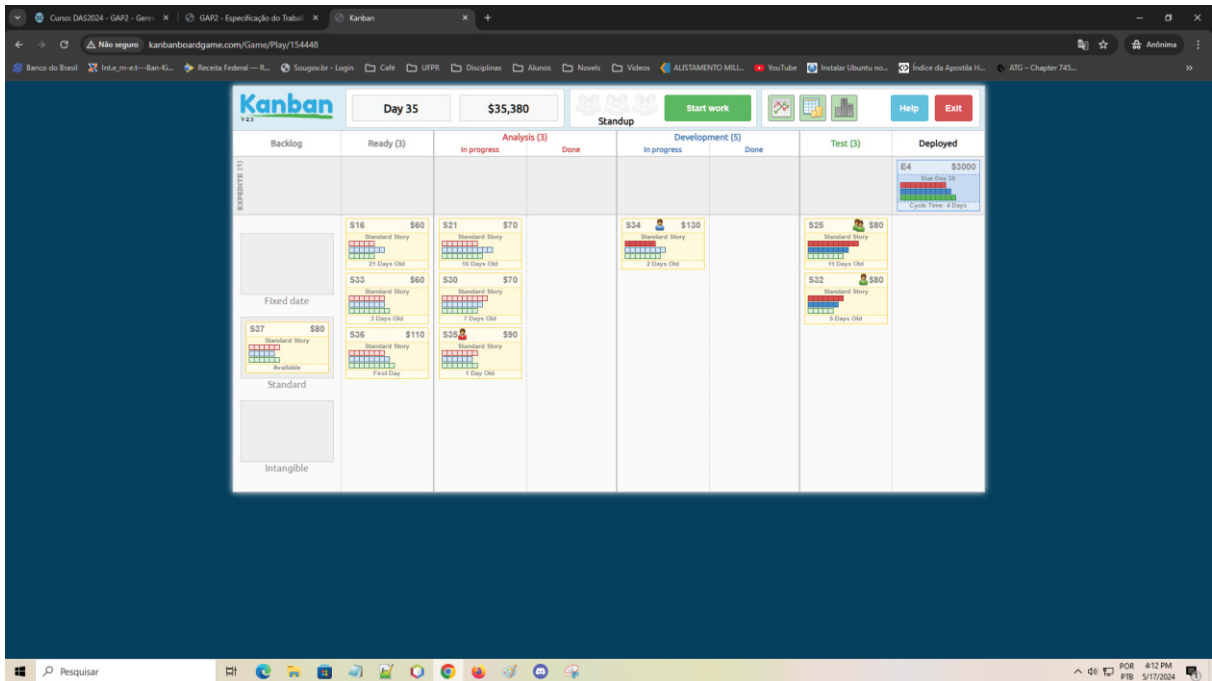
GAP2

FIGURA 16 – KANBAN DIA 15



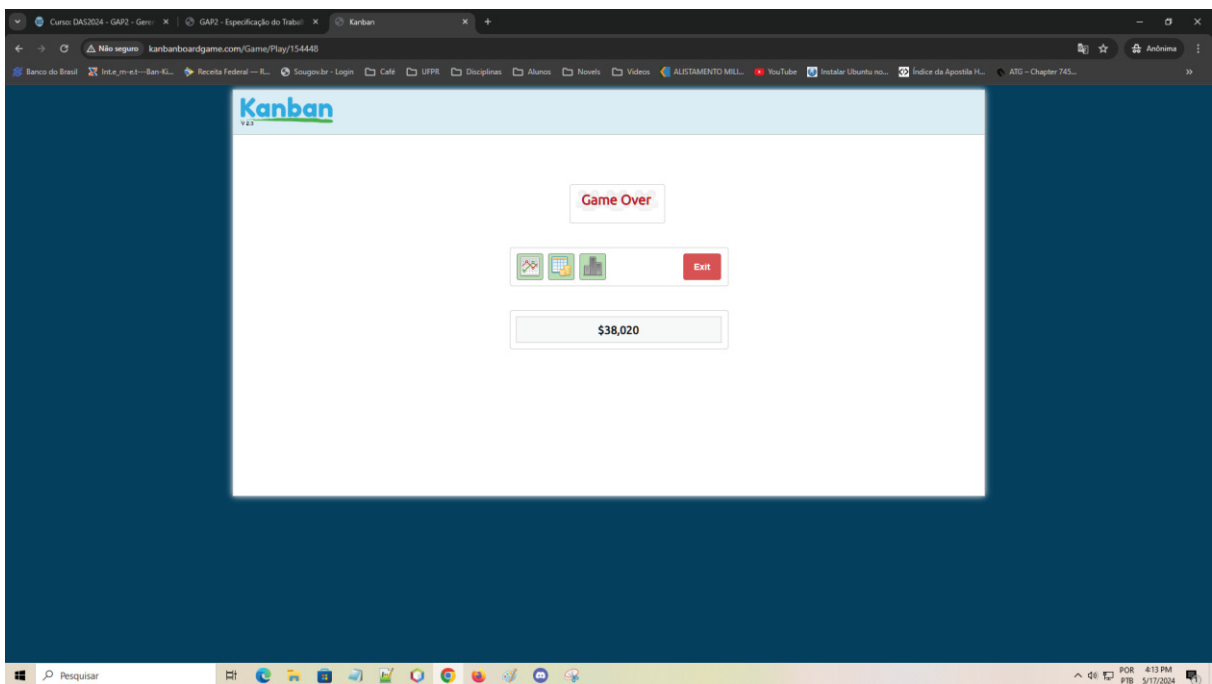
FONTE: O autor (2025)

FIGURA 17 – KANBAN DIA 35



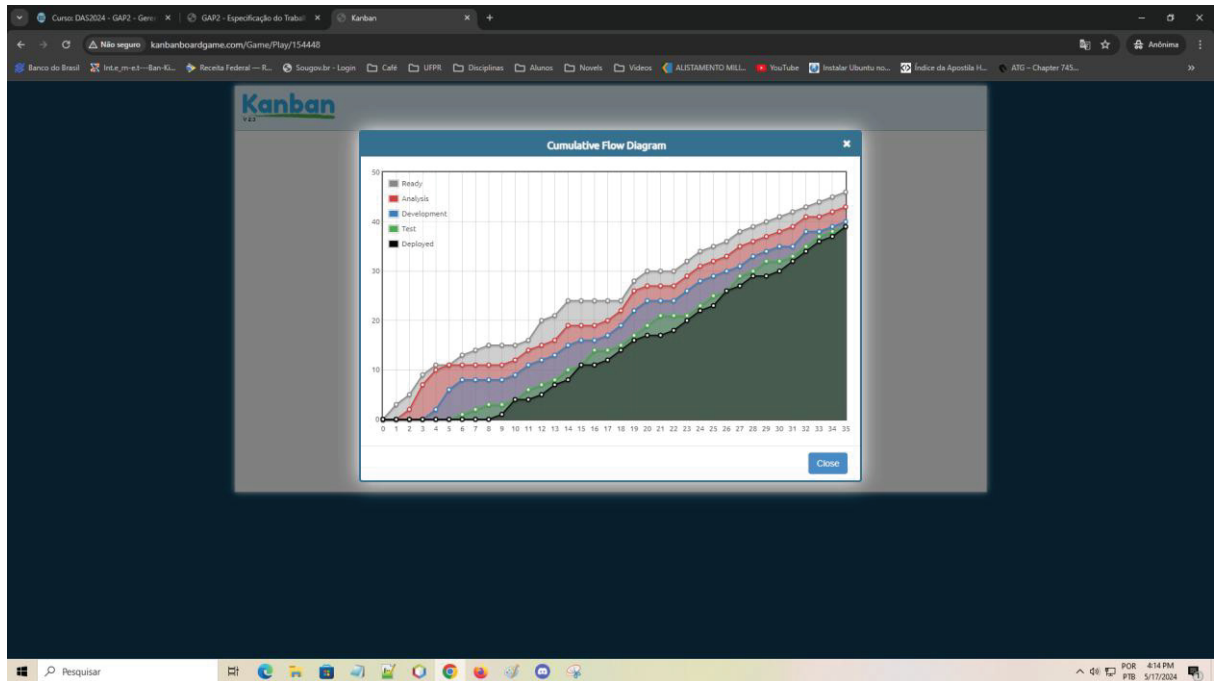
FONTE: O autor (2025)

FIGURA 18 – KANBAN FIM



FONTE: O autor (2025)

FIGURA 19 – KANBAN CFD



FONTE: O autor (2025)

## 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

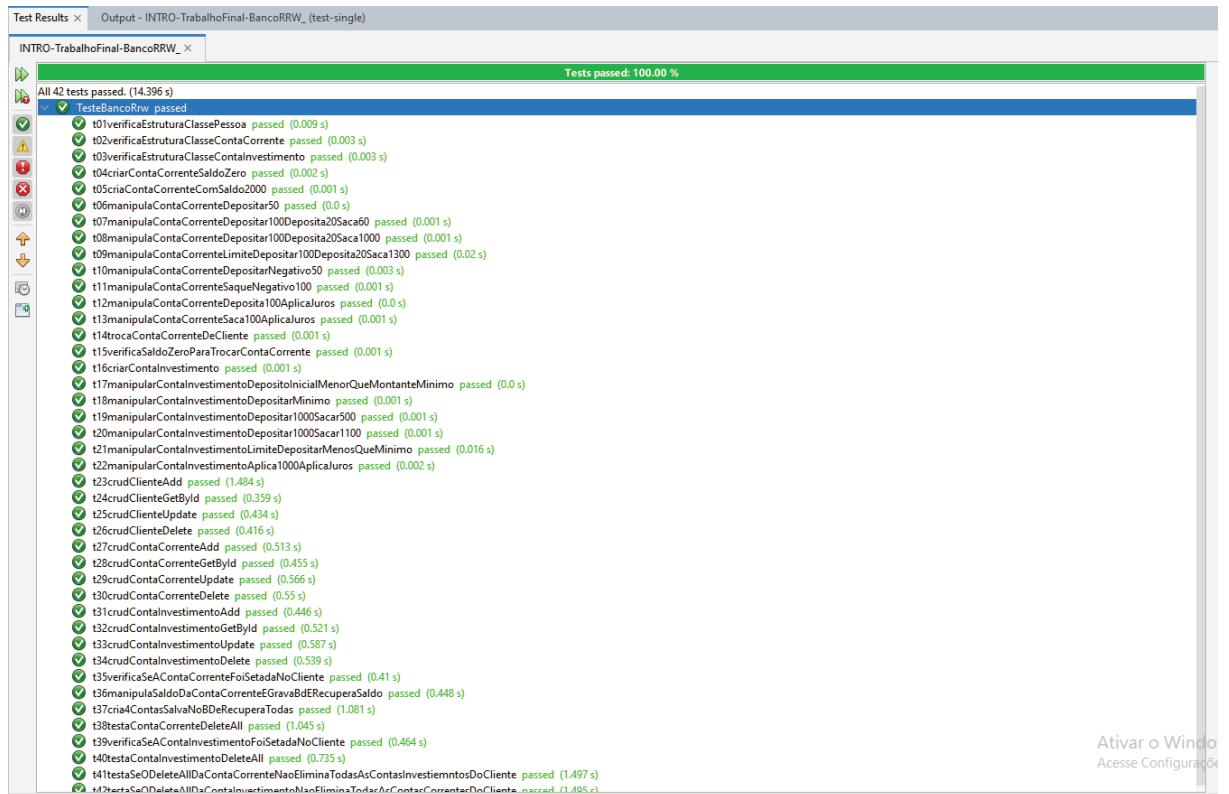
A disciplina de Introdução à Programação teve como projeto final a implementação de um sistema bancário simples em Java (2025) com o uso de *Test-Driven Development* (TDD) (Jeffries; Melnik, 2007). Este sistema permite gerenciar cadastros de clientes, conta-correntes e contas de investimentos. Foi fornecido o diagrama de classes, um *script* para criação de tabelas no MySQL (2025) e o projeto com classes vazias e os testes unitários que validam a corretude do funcionamento do projeto.

O uso dos 42 testes unitários guiou a programação, alertando o que faltava fazer e a corretude do que já havia sido feito. Com isso foi fácil progredir o projeto, com os testes pré-escritos reduzindo drasticamente a necessidade de tomada de decisões de alto nível durante a fase da programação, permitindo o foco total e sem distrações em apenas uma tarefa de cada vez.

Este projeto demonstrou a importância do uso de testes automatizados para agilidade, escalabilidade, robustez e eficácia do software desenvolvido. O uso de TDD também garante que mudanças e refatorações posteriores não alterem o funcionamento do produto final, tudo de maneira automatizada e reduzindo a falha humana.

## 5.1 ARTEFATOS DO PROJETO

FIGURA 20 – EVIDÊNCIA DE TESTES PASSADOS



FONTE: O autor (2025)

## 6 DISCIPLINA: BD – BANCO DE DADOS

O projeto final de Banco de Dados foi, baseado em um sistema de gerenciamento de biblioteca, a elaboração de um Modelo Entidade-Relacionamento (MER) Conceitual (Song; Froehlich, 1994), um Diagrama Entidade-Relacionamento (DER) no modelo Lógico (Song; Froehlich, 1994), contendo tabelas, chaves e associações, e então o *script* SQL (2025) para criação do banco de dados correspondente, assim como demonstrar o funcionamento do banco.

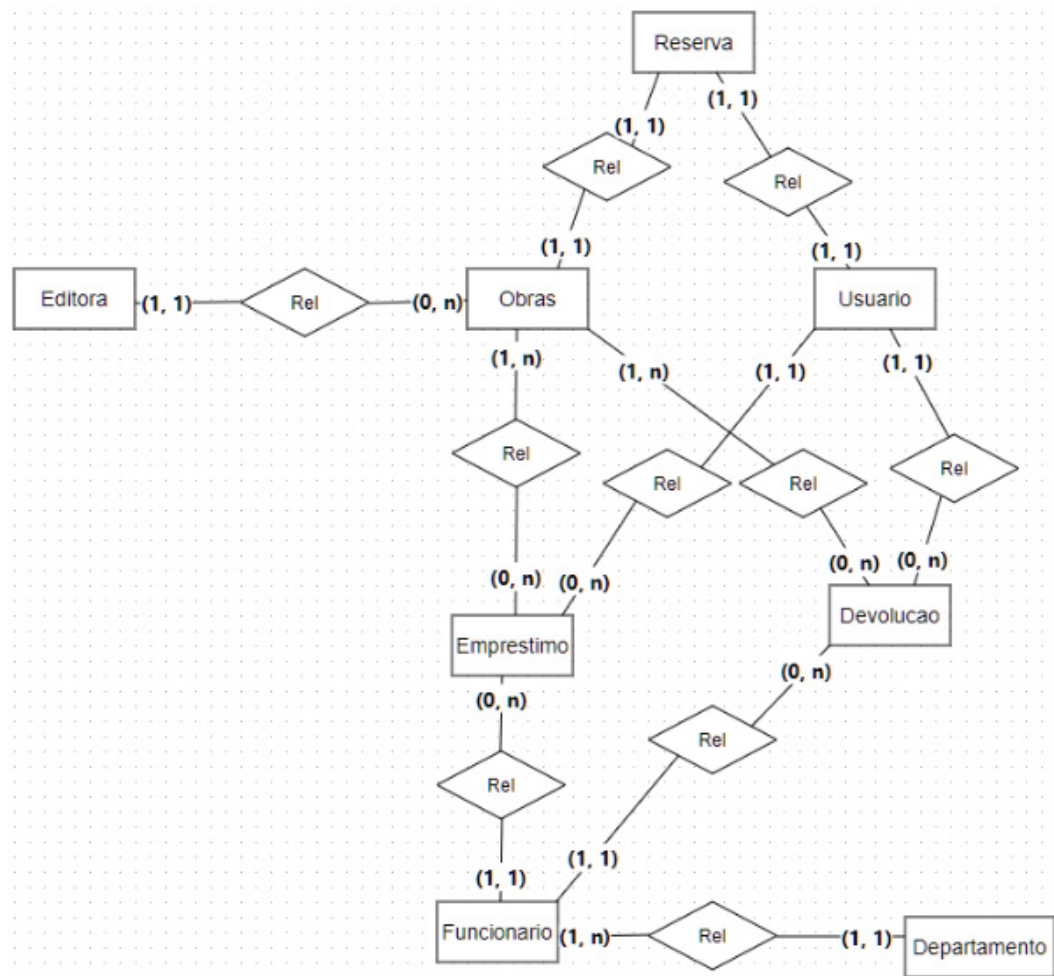
Um banco de dados bem construído e funcional é o que separa projetos simples de projetos com alto valor agregado. Só as informações acumuladas ao longo do tempo são uma mina de ouro para a análise de uso do produto e essencial na melhoria contínua, sem falar no quão fundamental o banco de dados é para o funcionamento de um sistema.

Este projeto demonstrou como erros de modelagem desencadeiam erros no banco de dados, que por sua vez causa falhas críticas ao sistema. Enquanto um banco de dados bem modelado serve como alicerce para todo o sistema, garantindo robustez e permanência dos dados.



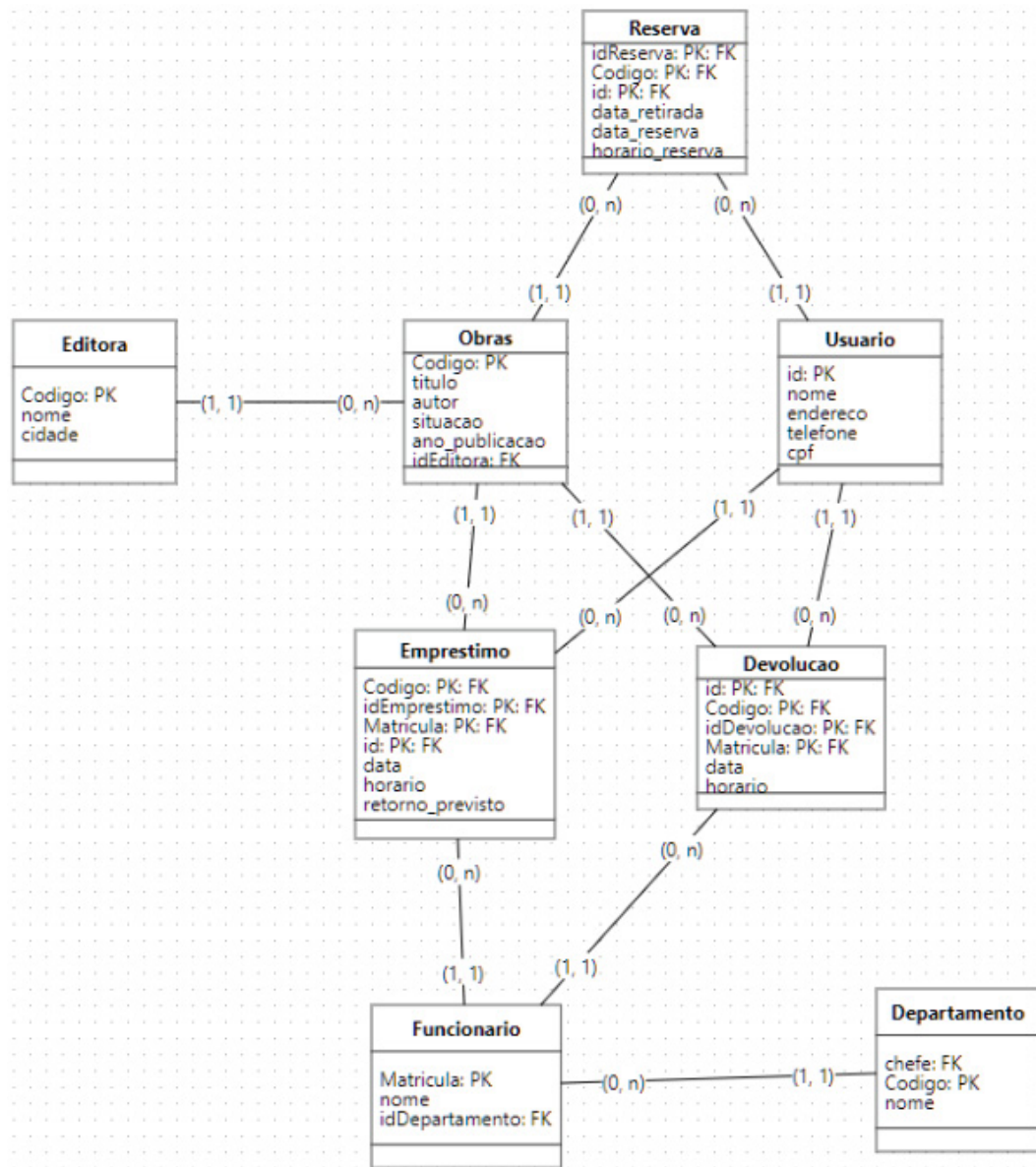
## 6.1 ARTEFATOS DO PROJETO

FIGURA 21 – MODELO CONCEITUAL



FONTE: O autor (2025)

FIGURA 22 – MODELO LÓGICO



FONTE: O autor (2025)

## SCRIPT SQL da Criação de Tabelas

```
CREATE TABLE Obras
(
  Codigo INT PRIMARY KEY,
  titulo INT,
  autor INT,
  situacao INT,
  ano_publicacao INT,
  idEditora INT,
);
```

```
CREATE TABLE Editora
(
```

```
Codigo INT PRIMARY KEY,
nome INT,
cidade INT,
);

CREATE TABLE Usuario
(
    id INT PRIMARY KEY,
    nome INT,
    endereco INT,
    telefone INT,
    cpf INT,
);

CREATE TABLE Funcionario
(
    Matricula INT PRIMARY KEY,
    nome INT,
    idDepartamento INT,
);

CREATE TABLE Departamento
(
    chefe INT NOT NULL,
    Codigo INT PRIMARY KEY,
    nome INT,
);

CREATE TABLE Reserva
(
    idReserva INT PRIMARY KEY,
    Codigo INT PRIMARY KEY,
    id INT PRIMARY KEY,
    data_retirada DATE NOT NULL,
    data_reserva DATE NOT NULL,
    horario_reserva INT NOT NULL,
);

CREATE TABLE Emprestimo
(
    Codigo INT PRIMARY KEY,
    idEmprestimo INT PRIMARY KEY,
    Matricula INT PRIMARY KEY,
    id INT PRIMARY KEY,
    data DATE NOT NULL,
    horario INT NOT NULL,
    retorno_previsto INT NOT NULL,
);

CREATE TABLE Devolucao
(
    id INT PRIMARY KEY,
    Codigo INT PRIMARY KEY,
    idDevolucao INT PRIMARY KEY,
    Matricula INT PRIMARY KEY,
```

```
data DATE NOT NULL,  
horario INT NOT NULL,  
);
```

```
ALTER TABLE Obras ADD FOREIGN KEY(idEditora) REFERENCES Editora  
(idEditora)  
ALTER TABLE Funcionario ADD FOREIGN KEY(idDepartamento) REFERENCES  
Departamento (idDepartamento)  
ALTER TABLE Departamento ADD FOREIGN KEY(chefe) REFERENCES  
Funcionario (chefe)  
ALTER TABLE Reserva ADD FOREIGN KEY(Codigo) REFERENCES Obras  
(Codigo)  
ALTER TABLE Reserva ADD FOREIGN KEY(id) REFERENCES Usuario (id)  
ALTER TABLE Emprestimo ADD FOREIGN KEY(Codigo) REFERENCES Obras  
(Codigo)  
ALTER TABLE Emprestimo ADD FOREIGN KEY(Matricula) REFERENCES  
Funcionario (Matricula)  
ALTER TABLE Emprestimo ADD FOREIGN KEY(id) REFERENCES Usuario (id)  
ALTER TABLE Devolucao ADD FOREIGN KEY(id) REFERENCES Usuario (id)  
ALTER TABLE Devolucao ADD FOREIGN KEY(Codigo) REFERENCES Obras  
(Codigo)  
ALTER TABLE Devolucao ADD FOREIGN KEY(Matricula) REFERENCES  
Funcionario (Matricula)
```

## 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

Na disciplina de Aspectos ágeis de Programação, o projeto final foi a refatoração do algoritmo de ordenação *Bubble Sort* (Min, 2010) utilizando princípios de *Clean Code* (Martin; *et al*, 2008).

O *Clean Code* é um conjunto de princípios e boas práticas que tem como objetivo deixar o código mais legível, modular e fácil de manter. O código limpo ajuda a diminuir erros e melhorar o entendimento do código entre diferentes membros da equipe, possibilitando um trabalho mais eficiente e produtivo.

O uso de boas práticas de escrita de software é fundamental para a compreensão, manutenabilidade e reutilização do código. Num ambiente ágil, onde mudanças ocorrem a todo o momento, tais características são muito prezadas, o que reforça a importância do *Clean Code* na programação ágil.

## 7.1 ARTEFATOS DO PROJETO

### ALGORITMO REFATORADO

```
// Implementação otimizada em Java do Bubble sort
// Código extraído de https://www.geeksforgeeks.org/bubble-sort/

import java.io.*;

class BubbleSort {
    // Implementação clean code em Java do Bubble sort
    // Código extraído de https://www.geeksforgeeks.org/bubble-sort/

    import java.io.*;

    class BubbleSort {
        //Modificacoes feitas conforme explicado na aula
        public static void main(String args[]) {
            int numeros[] = { 64, 34, 25, 12, 22, 90, 11 };
            bubbleSort(numeros);
            System.out.println("Array ordenado: ");
            imprimirNumeros(numeros);
        }

        static void bubbleSort(int numeros[]) {
            int tamanho = numeros.length;
            boolean foiTrocado;

            for (int i = 0; i < tamanho -1; i++) {
                foiTrocado = false;

                for (int j = 0; j < (tamanho - 1) - i; j++) foiTrocado |=
precisaTrocar(numeros, j);

                if (!foiTrocado) break;
            }
        }

        static boolean precisaTrocar(int numeros[], int j) {
            if (numeros[j] > numeros[j + 1]) {
                trocarComProximo(numeros, j);
                return true;
            }
        }
    }
}
```

```
    }  
    return false;  
}  
  
static void trocarComProximo(int numeros[], int j) {  
    int temp = numeros[j];  
    numeros[j] = numeros[j + 1];  
    numeros[j + 1] = temp;  
}  
  
static void imprimirNumeros(int numeros[]) {  
    for(int numero : numeros)  
        System.out.print(numero + " ");  
    System.out.println();  
}  
}  
  
// This code is contributed  
// by Nikita Tiwari.
```



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

Na disciplina WEB1, o projeto final consistia na criação de CRUDs (interfaces contendo as operações *Create*, *Read*, *Update* e *Delete*) para as entidades Aluno e Curso por meio do *framework* Angular (2025), incluindo interface, mas com persistência apenas em *Local Storage*.

Foi reforçada a importância de *frameworks* no desenvolvimento, automatizando processos burocráticos e tediosos comuns em todos os projetos de forma rápida e robusta, com apenas poucas linhas de código.

Já na disciplina WEB2 foi incrementado o projeto de WEB1 com a entidade Matrícula, incluindo um CRUD para tal e adicionando *backend* em Spring Boot (2025) e persistência em banco de dados PostgreSQL (2025). Este projeto realçou o quão rápido é a transição de um projeto local para um mais complexo, robusto e com persistência através das ferramentas adequadas e um processo metódico bem feito.

Esta disciplina mostrou como é possível a criação de aplicações web prontas para o mercado de forma rápida e prática, com ferramentas criando uma ponte entre ideia e prática, facilitando muitas coisas no caminho.

### 8.1 ARTEFATOS DO PROJETO

#### Script Angular inicial

```
ng new CRUD-AlunoCurso

cd ./CRUD-AlunoCurso
npm install --save bootstrap
npm install --save @fortawesome/fontawesome-free

ng g module Aluno
ng g module Curso
ng g module Matricula

ng g class shared/models/aluno --type=model
ng g class shared/models/curso --type=model
ng g class shared/models/matricula --type=model

ng g service aluno/services/aluno
ng g service curso/services/curso
ng g service matricula/services/matricula
```

```
ng g component aluno/listar-aluno
ng g component aluno/inserir-aluno
ng g component aluno/editar-aluno

ng g component curso/listar-curso
ng g component curso/inserir-curso
ng g component curso/editar-curso

ng g component matricula/listar-matricula
ng g component matricula/inserir-matricula
ng g component matricula/editar-matricula

ng g module Directives
ng g directive shared/directives/numerico

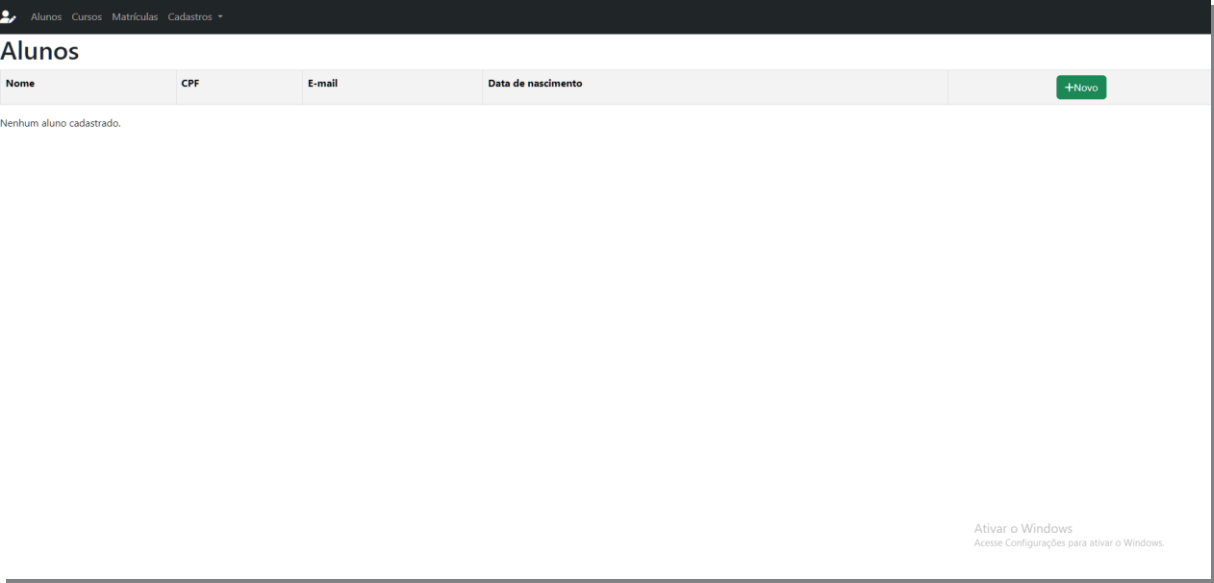
npm install --save ngx-mask
ng g pipe shared/pipes/caixa-alta

ng add @ng-bootstrap/ng-bootstrap
ng g component aluno/modal-aluno

npm install --save @ng-select/ng-select

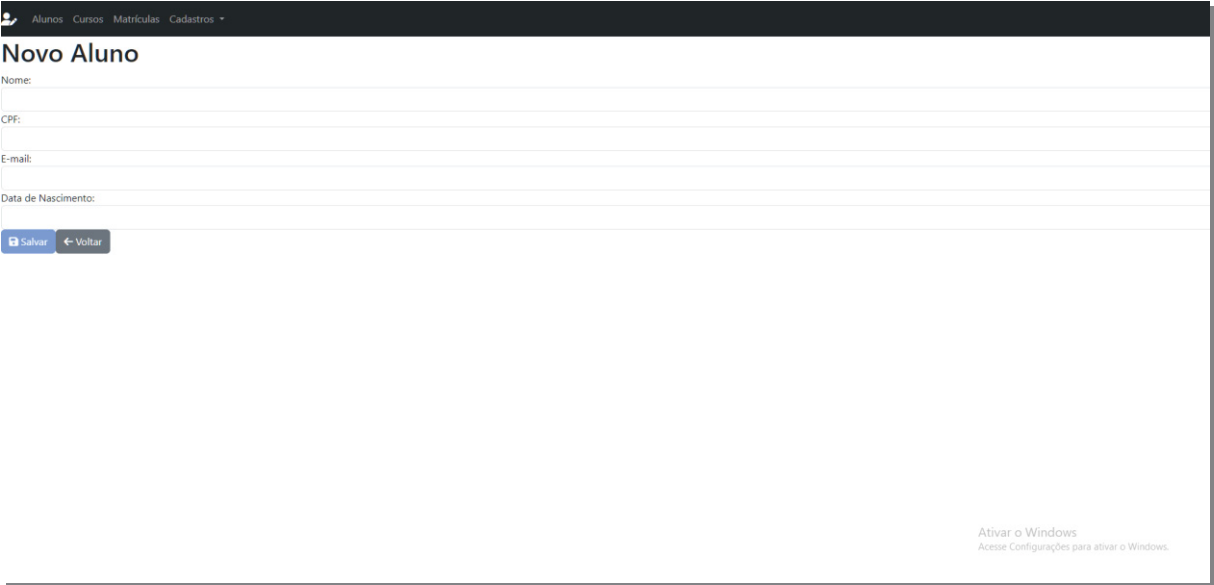
npm install rxjs -save
```

FIGURA 23 – TELA DE LISTA DE ALUNOS



FONTE: O autor (2025)

FIGURA 24 – TELA DE INSERÇÃO DE ALUNO



FONTE: O autor (2025)

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

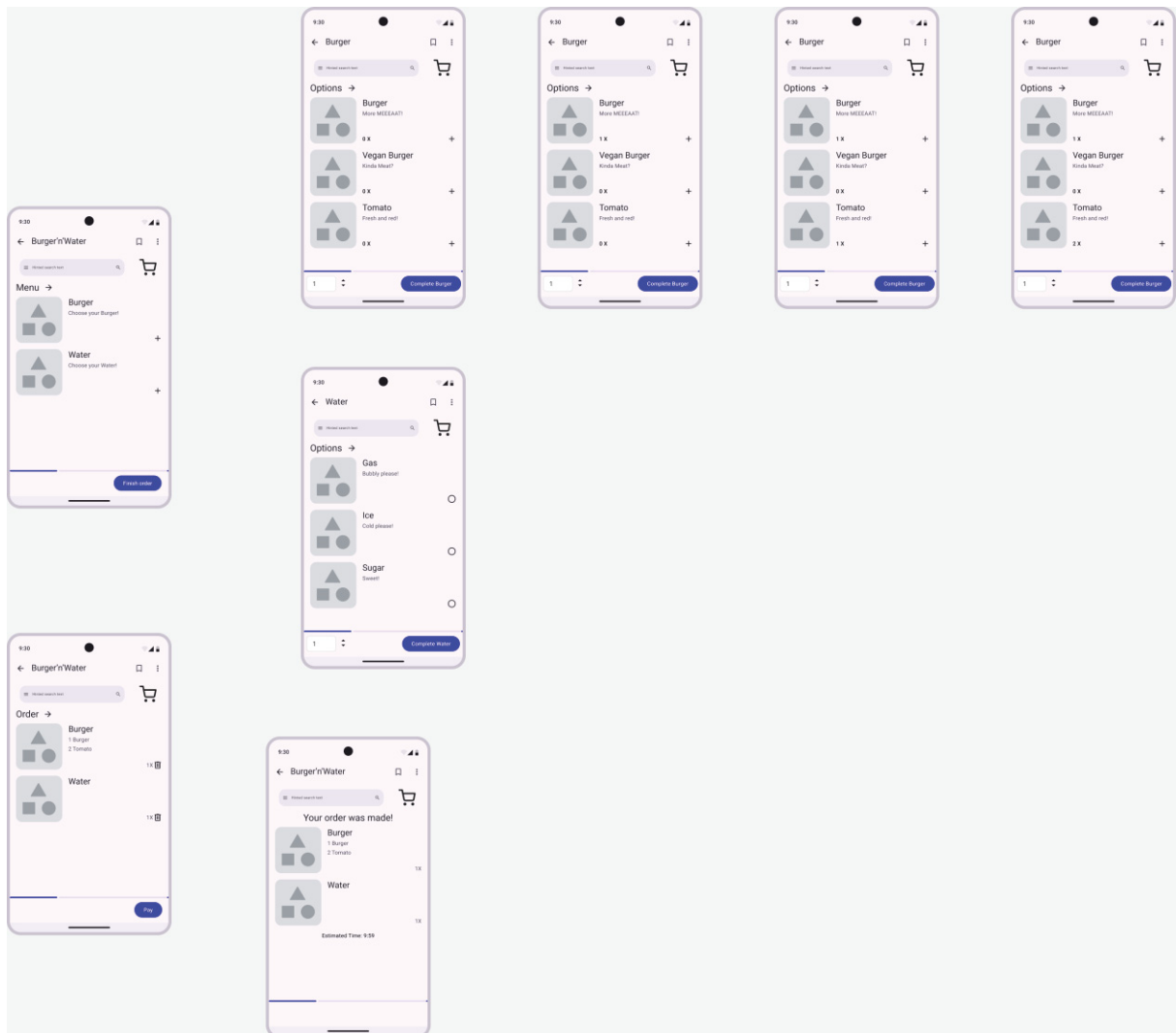
A disciplina de UX no Desenvolvimento Ágil de Software teve como projeto final a elaboração de telas de um aplicativo móvel hipotético com o uso de ferramentas de prototipagem. Os principais fatores nessas telas sendo interface intuitiva, com botões e fluxo de telas compreensível e estética concordante com boas práticas de UX (Teixeira, 2014).

O uso de cores contrastantes ou complementares, letras de tamanho adequado, símbolos reconhecíveis e intuitivos em botões, assim como aversão ao uso de cores verdes e vermelhas para pessoas daltônicas estão entre diversos fatores que impactam muito a experiência do usuário e subsequentemente o sucesso do produto.

Esta disciplina mostra a importância de UX no mercado. Mesmo fatores pequenos e aparentemente sutis podem ser o suficiente para um usuário preferir outro concorrente, às vezes nem mesmo conscientemente. O uso de ferramentas de prototipagem ajudam no *feedback* rápido do cliente e testadores, evitando retrabalho e acelerando o processo de refinamento do produto.

## 9.1 ARTEFATOS DO PROJETO

FIGURA 25 – TELAS



FONTE: O autor (2025)

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

As matérias de Desenvolvimento *Mobile* 1 e 2 tiveram projetos finais a criação de aplicações para dispositivos móveis com o uso da IDE Android Studio (2025).

Em MOB1, a aplicação deveria permitir cadastrar receitas e despesas para auxiliar no controle das finanças. Já em MOB2 a aplicação deveria consumir a API de Harry Potter HP-API (2025), aprofundando o aprendizado de *web services*. Tais projetos visavam colocar em prática o conhecimento visto em aula e familiarizar o aluno com a ferramenta.

Por ser opcional, nenhum dos dois trabalhos foram executados. Apesar disso pode-se ter uma grande apreciação pela ferramenta e pelas demonstrações em aula.

*Mobile* é um meio muito utilizado atualmente como plataforma de alcance mundial. O público-alvo pode abranger todos que possuam um celular com os devidos pré-requisitos, otimizando o sucesso de aplicações ágeis que são criadas e atingem retornos rapidamente.

Pela ampla concorrência no contexto *mobile*, diversos fatores como UX e *feedback* rápido são diferenciais importantes no mercado, necessitando reunir todas as disciplinas anteriores para gerar um produto digno de sucesso em tempo hábil.

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

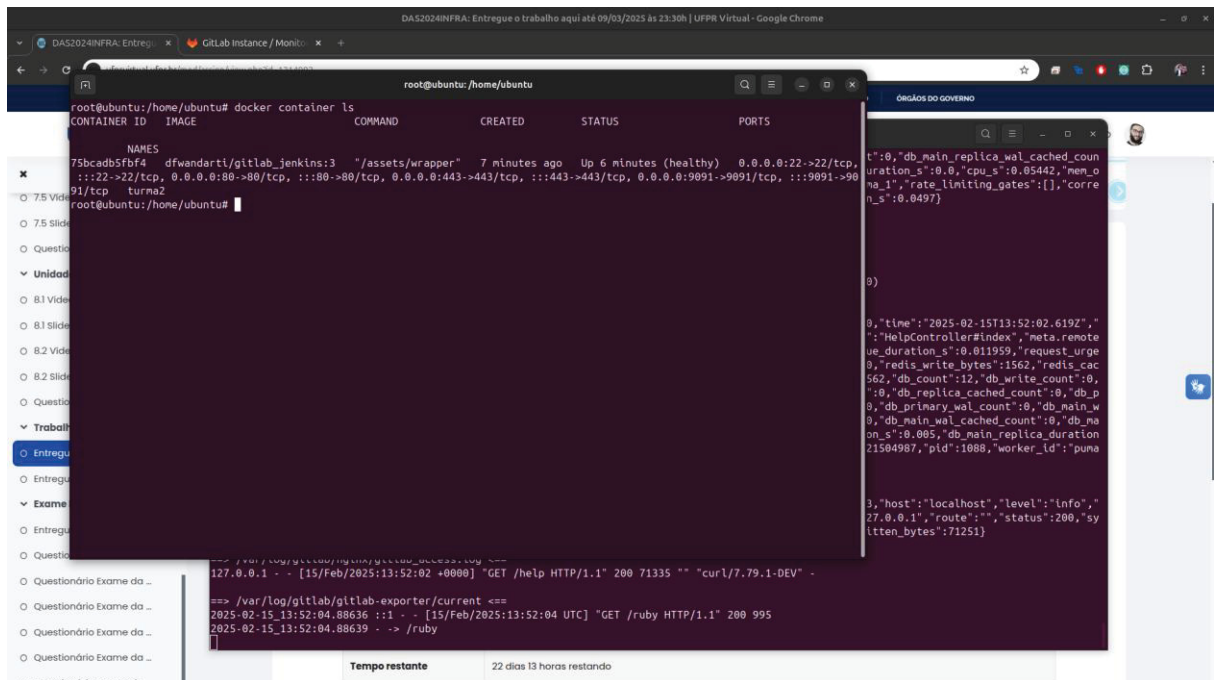
A disciplina de Infraestrutura para Desenvolvimento e Implantação de Software (DevOps) teve como projeto final encontrar a senha para o acesso ao GitLab (2025). Para achar tal senha, foi necessário criar um container a partir da imagem `dfwandarti/gitlab_jenkins:3`, nomeando-o com o nome do grupo e publicando as portas 22, 80, 443 e 9031, e então navegar dentro do container até encontrar a senha.

Tendo encontrado a senha, foi preciso criar um README contendo os nomes dos alunos do grupo e dar um commit de tal para o GitLab.

DevOps é fundamental no desenvolvimento ágil, pois é o que une desenvolvimento contínuo com automação de operações (Ebert *et al*, 2016). Um DevOps bom é o que permite a entrega contínua de código sem desperdício de tempo em processos repetitivos que podem ser automatizados. O DevOps pode também abranger operações como testes automatizados, assegurando a correteza do código produzido.

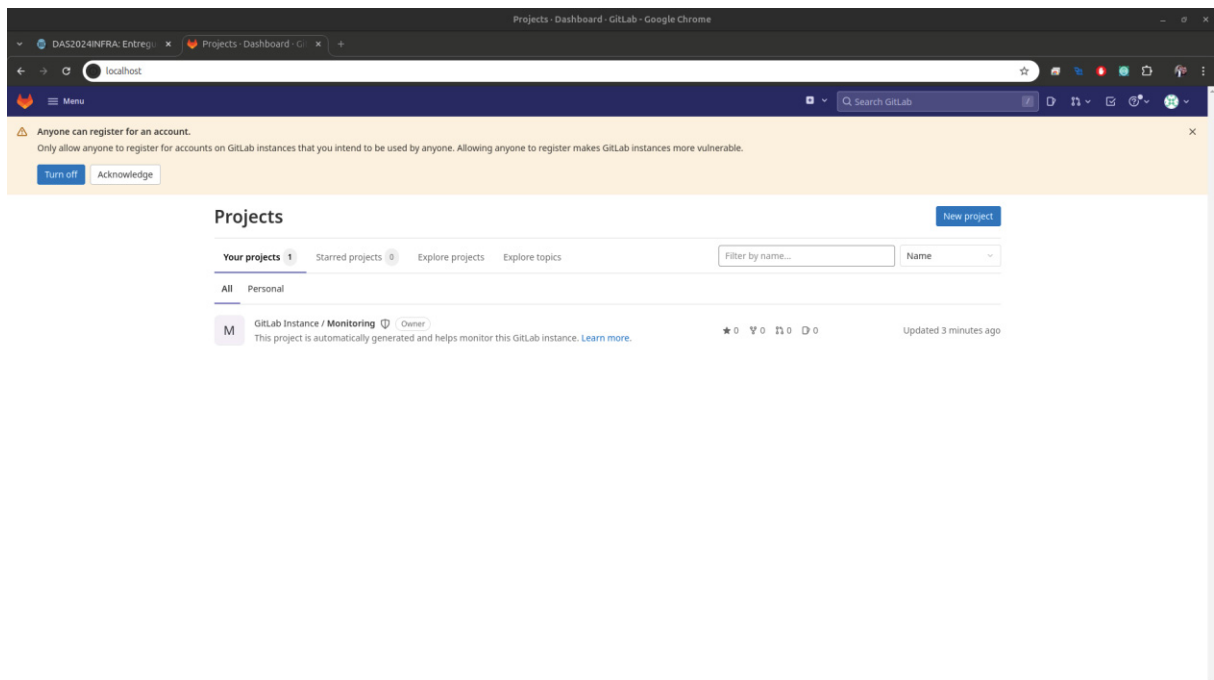
## 11.1 ARTEFATOS DO PROJETO

FIGURA 26 – CONTAINER



FONTE: O autor, em colaboração com colegas (2025)

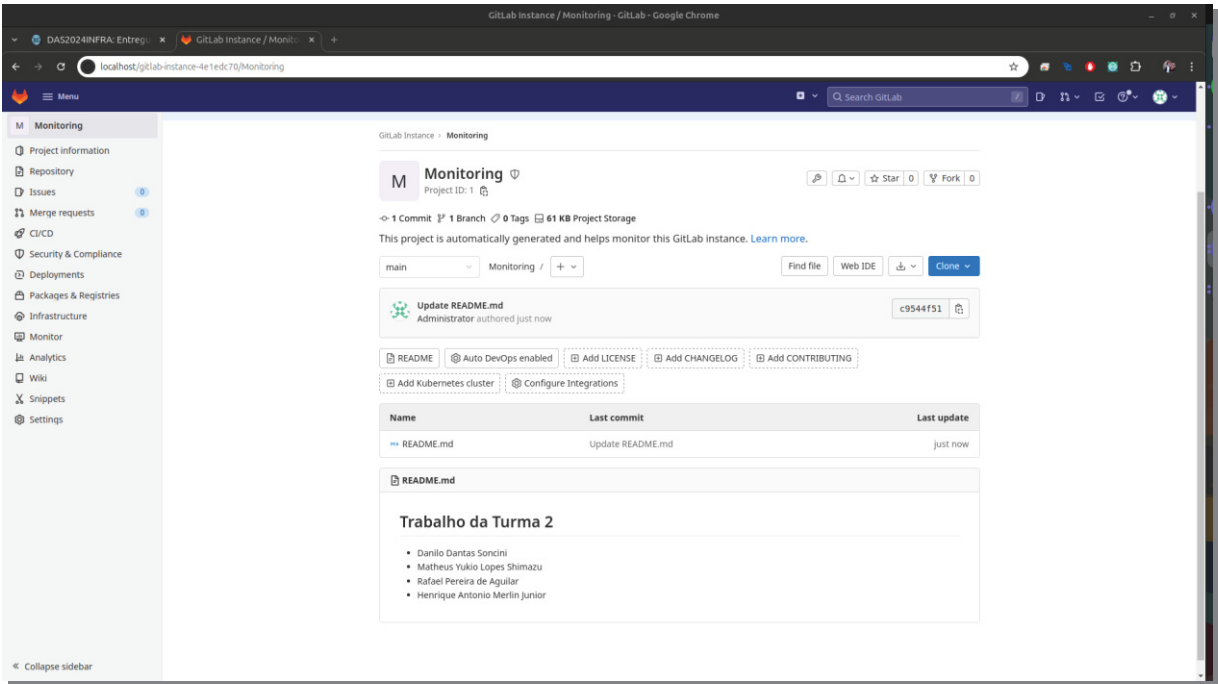
FIGURA 27 – GITLAB



FONTE: O autor, em colaboração com colegas (2025)



FIGURA 28 – COMMIT



FONTE: O autor, em colaboração com colegas (2025)

## 12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

Nesta última disciplina de Testes Automatizados, o trabalho final era, usando Playwright (2025), criar um teste em TypeScript (2025) que, através do Firefox, entra no site aNotepad (2009) e preenche o título com “Entrega trabalho TEST DAS 2024” e o conteúdo da nota com nome e matrícula do aluno.

O Playwright é uma ferramenta incrível que permite que o usuário crie testes baseado no fluxo de uso dele, através de cliques e entradas de dados em um site, por exemplo. Com isso é muito fácil a criação de testes automatizados de frontend, algo importante em DevOps para projetos ágeis.

A automação de testes reduz erros e garante que refatorações de código não impactem negativamente no comportamento esperado pelo usuário. Em um contexto ágil onde a entrega é contínua, a execução de testes é aumentada de forma inviável e perigosa se for feita manualmente. Desta forma a disciplina reforça a importância da união de DevOps e testes automatizados no desenvolvimento ágil de software e finaliza o curso de Especialização.

## 12.1 ARTEFATOS DO PROJETO

FIGURA 29 – CÓDIGO DO TESTE

```
1  import refrom playwright.sync_api  import Playwright, sync_playwright, expect
2
3  def run(playwright: Playwright) -> None:
4      browser = playwright.firefox.launch(headless=False)
5      context = browser.new_context()
6
7      page1 = context.new_page()  page1.goto("https://pt.anotepad.com/")
8
9      page1.get_by_role("textbox", name="Título da Nota").click()
10     page1.get_by_role("textbox", name="Título da Nota").fill("Entrega trabalho TEST DAS 2024")
11     page1.get_by_role("textbox", name="Conteúdo da Nota").click()
12     page1.get_by_role("textbox", name="Conteúdo da Nota").fill("Matheus Yukio Lopes Shimazu/GRR20171625")
13
14     context.close()
15     browser.close()
16
17  with sync_playwright() as playwright:
18     run(playwright)
```

FONTE: O autor (2025)

## 13 CONCLUSÃO

Este memorial de projetos apresentou o caminho educacional das disciplinas que juntas formam o curso de Desenvolvimento Ágil de Software, desde a origem e motivação do Manifesto Ágil, pelos processos de desenvolvimento e através de diversas ferramentas e técnicas para enfim chegar na aplicação prática dos preceitos ágeis nos ambientes WEB e *Mobile*.

O trajeto se iniciou com a base teórica e histórica das metodologias ágeis, sendo seguida por tópicos de gerenciamento de equipes, projetos e *releases* e, junto com as disciplinas AAP e UX, criou um aglomerado de princípios, valores, métodos e boas práticas gerais que podem ser adaptados e utilizados amplamente em variados contextos para melhorar e agilizar o processo de desenvolvimento de softwares.

Sobre esta base teórica foi ensinado programação prática em bancos de dados, WEB, *Mobile* e outros ambientes, utilizando-se ferramentas e *frameworks* como Angular, Spring Boot, Android Studio, Docker e incluindo DevOps e automação de testes.

Desenvolvimento *Mobile* em especial foi onde ficou clara a real necessidade de reunir todas as técnicas aprendidas no curso para ter competitividade no mercado de aplicações de celular.

Entre os desafios encontrados para a adoção do desenvolvimento ágil de software, os principais são:

- Agilidade e Burocracia: As metodologias ágeis aplicadas exageradamente de forma descuidada podem acabar criando mais burocracia do que resolvem, deve haver um equilíbrio e consenso da equipe sobre o que será utilizado;
- DevOps e Complexidade: De modo geral DevOps bem feito melhora a qualidade e confiabilidade dos processos, mas também acarreta um custo inicial técnico, temporal e financeiro por ser complexo.

Em resumo, o desenvolvimento ágil surgiu no começo do século 21, em um ambiente onde o ritmo de desenvolvimento começava a acelerar e métodos tradicionais eram insuficientes para suprir demandas de modo satisfatório. O surgimento de lojas de apps em 2008 criou uma explosão de competitividade, onde

pequenos detalhes, *releases* e *feedback* rápidos são diferenciais importantes. E é neste contexto que o Desenvolvimento Ágil brilha, suprimindo a necessidade de aplicações agradáveis e funcionais de maneira ágil.

## REFERÊNCIAS

- ANDROID. **Android Studio**. 2025. Disponível em: <https://developer.android.com/studio>. Acesso em: 22 nov. 2025.
- ANGULAR. **Angular**. 2025. Disponível em: <https://angular.dev/>. Acesso em: 22 nov. 2025.
- ANOTEPAD. **aNotepad**. 2009. Disponível em: <https://pt.anotepad.com>. Acesso em: 26 out. 2025.
- APPLE. **App Store**. 2025. Disponível em: <https://www.apple.com/br/app-store/>. Acesso em: 22 nov. 2025.
- BECK, K. *et al.* **Manifesto for Agile Software Development**. 2001. Disponível em: <https://agilemanifesto.org/>. Acesso em: 26 out. 2025.
- CHEN, L. **Continuous Delivery: Huge Benefits, but Challenges Too**. 2015. *IEEE Software*. v. 32, n.2, p. 50-54. Disponível em: <https://ieeexplore.ieee.org/document/7006384>. Acesso em: 22 nov. 2025.
- CNN. **Mobile apps overtake PC Web usage in U.S.** 28 fev. 2014. Disponível em: <https://fox4kc.com/news/mobile-apps-overtake-pc-web-usage-in-u-s/>. Acesso em: 26 out. 2025.
- COLEMAN, J.; VACANTI, D. **Kanban Guide**. *Kanban Guides (2025)*. Disponível em: <https://kanbanguides.org/the-kanban-guide/2025.5/>. Acesso em: 26 out. 2025.
- CONTE, N. Visualized: **Desktop VS. Mobile Global Web Traffic Over Time**. 16 ago. 2025. Disponível em: <https://www.visualcapitalist.com/desktop-vs-mobile-global-web-traffic/>. Acesso em: 26 out. 2025.
- EBERT, C.; GALLARDO, G.; SERRANO, N. **DevOps**. 2016. *IEEE Software*. v. 33, n. 2, p. 94-100. Disponível em: <https://ieeexplore.ieee.org/document/7458761>. Acesso em: 22 nov. 2025.
- ERDEM, A. **Mind Maps as a Lifelong Learning Tool**. 2017. *Universal Journal of Educational Research*. v. 5, n. n12A, p. 1-7. Disponível em: <https://eric.ed.gov/?id=EJ1165490>. Acesso em: 22 nov. 2025.
- GITLAB. **GitLab**. 2025. Disponível em: <https://about.gitlab.com/>. Acesso em: 26 out. 2025.
- HAIDABRUS, B.; GRABIS, J.; IVANOV, V.; DRUZHININ, E.; PSAROV, O. **Improving Agile teams effectiveness through the metrics**. 2023. *IEEE 64th International Scientific Conference on Information Technology and Management*

Science of Riga Technical University (ITMS). p. 1-5. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10317789>. Acesso em: 22 nov. 2025.

HP-API. **HP-API**. 2025. Disponível em: <https://hp-api.onrender.com/>. Acesso em: 26 out. 2025.

JAVA. **Java**. 2025. Disponível em: <https://www.java.com/pt-BR/>. Acesso em: 26 out. 2025.

JEFFRIES, R.; MELNIK, G. 2007. **Guest Editors' Introduction: TDD--The Art of Fearless Programming**. *IEEE Software*. v. 24, n. 3, p. 24-30. Disponível em: <https://ieeexplore.ieee.org/abstract/document/4163024>. Acesso em: 22 nov. 2025.

KANBAN SIMULATOR. **Kanban Simulator**. 2025. Disponível em: <http://www.kanbanboardgame.com/>. Acesso em: 26 out. 2025.

KONG, P.; LI, L.; BISSYANDÉ, T. F.; KLEIN, J. **Automated Testing of Android Apps: A Systematic Literature Review**. 2019. *IEEE Transactions on Reliability*. v. 68, n. 1, p. 45-66. Disponível em: <https://ieeexplore.ieee.org/document/8453877>. Acesso em: 22 nov. 2025.

LEAU, Y. B.; LOO, W. K.; THAM, W. Y.; TAN, S. F. **Software development life cycle AGILE vs traditional approaches**. 2012. *International Conference on Information and Network Technology*. p. 162-167. Disponível em: <https://ku-fpg.github.io/files/agile-traditional.pdf>. Acesso em: 22 nov. 2025.

MARTIN, R. C.; FEATHERS, M. C.; OTTINGER, T. R. 2008. **Clean Code: A Handbook of Agile Software Craftsmanship**. *Prentice Hall PTR*.

MIN, W. 2010. **Analysis on Bubble Sort Algorithm Optimization**. 2010 *International Forum on Information Technology and Applications*, Kunming, China. p. 208-211.

MYSQL. **MySQL**. 2025. Disponível em: <https://www.mysql.com/>. Acesso em: 26 out. 2025.

OMG. **OMG Unified Modeling Language 2.5.1**. *Object Management Group Document Number formal/2017-12-05*. Disponível em: <https://www.omg.org/spec/UML/2.5.1/PDF>. Acesso em: 26 out. 2025.

PLAYWRIGHT. **Playwright**. 2025. Disponível em: <https://playwright.dev/>. Acesso em: 26 out. 2025.

POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development: An Agile Toolkit**. *Boston: Addison-Wesley Professional*, 2003. ISBN 978-0-321-15078-3.

POSTGRESQL. **PostgreSQL**. 2025. Disponível em: <https://www.postgresql.org/>. Acesso em: 26 out. 2025.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Modeling Language Reference Manual**. 2005. *Pearson Education India*.

SHRIVASTAVA, A.; JAGGI, I.; KATOCH, N.; GUPTA, D; GUPTA, S. 2021. **A systematic review on extreme programming**. *Journal of Physics: Conference Series* v. 1969, n. 1, p. 012046. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1969/1/012046/meta>. Acesso em: 22 nov. 2025.

SONG, I. Y.; FROEHLICH, K. 1994. **A Practical Guide to Entity-Relationship Modeling**. *IEEE Potentials*. v. 13.5, p. 29-34. Disponível em: <https://cci.drexel.edu/faculty/song/courses/info%20605/appendix/AppendixA.PDF>. Acesso em: 22 nov. 2025.

SPRING BOOT. **Spring Boot**. 2025. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 26 out. 2025.

SQL. **SQL**. 2025. Disponível em: <https://support.microsoft.com/pt-br/topic/acessar-sql-conceitos-b%C3%A1sicos-vocabul%C3%A1rio-e-sintaxe-444d0303-cde1-424e-9a74-e8dc3e460671>. Acesso em: 26 out. 2025.

SUTHERLAND, J.; SCUMNIOTALES, J.; SCHWABER, K. **Agile Development: Lessons learned from the first Scrum**. 2004. Disponível em: [https://web.archive.org/web/20140630020607/http://www.scrumalliance.org/resource\\_download/35](https://web.archive.org/web/20140630020607/http://www.scrumalliance.org/resource_download/35). Acesso em: 26 out. 2025.

TEIXEIRA, F. 2014. **Introdução e boas práticas em UX Design**. Editora Casa do Código.

TYPESCRIPT. **TypeScript**. 2025. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 26 out. 2025.