

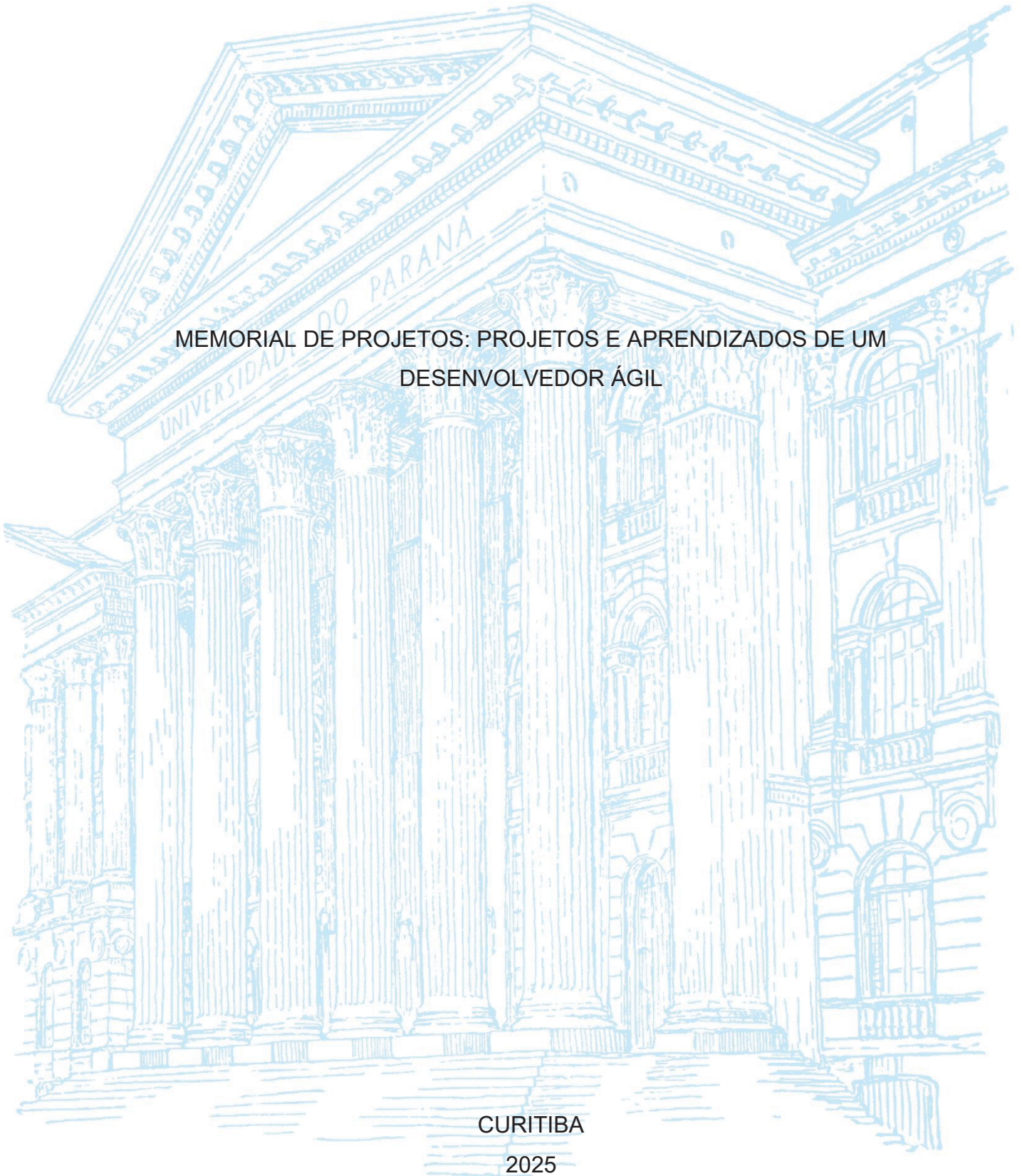
UNIVERSIDADE FEDERAL DO PARANÁ

JULIO CESAR MARIOTTO JUNIOR

MEMORIAL DE PROJETOS: PROJETOS E APRENDIZADOS DE UM
DESENVOLVEDOR ÁGIL

CURITIBA

2025



JULIO CESAR MARIOTTO JUNIOR

MEMORIAL DE PROJETOS: PROJETOS E APRENDIZADOS DE UM
DESENVOLVEDOR ÁGIL

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

Orientadora: Prof^a. Dr^a. Rafaela Mantovani Fontana

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 **JULIO CESAR MARIOTTO JUNIOR**, intitulada: **MEMORIAL DE PROJETOS: PROJETOS E APRENDIZADOS DE UM DESENVOLVEDOR ÁGIL**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa. A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 13 de Outubro de 2025.



RAFAELA MANTOVANI FONTANA
Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Este memorial tem como finalidade apresentar a trajetória de aprendizado vivida no curso de Especialização em Desenvolvimento Ágil de Software, destacando a conexão entre teoria e prática no desenvolvimento de sistemas. Durante as disciplinas, foram criados artefatos que representam as etapas fundamentais do ciclo de vida do software, como documentos de requisitos, modelagens funcionais e estruturais, planos de release, quadros Kanban, protótipos de interface, códigos-fonte, scripts de testes automatizados e configurações de infraestrutura utilizando Docker e GitLab. Esses materiais foram elaborados com base em princípios ágeis, como iteração contínua, foco no cliente, entrega incremental, colaboração entre equipes e rápida adaptação a mudanças. A disciplina de Métodos Ágeis proporcionou a fundamentação teórica que permeou todas as outras. As práticas de UX foram aplicadas na prototipação de soluções centradas no usuário, enquanto os conhecimentos de DevOps e Testes possibilitaram a integração e validação contínuas. O planejamento de projetos e a modelagem de sistemas asseguraram organização e coerência técnica nos desenvolvimentos realizados. A interação entre os conteúdos reforçou uma visão abrangente e prática do desenvolvimento ágil, aproximando o ambiente acadêmico da realidade de projetos modernos de software.

Palavras-chave: desenvolvimento ágil; métodos ágeis; software; sistemas; modelagem.

ABSTRACT

This report aims to present the learning experience of the Agile Software Development Specialization course, highlighting the connection between theory and practice in systems development. During the courses, artifacts representing the fundamental stages of the software lifecycle were created, such as requirements documents, functional and structural models, release plans, Kanban boards, interface prototypes, source code, automated test scripts, and infrastructure configurations using Docker and GitLab. These materials were developed based on agile principles such as continuous iteration, customer focus, incremental delivery, team collaboration, and rapid adaptation to change. The Agile Methods course provided the theoretical foundation that permeated all the others. UX practices were applied to the prototyping of user-centric solutions, while DevOps and Testing knowledge enabled continuous integration and validation. Project planning and systems modeling ensured organization and technical coherence in the developments carried out. The interaction between the content reinforced a comprehensive and practical vision of agile development, bringing the academic environment closer to the reality of modern software projects.

Keywords: agile development; agile methods; software; systems; modeling.

SUMÁRIO

| | |
|--|-----------|
| 1 PARECER TÉCNICO..... | 7 |
| 2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE..... | 9 |
| 2.1 ARTEFATOS DO PROJETO..... | 10 |
| 3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2..... | 19 |
| 3.1 ARTEFATOS DO PROJETO..... | 20 |
| 4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2 | 25 |
| 4.1 ARTEFATOS DO PROJETO..... | 26 |
| 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO..... | 27 |
| 5.1 ARTEFATOS DO PROJETO..... | 28 |
| 6 DISCIPLINA: BD – BANCO DE DADOS..... | 29 |
| 6.1 ARTEFATOS DO PROJETO..... | 30 |
| 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO | 34 |
| 7.1 ARTEFATOS DO PROJETO..... | 35 |
| 8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2 | 36 |
| 8.1 ARTEFATOS DO PROJETO..... | 37 |
| 9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE..... | 38 |
| 9.1 ARTEFATOS DO PROJETO..... | 39 |
| 10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2..... | 40 |
| 11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS) | 41 |
| 11.1 ARTEFATOS DO PROJETO..... | 42 |
| 12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS | 43 |
| 12.1 ARTEFATOS DO PROJETO..... | 44 |
| 13 CONCLUSÃO | 45 |
| REFERÊNCIAS..... | 46 |

1 PARECER TÉCNICO

O presente parecer técnico visa fornecer uma análise integrada dos projetos realizados durante as disciplinas da Especialização em Desenvolvimento Ágil de Software, destacando a contribuição de cada uma para a aplicação prática dos princípios ágeis. As atividades desenvolvidas refletem uma abordagem iterativa, incremental e colaborativa.

A disciplina de Métodos Ágeis de Desenvolvimento de Software (MADS) forneceu os fundamentos teóricos dos métodos Scrum, Kanban e Extreme Programming (XP), consolidando práticas como planejamento incremental, papéis e cerimônias ágeis, entrega contínua e adaptação com base em feedback. Esses conceitos formaram a base metodológica dos demais projetos. Segundo Prikladnicki (2014, p. 8), “a maior prioridade é satisfazer o cliente através da entrega contínua e rápida de um software com valor agregado”.

Nas disciplinas de Gerenciamento Ágil de Projetos I e II (GAP), foram criados artefatos como o plano de release, histórias de usuário e simulações utilizando o framework Kanban. Essas ferramentas ajudaram a organizar o escopo, priorizar entregas e otimizar o fluxo de trabalho, preparando o caminho para as aplicações práticas desenvolvidas em outras unidades curriculares. Como aponta Caroli (2018), é ingênuo iniciar o desenvolvimento do software sem antes gastar um tempo descobrindo o que fazer, porém com rapidez e eficiência. Foi com esse pensamento compreendemos a importância do gerenciamento ágil de projetos no desenvolvimento de software.

A integração dos conceitos foi fortalecida nas disciplinas de Modelagem Ágil de Software 1 e 2 (MAG), onde foram elaborados diagramas de casos de uso, diagramas de classes, sequência e atividades que ilustraram cenários reais de interação entre usuários e sistemas. Esses modelos facilitaram a comunicação entre as áreas técnica e de negócios, permitindo refinar e validar as soluções antes da implementação. Segundo Prikladnicki (2014), a documentação mantida é somente para guardar as informações importantes.

A disciplina de UX no Desenvolvimento Ágil de Software (UX) complementou este processo ao colocar o usuário no centro das decisões, criando expectativas significativas e satisfatórias por meio da criação de protótipos funcionais, levando em consideração fatores como usabilidade, acessibilidade, desempenho e estética. Essa

abordagem foi essencial para as decisões de design nos projetos das disciplinas de Desenvolvimento Web 1 e 2 (WEB) e Desenvolvimento Mobile 1 e 2 (MOB), resultando em aplicações navegáveis com interfaces intuitivas e experiências de usuário aprimorada (Grant, 2019).

Do ponto de vista técnico, a disciplina de Infraestrutura para desenvolvimento e implantação de Software (DevOps) possibilitou a aplicação de conceitos de integração e entrega contínua, configurando ambientes com Docker, GitLab e Jenkins. Segundo Wang (2022) essas práticas são vitais para manter a segurança, confiabilidade e escalabilidade em um ambiente ágil.

A qualidade do código foi aprimorada na disciplina de Aspectos Ágeis de Programação (AAP), por meio da aplicação de técnicas de *clean code* e refatorações, práticas que foram reforçadas nos Testes Automatizados (TEST), onde scripts foram desenvolvidos para validar funcionalidades em um ambiente controlado, garantindo segurança em entregas frequentes.

As bases da programação foram consolidadas na disciplina de Introdução à Programação (INTRO), com o desenvolvimento de sistemas simples que acessam banco de dados, enquanto a disciplina de Banco de Dados (BD) forneceu a estrutura necessária para a persistência de informações, utilizando modelagem relacional e SQL.

Todos esses projetos estão organizados neste memorial, por disciplina, evidenciando como os entregáveis se conectam em um fluxo de desenvolvimento ágil completo: da concepção à entrega, passando por planejamento, modelagem, codificação, testes, infraestrutura e validação com usuários. Como aponta Beck (2011, não p.), “a contínua atenção à excelência técnica e bom design aumenta a agilidade”. A constante interação entre teoria e prática aproximou o ambiente acadêmico da realidade de projetos modernos de software.

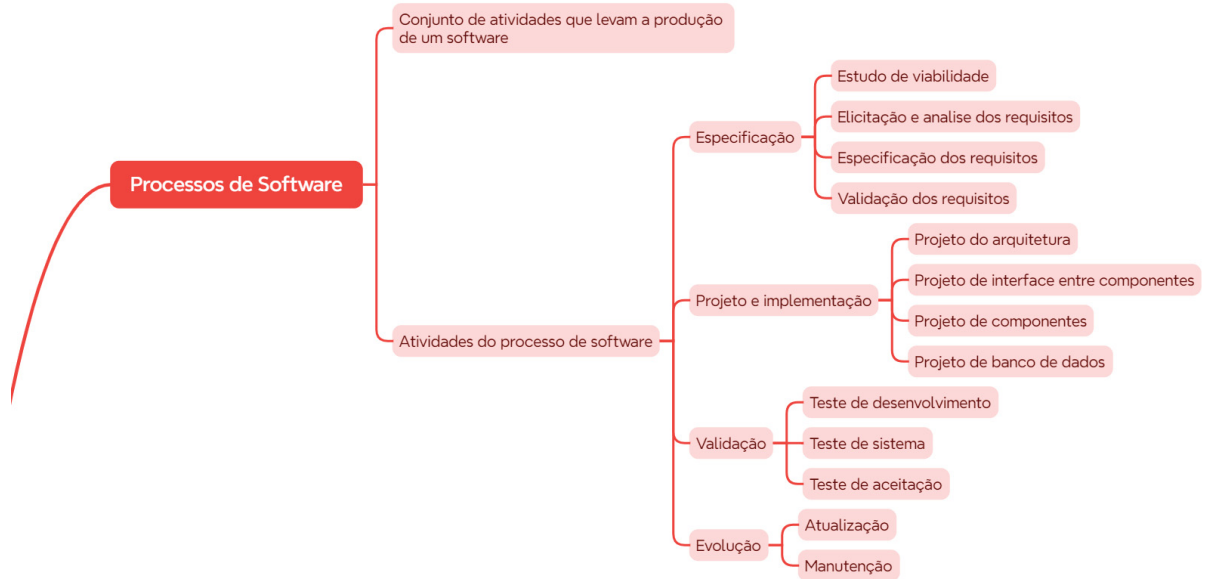
2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

A disciplina de Métodos Ágeis para Desenvolvimento de Software abordou diversos tópicos sobre o desenvolvimento de sistemas desde os processos de software e princípios ágeis, incluindo as metodologias tradicionais, manifesto ágil, Lean Software Development até as metodologias ágeis como Scrum, Extreme Programming (XP), Kanban e abordagens de entrega contínua. O projeto da disciplina se tratou de um mapa mental das duas unidades da disciplina onde é possível ter uma visão detalhada de todos os principais pontos abordado na disciplina.

O Scrum, Kanban e o Manifesto Ágil foram amplamente utilizados em outras disciplinas como UX, GAP 1 e 2, WEB 1 e 2, MOB 1 e 2, DevOps e TEST, promovendo uma abordagem colaborativa e iterativa que favorece a adaptação rápida às mudanças e a entrega contínua de valor.

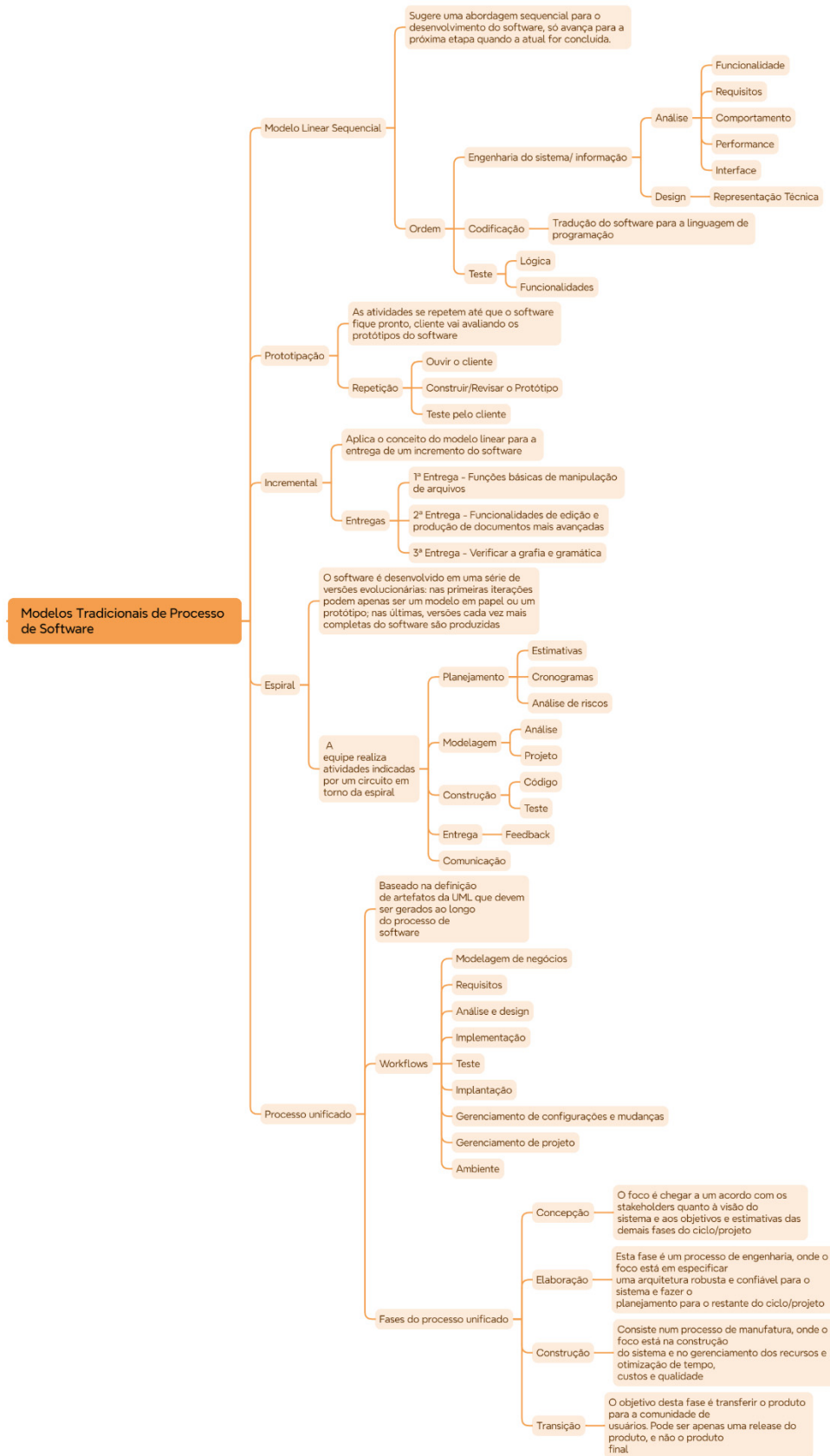
ARTEFATOS DO PROJETO

FIGURA 1 – MAPA MENTAL (1/9)



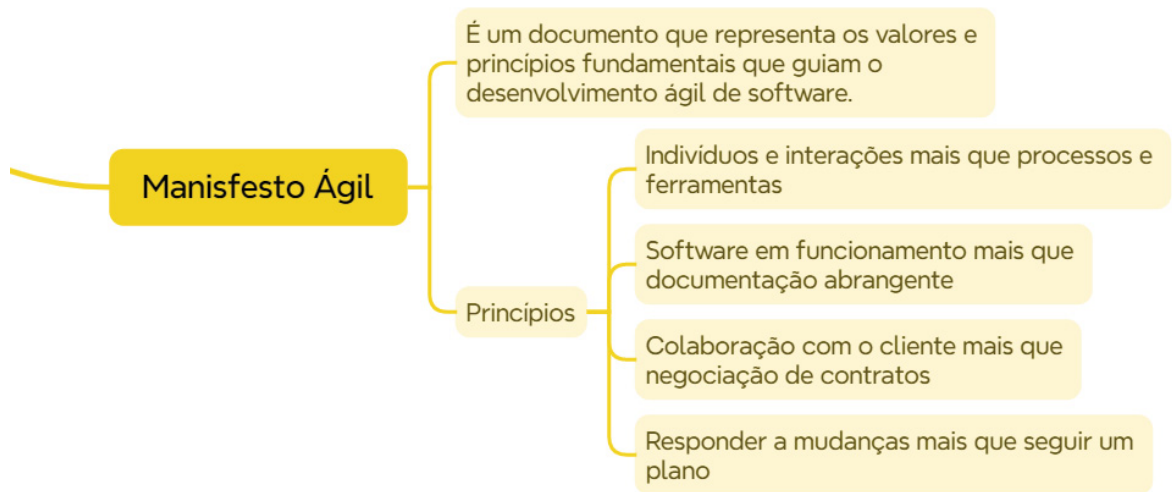
FONTE: O Autor (2025).

FIGURA 2 – MAPA MENTAL (2/9)



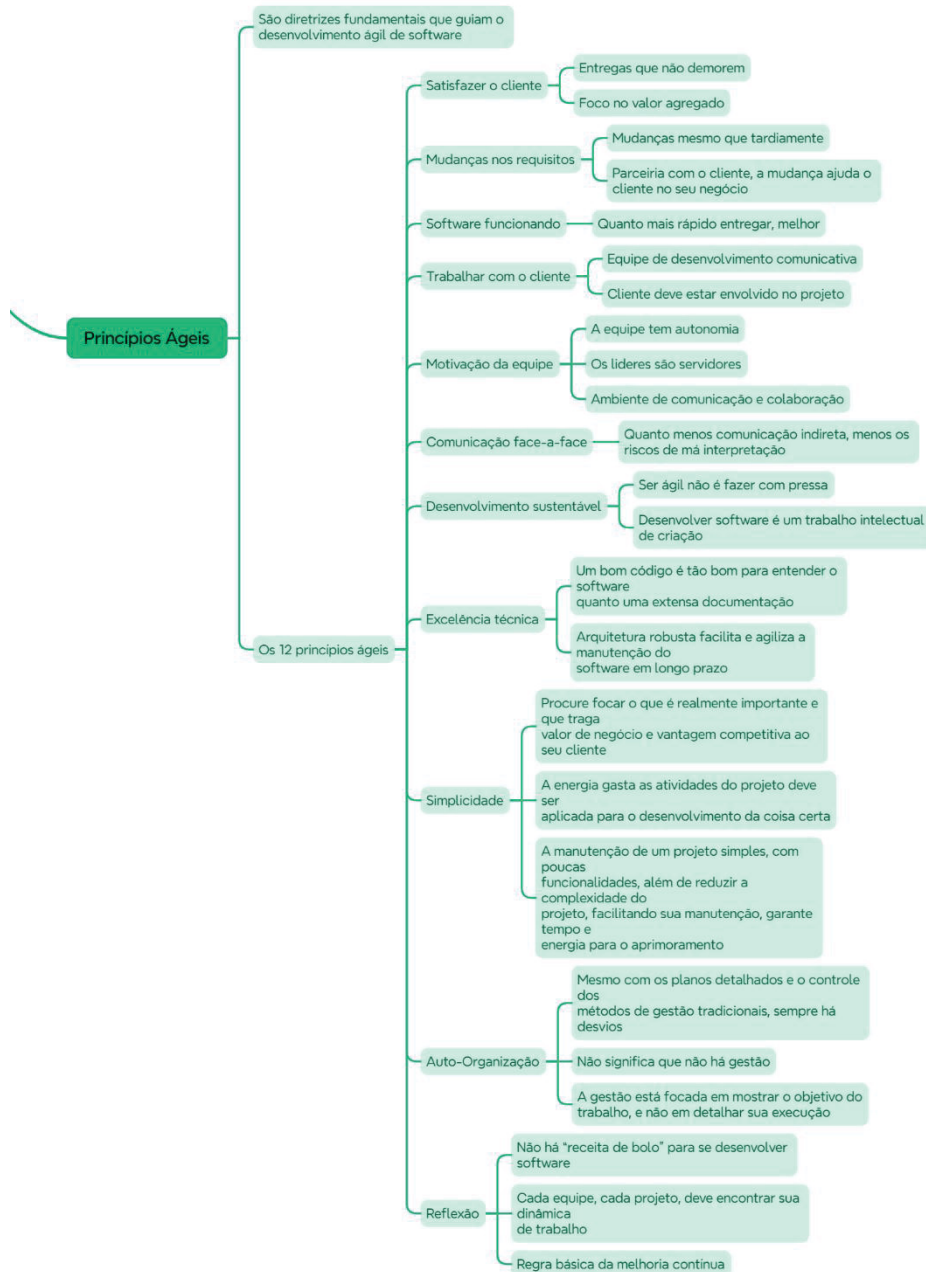
FONTE: O Autor (2025).

FIGURA 3 – MAPA MENTAL (3/9)



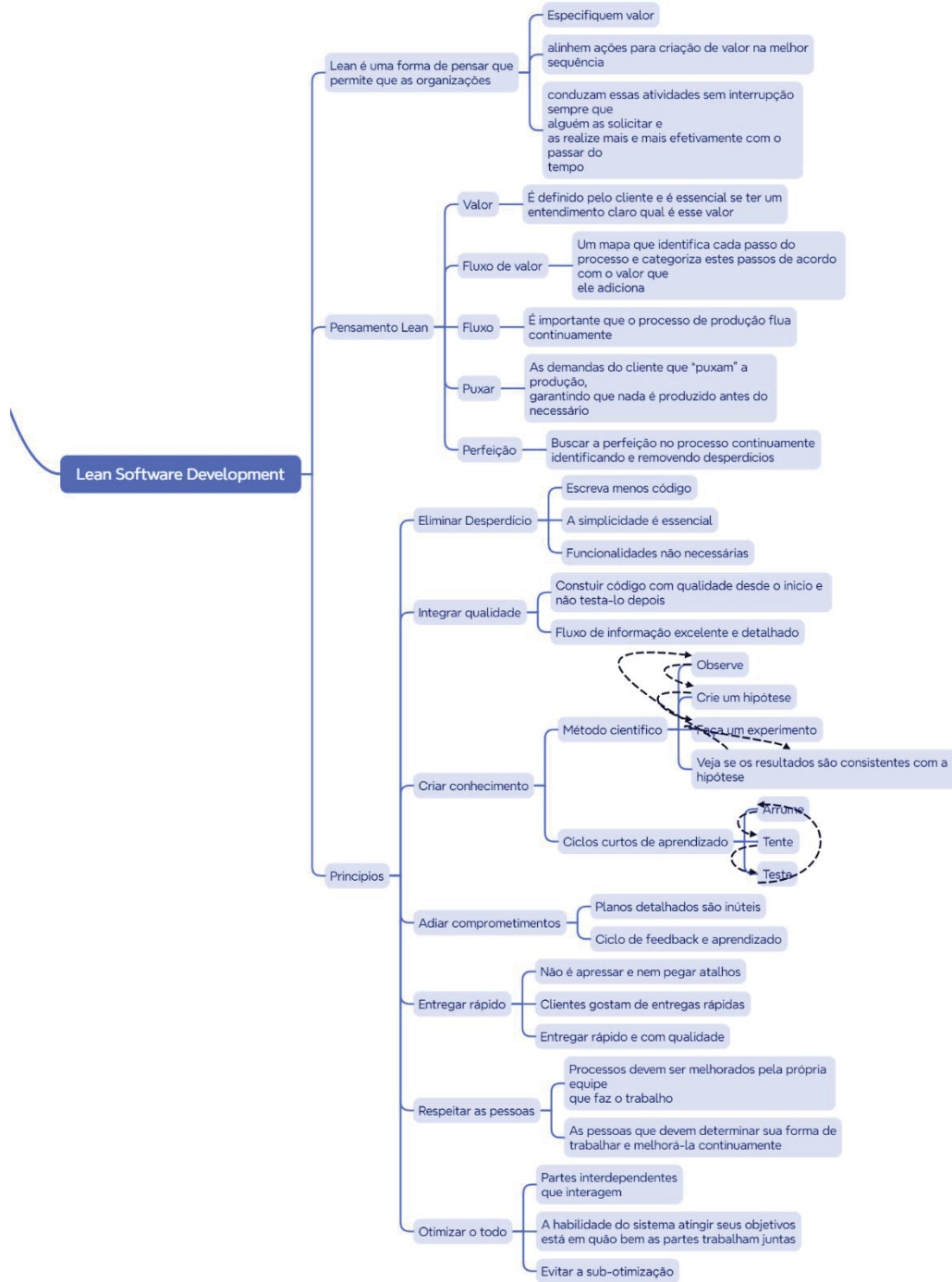
FONTE: O Autor (2025).

FIGURA 4 – MAPA MENTAL (4/9)



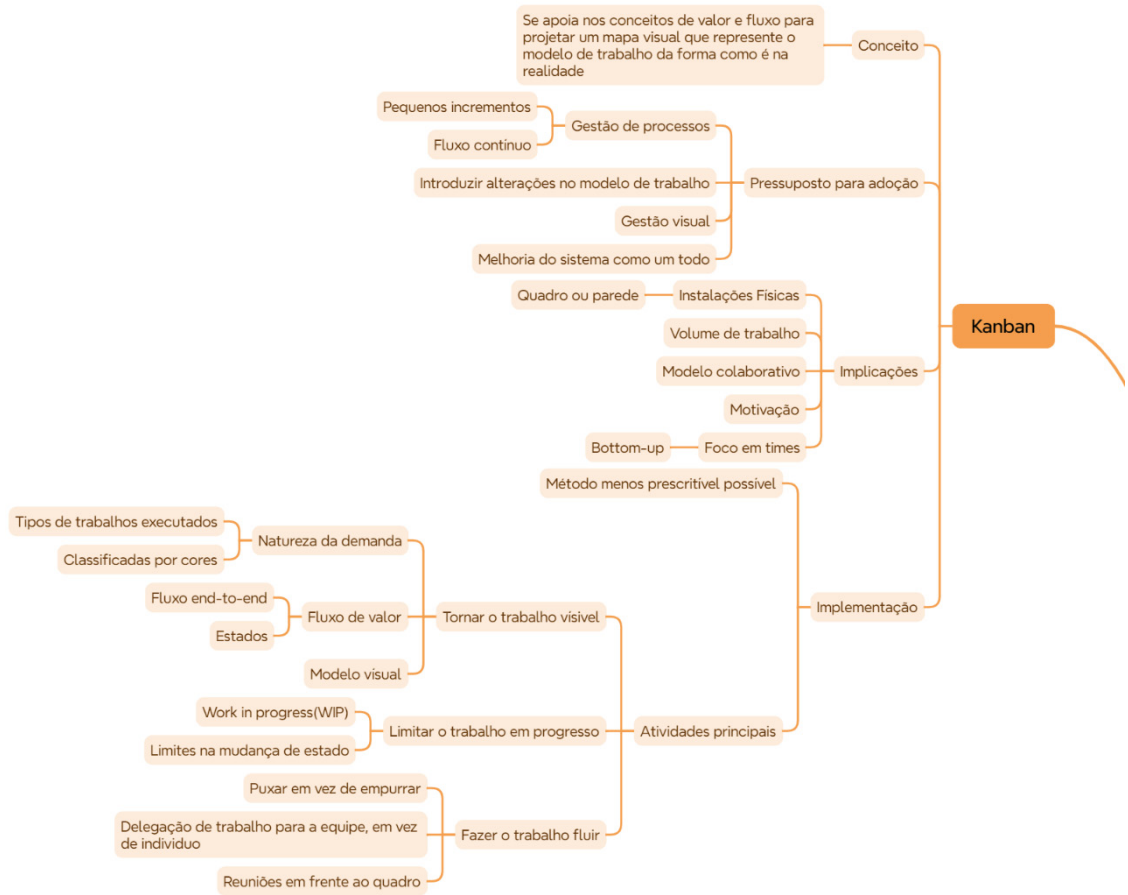
FONTE: O Autor (2025).

FIGURA 5 – MAPA MENTAL (5/9)



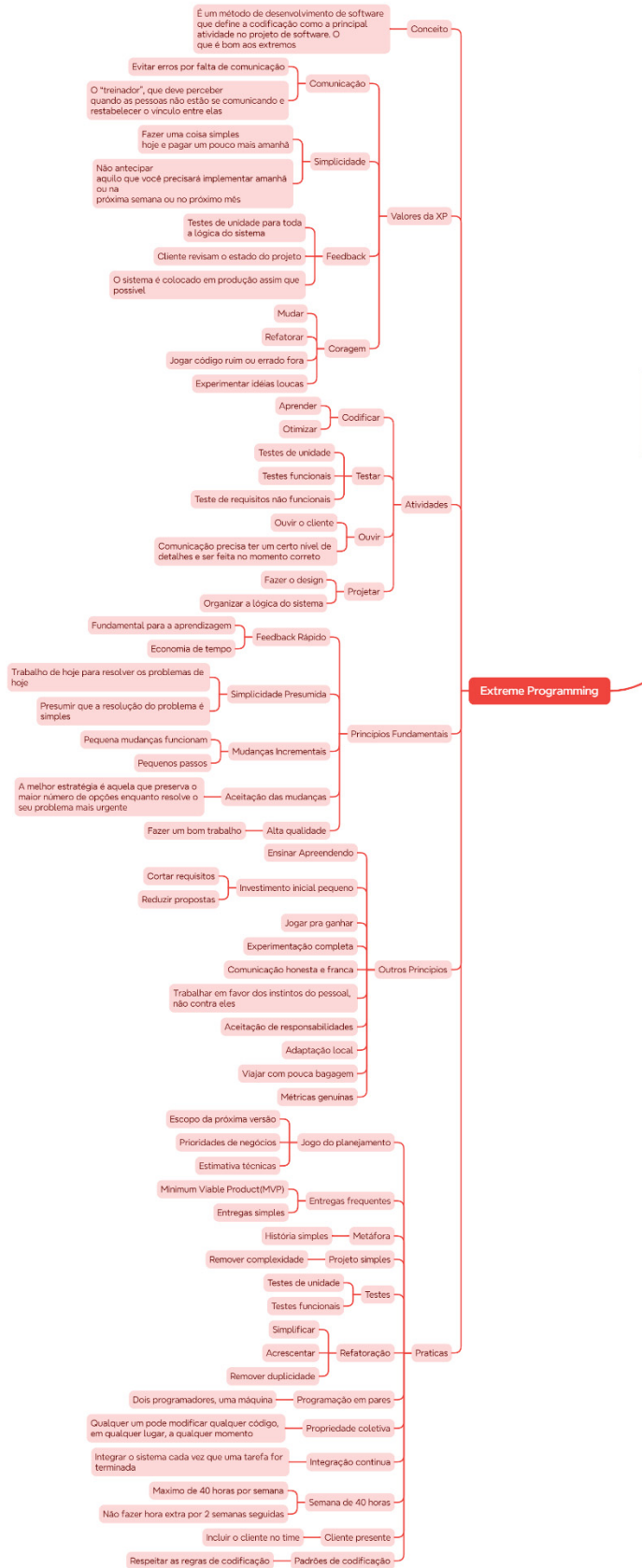
FONTE: O Autor (2025).

FIGURA 7 – MAPA MENTAL (7/9)



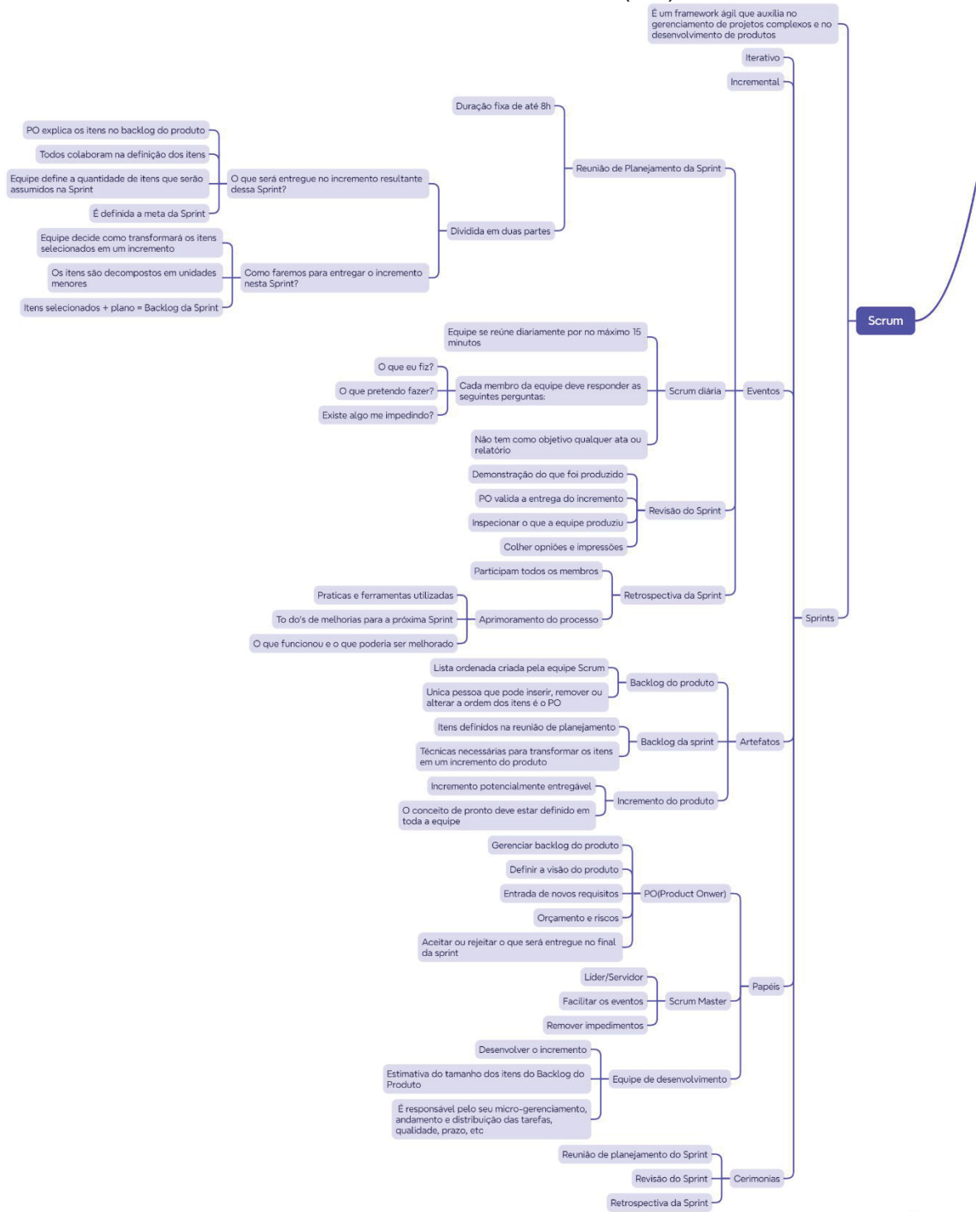
FONTE: O Autor (2025).

FIGURA 8 – MAPA MENTAL (8/9)



FONTE: O Autor (2025).

FIGURA 9 – MAPA MENTAL (9/9)



FONTE: O Autor (2025).

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

A disciplina MAG1 abordou os principais aspectos da modelagem ágil e sua aplicação no desenvolvimento de um software, vimos conceitos do UML, levantamento de requisitos, criação de casos de uso de nível 1 e 2, histórias de usuário, *domain driven design* (DDD).

As abordagens citadas acima foram implementadas em um projeto de sistema para gestão de um condomínio que permitiu a estruturação de forma lógica das funcionalidades esperadas da aplicação. Durante a modelagem funcional foram elaborados diagramas de caso de uso nível 1 e 2 (Figuras 10 e 11). Realizou-se a descrição de histórias de usuário em conjunto com a criação de desenhos de telas com seus critérios de aceitação (Figuras 12 e 13), que serviram para criação do backlog do produto.

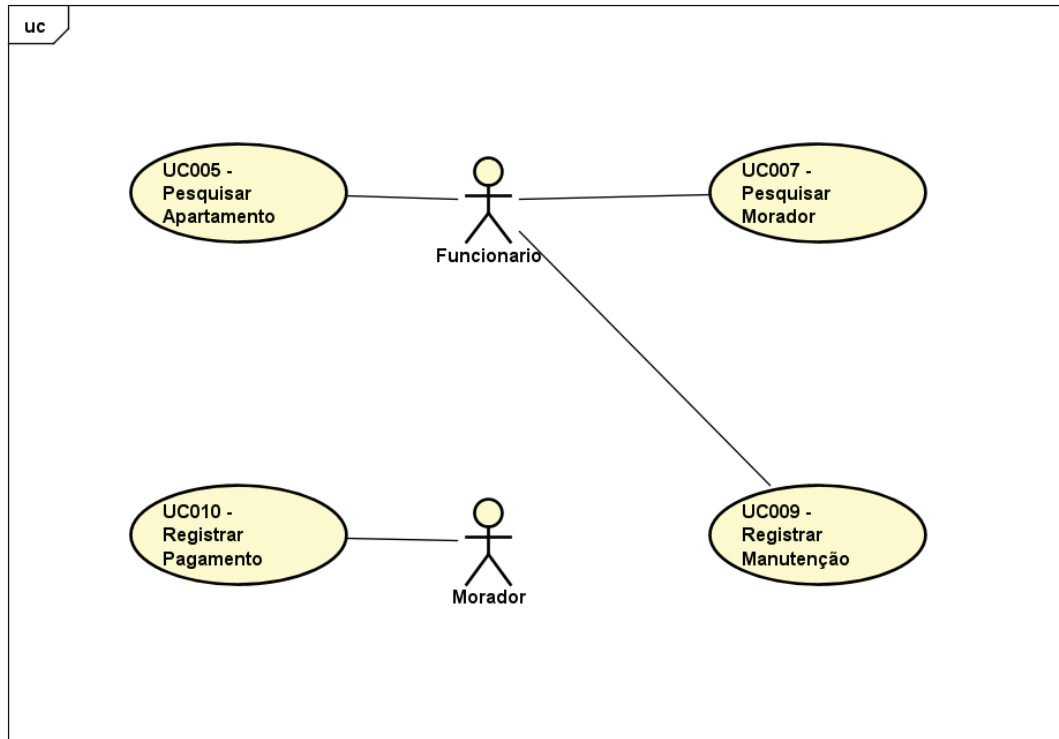
Na disciplina de MAG2, na unidade 1 vimos conceitos sobre objetos, classes, relacionamentos, herança, polimorfismos e a representação técnica do sistema através do diagrama de classes (Figura 14), onde demos continuidade ao projeto de gestão de um condomínio.

Na unidade 2 e 3, vimos o diagrama de sequência (Figura 15), o diagrama de atividades e o diagrama de transição de estados. Essas representações estruturadas facilitaram a transição para as etapas de implementação, orientando tanto a modelagem do banco de dados quanto a lógica de programação.

A integração entre os modelos funcionais e estruturais foi essencial para garantir coerência entre o que foi planejado e o que foi implementado

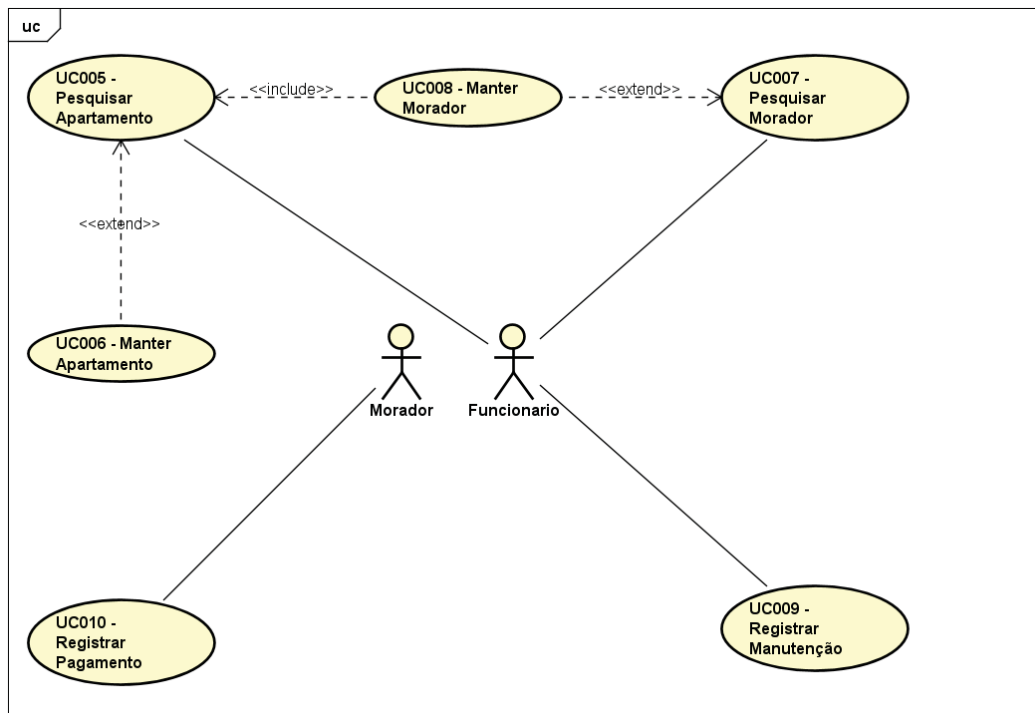
ARTEFATOS DO PROJETO

FIGURA 10 – CASO DE USO NÍVEL 1.



FONTE: O autor (2025).

FIGURA 11 – CASO DE USO NÍVEL 2



FONTE: O autor (2025).

FIGURA 12 – HISTÓRIA DE USUÁRIO

HU001 – PESQUISAR APARTAMENTO

SENDO o Funcionário
QUERO pesquisar os apartamentos
PARA fazer manutenções em seus dados

DESENHO DA TELA

Pesquisar Apartamento

Pesquisar:

| Bloco | Apartamento | Ação | |
|-------|-------------|--|--|
| BL01 | 101 | <input type="button" value="Alterar"/> | <input type="button" value="Excluir"/> |
| BL02 | 201 | <input type="button" value="Alterar"/> | <input type="button" value="Excluir"/> |
| BL02 | 202 | <input type="button" value="Alterar"/> | <input type="button" value="Excluir"/> |

LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve preencher a tabela com todos os apartamentos cadastrados
- 2) Deve permitir incluir um novo apartamento
- 3) Deve permitir alterar os dados de um apartamento cadastrado
- 4) Deve permitir excluir um apartamento cadastrado
- 5) Deve permitir consultar os dados de um apartamento cadastrado
- 6) Deve permitir pesquisar os apartamentos na tabela
- 7) Deve voltar a tela anterior

FONTE: O autor (2025).

FIGURA 13 – CRITÉRIOS DE ACEITAÇÃO

CRITÉRIOS DE ACEITAÇÃO – DETALHAMENTO

- 1) Deve preencher a tabela com todos os apartamentos cadastrados

| | |
|-----------------|--|
| Dado que | |
| Quando | A tela é apresentada |
| Então | A tabela deve ser preenchida com todos os apartamentos cadastrados no banco de dados |

- 2) Deve permitir incluir um novo apartamento

| | |
|-----------------|--|
| Dado que | Os apartamentos estão na tabela |
| Quando | O botão "Novo" é pressionado |
| Então | O sistema chama o UH002 – Manter Apartamento passando o parâmetro "Novo" |

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

| | |
|-----------------|--|
| Dado que | Os apartamentos estão na tabela |
| Quando | O botão "Alterar" é pressionado em uma determinada linha |
| Então | O sistema chama o UH002 – Manter Apartamento passando o parâmetro "Alterar" e os dados do apartamento da linha |

- 4) Deve permitir excluir um apartamento cadastrado

| | |
|-----------------|---|
| Dado que | Os apartamentos estão na tabela |
| Quando | O botão "Excluir" é pressionado em uma determinada linha |
| Então | O sistema pede uma confirmação, exclui o apartamento da base de dados e refaz a tabela da tela lendo novamente o banco de dados |

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

| | |
|-----------------|--|
| Dado que | Os apartamentos estão na tabela |
| Quando | O usuário clica em qualquer dado de um apartamento em uma determinada linha |
| Então | O sistema chama o UH002 – Manter Apartamento passando o parâmetro "Consultar" e os dados do apartamento da linha |

- 6) Deve permitir pesquisar os apartamentos na tabela

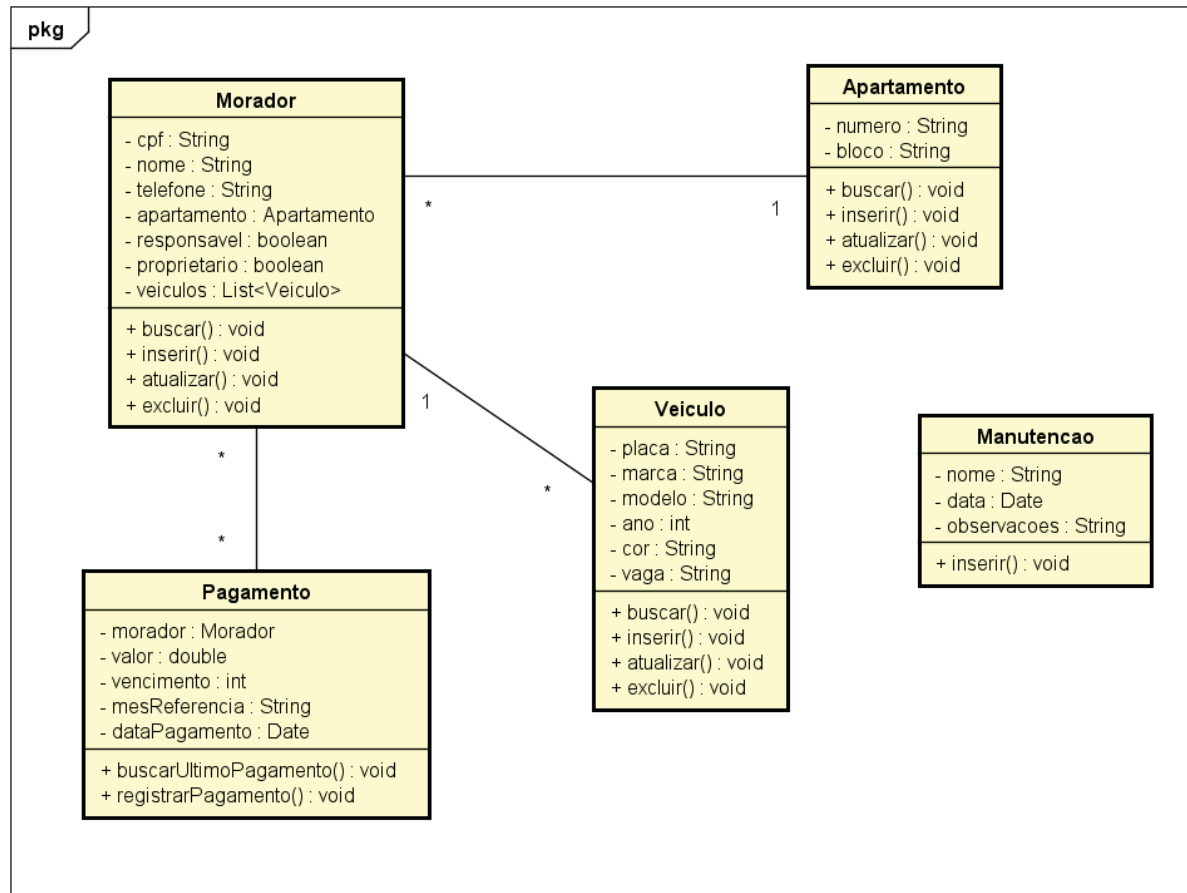
| | |
|-----------------|---|
| Dado que | Os apartamentos estão na tabela |
| Quando | For informado um argumento de pesquisa |
| Então | O sistema filtra e apresenta na tela somente os apartamentos que obedecem ao critério |

- 7) Deve voltar a tela anterior

| | |
|-----------------|--------------------------------------|
| Dado que | |
| Quando | O botão "Voltar" é pressionado |
| Então | O sistema volta para a tela anterior |

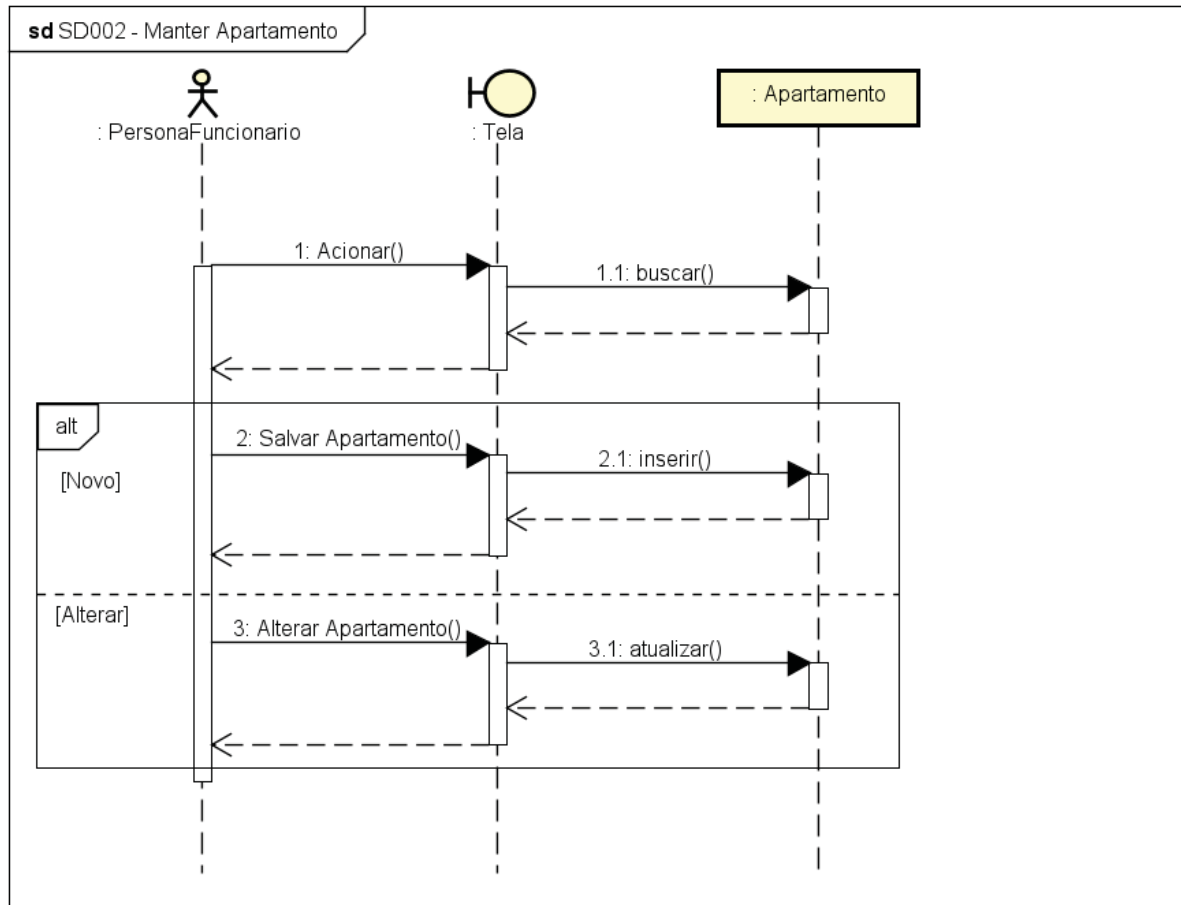
FONTE: O autor (2025).

FIGURA 14 – DIAGRAMA DE CLASSES



FONTE: O autor (2025).

FIGURA 15 – DIAGRAMA DE SEQUÊNCIA



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 de Software I e II abordaram os fundamentos e práticas relacionadas ao gerenciamento de projetos (PMBOK) no contexto de métodos ágeis, com ênfase na aplicação prática de frameworks como Scrum e Kanban.

Durante a disciplina de GAP1, o foco esteve na construção de um plano de release (Figura 16), para o sistema de gestão de condomínio. A atividade envolveu o levantamento, priorização e estimativa de histórias de usuário com base nas necessidades dos perfis de síndico, morador e administrador.

Na disciplina GAP2, os conceitos de fluxo de trabalho e melhoria contínua foram explorados de maneira mais profunda por meio da simulação prática com o Game Kanban (Figura 17). Essa atividade destacou os efeitos da limitação do trabalho em progresso (WIP), da visualização de gargalos e da relevância de um ciclo de feedback ágil. A dinâmica evidenciou a importância da colaboração entre a equipe, da autogestão e da adaptação constante como fundamentos para a eficiência operacional.

As duas disciplinas estabeleceram uma base sólida para o gerenciamento de escopo, tempo e entregas do projeto, interligando-se diretamente com os conteúdos abordados em UML, WEB 1 e 2, MOB 1 e 2 e DevOps. O planejamento das entregas guiou a organização do trabalho ao longo do curso, garantindo que estivesse alinhado com os princípios ágeis de transparência, inspeção e adaptação.

ARTEFATOS DO PROJETO

FIGURA 16 – PLANO DE RELEASE

Cálculo da Velocidade:

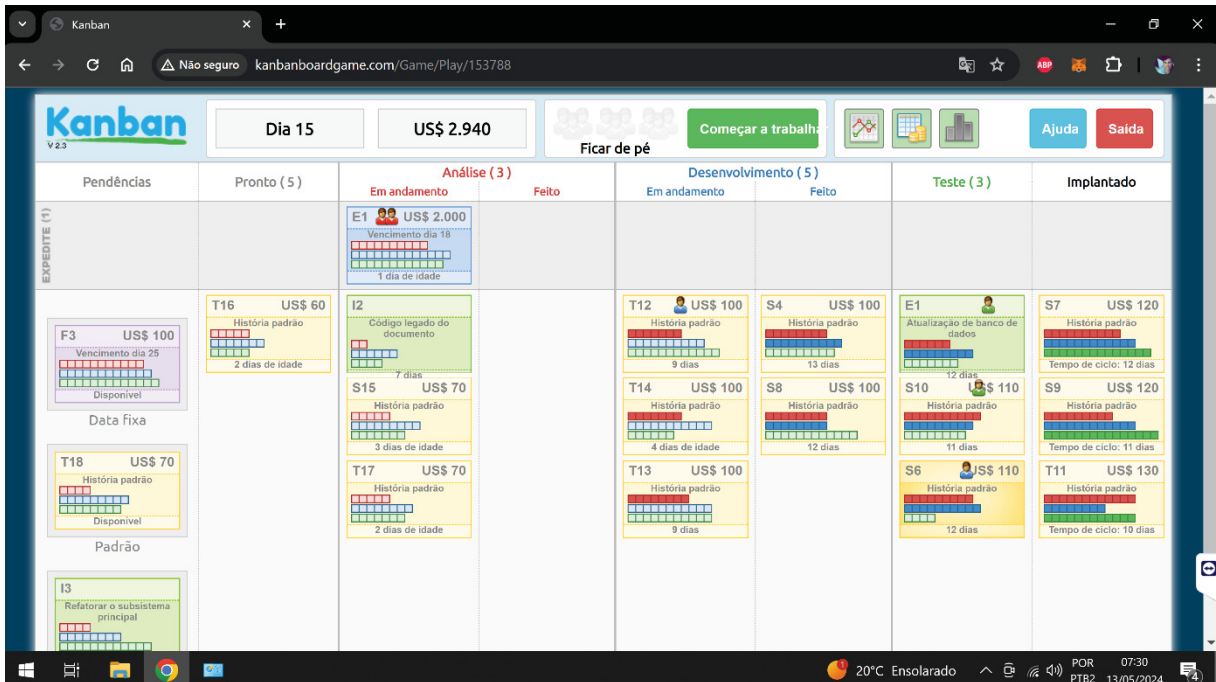
| | |
|----------------------------------|------------------------------|
| Horas disponíveis por dia: 8 | Tamanho da Sprint: 2 semanas |
| Horas disponíveis por Sprint: 80 | Velocidade: 10 |

Plano de Release:

| Iteração/Sprint 1 | Iteração/Sprint 2 | Iteração/Sprint 3 |
|---|---|--|
| Data Início: 06/05/24 Data Fim: 17/05/24 | Data Início: 20/05/24 Data Fim: 31/05/24 | Data Início: 03/06/24 Data Fim: 14/06/24 |
| <HU006 - Registrar Pagamento> SENDO o Morador QUERO registrar o pagamento do condomínio PARA estar regularizado perante o condomínio ESTIMATIVA (7) | <HU001 - Pesquisar Apartamento> SENDO o Funcionário QUERO pesquisar os apartamentos PARA fazer manutenções em seus dados ESTIMATIVA (5) | <HU002 - Manter Apartamento > SENDO o Funcionário QUERO manter os dados do apartamento PARA que seus dados fiquem atualizados ESTIMATIVA (4) |
| <HU005 - Registrar Manutenção> SENDO o Funcionário QUERO registrar manutenções PARA manter registros no banco de dados ESTIMATIVA (3) | <HU003 - Pesquisar Morador> SENDO o Funcionário QUERO pesquisar os moradores PARA fazer manutenção em seus dados ESTIMATIVA (5) | <HU004 - Manter Morador> SENDO o Funcionário QUERO manter os dados do morador PARA que seus dados fiquem atualizados ESTIMATIVA (5) |

FONTE: O autor (2025).

FIGURA 17 – KANBAN GAME



FONTE: O autor (2025).

5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

A disciplina de Introdução à Programação abordou os fundamentos da lógica computacional e das estruturas básicas de programação. Utilizando JAVA como linguagem de programação, a proposta pedagógica focou na prática da escrita de algoritmos, no uso de estruturas condicionais e de repetição, além da manipulação de dados em bancos relacionais e da aplicação de testes unitários com JUnit.

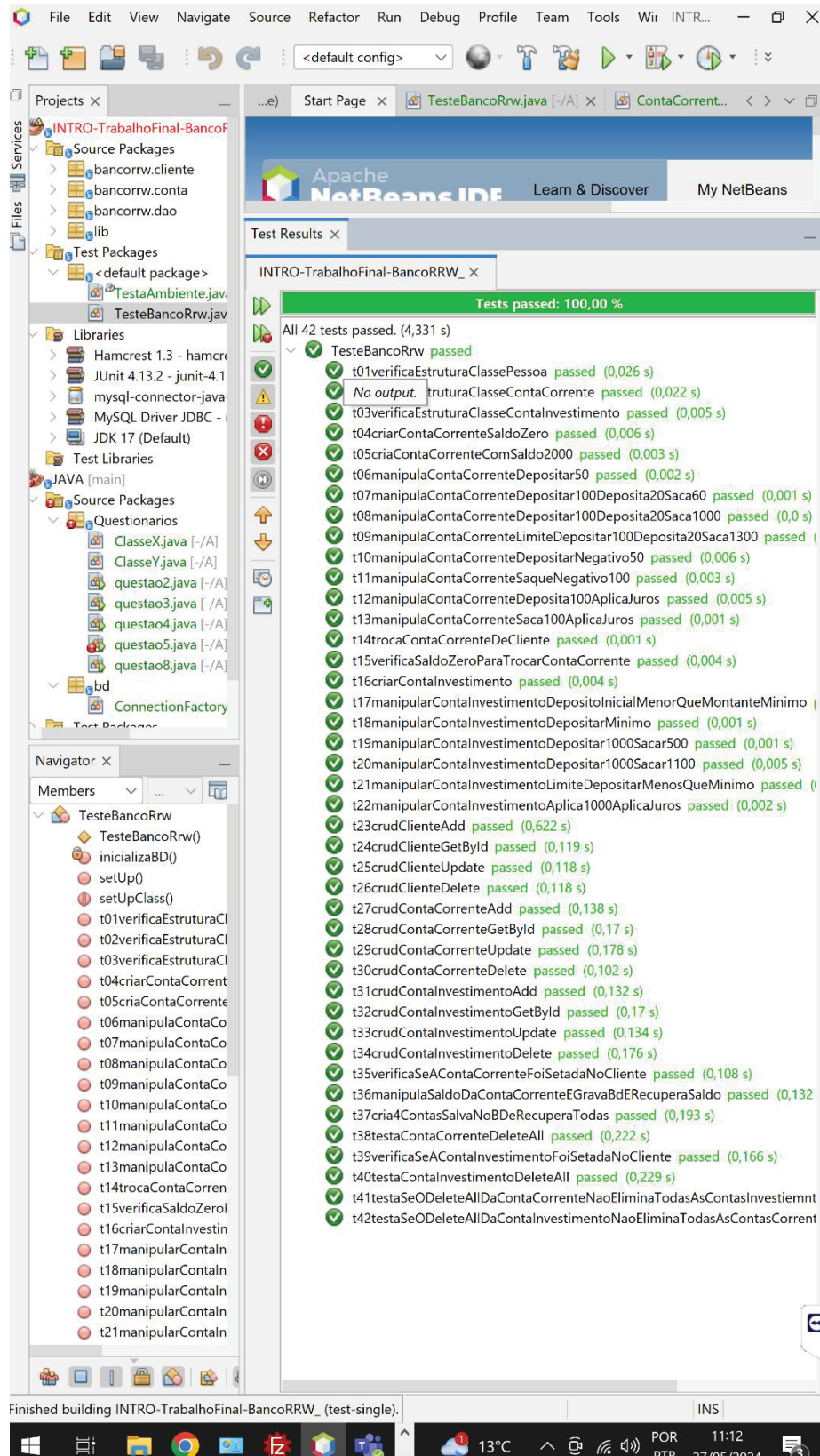
Como atividade avaliativa, os alunos desenvolveram o *back-end* de um sistema bancário simplificado, que suportava operações como cadastro de clientes, criação de contas correntes e de conta investimento, além de movimentações de crédito e débito, com persistência em um banco de dados MySQL. O projeto incluía um diagrama de classes e um conjunto de 42 casos de teste automatizados previamente definidos, com a meta de que pelo menos 40 desses testes fossem aprovados simultaneamente (Figura 18), utilizando o paradigma orientado a objetos e os princípios de desenvolvimento orientado por testes (TDD).

A execução desse projeto possibilitou a aplicação dos conteúdos teóricos em um ambiente prático, simulando o desenvolvimento real de um sistema com validações automatizadas. A prática de TDD foi fundamental para entender a importância dos testes desde as primeiras fases do ciclo de vida do software, um conceito que se mostrou essencial nas disciplinas subsequentes de Aspectos Ágeis de Programação, Testes Automatizados e DevOps.

Além disso, o desenvolvimento do raciocínio lógico e a habilidade de decompor problemas em funções e classes reutilizáveis proporcionaram uma base sólida para a criação de soluções mais complexas nas disciplinas de desenvolvimento Web e Mobile.

ARTEFATOS DO PROJETO

FIGURA 18 – TESTE AUTOMATIZADOS NO JAVA



FONTE: O Autor (2025).

6 DISCIPLINA: BD – BANCO DE DADOS

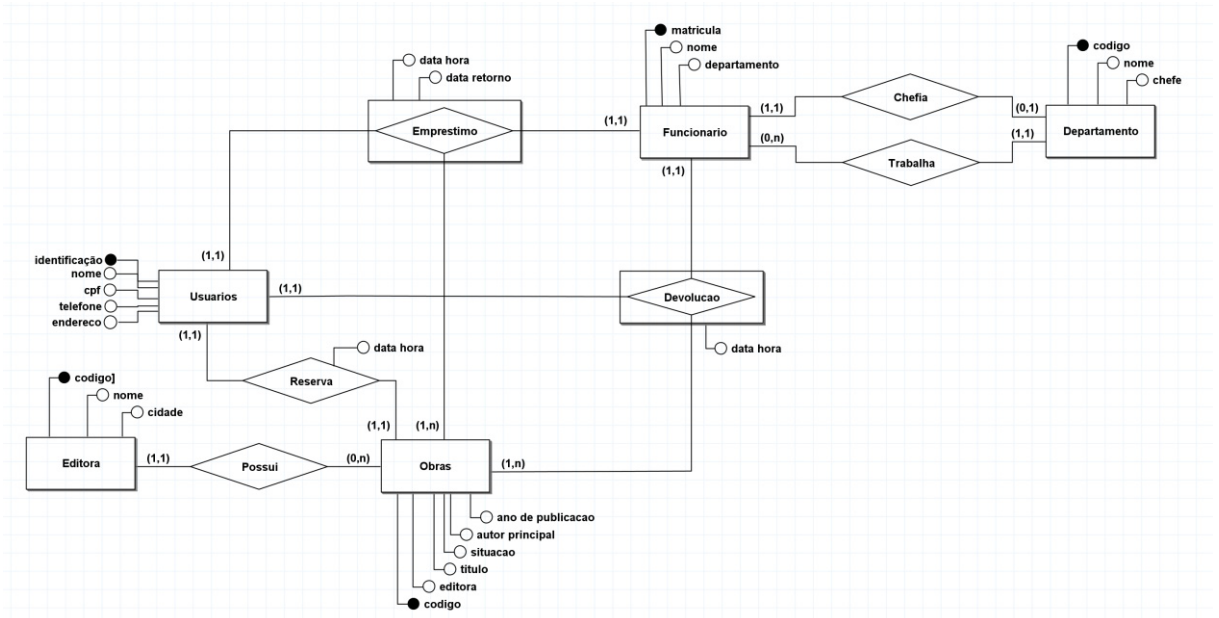
A disciplina de Banco de Dados concentrou-se na compreensão e aplicação das etapas essenciais para a modelagem e implementação de sistemas de banco de dados. Os conteúdos abordados incluíram a modelagem conceitual e lógica seguindo o modelo entidade relacionamento, assim como o estudo da linguagem SQL com foco em boas práticas de normalização, integridade referencial e estruturação dos dados.

O trabalho desenvolvido na disciplina foi dividido em duas questões. A primeira questão envolveu a modelagem conceitual e lógica de um sistema de controle de biblioteca (Figura 19 e 20). A segunda parte, permitiu a escolha de um tema livre, o qual foi escolhido o controle de estoque, onde foram exploradas relações 1:1, 1:N e N:N entre tabelas como Produtos, Categoria, Fornecedores e Pedidos. Essa atividade incluiu a elaboração de um modelo lógico (Figura 21) e scripts para a criação de tabelas (Figura 22), a definição de chaves primárias e estrangeiras, além da implementação de restrições de dados e criação de valores reais para cada tabela.

O conteúdo abordado na disciplina foi amplamente utilizado nos projetos das matérias seguintes como introdução a programação, web e mobile. A clareza na definição do modelo de dados, contribui significativamente para a agilidade na criação do banco de dados, garantindo consistência, robustez e eficiência que são fatores fortemente abordados no desenvolvimento ágil de software.

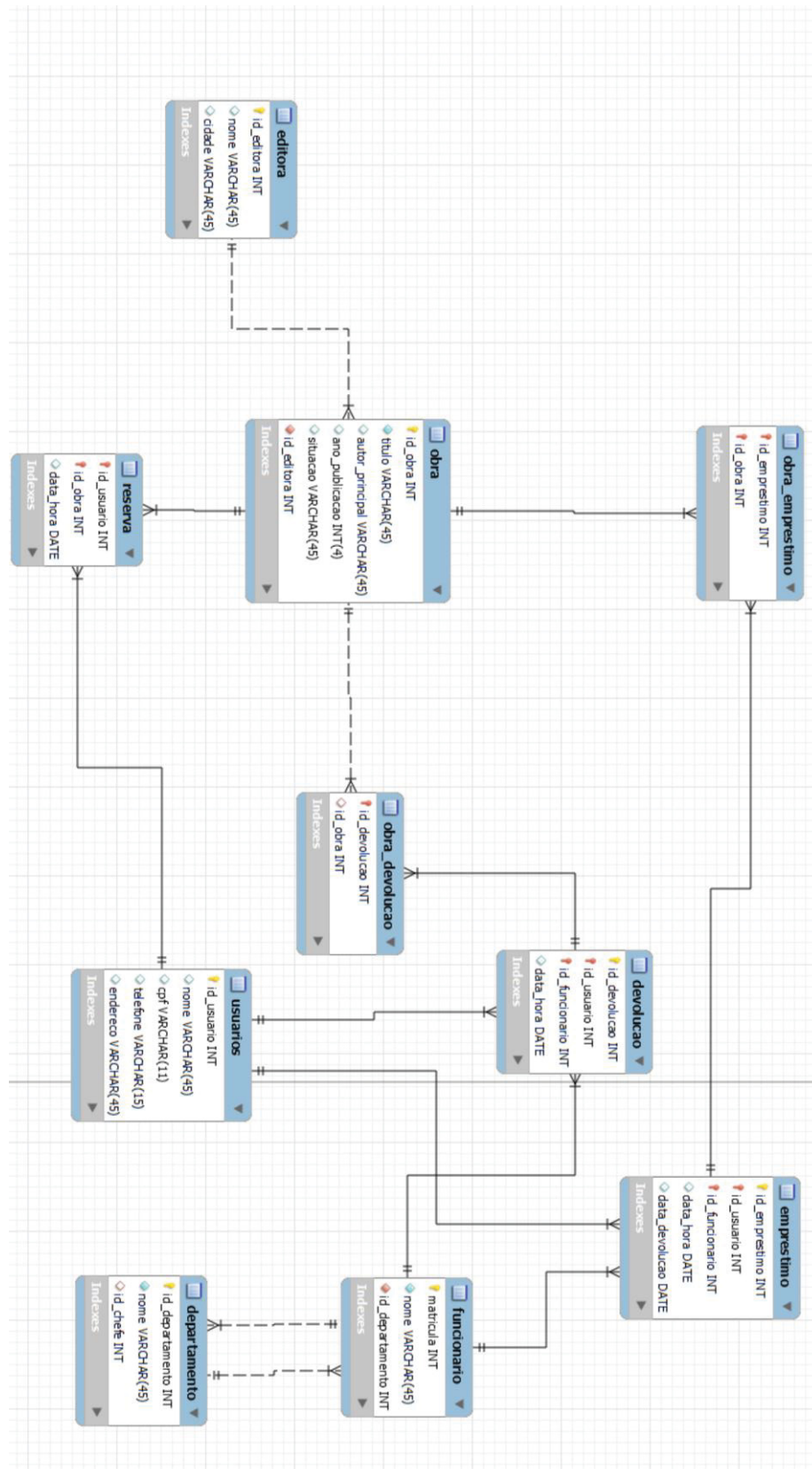
ARTEFATOS DO PROJETO

FIGURA 19 – MODELO CONCEITUAL



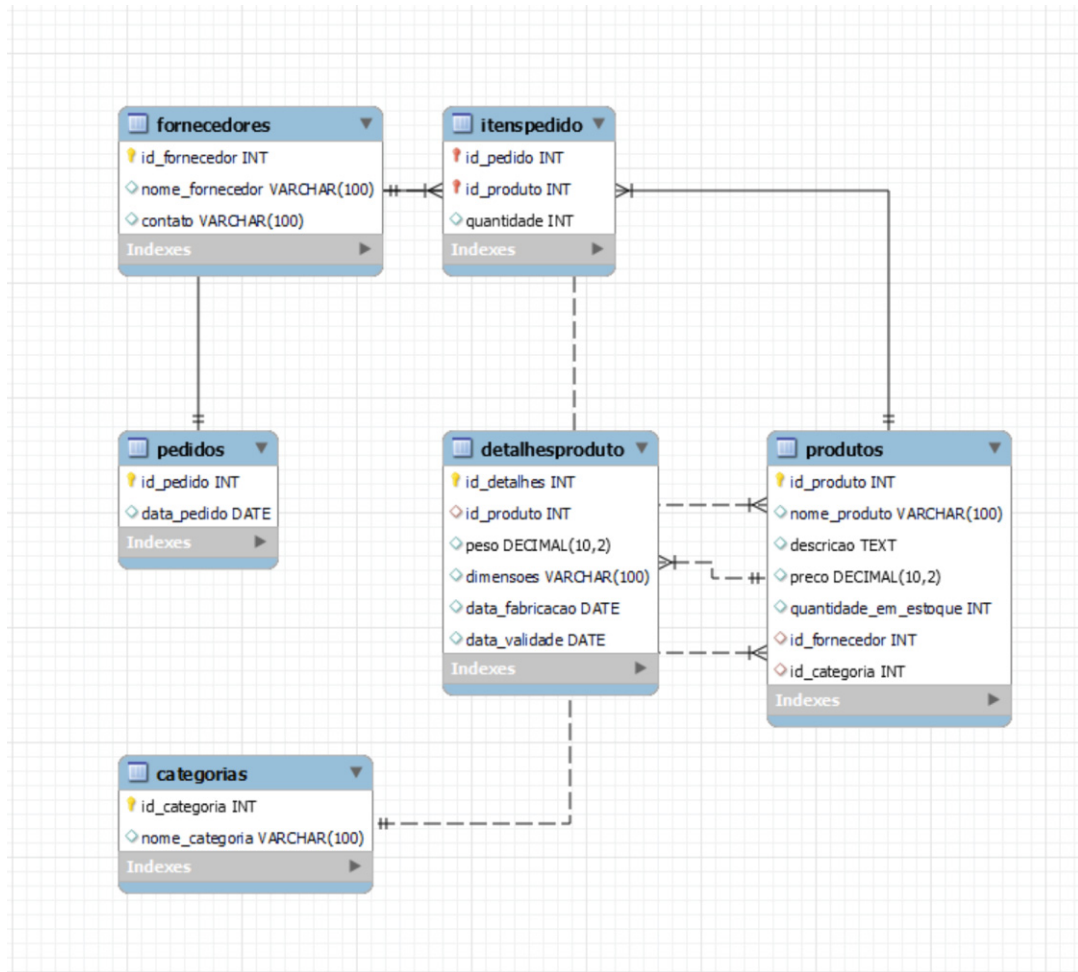
FONTE: O Autor (2025).

FIGURA 20 – MODELO LÓGICO QUESTÃO 1



FONTE: O Autor (2025).

FIGURA 21 – MODELO LÓGICO QUESTÃO 2



FONTE: O Autor (2025).

FIGURA 22 – SCRITPS SQL

Script do banco de dados:

```

CREATE TABLE Fomecedores (
  id_fomecedor INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nome_fomecedor VARCHAR(100) NOT NULL,
  contato VARCHAR(100) NOT NULL
)

CREATE TABLE Categorias (
  id_categoria INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nome_categoria VARCHAR(100) NOT NULL
)

CREATE TABLE Produtos (
  id_produto INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nome_produto VARCHAR(100) NOT NULL,
  descricao TEXT NOT NULL,
  preco DECIMAL(10, 2) NOT NULL,
  quantidade_em_estoque INT NOT NULL,
  id_fomecedor INT NOT NULL,
  id_categoria INT NOT NULL,
  FOREIGN KEY (id_fomecedor) REFERENCES Fomecedores(id_fomecedor)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (id_categoria) REFERENCES Categorias(id_categoria)
  ON DELETE SET NULL
  ON UPDATE CASCADE
)

CREATE TABLE Pedidos (
  id_pedido INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  data_pedido DATE NOT NULL
)

CREATE TABLE ItensPedido (
  id_pedido INT NOT NULL,
  id_produto INT NOT NULL,
  quantidade INT NOT NULL,
  PRIMARY KEY (id_pedido, id_produto),
  FOREIGN KEY (id_pedido) REFERENCES Pedidos(id_pedido)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  FOREIGN KEY (id_produto) REFERENCES Produtos(id_produto)
  ON DELETE CASCADE
  ON UPDATE CASCADE
)

CREATE TABLE DetalhesProduto (
  id_detalhes INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  id_produto INT NOT NULL UNIQUE,
  peso DECIMAL(10, 2) NOT NULL,
  dimensoes VARCHAR(100) NOT NULL,
  data_fabricacao DATE NOT NULL,
  data_validade DATE NOT NULL,
  FOREIGN KEY (id_produto) REFERENCES Produtos(id_produto)
  ON DELETE CASCADE
  ON UPDATE CASCADE
)

INSERT INTO Fomecedores (nome_fomecedor, contato) VALUES
('Fomecedor A', 'contatoA@example.com'),
('Fomecedor B', 'contatoB@example.com'),
('Fomecedor C', 'contatoC@example.com'),
('Fomecedor D', 'contatoD@example.com'),
('Fomecedor E', 'contatoE@example.com');

INSERT INTO Categorias (nome_categoria) VALUES
('Eletrônicos'),
('Móveis'),
('Roupas'),
('Alimentos'),
('Brinquedos');

INSERT INTO Produtos (nome_produto, descricao, preco, quantidade_em_estoque, id_fomecedor, id_categoria) VALUES
('TV 42"', 'Smart TV 42 polegadas', 1500.00, 50, 1, 1),
('Sofá', 'Sofá 3 lugares', 700.00, 20, 2, 2),
('Camiseta', 'Camiseta de algodão', 30.00, 100, 3, 3),
('Arroz 5kg', 'Pacote de arroz 5kg', 20.00, 200, 4, 4),
('Boneca', 'Boneca de pano', 40.00, 80, 5, 5);

INSERT INTO Pedidos (data_pedido) VALUES

```

FONTE: O Autor (2025).

7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

A disciplina de Aspectos Ágeis de Programação teve como foco principal a aplicação de boas práticas de codificação em consonância com os princípios do desenvolvimento ágil. Os conteúdos abordaram temas como refatoração, testes automatizados, versionamento com Git, princípios SOLID, programação orientada a objetos, TDD além de padrões de projeto e métricas de qualidade de software. Esses conceitos são essenciais para manter um código limpo, reutilizável, testável e adaptável, características fundamentais em ciclos iterativos e incrementais.

O projeto na disciplina consistiu na aplicação dos princípios de *clean code* em um algoritmo de ordenação Bubble Sort (Figura 23), com a exigência de pelo menos três alterações significativas no código original. Foram implementadas melhorias como renomear os métodos e variáveis para nomes mais descritivos, remoção de variáveis não utilizadas, alterações em condicionais booleanas e a utilização do *foreach* para impressão de *arrays*. Essas mudanças destacaram a importância da clareza, legibilidade e simplicidade no código-fonte, fatores que favorecem a manutenção e a evolução contínua do software.

A disciplina estabeleceu conexões diretas com outras unidades curriculares, como Introdução à Programação, ao reforçar os fundamentos da codificação estruturada e orientada a objetos, e com Testes Automatizados, ao antecipar práticas de validação e cobertura de código. Além disso, os princípios estudados foram aplicados nas fases de construção dos sistemas desenvolvidos nas disciplinas de desenvolvimento web, mobile e DevOps, contribuindo para entregas mais confiáveis e sustentáveis ao longo das sprints planejadas nas disciplinas de gerenciamento ágil. Assim, Aspectos Ágeis de Programação consolidou conhecimentos que permeiam todo o ciclo de desenvolvimento, sublinhando a importância da qualidade técnica como parte integrante da agilidade no processo de entrega de software.

ARTEFATOS DO PROJETO

FIGURA 23 – CODIGO BUBBLE SORT

```
import java.io.*;

class BubbleSort {

    static void sort(int[] array) {
        int length = array.length;
        for (int i = 0; i < length - 1; i++) {
            boolean trocado = false;
            for (int j = 0; j < length - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                    trocado = true;
                }
            }
            if (!trocado) {
                break;
            }
        }
    }

    static void printArray(int[] array) {
        for (int value : array) {
            System.out.print(value + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int[] array = {64, 34, 25, 12, 22, 11, 90};
        sort(array);
        System.out.println("Sorted array:");
        printArray(array);
    }
}
```

FONTE: O Autor (2025).

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

As disciplinas de Desenvolvimento Web 1 e 2 foram fundamentais para fixar os conceitos sobre a prática de desenvolvimento em projetos web, com ênfase na construção de sistemas completos baseados em frameworks modernos. Em WEB1, abordamos a criação de aplicações *front-end* com a utilização do angular, explorando conceitos como componentes, serviços, roteamento e manipulação de dados com persistência via *Local Storage*. O trabalho da disciplina consistiu na implementação de dois CRUDs (Alunos e Cursos) como mostra as Figuras 24 e 25 que foram criadas com Bootstrap.

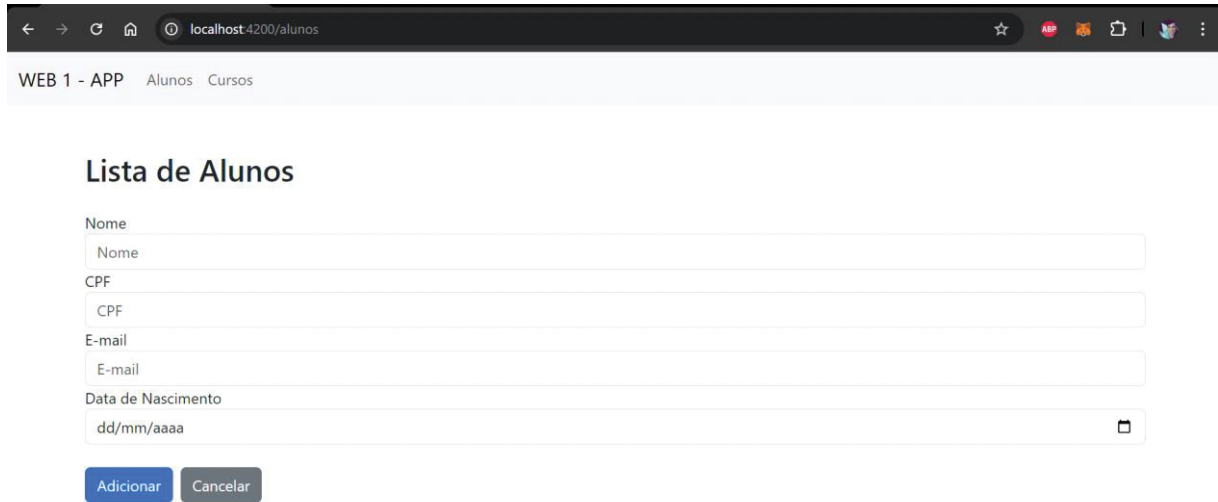
Em WEB2, o foco foi na inclusão do *back-end* em Java no mesmo projeto criado em WEB 1, utilizando o framework Spring Boot com persistência em banco de dados PostgreSQL. Além dos CRUDs de alunos e cursos, foi desenvolvido um CRUD para matrículas, relacionando as entidades. A arquitetura passou a ter uma separação mais clara entre as camadas, incluindo APIs REST para comunicação entre *front-end* e *back-end*.

Esses projetos proporcionaram vivência prática em construção de sistemas web reais, com atenção à usabilidade, organização de código e integração entre tecnologias. A estruturação em Angular e Spring Boot favoreceu a modularização e a escalabilidade da aplicação, alinhando-se às práticas recomendadas em ambientes ágeis.

A experiência obtida em WEB1 e WEB2 conectou-se diretamente com outras disciplinas do curso, como BD (pela modelagem e persistência), UX (pela interface e fluxo do usuário), DevOps (pela organização do projeto e boas práticas de entrega), além de Testes Automatizados e Aspectos Ágeis de Programação, que reforçaram a importância da qualidade do código e validação das funcionalidades. Assim, essas disciplinas contribuíram para o desenvolvimento de um sistema funcional completo, refletindo os princípios do desenvolvimento ágil de software.

ARTEFATOS DO PROJETO

FIGURA 24 – CRUD ALUNOS



← → ↻ 🏠 localhost:4200/alunos ☆ ASP 🐱 📄 | 🐦 ⋮

WEB 1 - APP Alunos Cursos

Lista de Alunos

Nome

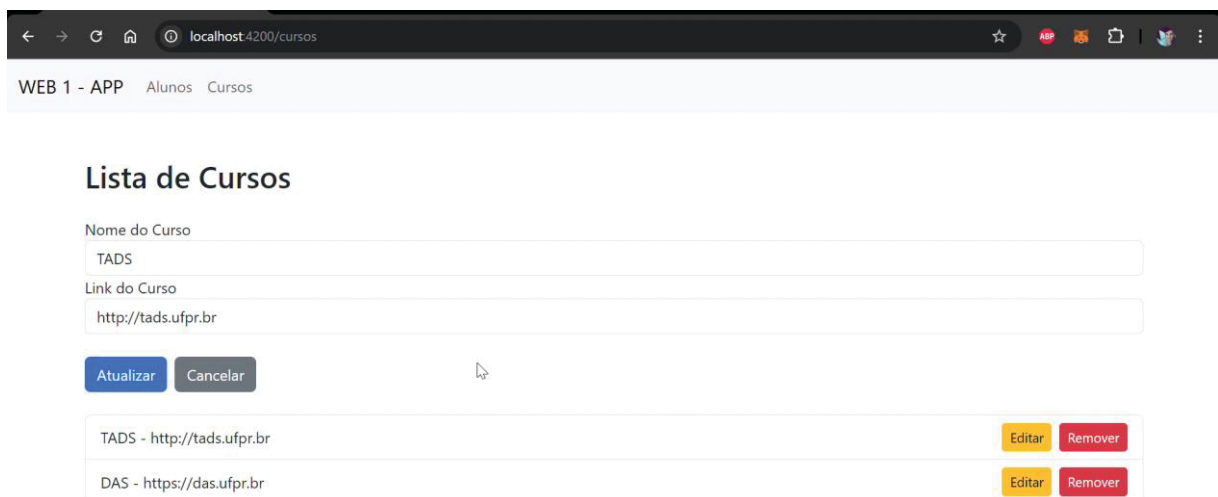
CPF

E-mail

Data de Nascimento
 📅

FONTE: O Autor (2025).

FIGURA 25 – CRUD CURSOS



← → ↻ 🏠 localhost:4200/cursos ☆ ASP 🐱 📄 | 🐦 ⋮

WEB 1 - APP Alunos Cursos

Lista de Cursos

Nome do Curso

Link do Curso

| | | |
|----------------------------|---------------------------------------|--|
| TADS - http://tads.ufpr.br | <input type="button" value="Editar"/> | <input type="button" value="Remover"/> |
| DAS - https://das.ufpr.br | <input type="button" value="Editar"/> | <input type="button" value="Remover"/> |

FONTE: O Autor (2025).

9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

A disciplina de UX no Desenvolvimento Ágil de Software teve como foco a integração da experiência do usuário ao processo de desenvolvimento de software, destacando a importância de projetar soluções centradas nas necessidades reais dos usuários. O conteúdo abordou conceitos e técnicas envolvidos no UX design como usabilidade, design de interação, arquitetura da informação, testes com usuários e prototipação, alinhando-se aos valores ágeis de entrega contínua de valor, validação com usuários e adaptação rápida com base em feedback.

Como projeto final, foi desenvolvido o protótipo de um site institucional de uma empresa de informática (Julio Informática), com o principal objetivo mostrar ao cliente as principais atividades da empresa, sua história e os contatos.

Foram elaboradas cinco telas no projeto: Tela Inicial (Figura 26), Serviços, Sobre Portifólio e Contato. Todos os aspectos visuais do site foram feitos seguindo as diretrizes abordadas na disciplina, como o tamanho e a cor da fonte, espaçamento das letras, hierarquia visual, adaptabilidade, *branding* entre outros.

A disciplina se integrou de forma direta com áreas como desenvolvimento web e gerenciamento de projetos, uma vez que os protótipos gerados serviram de base para implementação e planejamento técnico. Assim, UX contribuiu significativamente para a entrega de software orientado ao valor e centrado no usuário, alinhando-se aos princípios fundamentais do desenvolvimento ágil.

ARTEFATOS DO PROJETO

FIGURA 26 - SITE INSTITUCIONAL



FONTE: O Autor (2025).

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

As disciplinas de Desenvolvimento Mobile 1 e 2 (MOB1 e MOB2) tiveram como principal objetivo capacitar os alunos a projetar, desenvolver e testar aplicações móveis nativas utilizando a linguagem Kotlin e a plataforma Android. No contexto do desenvolvimento ágil de software, essas disciplinas mostraram-se essenciais ao promoverem a aplicação prática de conceitos como prototipação rápida, iteração contínua e entrega incremental de valor por meio de funcionalidades bem definidas.

Em MOB2, abordamos as requisições HTTP assíncronas, consumindo *endpoints* para buscar os dados, o uso de bibliotecas como Retrofit para chamadas HTTP e a importância da separação de responsabilidades entre camadas da aplicação, contribuindo com a compreensão de arquiteturas como MVC (*Model-View-Controller*) / MVVM (*Model-View-ViewModel*), que foi fundamental para integração com *back-ends*, como os abordados nas disciplinas de WEB 1, WEB 2 e Banco de Dados.

Os trabalhos das duas disciplinas eram opcionais, e optou-se por não os entregar.

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

A disciplina teve como foco principal a aplicação dos conceitos de DevOps no contexto do desenvolvimento ágil, promovendo a integração entre desenvolvimento, entrega contínua e infraestrutura automatizada. Os conteúdos abordados trataram de virtualização, contêineres, versionamento de infraestrutura, automação de pipelines e integração contínua, pilares fundamentais para ciclos de entrega curtos, seguros e sustentáveis.

O projeto prático da disciplina consistiu na implantação de um ambiente integrado com GitLab e Jenkins a partir da imagem `dfwandarti/gitlab_jenkins:3` executadas via Docker. O exercício incluiu a criação de um container nomeado conforme a matrícula do aluno Figura (27), com publicação das portas 22, 80, 443 e 9091, além da autenticação no GitLab via root e recuperação da senha inicial dentro do container.

Essa atividade promoveu o entendimento prático de automação e versionamento, fundamentais na cultura DevOps e também o contato direto com ferramentas amplamente utilizadas em ambientes ágeis, como Docker, GitLab e Jenkins, reforçando práticas de CI/CD (integração e entrega contínua). Além disso, possibilitou a reflexão sobre automação de ambientes, gerenciamento de pipelines e versionamento de código, temas que impactam diretamente a confiabilidade e velocidade de entrega dos produtos desenvolvidos.

ARTEFATOS DO PROJETO

FIGURA 27 - DOCKER

The screenshot displays the Docker Desktop interface. The top navigation bar includes the Docker logo, a search bar, and system icons. A notification banner at the top reads "More Docker. Easy Access. New Streamlined Plans. [Learn more](#)".

The left sidebar contains navigation options: Containers (selected), Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions.

The main area is titled "Containers" and shows a table of running containers:

| Name | Container ID | Image | Port(s) | CPU ... | Last ... | Actions |
|-------------|--------------|-----------------------------|---|---------|----------------|---------------------------|
| GRR20154017 | 7454165acf93 | dfwandarti/gitlab_jenkins:3 | 22:22 Show all ports (4) | 84.85% | 53 seconds ago | [Stop] [Refresh] [Delete] |

Below the table is a "Terminal" window showing the command used to start the container and its output:

```
PS C:\Users\Julio Mariotto> docker run -d --name GRR20154017 -p 22:22 -p 80:80 -p 443:443 -p 9091:9091 dfwandarti/gitlab_jenkins:3
7454165acf93382f496e73a831b85975583bd90eb7924ca0854c81781206f98
PS C:\Users\Julio Mariotto> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS
7454165acf93   dfwandarti/gitlab_jenkins:3        "/assets/wrapper"       14 seconds ago  Up 13 seconds (health: starting)  0.0.0.0:22->22/tcp, 0
.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:9091->9091/tcp
GRR20154017
PS C:\Users\Julio Mariotto>
```

The bottom status bar indicates "Engine running" and system resource usage: RAM 2.30 GB, CPU 12.33%, Disk 7.45 GB used (limit 1006.85 GB). The terminal window title is "Terminal" and the version is "v4.38.0".

FONTE: O Autor (2025).

12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

A disciplina de Testes Automatizados teve como objetivo central consolidar práticas de verificação e validação de software, fundamentais para garantir qualidade, confiabilidade e robustez em ciclos de desenvolvimento ágil. O conteúdo abordado introduziu os principais conceitos de testes em diferentes níveis (unitário, integração, sistema e aceitação), além de técnicas como TDD (*Test-Driven Development*) e BDD (*Behavior-Driven Development*), com foco na automação contínua do processo de testes.

O projeto da disciplina consistiu na criação de um script automatizado, utilizando a biblioteca Puppeteer, capaz de acessar a plataforma Anotepad, preencher automaticamente o título e o conteúdo de uma nota, e em seguida encerrar a execução. O script foi desenvolvido em JavaScript, e demonstrou a habilidade de interagir com elementos da interface gráfica do navegador, simulando o comportamento de um usuário real (Figura 28).

A relevância desse exercício prático está na consolidação do conceito de testes de interface (E2E – *end-to-end*) e na aplicação de automação em processos de validação de requisitos. Em um contexto de desenvolvimento ágil, esse tipo de abordagem reduz falhas manuais, acelera a entrega e contribui diretamente para o feedback rápido e contínuo, promovendo maior confiança nas funcionalidades implementadas.

Além disso, a disciplina apresentou integração significativa com outras áreas do curso, como Programação, DevOps, Web, Mobile e UX, pois a automação de testes foi aplicada tanto no *back-end* quanto no *front-end*, reforçando a importância de um pipeline de entrega completo e confiável. Assim, a disciplina de Testes Automatizados consolidou-se como uma etapa essencial na garantia da qualidade em um fluxo de desenvolvimento ágil e iterativo.

ARTEFATOS DO PROJETO

FIGURA 28 – CODIGO JAVA SCRIPT

```
1  const puppeteer = require('puppeteer');
2
3  (async () => {
4
5      const browser = await puppeteer.launch({ headless: false });
6      const page = await browser.newPage();
7
8      await page.goto('https://pt.anothepad.com/');
9
10
11     await page.type('#edit_title', 'Entrega trabalho TEST DAS 2024');
12
13
14     await page.type('#edit_textarea', `Nome: Julio Cesar Mariotto Junior\nMatrícula: GRR20154017`);
15
16
17     await new Promise(resolve => setTimeout(resolve, 60000));
18
19     await browser.close();
20 })();
```

FONTE: O Autor (2025).

13 CONCLUSÃO

Este memorial resume a experiência adquirida durante a Especialização em Desenvolvimento Ágil de Software, destacando a aplicação prática dos conceitos abordados em cada disciplina. A conexão entre teoria e prática foi constante, abrangendo desde os fundamentos dos métodos ágeis até a entrega contínua de soluções funcionais, incluindo áreas como modelagem, programação, UX, testes, DevOps, gerenciamento de projetos, bancos de dados e desenvolvimento web e mobile.

A disciplina de Métodos Ágeis estabeleceu as bases conceituais que orientaram os projetos, enquanto as outras disciplinas complementaram esse conjunto com ferramentas e técnicas essenciais para o desenvolvimento ágil. A experiência prática possibilitou compreender não apenas os aspectos técnicos, mas também os desafios humanos e organizacionais na implementação de uma cultura ágil.

Entre os principais desafios enfrentados, destacam-se a necessidade de mudança de mentalidade nas equipes, a disciplina na execução dos rituais, a integração entre as áreas e o domínio das ferramentas de automação. Superar esses obstáculos exigiu comunicação eficaz, colaboração contínua e foco na entrega de valor, fatores que se mostraram cruciais para o êxito dos projetos.

Dessa forma, esse percurso formativo não apenas consolidou conhecimentos técnicos, mas também reforçou a importância da agilidade como uma abordagem estratégica para enfrentar a complexidade e a constante evolução no desenvolvimento de software. Concluir este ciclo é reconhecer que o aprendizado contínuo é o verdadeiro motor da inovação, e que ser ágil é, acima de tudo, estar pronto para a mudança.

REFERÊNCIAS

BECK et al. Manifesto Ágil, 2011. Disponível em <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 10 out. 2025.

CAROLI, P. **Lean Inception**: como alinhar pessoas e construir o produto certo. São Paulo: Editora Caroli, 2018.

GRANT, Will. **UX Design**: guia definitivo com as Melhores Práticas de UX. São Paulo: Novatec, 2019.

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

WANG, Rosemary. **Infrastructure as Code, Patterns and Practices**. Nova Iorque: Manning Publications, 2022.