

UNIVERSIDADE FEDERAL DO PARANÁ

RAHUANA MARIA RIBEIRO FUJIOKA

MEMORIAL DE PROJETOS: DESENVOLVIMENTO DE SOFTWARE
INTEGRADO COM AGILIDADE

CURITIBA

2025

RAHUANA MARIA RIBEIRO FUJIOKA

MEMORIAL DE PROJETOS: DESENVOLVIMENTO DE SOFTWARE
INTEGRADO COM AGILIDADE

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

Orientador: Prof. Dr. Razer Anthom Nizer Rojas Montaña

CURITIBA

2025

RESUMO

Este memorial apresenta os trabalhos desenvolvidos ao longo do curso de Especialização em Desenvolvimento Ágil de Software da UFPR. Sendo organizado de acordo com a ordem das disciplinas, mostra a conexão do conteúdo com o desenvolvimento ágil de software, unindo a teoria à prática. Iniciando com o entendimento dos requisitos, planejamento, apresentando os diagramas desenvolvidos para embasar o desenvolvimento, além da compreensão dos conceitos e melhores práticas de programação, de infraestrutura e de testes, levando ao desenvolvimento ágil de software. O objetivo é apresentar os trabalhos desenvolvidos e a ligação entre os temas apresentados nas disciplinas. Em cada seção estão incluídos artefatos como diagramas, tabelas, listas de critérios, imagens das atividades, trechos de códigos e de testes entregues durante a realização das disciplinas.

Palavras-chave: Métodos ágeis, desenvolvimento de software, gerenciamento de projetos, processos de software.

ABSTRACT

This memorial presents the projects developed throughout the Specialization Course in Agile Development at UFPR. Organized according to the sequence of subjects, it demonstrates the connection between the course content and the agile software development, linking theory and practice. Beginning with the requirements understanding, planning, and presenting the diagrams created to support the development process, along with the comprehension of concepts and best practices in programming, infrastructure and testing, leading to the agile development. The objective is to showcase the projects and the relation between the topics studied during the course. Artifacts, such as diagrams, tables, criteria lists, activity images, code excerpts, and test scripts delivered during the coursework are included in each section.

Keywords: Agile methods, software development, project management, software process

SUMÁRIO

1 PARECER TÉCNICO.....	8
2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....	10
2.1 ARTEFATOS DO PROJETO.....	10
3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2..	16
3.1 ARTEFATOS DO PROJETO – MAG1.....	16
3.1.1 DIAGRAMA DE CASO DE USO DE NÍVEL 1	16
3.1.2 DIAGRAMA DE CASO DE USO DE NÍVEL 2	17
3.1.3 HISTÓRIAS DE USUÁRIO.....	17
3.2 ARTEFATOS DO PROJETO – MAG2.....	19
3.2.1 DIAGRAMA DE CLASSES.....	19
3.2.2 DIAGRAMA DE SEQUÊNCIA	20
4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2	23
4.1 ARTEFATOS DO PROJETO – GAP 1	23
4.2 ARTEFATOS DO PROJETO – GAP 2	25
5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....	27
5.1 ARTEFATOS DO PROJETO.....	27
6 DISCIPLINA: BD – BANCO DE DADOS.....	29
6.1 ARTEFATOS DO PROJETO.....	29
6.1.1 MODELO ENTIDADE-RELACIONAMENTO CONCEITUAL	29
6.1.2 MODELO LÓGICO (DIAGRAMA ENTIDADE RELACIONAMENTO)	31
6.1.3 MODELO EER	33
6.1.4 SCRIPT	33
6.1.5 SELECT TABELAS	35
7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO	36
7.1 ARTEFATOS DO PROJETO.....	36
7.1.1 CÓDIGO BUBBLE SORT REFATORADO	36
8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2	38
8.1 ARTEFATOS DO PROJETO.....	38
9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE.....	40

9.1 ARTEFATOS DO PROJETO.....	40
9.1.1 TELAS:	40
9.1.2 ESCOLHAS FEITAS:	44
10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....	45
10.1 ARTEFATOS DO PROJETO.....	45
11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)	48
11.1 ARTEFATOS DO PROJETO.....	48
12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS	51
12.1 ARTEFATOS DO PROJETO.....	51
12.1.1 CÓDIGO UTILIZADO NO PLAYWRIGHT	51
13 CONCLUSÃO	52
REFERÊNCIAS.....	53

1 PARECER TÉCNICO

Este parecer técnico apresenta uma análise dos projetos desenvolvidos durante a realização do curso, iniciando com disciplinas que fundamentaram o conhecimento teórico das metodologias ágeis, modelagem de dados e análise de requisitos, seguidas dos conteúdos práticos como desenvolvimento de software, infraestrutura e testes.

A disciplina de Métodos Ágeis para Desenvolvimento de Software (MADS) foi fundamental para o entendimento dos princípios do Manifesto Ágil (Highsmith, 2002) e suas aplicações práticas nas demais disciplinas, além da introdução aos métodos ágeis, como Scrum (Schwaber e Sutherland, 2020), Kanban, XP, entre outros (Cohn, 2011) e as características de cada um. Foram adquiridos conhecimentos de modelagem nas disciplinas de MAG1 e MAG2 que contribuíram significativamente para o entendimento da documentação que é realmente necessária para um projeto ágil, sendo criados os diagramas da UML que facilitam o entendimento da solicitação e descoberta de gargalos ainda no planejamento, proporcionando mais clareza para o desenvolvimento. Na disciplina de Gerenciamento Ágil de Projetos de Software – GAP1 foram abordadas as diferenças entre projeto e projeto de software e também entre a gestão tradicional de projetos, baseada no PMBOK (PMI, 2017) e a gestão ágil de projetos, que é mais adaptável. Também foi estudado sobre a utilização da ferramenta de quadro Kanban (Poppendieck, 2003) como auxílio para melhorar a visibilidade e transparência das atividades realizadas, além de possibilitar a limitação de trabalho em paralelo, reforçando a frase “pare de começar e comece a terminar” (Kanban University, The Official Kanban Guide, p. 12), com foco na conclusão das atividades antes do início das próximas.

A seguir, foram iniciadas as disciplinas práticas, como Introdução à Programação, Desenvolvimento Web e Mobile, Banco de Dados e Aspectos Ágeis de Programação, nas quais foram apresentados conteúdos relevantes e integrados com ferramentas modernas, utilizando Angular, Spring Boot e técnicas como Test-Driven-Development (Ambler, 2013), proporcionando mais responsividade das aplicações e aplicando a teoria Clean Code, para facilitar a manutenção. Além destas, na disciplina de UX, foram aplicados os princípios de design e conceitos psicológicos para a criação de protótipos e o entendimento da importância de

entender a jornada e receber *feedback* dos usuários para que o desenvolvimento seja iterativo e as correções rápidas.

Para finalizar, foram aplicados os conhecimentos adquiridos nas disciplinas de Infraestrutura para Desenvolvimento e Implantação de Software (INFRA) e Testes Automatizados (TEST), nas quais foram reforçados os conceitos de DevOps e seus atos e automação. Foi entendido sobre o ciclo de vida do desenvolvimento, incluindo fases desde o planejamento até a implantação e manutenção do software, focando na segurança, qualidade e *deploy* contínuo. Dessa forma, foi possível visualizar a necessidade da integração entre a teoria e a prática durante todo o ciclo de desenvolvimento do software, aplicando os conceitos das metodologias ágeis.

O título deste trabalho enfatiza os conceitos compreendidos durante a realização do curso, como a importância da integração entre as metodologias ágeis, entre si, como a associação entre o Scrum e Kanban para melhorar a visualização das atividades e com o restante do ciclo de desenvolvimento. As fases iniciais, de planejamento e análise de requisitos são fundamentais para que o desenvolvimento ágil ocorra, pois nelas são realizadas o refinamento da solicitação, para que o resultado do desenvolvimento seja satisfatório para o usuário, promovendo clareza para os envolvidos, além da melhoria contínua.

Assim, o conteúdo abordado durante o curso proporciona ao aluno o preenchimento das lacunas teóricas e práticas para que seja executado o desenvolvimento de software de forma integrada e contínua, além de enfatizar a colaboração entre os interessados, desde o solicitante ao tester.

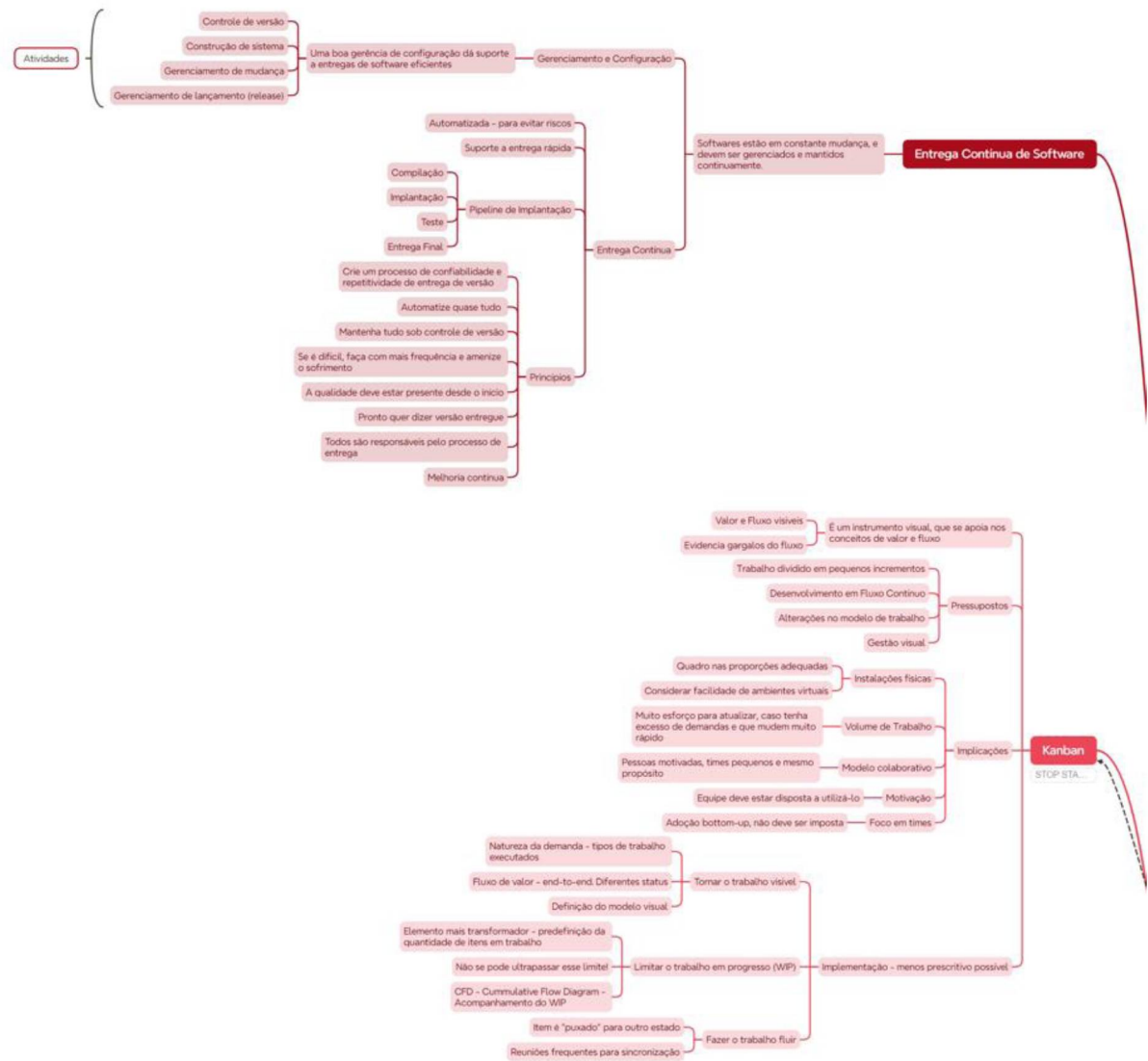
2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

O projeto proposto para esta disciplina foi um mapa mental, abordando os conteúdos apresentados: Processo de Software, Modelos Tradicionais de Processo de Software, Manifesto Ágil, Princípios Ágeis, Lean Software Development (Poppendieck, 2003), Scrum, Extreme Programming, Kanban, Entrega Contínua de Software (Cohn, 2011) e contendo o detalhamento de cada um. Foram detalhados os conceitos e listados processos e princípios de cada tópico, possibilitando o embasamento teórico para as disciplinas posteriores.

O conteúdo é importante para vincular conhecimentos teóricos com o que é aplicado na prática nos processos e projetos de desenvolvimento de software, tornando a atuação no mercado em conformidade com a metodologia ágil. Além disso, foram compartilhadas experiências de como funcionam e foram implementadas as metodologias nas empresas em que os acadêmicos trabalham, expondo os diversos modos de trabalho proporcionados por esta área.

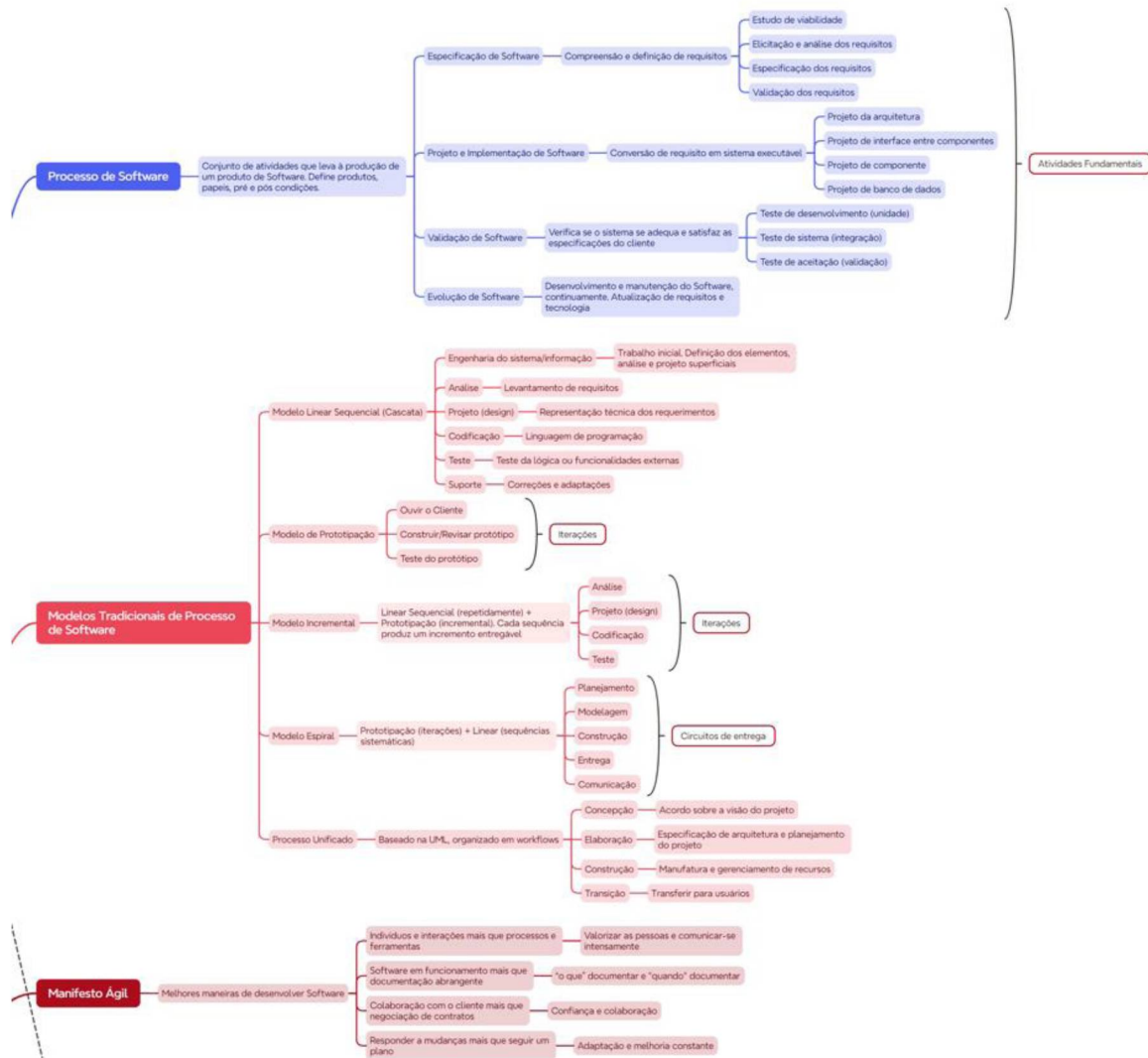
2.1 ARTEFATOS DO PROJETO

Figura 1 - Mapa mental MADS parte 1



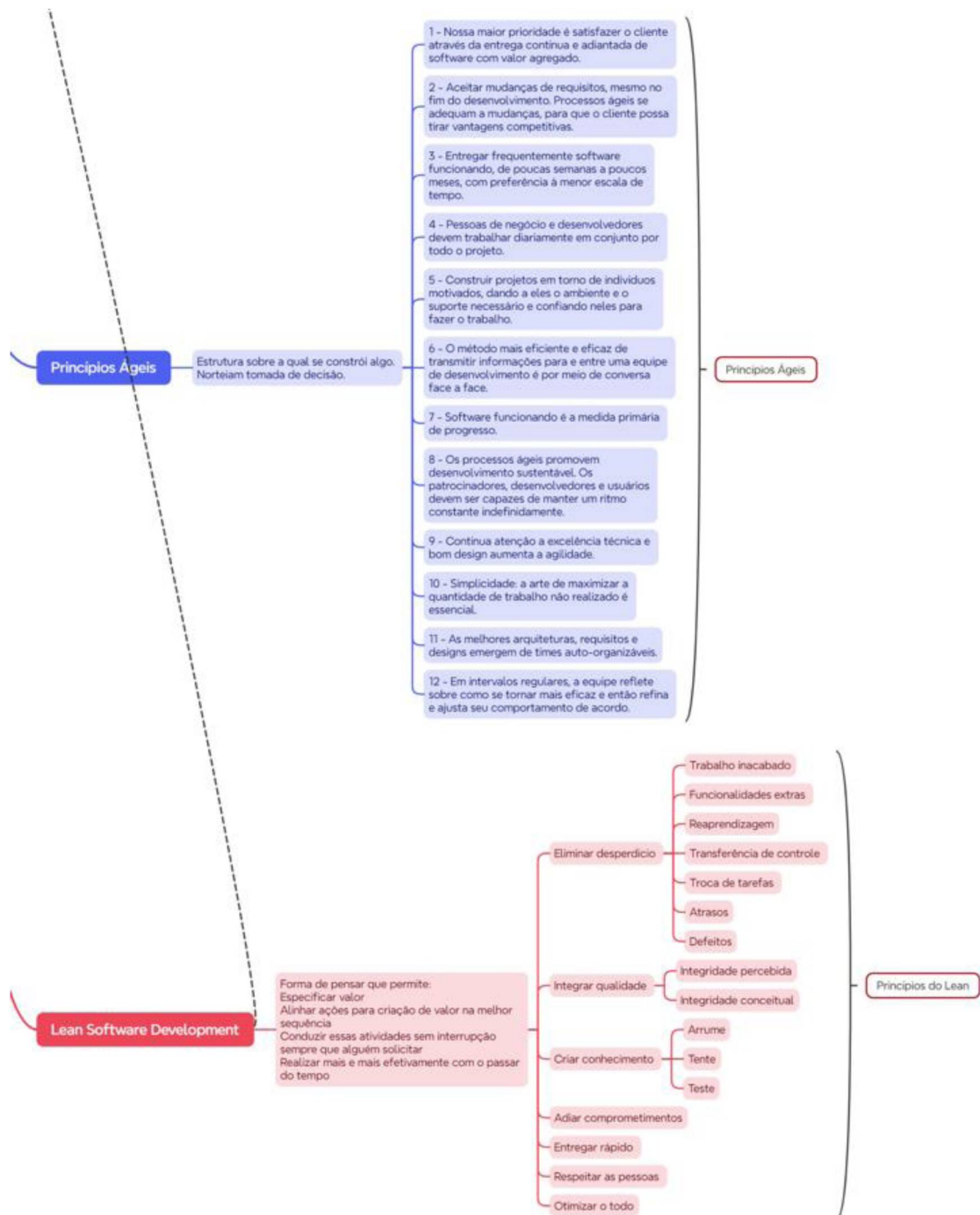
FONTE: A autora (2025)

Figura 2 - Mapa mental MADS parte 2



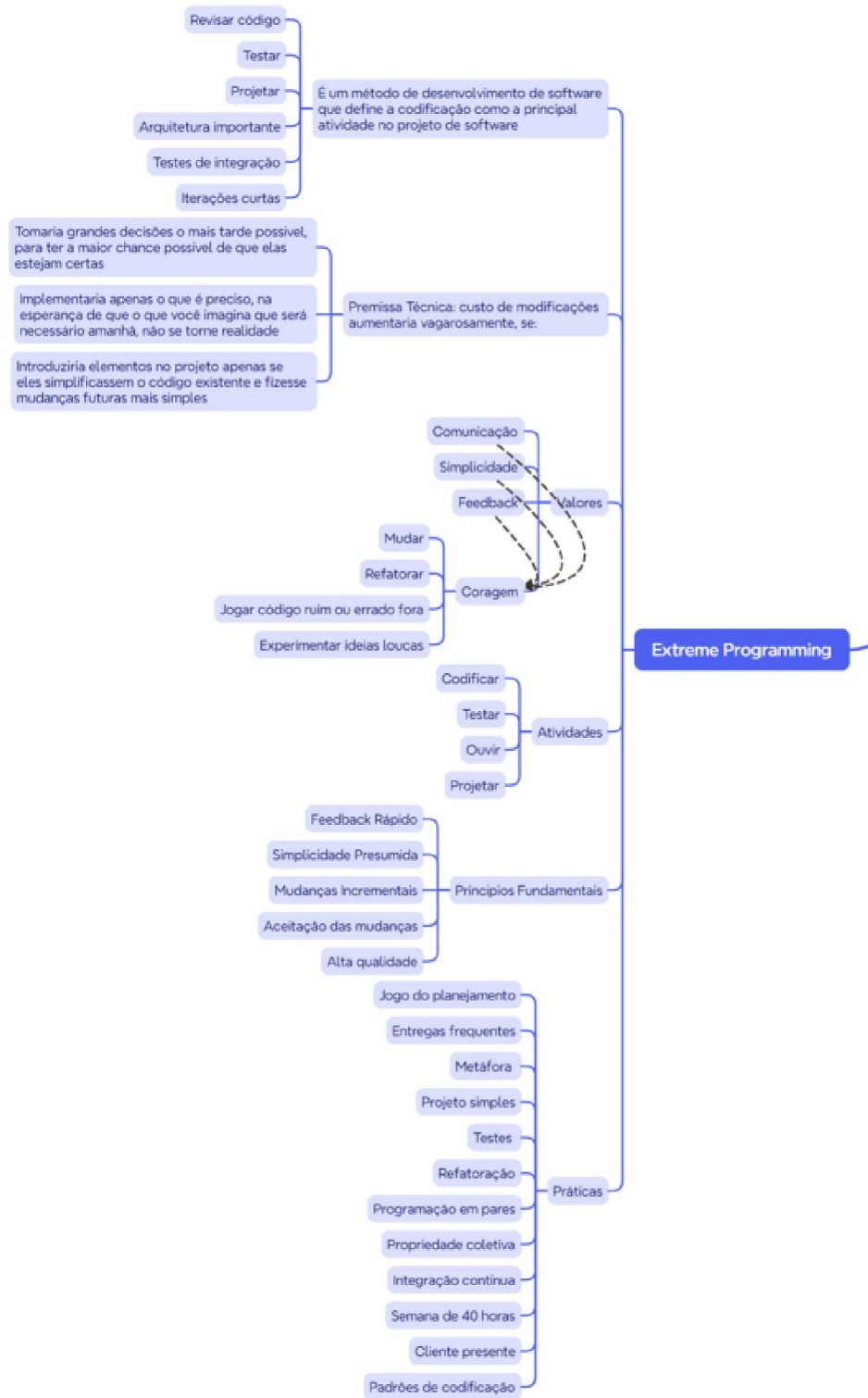
FONTE: A autora (2025)

Figura 3 - Mapa mental MADS parte 3



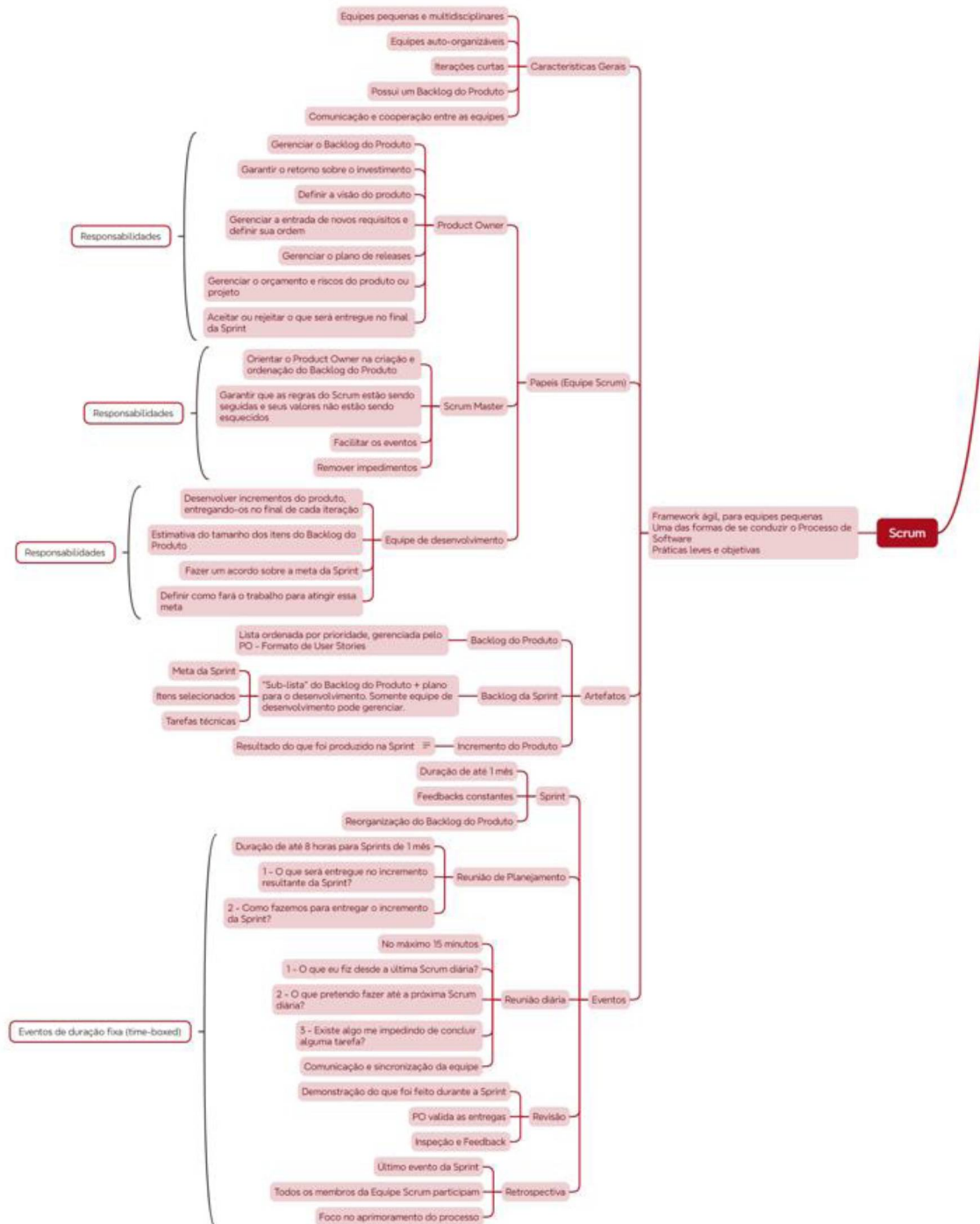
FONTE: A autora (2025)

Figura 4 - Mapa mental MADS parte 4



FONTE: A autora (2025)

Figura 5 - Mapa mental MADS parte 5



FONTE: A autora (2025)

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

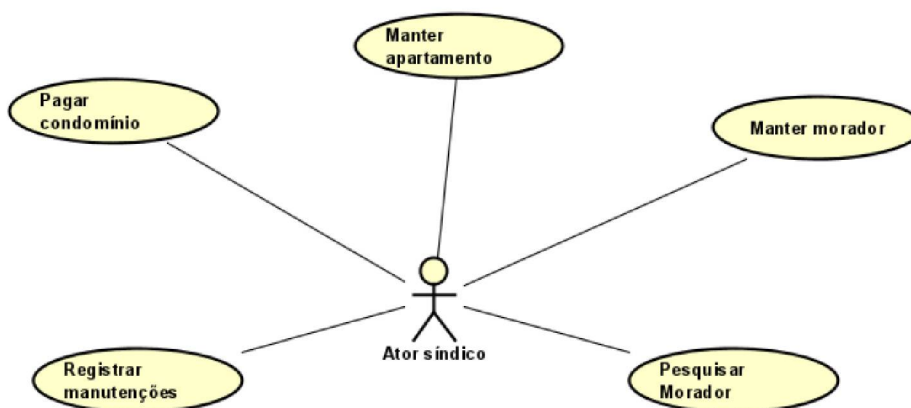
Nas disciplinas de Modelagem Ágil de Software 1 e 2, foi realizada a modelagem de um sistema de gestão de condomínio, em que foram criados os artefatos necessários para basear o desenvolvimento do software. Em MAG1, foram produzidos os diagramas de casos de uso, níveis 1 e 2 e as histórias de usuário, com os devidos critérios de aceitação. Sucedido por MAG2, em que foram produzidos os Diagramas de Classes e de Sequência (Starr, 2008), fundamentais para o entendimento dos requisitos e mantendo a documentação essencial para o projeto. Assim, foi possível definir as regras de negócio e relacionamento entre artefatos, permitindo uma abordagem transparente para o desenvolvimento do software e assegurando as informações relevantes para os testes.

Foi possível evidenciar a necessidade de uma documentação enxuta, mas essencial para que a etapa de programação seja feita com maior assertividade. Dessa forma, é possível visualizar a importância de um planejamento estruturado para que o resultado do desenvolvimento atenda às expectativas e erros sejam evitados, demonstrando como as disciplinas estão diretamente relacionadas.

3.1 ARTEFATOS DO PROJETO – MAG1

3.1.1 DIAGRAMA DE CASO DE USO DE NÍVEL 1

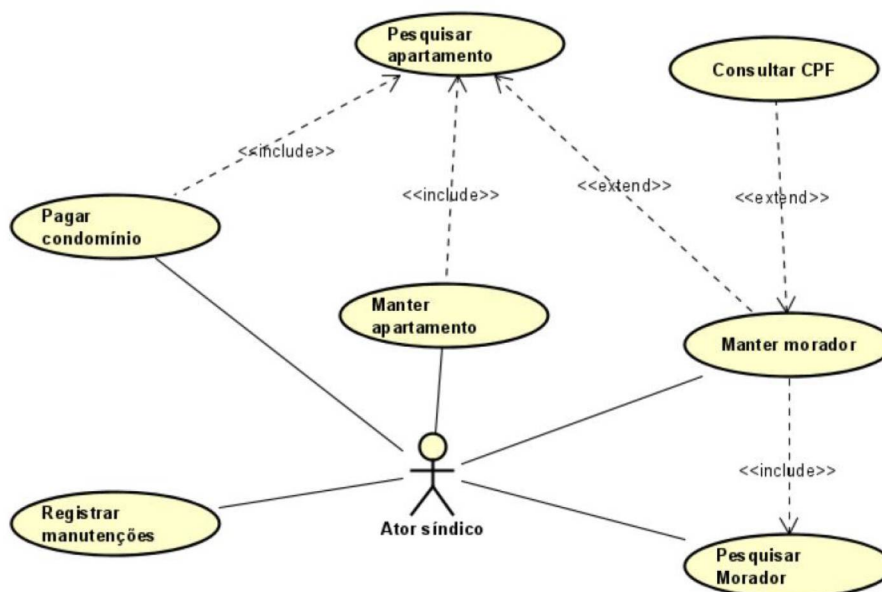
Figura 6 – Diagrama de Caso de Uso de Nível 1



FONTE: A autora (2025)

3.1.2 DIAGRAMA DE CASO DE USO DE NÍVEL 2

Figura 7 – Diagrama de Caso de Uso de Nível 2



FONTE: A autora (2025)

3.1.3 HISTÓRIAS DE USUÁRIO

HU001 – Gestão de Condomínio

SENDO O Síndico

QUERO Um sistema para gerenciamento de condomínio

PARA Automatizar os serviços

LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve apresentar a tabela com os moradores cadastrados
- 2) Deve permitir incluir um novo morador
- 3) Deve permitir alterar os dados de um morador cadastrado
- 4) Deve permitir excluir um morador cadastrado
- 5) Deve permitir consultar os dados de um morador cadastrado
- 6) Deve pesquisar os moradores na tabela da tela
- 7) Deve permitir gerenciar os apartamentos
- 8) Deve permitir gerenciar os pagamentos
- 9) Deve permitir cadastrar manutenções

HU002 – Manter Morador**SENDO o SÍNDICO**

QUERO manter os dados dos moradores

PARA que seus dados fiquem atualizados

LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve receber o parâmetro da tela de pesquisa e configurar a tela
- 2) Deve voltar à tela anterior
 - Critérios exclusivos para (Parâmetros Novo e Alterar)
- 3) Não deve prosseguir com CPF inconsistente
- 4) Não deve prosseguir com NOME em branco
- 5) Não deve prosseguir com Número do Apartamento inexistente
- 6) Deve preencher as informações RESPONSÁVEL e PROPRIETÁRIO
- 7) Deve salvar os dados do morador

HU003 – Manter Apartamento**SENDO o SÍNDICO**

QUERO manter as informações de apartamentos

PARA que seja possível associar com os dados do morador

LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve receber o parâmetro da tela de pesquisa e configurar a tela
- 2) Deve voltar à tela anterior
- 3) Não deve prosseguir com campos vazios
- 4) Não deve prosseguir caso o apartamento informado já exista no banco de dados
- 5) Deve salvar os dados do apartamento

HU004 – Pagar condomínio**SENDO o SÍNDICO**

QUERO manter as informações de pagamentos

PARA que seja possível visualizar inadimplências

LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve apresentar a tela com a tabela de pagamentos
- 2) Deve apresentar a tela com a tabela de pagamentos conforme número do apartamento pesquisado
- 3) Não deve salvar o registro se houver pagamentos em aberto
- 4) Deve voltar à tela anterior
- 5) Deve salvar os dados do pagamento

HU005 – Registrar Manutenções

SEND O SÍNDICO

QUERO registrar as manutenções do condomínio

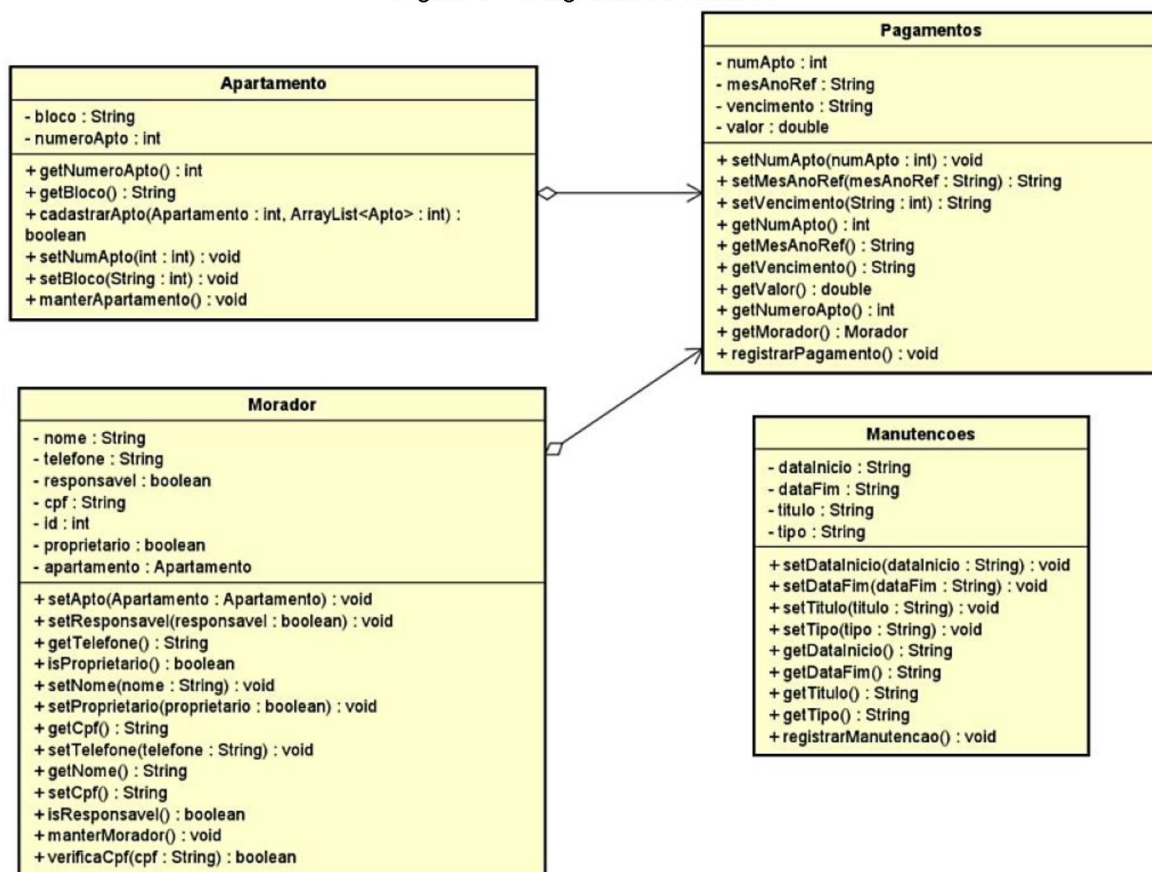
PARA que seja possível manter um histórico e informar os moradores

LISTA DE CRITÉRIOS DE ACEITAÇÃO

- 1) Deve apresentar uma tabela contendo o histórico de manutenções
- 2) Deve apresentar a tela com os campos em branco
- 3) Não deve salvar o registro se houver campos vazios
- 4) Deve listar os tipos de manutenções, conforme R2
- 5) Deve voltar à tela anterior
- 6) Deve salvar as informações da manutenção

3.2 ARTEFATOS DO PROJETO – MAG2**3.2.1 DIAGRAMA DE CLASSES**

Figura 8 – Diagrama de Classes

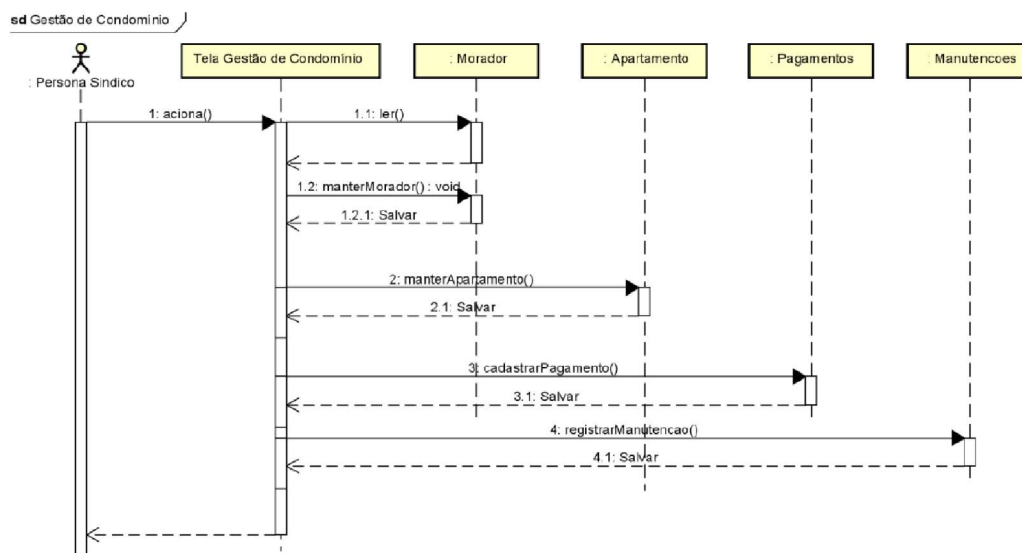


FONTE: A autora (2025)

3.2.2 DIAGRAMA DE SEQUÊNCIA

HU001 - Gestão de Condomínio

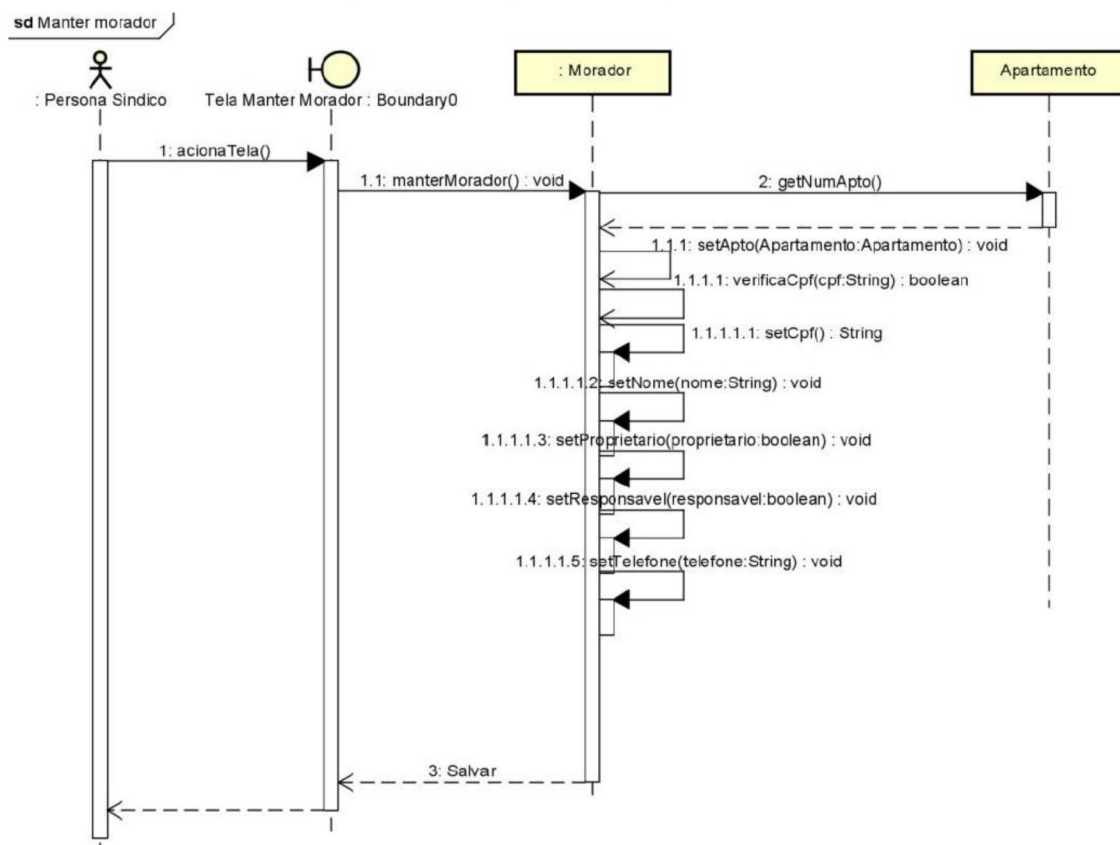
Figura 9 – Diagrama de Sequência HU001



FONTE: A autora (2025)

HU002 – Manter Morador

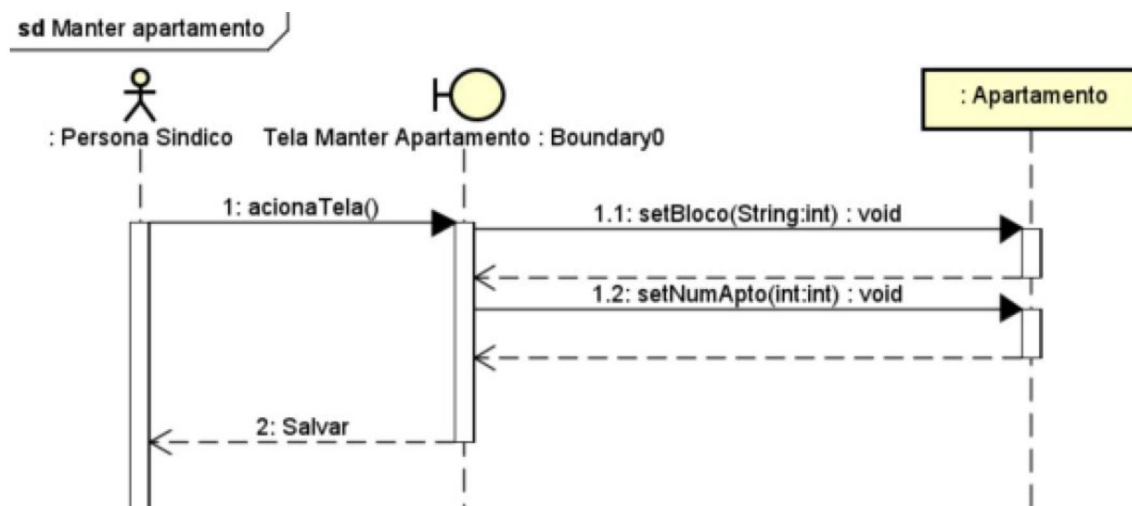
Figura 10 – Diagrama de Sequência HU002



FONTE: A autora (2025)

HU003 – Manter Apartamento

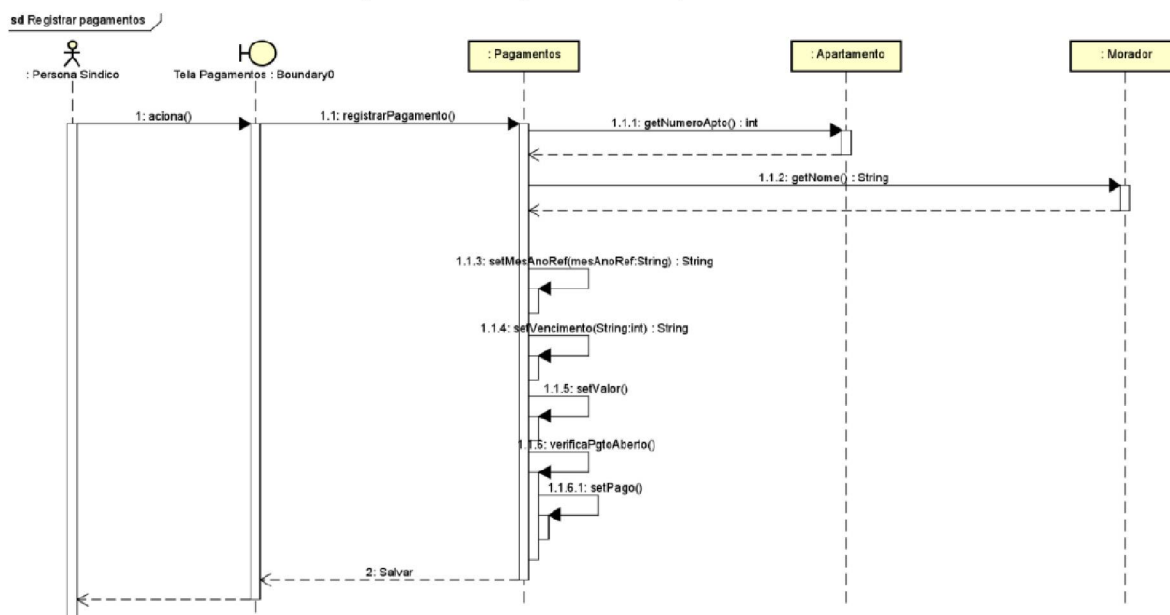
Figura 11 – Diagrama de Sequência HU003



FONTE: A autora (2025)

HU004 – Pagar condomínio

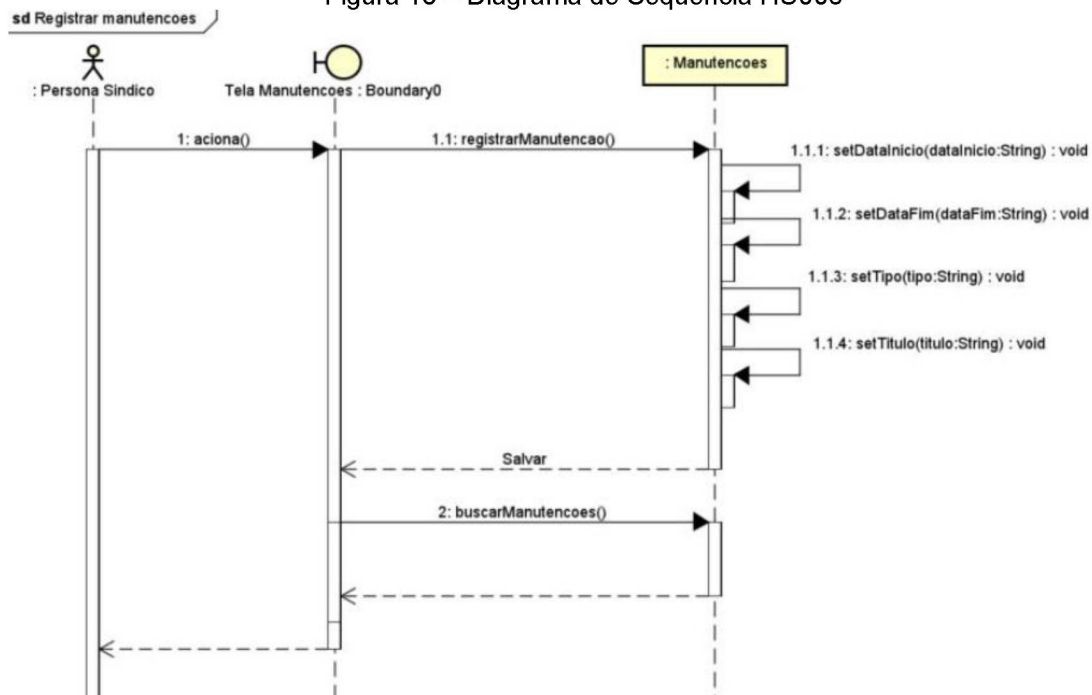
Figura 12 – Diagrama de Sequência HU004



FONTE: A autora (2025)

HU005 – Registrar Manutenções

Figura 13 – Diagrama de Sequência HU005



FONTE: A autora (2025)

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

O projeto desenvolvido para a disciplina de Gerenciamento Ágil de Projetos Software 1 consistia em um plano de release para um aplicativo de organização residencial, tema escolhido pelo aluno. Foi realizado o cálculo de velocidade, elaborada a tabela de release contendo datas e estimativas para cada *sprint*.

Para a disciplina de Gerenciamento Ágil de Projetos Software 2, foi realizada a simulação de gestão de projetos, utilizando um jogo online para montagem do Kanban (Kanban University, 2021), demonstrando a importância de se ter visualização das atividades e de finalizá-las antes de iniciar as próximas e evidenciando o conceito de 'puxar' as demandas para cada coluna de desenvolvimento. A atividade foi finalizada com a análise do Diagrama de Fluxo Cumulativo (CFD) (Kanban University, 2021), proporcionando a visualização dos resultados.

As disciplinas foram essenciais para o entendimento dos conceitos e aplicações práticas do conteúdo, aperfeiçoando o conhecimento e servindo de base para o planejamento de atividades das próximas etapas do desenvolvimento de software.

4.1 ARTEFATOS DO PROJETO – GAP 1

Aplicativo para organização residencial.

Cálculo da velocidade:

Horas disponíveis por dia:	8	Tamanho da <i>sprint</i> :	2 semanas (10 dias úteis)
Horas disponíveis por <i>sprint</i> :	80	Velocidade:	10 pontos/ <i>sprint</i>

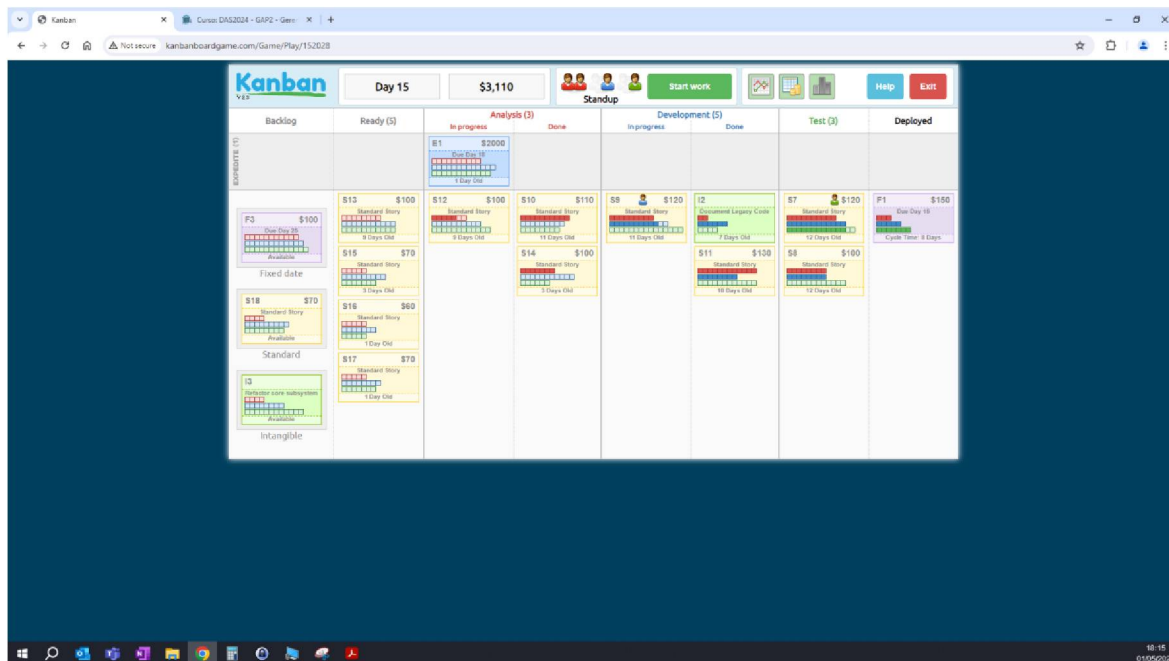
Plano de Release:

Iteração/ <i>Sprint</i> 1	Iteração/ <i>Sprint</i> 2	Iteração/ <i>Sprint</i> 3
Data Início: 01/04/2024	Data Início: 15/04/2024	Data Início: 29/04/2024
Data Fim: 12/04/2024	Data Fim: 26/04/2024	Data Fim: 10/05/2024
<HU0001 – Tela de Login> SENDO a principal usuária do sistema QUERO uma tela PARA fazer login no aplicativo	<HU0006 – Inserir filtro na lista de tarefas> SENDO usuária do sistema QUERO um filtro na lista de tarefas	<HU0011 – Tela de Relatório> SENDO administradora do sistema QUERO uma tela PARA visualizar relatórios ESTIMATIVA: 0.5

ESTIMATIVA: 1	PARA visualizar somente as tarefas de determinada categoria ou usuário responsável ESTIMATIVA: 2	
<HU0002 – Tela de Administração> SENDO a administradora do sistema QUERO uma tela PARA cadastrar usuários e alterar permissões ESTIMATIVA: 3	<HU0007 – Concluir tarefa> SENDO usuária do sistema QUERO um ícone na lista de tarefas PARA indicar que determinada tarefa foi concluída ESTIMATIVA: 1	<HU0012 – Relatório de atividades por usuário> SENDO administradora do sistema QUERO um relatório PARA visualizar as atividades concluídas e andamento de cada usuário ESTIMATIVA: 3
<HU0003 – Listar Usuários> SENDO a administradora do sistema QUERO uma tela PARA visualizar a lista de usuários do aplicativo, alterar as permissões de cada um e que seja possível deletá-los ESTIMATIVA: 2	<HU0008 – Informar usuário responsável> SENDO usuária do sistema QUERO que seja possível informar quando alguma tarefa for criada para o meu usuário ou eu tenha alguma alteração PARA obter informações sem necessidade de abrir o aplicativo ESTIMATIVA: 1,5	<HU0013 – Relatório de atividades por categoria> SENDO administradora do sistema QUERO um relatório PARA visualizar as atividades concluídas e em andamento de cada categoria ESTIMATIVA: 3
<HU0004 – Cadastrar Tarefas> SENDO o administrador do sistema QUERO uma tela PARA cadastrar as tarefas ESTIMATIVA: 2	<HU0009 – Criar tarefas recorrentes> SENDO usuária do sistema QUERO um método PARA criar tarefas com recorrência diária, semanal, mensal, semestral e anual ESTIMATIVA: 2,5	<HU0014 – Relatório de atividades por período> SENDO administrador do sistema QUERO um relatório PARA visualizar as atividades concluídas e em andamento por período de tempo ESTIMATIVA: 3,5
<HU0005 – Listar Tarefas> SENDO o administrador do sistema QUERO uma tela PARA visualizar as tarefas, contendo a lista de tarefas, suas categorias e a pessoa responsável ESTIMATIVA: 2	<HU0010 – Avisar data da atividade> SENDO usuária do sistema QUERO ser avisada quando chegar a data de realização de alguma atividade recorrente PARA que seja possível realizá-la no prazo ESTIMATIVA: 3	

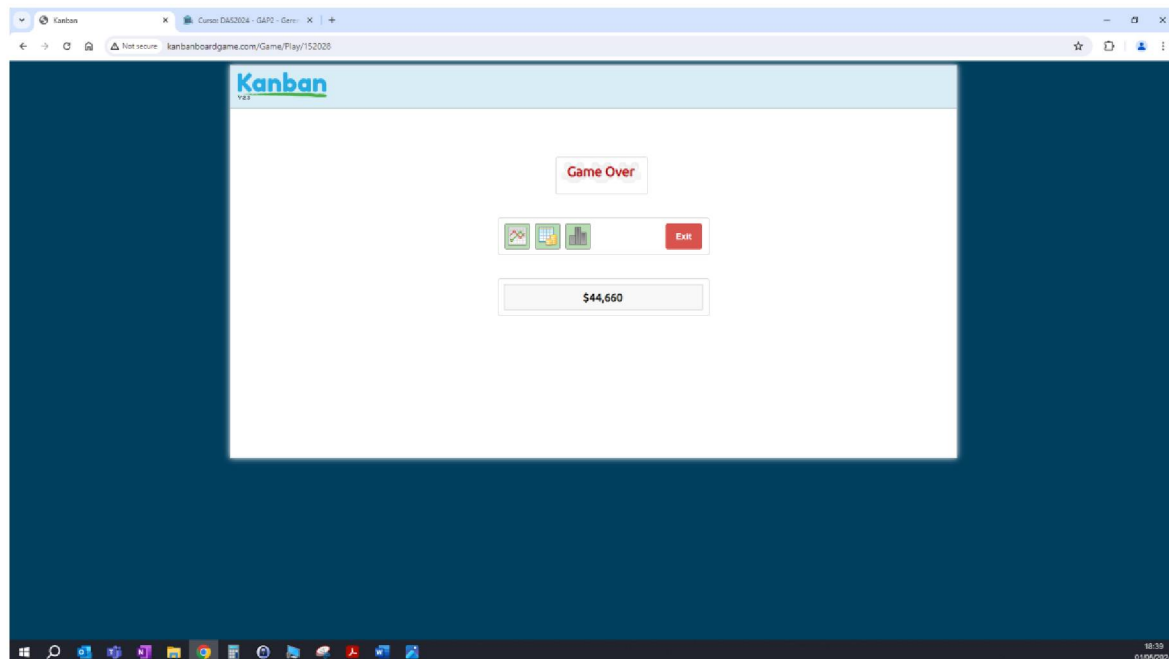
4.2 ARTEFATOS DO PROJETO – GAP 2

Figura 14 – Tela do dia 15 da execução do jogo no site *Kanbanboardgame.com*



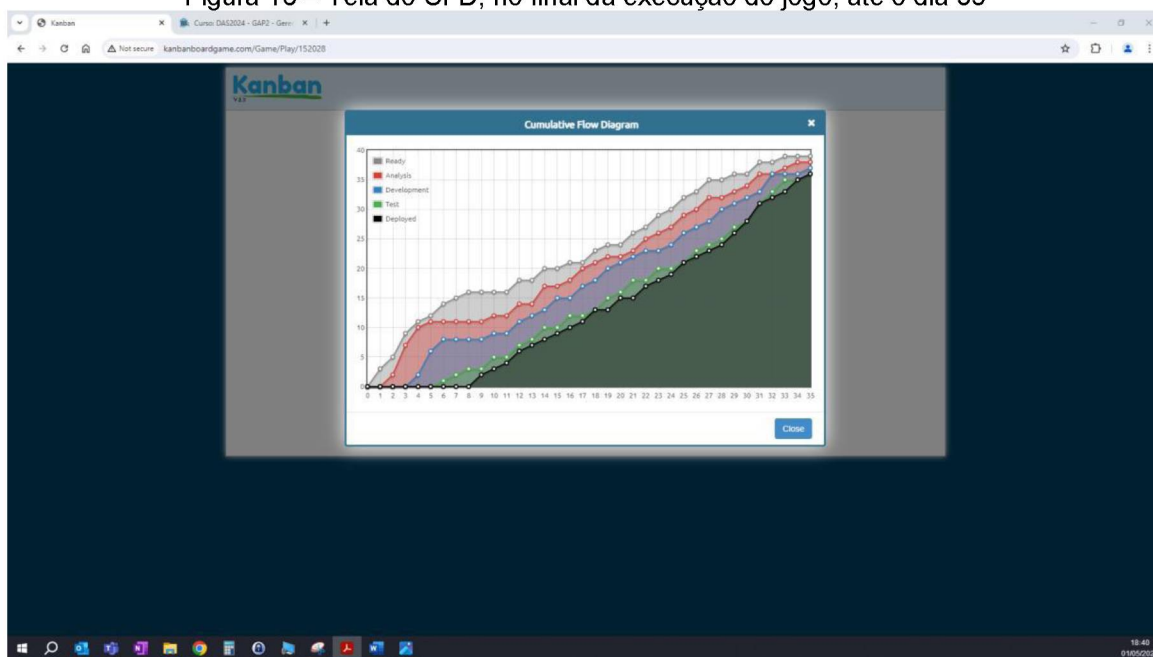
FONTE: A autora (2025)

Figura 15 – Tela do resultado final mostrando a receita final obtida



FONTE: A autora (2025)

Figura 16 – Tela do CFD, no final da execução do jogo, até o dia 35



FONTE: A autora (2025)

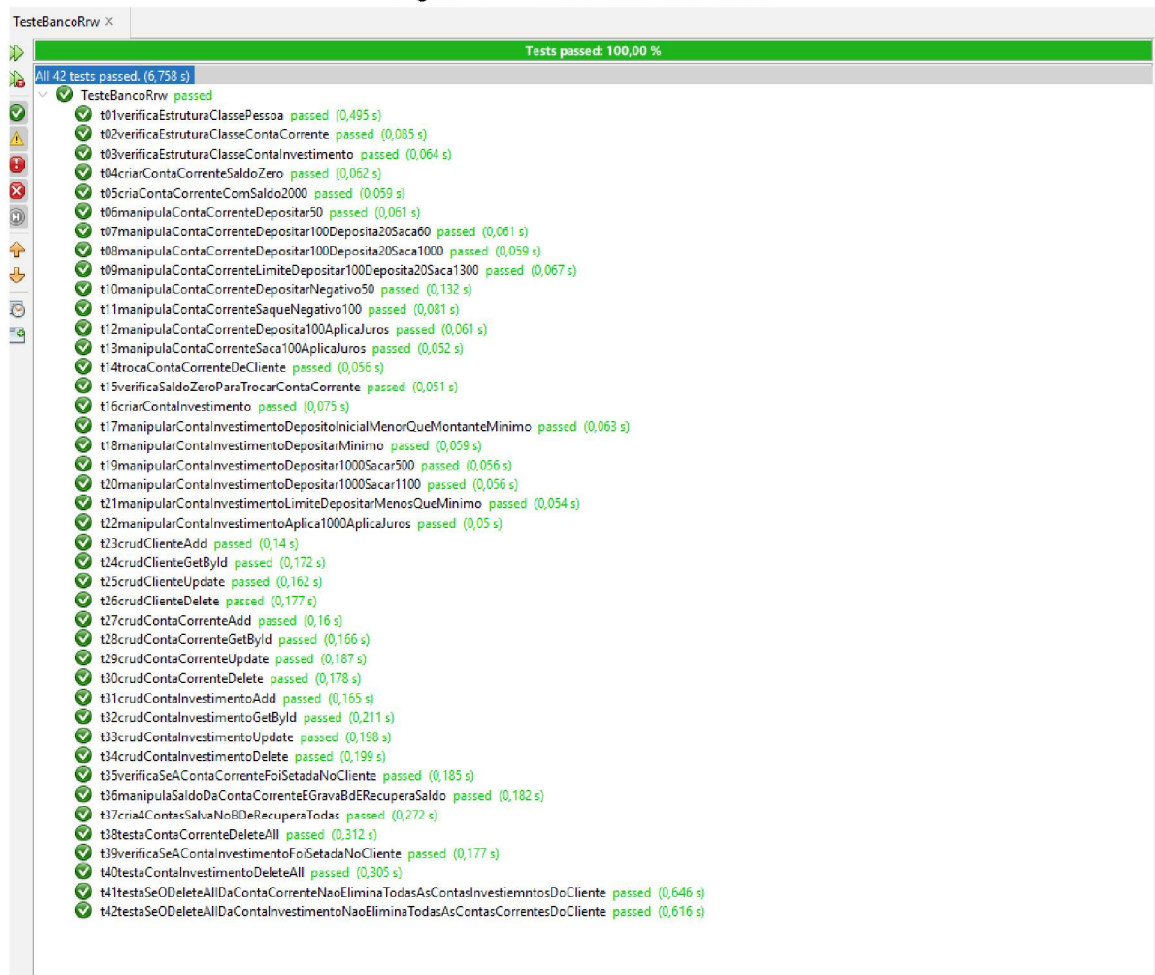
5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

O projeto proposto para esta disciplina foi a implementação de um sistema de conta corrente em Java, contendo cadastro de clientes, contas e investimentos. Foram fornecidos os Diagramas de Classes, o projeto para NetBeans contendo os testes unitários e o script DDL para criação das tabelas no MySQL, através da ferramenta Command Line. Foi necessário implementar as classes de forma que todos os testes unitários em Junit fossem satisfatórios e apresentassem a cor verde.

Foram aplicados os conceitos de Test-Driven Development (TDD), que consiste na escrita dos testes antes do código, apresentando o enfoque ágil. Além disso, foi reforçada a importância da realização dos testes para que uma aplicação seja bem construída e confiável e também a necessidade das boas práticas de desenvolvimento para facilitar futuras manutenções no código e tratamento de erros.

5.1 ARTEFATOS DO PROJETO

Figura 17 – Resultado dos testes



FONTE: A autora (2025)

6 DISCIPLINA: BD – BANCO DE DADOS

No trabalho da disciplina de Banco de Dados, foram desenvolvidas duas questões, a primeira foi a modelagem de um sistema de controle de biblioteca e a segunda à implementação em SQL de um sistema de controle de vacinação, tema escolhido pelos alunos.

Para a primeira questão, foi elaborado o Modelo Entidade-Relacionamento (MER), contendo as entidades e os relacionamentos e o Diagrama Entidade-Relacionamento (DER) (IBM, 2024), detalhando o MER em tabelas, campos, chaves e relacionamentos. Para a segunda questão, foi desenvolvido o modelo lógico de um sistema de controle de vacinas, seguido pelo script para criação das tabelas, que foram populadas com dados fictícios para evidenciar os diferentes tipos de relacionamentos.

Os conceitos aprendidos foram essenciais para o entendimento dos modelos de dados, que são essenciais para o desenvolvimento e planejamento do banco de dados, além de reforçar o relacionamento com as disciplinas estudadas anteriormente e servindo de apoio para as atividades das próximas fases de desenvolvimento do software, estudadas nas disciplinas subsequentes.

6.1 ARTEFATOS DO PROJETO

6.1.1 MODELO ENTIDADE-RELACIONAMENTO CONCEITUAL

1. Obras
 - Código (PK)
 - Título
 - Autor Principal
 - Ano de Publicação
 - Situação (Disponível, Emprestada)
 - Editora (FK)
2. Editoras
 - Código (PK)
 - Nome
 - Cidade
3. Usuários
 - Identificação Única (PK)
 - Nome

- Endereço Completo
- Telefone de Contato
- CPF

4. Funcionários

- Número de Matrícula (PK)
- Nome Completo
- Departamento (FK)

5. Departamentos

- Código (PK)
- Nome
- Chefe (FK - Funcionário)

6. Empréstimos

- Código (PK)
- Usuário (FK)
- Data e Horário do Empréstimo
- Data Prevista de Retorno
- Funcionário que Realizou o Empréstimo (FK)

7. ItensEmpréstimo

- Código (PK)
- Empréstimo (FK)
- Obra (FK)
- Data e Horário da Devolução
- Funcionário que Recebeu a Devolução (FK)

8. Reservas

- Código (PK)
- Usuário (FK)
- Obra (FK)
- Data e Horário da Reserva
- Data de Retirada

6.1.2 MODELO LÓGICO (DIAGRAMA ENTIDADE RELACIONAMENTO)

BANCO: biblioteca

```
CREATE SCHEMA biblioteca;
```

Tabela: Editoras

```
CREATE TABLE Editoras (
  Codigo INT PRIMARY KEY,
  Nome VARCHAR(255) NOT NULL,
  Cidade VARCHAR(255) NOT NULL
);
```

Tabela: Obras

```
CREATE TABLE Obras (
  Codigo INT PRIMARY KEY,
  Titulo VARCHAR(255) NOT NULL,
  Autor_Principal VARCHAR(255) NOT NULL,
  Ano_Publicacao INT NOT NULL,
  Situacao VARCHAR(50) CHECK (Situacao IN ('Disponível',
'Emprestada')),
  Editora INT,
  FOREIGN KEY (Editora) REFERENCES Editoras(Codigo)
);
```

Tabela: Usuários

```
CREATE TABLE Usuarios (
  Identificacao_Unica INT PRIMARY KEY,
  Nome VARCHAR(255) NOT NULL,
  Endereco_Completo TEXT NOT NULL,
  Telefone_Contato VARCHAR(20) NOT NULL,
  CPF VARCHAR(11) NOT NULL UNIQUE
);
```

Tabela: Departamentos

```
CREATE TABLE Departamentos (
  Codigo INT PRIMARY KEY,
  Nome VARCHAR(255) NOT NULL,
  Chefe INT,
  FOREIGN KEY (Chefe) REFERENCES Funcionarios(Numero_Matricula)
);
```

Tabela: Funcionários

```
CREATE TABLE Funcionarios (
  Numero_Matricula INT PRIMARY KEY,
  Nome_Completo VARCHAR(255) NOT NULL,
  Departamento INT,
  FOREIGN KEY (Departamento) REFERENCES Departamentos(Codigo)
);
```

Tabela: Empréstimos

```
CREATE TABLE Emprestimos (
  Codigo INT PRIMARY KEY,
```

```

        Usuario INT,
        Data_Horario_Emprestimo TIMESTAMP NOT NULL,
        Data_Prevista_Retorno DATE NOT NULL,
        Funcionario_Emprestimo INT,
        FOREIGN KEY (Usuario) REFERENCES Usuarios(Identificacao_Unica),
        FOREIGN      KEY      (Funcionario_Emprestimo)      REFERENCES
Funcionarios(Numero_Matricula)
    );

```

Tabela: ItensEmpréstimo

```

CREATE TABLE ItensEmprestimo (
    Codigo INT PRIMARY KEY,
    Emprestimo INT,
    Obra INT,
    Data_Horario_Devolucao TIMESTAMP,
    Funcionario_Devolucao INT,
    FOREIGN KEY (Emprestimo) REFERENCES Emprestimos(Codigo),
    FOREIGN KEY (Obra) REFERENCES Obras(Codigo),
    FOREIGN      KEY      (Funcionario_Devolucao)      REFERENCES
Funcionarios(Numero_Matricula)
    );

```

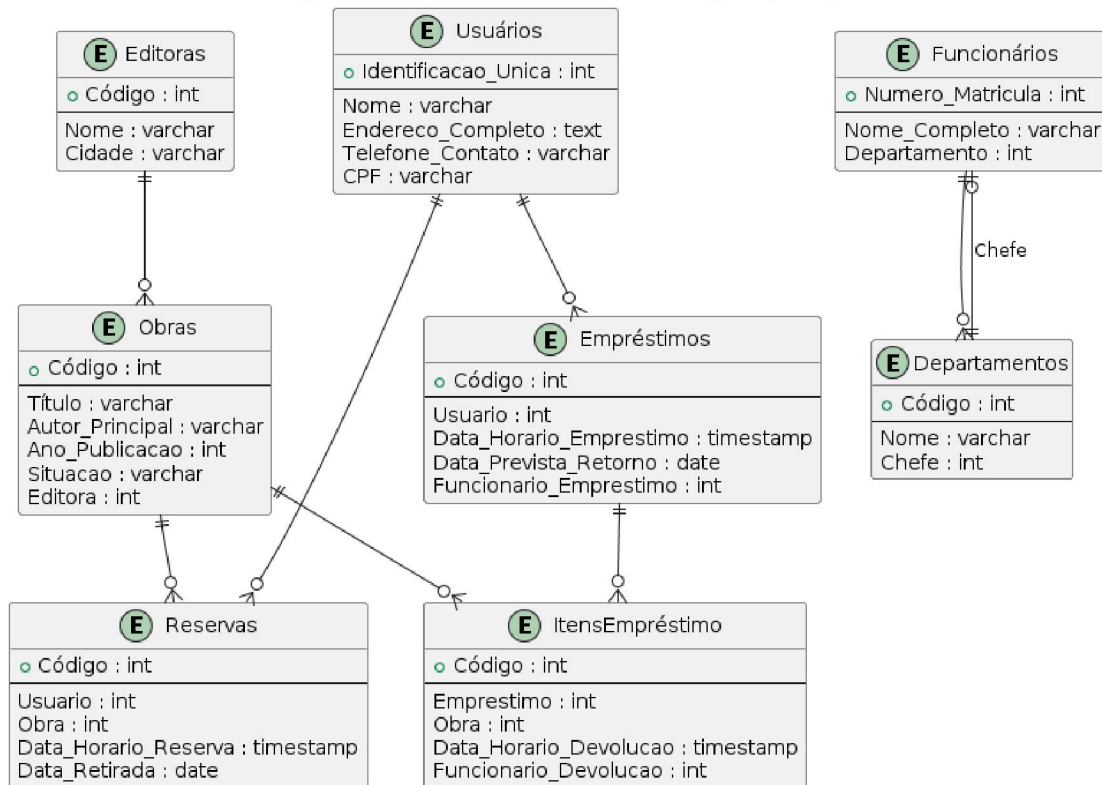
Tabela: Reservas

```

CREATE TABLE Reservas (
    Codigo INT PRIMARY KEY,
    Usuario INT,
    Obra INT,
    Data_Horario_Reserva TIMESTAMP NOT NULL,
    Data_Retirada DATE NOT NULL,
    FOREIGN KEY (Usuario) REFERENCES Usuarios(Identificacao_Unica),
    FOREIGN KEY (Obra) REFERENCES Obras(Codigo)
    );

```

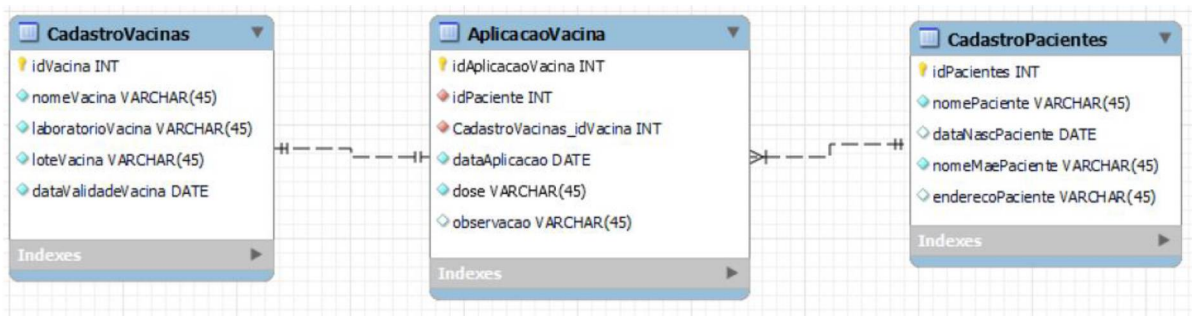
Figura 18 – Detalhamento modelo lógico (DER)



FONTE: A autora (2025)

6.1.3 MODELO EER

Figura 19 – Modelo Entidade-Relacionamento (MER)



FONTE: A autora (2025)

6.1.4 SCRIPT

```
-- MySQL Workbench Forward Engineering
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

--
-- Schema vaccinacao
--
--
--
-- Schema vaccinacao
```

```

-----
CREATE SCHEMA IF NOT EXISTS `vacinacao` DEFAULT CHARACTER SET utf8 ;
USE `vacinacao` ;
-----
-- Table `vacinacao`.`CadastroVacinas`
-----
CREATE TABLE IF NOT EXISTS `vacinacao`.`CadastroVacinas` (
  `idVacina` INT NOT NULL AUTO_INCREMENT,
  `nomeVacina` VARCHAR(45) NOT NULL,
  `laboratorioVacina` VARCHAR(45) NOT NULL,
  `loteVacina` VARCHAR(45) NOT NULL,
  `dataValidadeVacina` DATE NOT NULL,
  PRIMARY KEY (`idVacina`))
ENGINE = InnoDB;
-----
-- Table `vacinacao`.`CadastroPacientes`
-----
CREATE TABLE IF NOT EXISTS `vacinacao`.`CadastroPacientes` (
  `idPacientes` INT NOT NULL AUTO_INCREMENT,
  `nomePaciente` VARCHAR(45) NOT NULL,
  `dataNascPaciente` DATE NULL,
  `nomeMaePaciente` VARCHAR(45) NOT NULL,
  `enderecoPaciente` VARCHAR(45) NULL,
  PRIMARY KEY (`idPacientes`),
  UNIQUE INDEX `dataNascPaciente_UNIQUE` (`dataNascPaciente` ASC) VISIBLE,
  UNIQUE INDEX `idPacientes_UNIQUE` (`idPacientes` ASC) VISIBLE)
ENGINE = InnoDB;
-----
-- Table `vacinacao`.`AplicacaoVacina`
-----
CREATE TABLE IF NOT EXISTS `vacinacao`.`AplicacaoVacina` (
  `idAplicacaoVacina` INT NOT NULL AUTO_INCREMENT,
  `idPaciente` INT NOT NULL,
  `CadastroVacinas_idVacina` INT NOT NULL,
  `dataAplicacao` DATE NOT NULL,
  `dose` VARCHAR(45) NOT NULL,
  `observacao` VARCHAR(45) NULL,
  PRIMARY KEY (`idAplicacaoVacina`),
  UNIQUE INDEX `dataAplicacao_UNIQUE` (`dataAplicacao` ASC) VISIBLE,
  INDEX `fk_AplicacaoVacina_CadastroPacientes1_idx` (`idPaciente` ASC) VISIBLE,
  INDEX `fk_AplicacaoVacina_CadastroVacinas1_idx` (`CadastroVacinas_idVacina`
  ASC) VISIBLE,
  CONSTRAINT `fk_AplicacaoVacina_CadastroPacientes1`
  FOREIGN KEY (`idPaciente`)
  REFERENCES `vacinacao`.`CadastroPacientes` (`idPacientes`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_AplicacaoVacina_CadastroVacinas1`
  FOREIGN KEY (`CadastroVacinas_idVacina`)
  REFERENCES `vacinacao`.`CadastroVacinas` (`idVacina`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
USE `vacinacao` ;
-----
-- Placeholder table for view `vacinacao`.`view1`
-----
CREATE TABLE IF NOT EXISTS `vacinacao`.`view1` (`id` INT);
-----
-- View `vacinacao`.`view1`
-----
DROP TABLE IF EXISTS `vacinacao`.`view1`;

USE `vacinacao`;

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

6.1.5 SELECT TABELAS

Figura 20 – Evidência dados tabela CadastroPacientes

9 `SELECT * FROM controleVacinas.CadastroPacientes`

idPacientes	nomePaciente	dataNascPaciente	nomeMaePaciente	endereçoPaciente
1	Jorge da Silva	1999-05-10	Jurema da Silva	Rua Goiás, 25, Centro, Curitiba - PR
2	Carlos Oliveira	2005-12-10	Maria Oliveira	Rua Guarapuava, 381, Centro, Curitiba - PR
3	Eduardo Martins	1995-06-07	Suzana Martins	Rua Rio Branco, 55, Centro, Curitiba - PR
4	Luana Gomes	1997-08-02	Marcia Gomes	Rua Portugal, 14, Centro, Curitiba - PR
5	Julia Lopes	2001-01-25	Marli Lopes	Rua Sprada, 995, CIC, Curitiba - PR
NULL	NULL	NULL	NULL	NULL

FONTE: A autora (2025)

Figura 21 – Evidência dados tabela CadastroVacinas

17 `SELECT * FROM controleVacinas.CadastroVacinas`

idVacina	nomeVacina	laboratorioVacina	loteVacina	dataValidadeVacina
1	dTpa	Fiocruz	12345	2028-12-31
2	BCG	Butantan	78945	2030-12-31
3	Hepatite	Astrazeneca	456321	2026-12-31
4	Influenza	Pfizer	452189	2025-12-31
5	CoronaVac	Butantan	785214	2025-12-31
NULL	NULL	NULL	NULL	NULL

FONTE: A autora (2025)

Figura 22 – Evidência dados tabela AplicacaoVacina

25 `SELECT * FROM controleVacinas.AplicacaoVacina`

idAplicacaoVacina	idPaciente	CadastroVacinas_idVacina	dataAplicacao	dose	observacao
1	1	3	2024-02-15	2	
2	5	4	2023-05-10	1	PrimeiraDose
3	2	4	2024-11-20	Unica	
4	3	5	2022-03-30	1	N/A
5	4	4	2023-06-05	1	
6	1	3	2021-08-01	1	
NULL	NULL	NULL	NULL	NULL	

FONTE: A autora (2025)

7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

O trabalho da disciplina de Aspectos Ágeis de Programação consistia na reestruturação do código de ordenação Bubble Sort em Java, fornecido pelo professor, para uma versão utilizando os conceitos de Clean Code (Martin, 2008), melhorando o algoritmo em 3 pontos. Para isso, foram aplicados os conceitos aprendidos, como a criação de nomes significativos para as variáveis, inserção e/ou remoção de comentários úteis e a refatoração dos métodos com ações únicas.

A recriação das variáveis com nomes apresentáveis é importante para facilitar sua identificação, evitando a necessidade de explicação em formato de comentários, muitas vezes desnecessários. A refatoração dos métodos, evita que sejam construídas funções muito longas, que dificultam o entendimento e a manutenção do código.

O conceitos de Clean Code relacionam-se às demais disciplinas de programação, reforçando as boas práticas no desenvolvimento de software, para que as aplicações sejam desenvolvidas de forma robusta, mas bem estruturadas, facilitando a manutenção do código.

7.1 ARTEFATOS DO PROJETO

7.1.1 CÓDIGO BUBBLE SORT REFATORADO

```
/*De acordo com o que foi realizado na aula*/

import java.io.*;

class BubbleSort {
    static void bubbleSort(int array[], int arraySize){
        int i, nextIndex, temp;
        boolean isSwapped;
        for (i = 0; i < arraySize - 1; i++) {
            isSwapped = false;
            for (nextIndex = 0; nextIndex < arraySize - i - 1; nextIndex++) {
                if (array[nextIndex] > array[nextIndex + 1]) {
                    exchangeValues(array, nextIndex);
                    isSwapped = true;
                }
            }
            if (!isSwapped)
                break;
        }
    }
}
```

```
private static void exchangeValues(int array[], int nextIndex){
    int temp;
    temp = array[nextIndex];
    array[nextIndex] = array[nextIndex + 1];
    array[nextIndex + 1] = temp;
}

static void printArray(int array[], int arraySize){
    int i;
    for (i = 0; i < arraySize; i++)
        System.out.print(array[i] + " ");
    System.out.println();
}

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

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

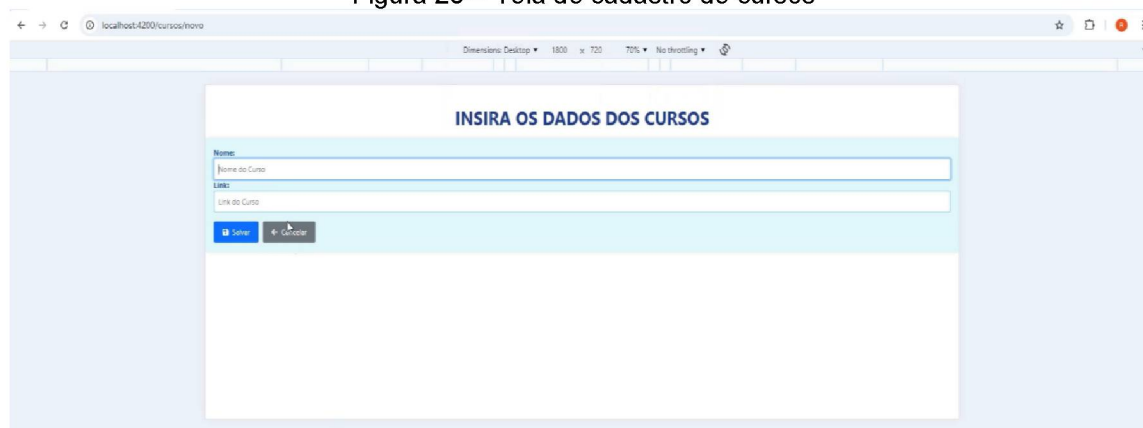
O trabalho para a disciplina de Desenvolvimento WEB 1 consistia na criação de um projeto para aplicação do conceito de CRUD: *Create*, *Read*, *Update* e *Delete* para relacionar informações de alunos e cursos. Para isso, foram implementadas telas de inserção, listagem, alteração e remoção dos dados e utilizada a API Local Storage (MDN Web Docs, 2024) para armazenamento das informações.

Para a disciplina de Desenvolvimento WEB 2, o projeto proposto foi a continuação do anterior, agora adicionando os dados de matrícula e seus devidos relacionamentos com as tabelas de dados previamente implementadas. Para este trabalho, foi solicitado o desenvolvimento do *back-end* em Spring e armazenamento em banco de dados PostgreSQL (The PostgreSQL Global Development Group, 2025) e não mais em Local Storage. Além das funções adicionais referentes à matrícula, deveria ser implementado o menu para navegação entre as telas. Apesar de não ter sido realizada a entrega, houve discussão e troca de ideias entre os integrantes da equipe sobre o desenvolvimento da segunda fase do sistema, mostrando a complexidade deste tema.

O conteúdo e atividades propostas foram essenciais para aplicação da teoria, reforçando o vínculo com as outras disciplinas e a importância do Desenvolvimento WEB nas práticas de desenvolvimento ágil.

8.1 ARTEFATOS DO PROJETO

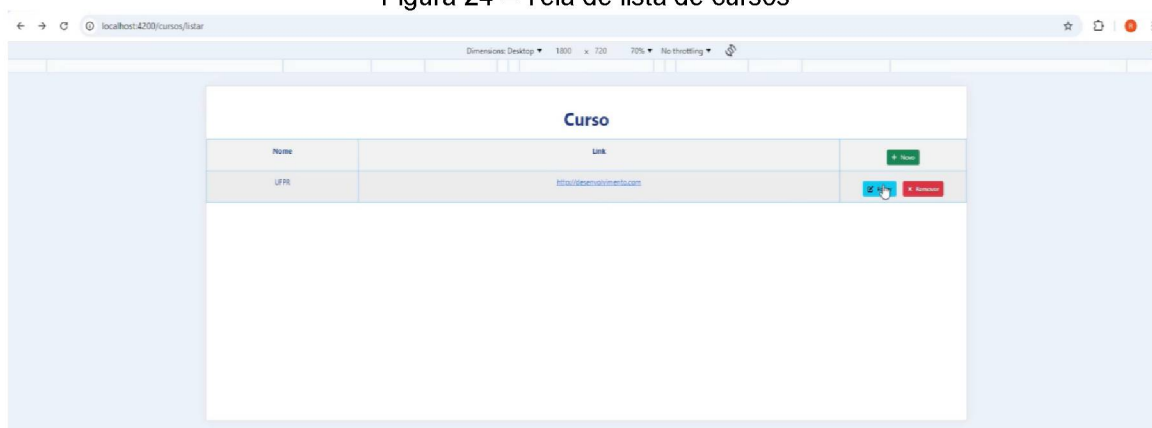
Figura 23 – Tela de cadastro de cursos



A imagem mostra uma interface web em um navegador com o endereço `localhost:4200/cursos/novo`. O formulário principal, intitulado "INSIRA OS DADOS DOS CURSOS", contém dois campos de entrada: "Nome" (com o placeholder "Nome do Curso") e "Link" (com o placeholder "Link do Curso"). Abaixo desses campos, há dois botões: "Salvar" em azul e "Cancelar" em cinza. A interface é limpa, com uma barra de navegação superior e uma barra de status inferior.

FONTE: A autora (2025)

Figura 24 – Tela de lista de cursos



FONTE: A autora (2025)

Figura 25 – Tela de cadastro de alunos

INSIRA OS DADOS DOS ALUNOS

Nome:

 Atenção! Digite o nome do aluno.

CPF:

 Atenção! Digite o CPF do aluno.
 Digite um CPF válido.

E-mail:

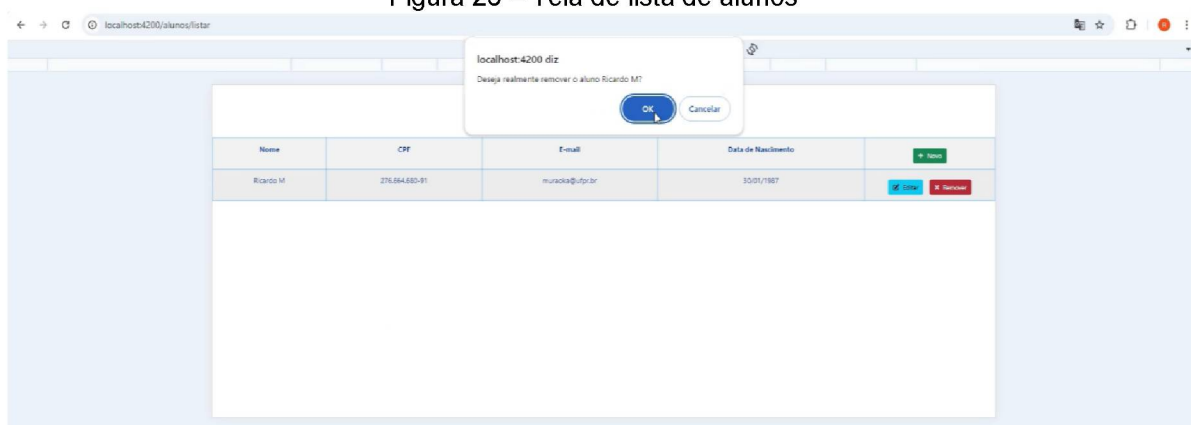
 Atenção! Digite o e-mail do aluno.

Data de Nascimento:

 Atenção! Digite a data de nascimento do aluno.
 Digite a data de nascimento.

FONTE: A autora (2025)

Figura 26 – Tela de lista de alunos



FONTE: A autora (2025)

9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

O projeto final para a disciplina de UX no Desenvolvimento Ágil de Software, consistiu na elaboração de telas para um aplicativo de gestão de condomínio, tema escolhido pelo aluno. Além da criação das telas, foram explicadas as escolhas de cores, modelo de layout e fonte de letras, para justificar a aplicação dos conceitos de UX para que a aplicação fosse intuitiva e funcional ao usuário. Para a criação das telas foi utilizado o Figma (Figma Learn, 2025), uma ferramenta de design moderna, que permite criação de protótipos, com telas interativas entre si, facilitando o desenvolvimento e reforçando os conceitos de agilidade no desenvolvimento do protótipo.

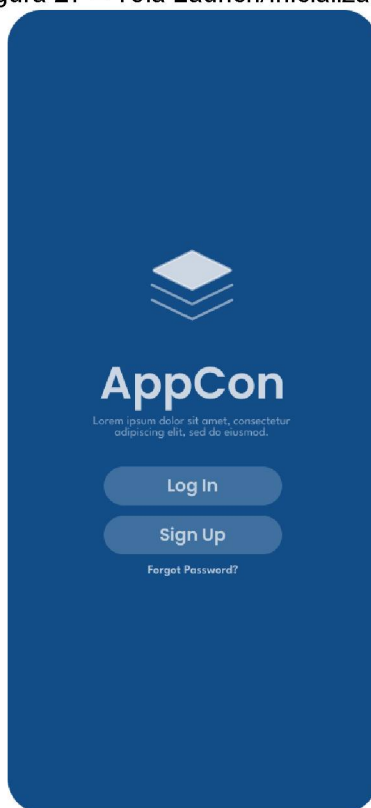
Além disso, havia uma etapa para solicitação de *feedback* de algum possível usuário, reforçando a necessidade desta etapa no desenvolvimento ágil, para garantir que a solução atenda às expectativas e as alterações sejam implementadas em cada ciclo.

Esta disciplina contribuiu para conscientização da importância da experiência do usuário e de solicitar *feedback* constante nas iterações durante o desenvolvimento ágil.

9.1 ARTEFATOS DO PROJETO

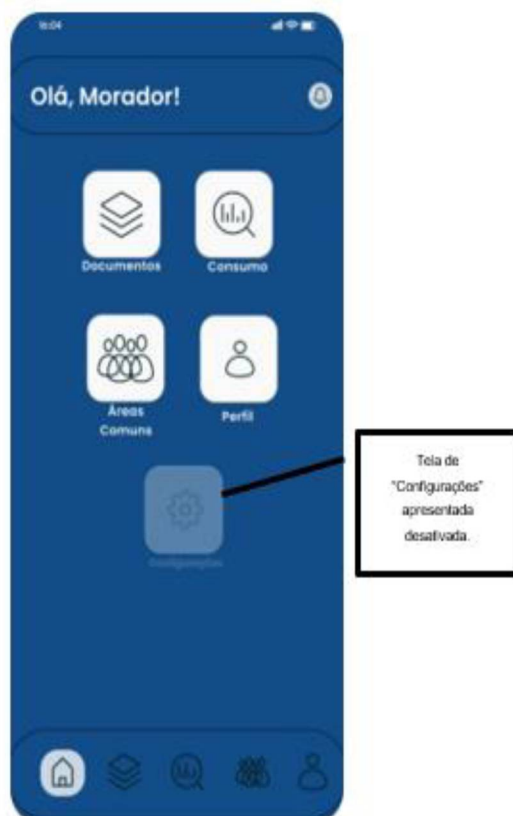
9.1.1 TELAS:

Figura 27 – Tela Launch/Inicialização



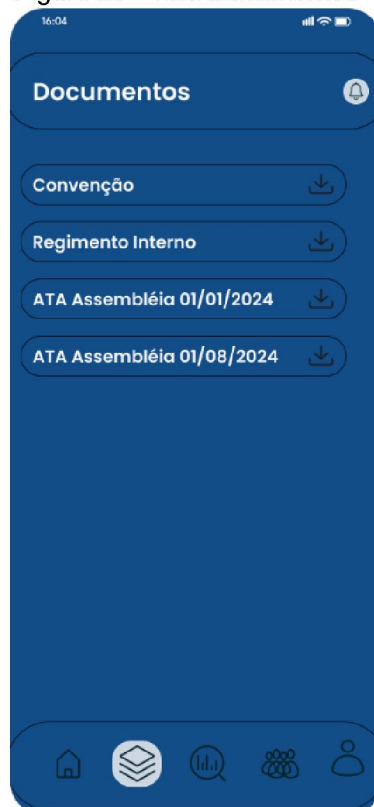
FONTE: A autora (2025)

Figura 28 – Tela Home/Inicial



FONTE: A autora (2025)

Figura 29 –Tela Documentos



FONTE: A autora (2025)

Figura 30 –Tela Consumo



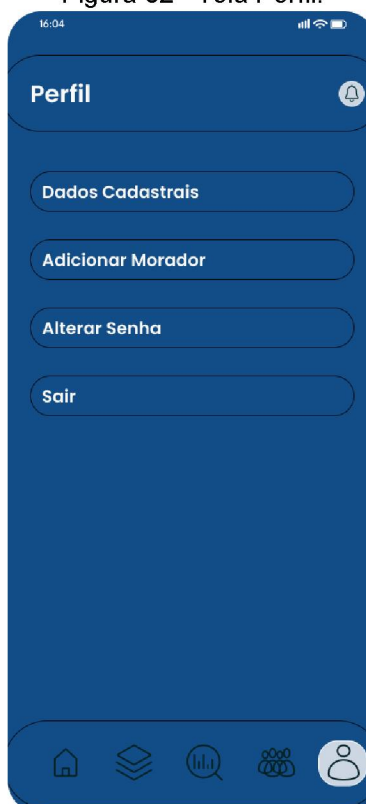
FONTE: A autora (2025)

Figura 31 –Tela Áreas Comuns



FONTE: A autora (2025)

Figura 32 –Tela Perfil:



FONTE: A autora (2025)

9.1.2 ESCOLHAS FEITAS:

- a. Cores:
 - i. Azul: Para demonstrar tranquilidade e integridade e gerar confiança no usuário/cliente.
 - ii. Branco: Para dar o contraste dos botões e textos (havia sido testado o mesmo cinza da tela de inicialização, mas o contraste não foi suficientemente alto para leitura inclusiva de todos os usuários). Para a tela selecionada na barra de navegação inferior, foi utilizado um pouco de transparência no branco, pois é uma indicação de localização do usuário e não a funcionalidade principal da tela.
 - iii. Cinza: Utilizado na tela inicial, para demonstrar profissionalismo e seriedade, harmonizando com o azul.
- b. Layout:
 - i. Similar à vários aplicativos utilizados atualmente, com a barra de navegação na parte inferior e o ícone de notificação na parte superior direita da tela.
 - ii. Linhas pretas e com bordas arredondadas para delimitar as telas e as funcionalidades e textos secundários.
- c. Fonte:
 - i. Arredondada para gerar mais suavidade na leitura e não tanta rigidez que a seriedade das cores pode causar.

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

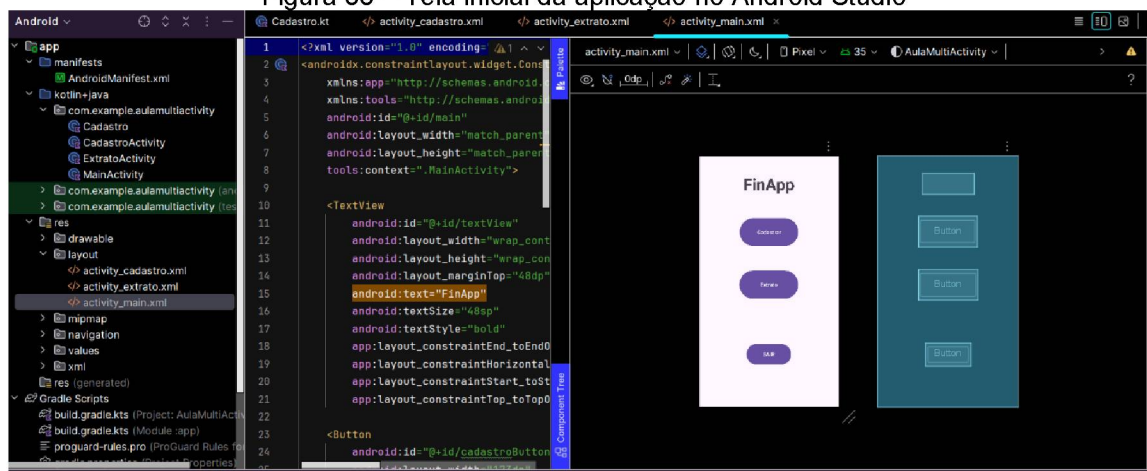
O projeto proposto em Desenvolvimento Mobile 1 foi o desenvolvimento de um sistema Android para organização financeira, incluindo cadastro e visualização do extrato de gastos. Foram aplicados os conceitos de Activity, memória, listas, *design* do *layout* e transferência de dados entre telas, compreendido na aula de MultiActivity (Android Developers, 2024). Foi necessária a utilização da ferramenta Android Studio e entendimento da estrutura dessa aplicação, conforme visto nas aulas. Além dos conteúdos aplicados durante o desenvolvimento do trabalho, também foi necessário o entendimento do Git Hub para fazer o *deploy* do código nesta ferramenta.

Para a disciplina de MOB 2, o trabalho solicitado envolvia a aplicação dos conceitos aprendidos de corrotinas e web services, utilizando a API fornecida e permitindo listar e filtrar as informações apresentadas, com o tema Harry Potter.

Apesar de não ter sido entregue o trabalho de MOB 2, a teoria foi estudada e aplicada nos questionários propostos para aprovação na disciplina, sendo possível reforçar a integração com diversas disciplinas do curso, principalmente Introdução à Programação e Banco de Dados, além da aplicação de conceitos de desenvolvimento ágil.

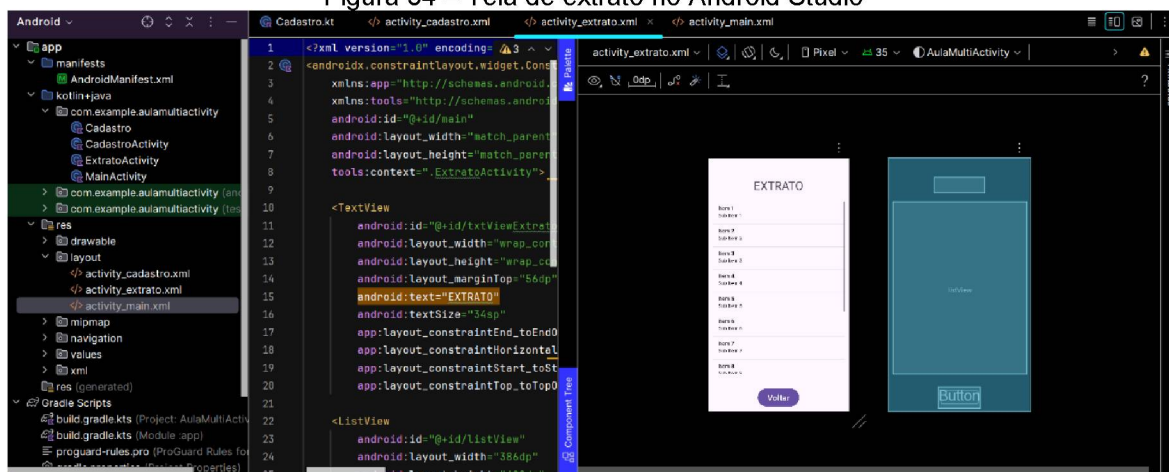
10.1 ARTEFATOS DO PROJETO

Figura 33 – Tela inicial da aplicação no Android Studio



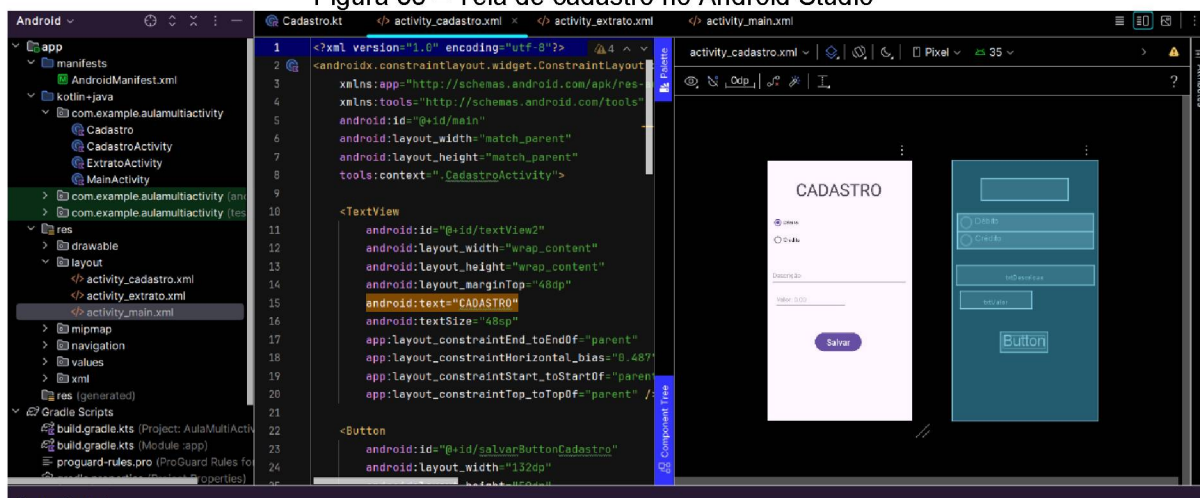
FONTE: A autora (2025)

Figura 34 – Tela de extrato no Android Studio



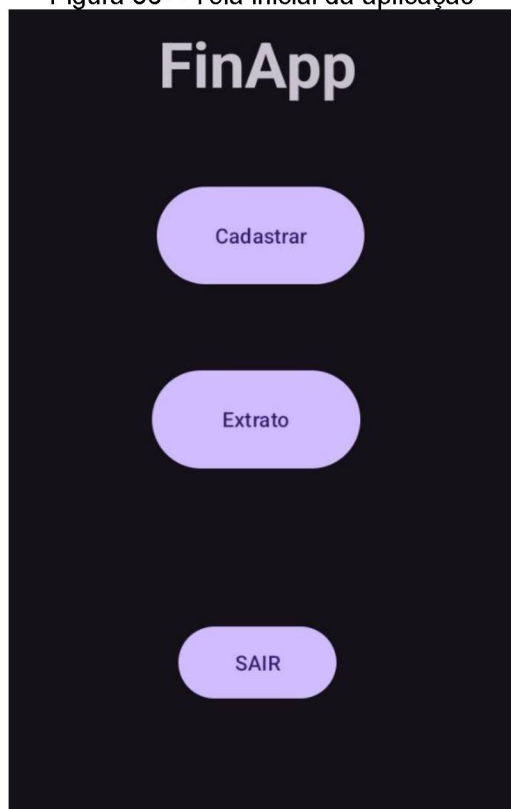
FONTE: A autora (2025)

Figura 35 – Tela de cadastro no Android Studio



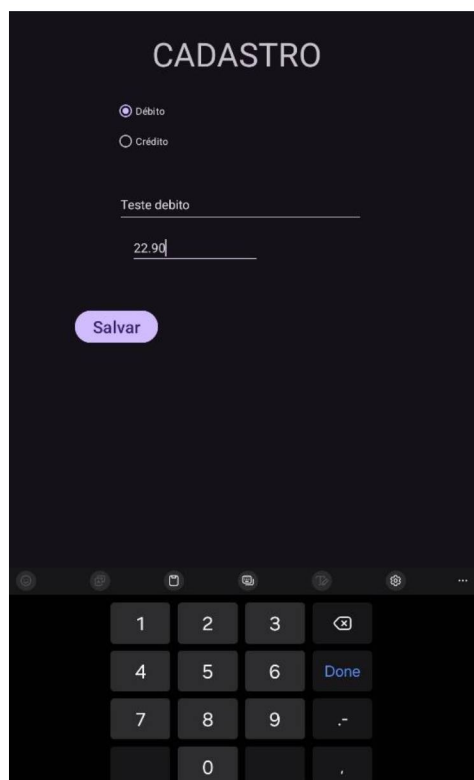
FONTE: A autora (2025)

Figura 36 – Tela inicial da aplicação



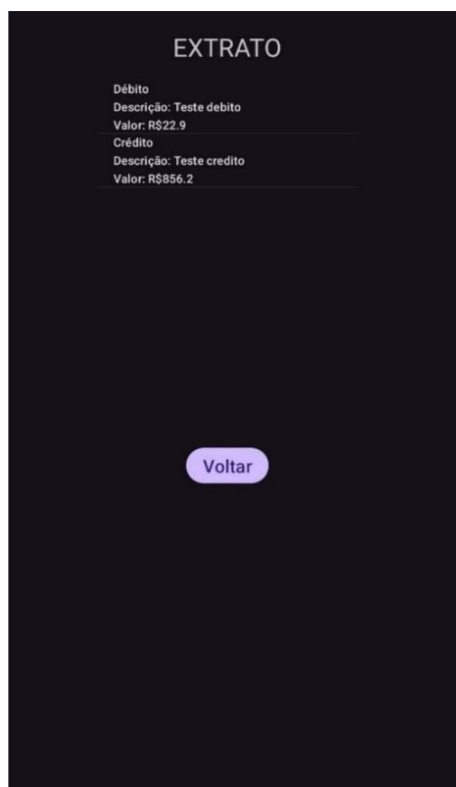
FONTE: A autora (2025)

Figura 37 – Tela de cadastro da aplicação



FONTE: A autora (2025)

Figura 38 – Tela de extrato da aplicação



FONTE: A autora (2025)

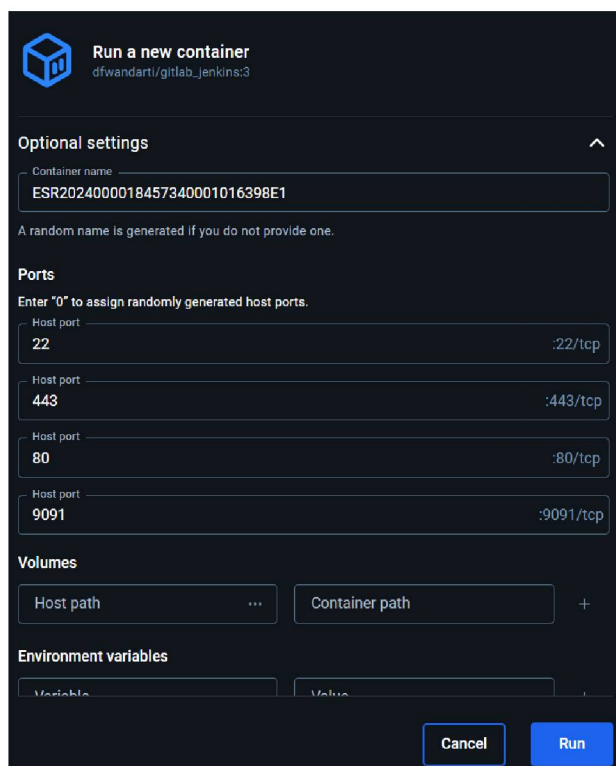
11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)

O trabalho final para a disciplina Infraestrutura para Desenvolvimento e Implantação de Software (DevOps) (Atlassian, 2007) consistia na criação de um contêiner (Docker) e utilização do GitLab, sendo solicitada a utilização de uma imagem fornecida pelo professor, além da publicação nas portas solicitadas. Foi possível aplicar a teoria tanto na interface quanto em linha de comando para fixação do conteúdo em diferentes métodos.

Durante o estudo da disciplina, foi possível fortalecer a teoria sobre DevOps, principalmente sobre o ciclo de vida de um produto ou aplicação, desde o planejamento até a manutenção, passando ainda pelas fases de projeto, implementação, teste e implantação. Além disso, foi estudado sobre as métricas para verificar o desempenho e auxiliar na identificação e tratamento de gargalos, sendo fundamentais para o desenvolvimento ágil de software.

11.1 ARTEFATOS DO PROJETO

Figura 39 – Tela de configuração do container



The screenshot shows the 'Run a new container' dialog in GitLab. The title bar says 'Run a new container' with the image 'dfwandarti/gitlab_jenkins:3'. Below the title bar, there is a section for 'Optional settings' with a dropdown arrow. The 'Container name' field is filled with 'ESR2024000018457340001016398E1'. Below this, a note says 'A random name is generated if you do not provide one.' The 'Ports' section has a note 'Enter "0" to assign randomly generated host ports.' and four rows of port mappings: '22' to ':22/tcp', '443' to ':443/tcp', '80' to ':80/tcp', and '9091' to ':9091/tcp'. The 'Volumes' section has two input fields: 'Host path' and 'Container path', with a plus sign button. The 'Environment variables' section has two input fields: 'Variable' and 'Value'. At the bottom right, there are 'Cancel' and 'Run' buttons.

FONTE: A autora (2025)

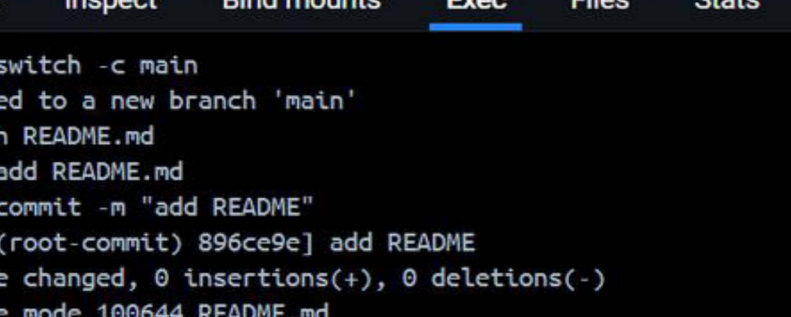
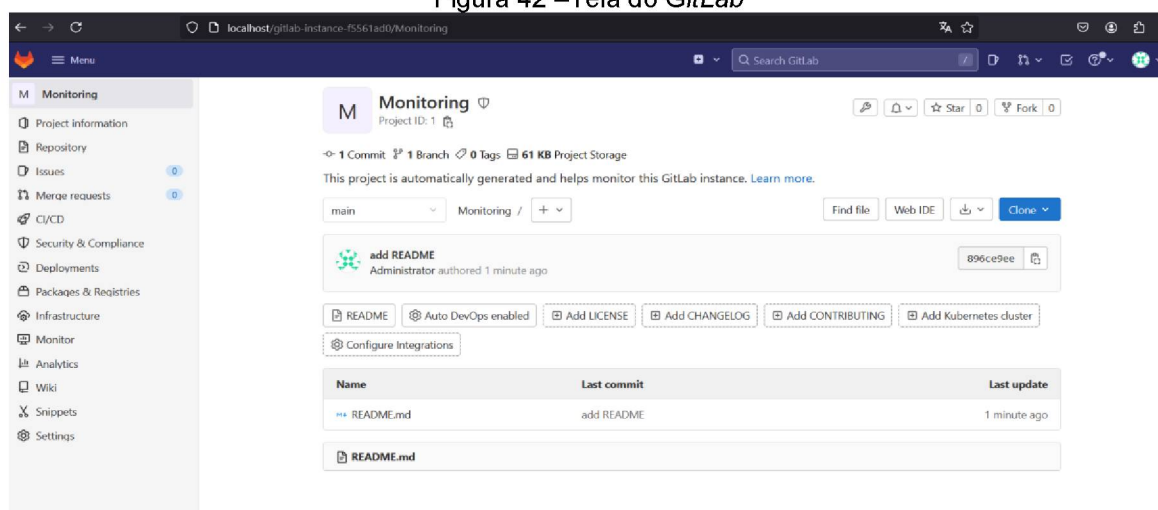


Figura 41 – Tela de execução

Logs	Inspect	Bind mounts	Exec	Files	Stats
------	---------	-------------	------	-------	-------

```
# git switch -c main
Switched to a new branch 'main'
# touch README.md
# git add README.md
# git commit -m "add README"
[main (root-commit) 896ce9e] add README
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
# git push -u origin main
Username for 'http://3336b0baf637': root
Password for 'http://root@3336b0baf637':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 215 bytes | 215.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://3336b0baf637/gitlab-instance-f5561ad0/Monitoring.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

FONTE: A autora (2025)

Figura 42 –Tela do *GitLab*

FONTE: A autora (2025)

12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

Na disciplina de Testes Automatizados foi proposto como trabalho a criação de um código para ser utilizado na ferramenta Playwright, escrevendo nome e matrícula no anoteпад e considerando os três navegadores: Chromium, Webkit (Safari) e Firefox (Playwright, 2020). Esta prática demonstra um exemplo de automação do início ao fim, garantindo que a aplicação está de acordo com os requisitos.

A aplicação de testes na metodologia ágil é essencial para que tanto a manutenção da aplicação seja facilitada quanto às necessidades do usuário sejam atendidas, garantindo assim que a entrega agregue valor.

12.1 ARTEFATOS DO PROJETO

12.1.1 CÓDIGO UTILIZADO NO PLAYWRIGHT

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

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


13 CONCLUSÃO

Este memorial expôs os trabalhos desenvolvidos durante o decorrer do curso de pós graduação em Desenvolvimento Ágil de Software, apresentando de forma sequencial como foram desenvolvidos os projetos de forma a integrar com as disciplinas subsequentes, partindo da análise de projeto até a execução e automação de testes.

Inicialmente foram estudados os conceitos de gestão de projetos e modelagem ágil, focando na documentação essencial para um projeto ágil, além da escrita e detalhamento de histórias de usuário e diagramas teóricos para embasar o planejamento dos projetos. Então, foram abordados conteúdos práticos de desenvolvimento e tecnologias úteis para o desenvolvimento ágil de software, focando nas melhores práticas para implementação ágil. Utilizando-se ferramentas e *frameworks* atuais, foi possível entender o conceito de integração entre as tecnologias de desenvolvimento web, mobile, relacionadas à infraestruturas apresentadas, ressaltando a importância da execução e automação de testes para garantir que o produto segue os requisitos propostos e atende às necessidades dos usuários e reforçando a necessidade de implementação das práticas de DevOps.

Além disso, com o conteúdo de UX, foi possível perceber a importância de se utilizar protótipos para criação de interfaces no desenvolvimento iterativo, obtendo *feedback* constante e permitindo correções e melhorias de forma rápida. Também foram entendidos os conceitos de design, proporcionando o entendimento da relevância deste conteúdo para desenvolver um produto assertivo, oferecendo experiência satisfatória ao usuário.

De modo geral, a trajetória percorrida durante o curso enfatiza a importância das práticas de metodologia ágil durante todo o desenvolvimento de um software, proporcionando clareza nas entregas e oportunidades de melhoria contínua, garantindo a qualidade do produto.

REFERÊNCIAS

AMBLER, Scott W., Introdução ao Desenvolvimento Orientado a Testes (TDD), 2013. Disponível em: <https://agiledata.org/essays/tdd.html>. Acesso em: 12 ago. 2025.

ANDROID DEVELOPERS. Activity, 2024. Disponível em: <https://developer.android.com/reference/kotlin/android/app/Activity>. Acesso em: 14 ago. 2025.

ATLASSIAN. O que é DevOps?, 2007. Disponível em: <https://www.atlassian.com/br/devops>. Acesso em: 14 ago. 2025.

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

FIGMA LEARN. Figma Design for beginners, 2025. Disponível em: <https://help.figma.com/hc/en-us/sections/30880632542743>. Acesso em: 14 ago. 2025

HIGHSMITH, J. Agile Project Management: Creating Innovative Products. Boston: Addison-Wesley, 2002.

HIGHSMITH, J. História: O Manifesto Ágil, 2002. Disponível em: <https://agilemanifesto.org/history.html>. Acesso em: 28 jul. 2025.

IBM. O que é um diagrama de relacionamento entre entidades?, 2024. Disponível em: <https://www.ibm.com/br-pt/think/topics/entity-relationship-diagram>. Acesso em: 14 ago. 2025.

KANBAN UNIVERSITY, The Official Kanban Guide. Disponível em: <https://kanban.university/kanban-guide>, 2021. Acesso em: 11 ago. 2025.

MARTIN, R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Upper Saddle River: Prentice Hall, 2008. Disponível em: <https://www.pearson.com/en-us/subject-catalog/p/clean-code-a-handbook-of-agile-software-craftsmanship/P200000002226>. Acesso em: 21 jul. 2025.

MDN WEB DOCS. Window: localStorage property, 2024. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>. Acessado em: 14 ago. 2025.

PLAYWRIGHT, Test generator, 2020. Disponível em: <https://playwright.dev/docs/codegen>. Acesso em: 12 ago. 2025.

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

PROJECT MANAGEMENT INSTITUTE, Project Management Body of Knowledge (PMBOK guide). Pennsylvania. 6ª ed., 2017.

SCHWABER, K.; SUTHERLAND, J. The 2020 Scrum Guide. Disponível em: <https://scrumguides.org/scrum-guide.html>. Acesso em: 04 jul. 2025.

STARR, Leon. How to Build Articulate Class Models and get Real Benefits from UML, 2008. Disponível em: <https://www.uml.org/news.htm>. Acesso em: 03 jul. 2025.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL 17.6 Documentation, 2025. Disponível em: <https://www.postgresql.org/docs/current/>. Acesso em: 14 ago. 2025.