

UNIVERSIDADE FEDERAL DO PARANÁ

RICARDO MURAOKA

MEMORIAL DE PROJETOS: DESENVOLVIMENTO ÀGIL DE *SOFTWARE*
UTILIZANDO *FRAMEWORKS* MODERNOS

CURITIBA
2025

RICARDO MURAOKA

MEMORIAL DE PROJETOS: DESENVOLVIMENTO ÁGIL DE *SOFTWARE*
UTILIZANDO *FRAMEWORKS* MODERNOS

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 Montañó

CURITIBA
2025



MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO DESENVOLVIMENTO ÁGIL
DE SOFTWARE - 40001016398E1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação Desenvolvimento Ágil de Software da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de RICARDO MURAGNA, intitulada: **MEMORIAL DE PROJETOS: DESENVOLVIMENTO ÁGIL DE SOFTWARE UTILIZANDO FRAMEWORKS MODERNOS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua _____ no rito de defesa.

A outorga do título de especialista está sujeita a 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, 28 de Outubro de 2025.


RAFAELINO DE AGUIAR MONTANO
Presidente da Banca Examinadora


JAIME WOLCIEWSKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

O objetivo deste memorial é demonstrar a trilha de aprendizado experienciada ao longo do curso de especialização em Desenvolvimento Ágil de Software. Essa jornada envolve a integração entre as diversas disciplinas cursadas, com aplicação prática dos conceitos, sendo os artefatos gerados a o resultado final das atividades desenvolvidas em cada disciplina. Todas as disciplinas adotaram como foco central os princípios do desenvolvimento ágil, adaptando-os a contextos específicos para otimizar o ciclo de vida do software. Durante cada etapa percorrida nesta especialização — inclusive aquelas com ênfase em codificação —, foi possível observar como o conhecimento teórico e prático pode transformar processos tradicionais. Ao incorporar práticas ágeis, as etapas de desenvolvimento ganham maior qualidade, fomentam a colaboração entre equipes e, se bem aplicadas, reduzem custos operacionais, promovendo um ambiente mais eficiente e inovador. Essa abordagem que nos permite visualizar todo o processo através de um ponto de visto macro, passando pelas diversas etapas, que inicia com os fundamentos dos Métodos Ágeis no Desenvolvimento de Software e avança até as disciplinas de codificação em frameworks para Web e Mobile, constrói pilares sólidos para a formação de profissionais mais competentes. Esses conhecimentos equipam os participantes para enfrentar as demandas crescentes do mercado de tecnologia, onde a agilidade e a integração interdisciplinar são essenciais. Assim, o curso não apenas transmite habilidades técnicas, mas também cultiva uma mentalidade adaptável, preparando-os para contribuir de forma impactante em projetos reais.

Palavras-chave: desenvolvimento ágil de software; frameworks ágeis; Melhoria contínua; modelagem de sistemas, entrega de valor em softwares.

ABSTRACT

The objective of this memorial is to demonstrate the learning path experienced throughout the specialization course in Agile Software Development. This journey involves the integration among the various disciplines taken, with practical application of the concepts, the generated artifacts being the final result of the activities developed in each discipline. All disciplines adopted as central focus the principles of agile development, adapting them to specific contexts to optimize the software lifecycle. During each stage traversed in this specialization—including those with emphasis on coding—it was possible to observe how theoretical and practical knowledge can transform traditional processes. By incorporating agile practices, the development stages gain greater quality, foster collaboration among teams and, if well applied, reduce operational costs, promoting a more efficient and innovative environment. This approach allows us to visualize the entire process from a macro viewpoint, going through the various stages, which begins with the fundamentals of Agile Methods in Software Development and advances to the disciplines of coding in frameworks for Web and Mobile, building solid pillars for the formation of more competent professionals. These knowledges equip the participants to face the growing demands of the technology market, where agility and interdisciplinary integration are essential. Thus, the course not only transmits technical skills, but also cultivates an adaptable mindset, preparing them to contribute impactfully in real projects.

Keywords: agile software development; agile frameworks; continuous improvement; systems modeling, value delivery in software.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – MAPA MENTAL	11
FIGURA 2 - ENTREGA CONTÍNUA.....	11
FIGURA 3 - KANBAN	12
FIGURA 4 - EXTREME PROGRAMMING.....	12
FIGURA 5 - SCRUM	12
FIGURA 6 - LEAN SOFTWARE DEVELOPMENT	13
FIGURA 7 - PRINCÍPIOS ÁGEIS	13
FIGURA 8 - MANIFESTO ÁGIL.....	13
FIGURA 9 - MODELOS TRADICIONAIS DE PROCESSO DE SOFTWARE	14
FIGURA 10 - PROCESSO DE SOFTWARE	14
FIGURA 11 - CASO DE USO DE NÍVEL 1.....	16
FIGURA 12 - CASO DE USO DE NÍVEL 2.....	16
FIGURA 13 - HISTÓRIA DE USUÁRIOS – GESTÃO DE CONDOMÍNIO.....	17
FIGURA 14 - HISTÓRIA DE USUÁRIOS – MANTER MORADOR	17
FIGURA 15 - DIAGRAMA DE CLASSES	17
FIGURA 16 - DIAGRAMA DE SEQUENCIA – GESTAO DE CONDOMINIO	18
FIGURA 17 - DIAGRAMA DE SEQUÊNCIA – MANTER MORADOR.....	18
FIGURA 18 - DIAGRAMA DE SEQUÊNCIA – MANTER APARTAMENTO	19
FIGURA 19 - DIAGRAMA DE SEQUENCIA – PAGAR CONDOMINIO	19
FIGURA 20 - DIAGRAMA DE SEQUÊNCIA – REGISTRAR MANUTENÇÕES	19
FIGURA 21 - QUADRO KANBAN 15 DIAS	22
FIGURA 22 - VALOR FINAL	22
FIGURA 23 - TESTES COM JUNIT	25
FIGURA 24 - TABELA CADASTROPACIENTES.....	29
FIGURA 25 - TABELA CADASTROVACINAS	29
FIGURA 26 - TABELA APLICACAOVACINA	29
FIGURA 27 - CRUD CURSO	33
FIGURA 28 - CREATE CURSO	33
FIGURA 29 - CREATE ALUNO.....	33
FIGURA 30 - CRUD ALUNO.....	33
FIGURA 31 - PÁGINA INICIAL TOP PET	35
FIGURA 32 - PÁGINA DE AGENDAMENTO TOP PET.....	35
FIGURA 33 - CONTATO TOP PET.....	36
FIGURA 34 - SOBRE A CLINICA TOP PET	36
FIGURA 35 - SERVIÇOS OFERECIDOS TOP PET	37
FIGURA 36 - COMANDO DOCKER.....	40
FIGURA 37 - IMAGEM DO CONSOLE	40
FIGURA 38 - IMAGEM DO BROWSER	40
FIGURA 39 - COMMIT REALIZADO	41
FIGURA 40 - IMAGEM DO BROWSER 2	41

SUMÁRIO

1 PARECER TÉCNICO.....	8
2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....	10
2.1 ARTEFATOS DO PROJETO.....	11
3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2.....	15
3.1 ARTEFATOS DO PROJETO.....	16
4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2	20
4.1 ARTEFATOS DO PROJETO.....	21
5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO	24
5.1 ARTEFATOS DO PROJETO.....	25
6 DISCIPLINA: BD – BANCO DE DADOS.....	27
6.1 ARTEFATOS DO PROJETO.....	28
7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO	30
7.1 ARTEFATOS DO PROJETO.....	31
8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2	32
8.1 ARTEFATOS DO PROJETO.....	33
9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE.....	34
9.1 ARTEFATOS DO PROJETO.....	35
10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....	38
11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)	39
11.1 ARTEFATOS DO PROJETO.....	40
12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS	42
12.1 ARTEFATOS DO PROJETO.....	43
13 CONCLUSÃO	44
REFERÊNCIAS.....	45

1 PARECER TÉCNICO

O presente parecer técnico tem como objetivo demonstrar como os projetos/trabalhos desenvolvidos ao longo das disciplinas interagem para formar um processo de desenvolvimento ágil de *software* realmente integrado, se complementando a cada disciplina desenvolvida, demonstrada através dos artefatos gerados por cada uma das atividades práticas realizadas – do *deployment* de *pipelines* *DevOps* através de (Docker, 2025), GitLab (Gitlab, 2025) e Jenkins (Jenkins, 2025), à implementação das suítes de testes automatizados com Selenium (Selenium, 2025) e Maven (Apache, 2025). Conforme Farley, “Liberar *software* é muitas vezes uma arte; deveria ser uma disciplina de engenharia”. Apesar de desenvolvidos separadamente, é possível enxergar como caminham para um mesmo objetivo comum, e em contextos reais, quando integrados, geram ciclos iterativos e incrementais, garantindo entregas discretas, retroalimentação rápida e flexibilidade para a mudança. Além disso, tornam possível a entrega de maior qualidade, em menor tempo e com menor custo.

Todo este processo depende da integração das diversas disciplinas que, em conjunto, formam um ciclo contínuo de entrega de valor. A Modelagem Ágil garante que requisitos sejam traduzidos em artefatos de design adaptáveis (Ambler, 2002, p. 23), enquanto o Gerenciamento Ágil de Projetos estrutura frameworks e metodologias ágeis ao dia a dia de uma equipe de desenvolvimento (Schwaber; Sutherland, 2020). A disciplina de banco de dados foi responsável por todo o alicerce da camada de persistência de dados, e possui 4 componentes básicos: dados, *hardware*, *software* e usuários. Esta conceituação é necessária para entender a integração com outras disciplinas, e pode ser evidenciada no uso dos *frameworks* WEB e mobile vistos nesta especialização, quando integrados ao banco.

Além disso, as disciplinas de Desenvolvimento Web e Mobile habilitam a codificação em diferentes *frameworks* para diferentes dispositivos, enquanto UX no Desenvolvimento Ágil proporciona uma melhor experiência ao usuário, que através de seu feedback, assegura que interfaces sejam validadas e refinadas a cada iteração. Juntas, essas áreas — infraestrutura, modelagem, programação, banco de dados, web/mobile, testes e UX — buscam um objetivo final único, através de um processo colaborativo, flexível e orientado à qualidade, capacitando equipes a responder com agilidade às mudanças de requisitos e a entregar soluções robustas” (Sommerville, 2018, p. 32). Através do processo colaborativo, flexível e orientado à qualidade,

capacitando equipes a responder com agilidade às mudanças de requisitos e a entregar soluções robustas, de qualidade, e centradas no usuário final, que deve perceber, ao final, um *software* que não tenha dificuldades no seu uso no dia-a-dia.

O projeto Infraestrutura para Desenvolvimento e Implantação definiu a fundação para automação de builds, testes e *deploys*, garantindo que as equipes possam aproveitar ambientes reproduzíveis e padronizados (Newman, 2015). Essa padronização, com gerenciamento de containers configurados de forma idêntica às instâncias de produção, pode ter integrados Testes automatizados, que reduzem falsos positivos e garantem maior confiabilidade nas respostas (Fowler, 2006). Similarmente, todo o processo de criação e gestão do desenvolvimento, é demonstrado pelas disciplinas de Modelagem Ágil e Gerenciamento de Projetos

A disciplina de Testes Automatizados reforçou práticas de validação rápida de evoluções nos *softwares*. A execução de testes a cada *commit* fortalece o alinhamento com os valores ágeis, pois oferece retorno imediato não apenas ao desenvolvedor, mas a todo o time, impulsionando correções incrementais e evitando a propagação de defeitos (Duvall *et al.*, 2007). Sendo assim, a realização de testes a cada entrega, assegura que alterações em componentes de infraestrutura ou da lógica de negócio não quebrem funcionalidades existentes.

Automatizar seus testes repetitivos pode ser um grande divisor de águas na sua vida como desenvolvedor de *software*. Automatize esses testes e você não precisará mais seguir protocolos de cliques sem pensar para verificar se seu *software* ainda funciona corretamente. Automatize seus testes e você poderá mudar sua base de código sem piscar. (Vocke, 2018, par. 10, tradução nossa)

Por fim, todo o conjunto de projetos aqui demonstrados, tem em cada disciplina a contribuição de elementos indispensáveis para a construção de um ciclo ágil robusto. A disciplina de Infraestrutura proporciona o contexto de execução estável, a de Modelagem Ágil estabelece as bases conceituais, as de Programação e Banco de Dados materializam a lógica e o armazenamento, e os Testes Automatizados monitoram continuamente a saúde do sistema. A sinergia entre as disciplinas aqui vivenciadas é peça-chave para o sucesso de projetos de *software* ágil.

2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

A disciplina de métodos ágeis para desenvolvimento de *software* trouxe conceitos fundamentais que devem ser aplicados ao desenvolvimento de *softwares*. Abordou modelos tradicionais do processo de *software*, bem como trouxe princípios, ferramentas, *frameworks* e metodologias, muito utilizados no desenvolvimento de *softwares* modernos.

O artefato produzido nesta disciplina é uma coletânea de todos estes conceitos, que podem ser aplicados em todas as fases do desenvolvimento de *software*. O quadro Kanban por exemplo, conceituado nesta disciplina, é utilizado de forma prática na disciplina de gerenciamento ágil de projetos de *software*. Kanban é um sistema que permite visualizar o fluxo de trabalho, limitar o trabalho em progresso e maximizar o fluxo” (Anderson, 2010, p. 15). Os princípios ágeis, permeiam toda a trilha percorrida nesta especialização, a simplicidade, experiência técnica e auto-organização pode ser visto de forma contundente no refatoramento *clean code* do artefato produzido na disciplina de aspectos ágeis. A satisfação do cliente, é a base da disciplina de UX, ou experiência do usuário. Metodologias como *extreme programming*, podem facilmente ser aplicadas durante a codificação, experiência prática trazida por disciplina como desenvolvimento WEB (forma abreviada e popular para se referir a *World Wide Web*) e desenvolvimento *mobile*.

Pode-se dizer que, enquanto outras disciplinas trazem conhecimento técnicos mais evidenciados, a disciplina de métodos ágeis traz toda a conceituação dos princípios e ferramentas que guiarão a aplicação prática dos conhecimentos trazidos em outras disciplinas.

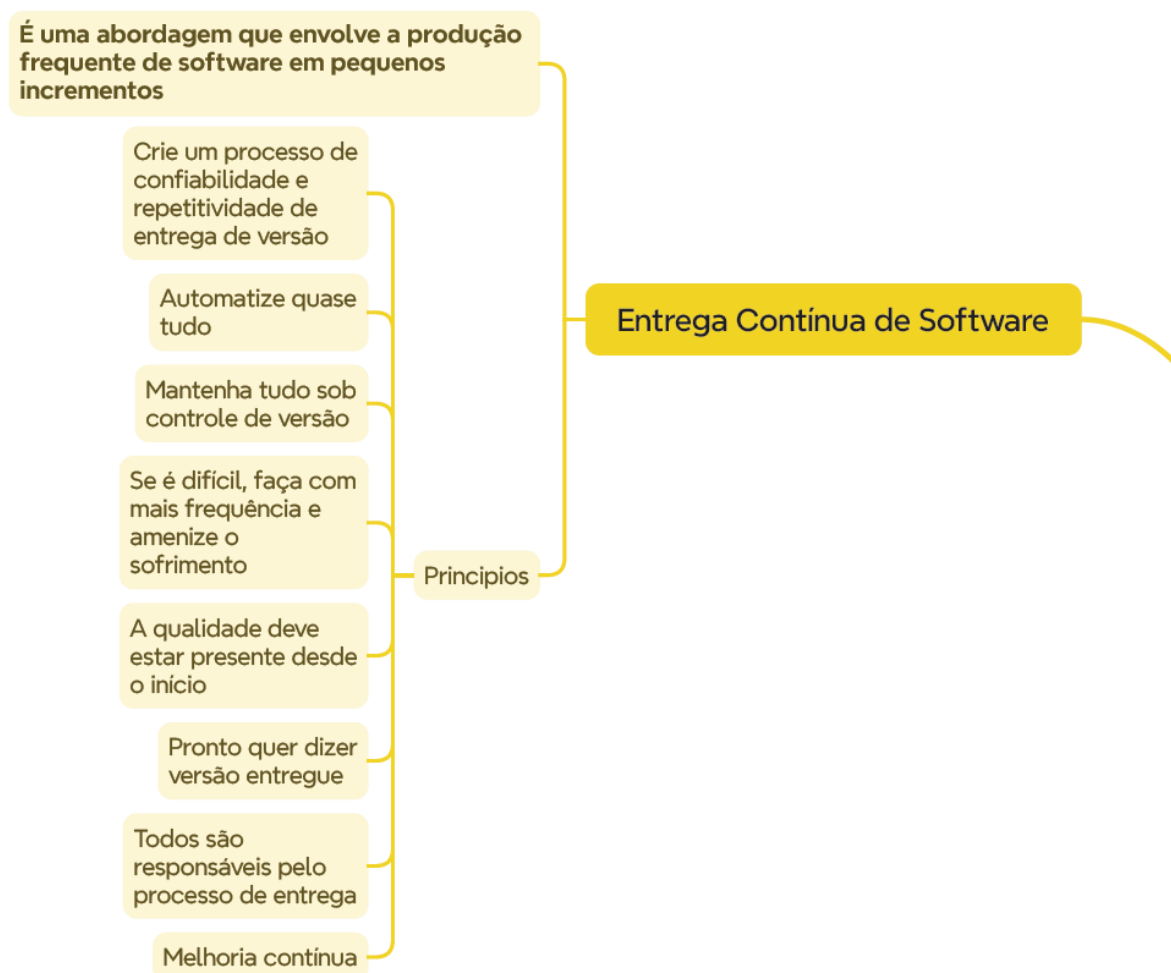
2.1 ARTEFATOS DO PROJETO

FIGURA 1 – MAPA MENTAL



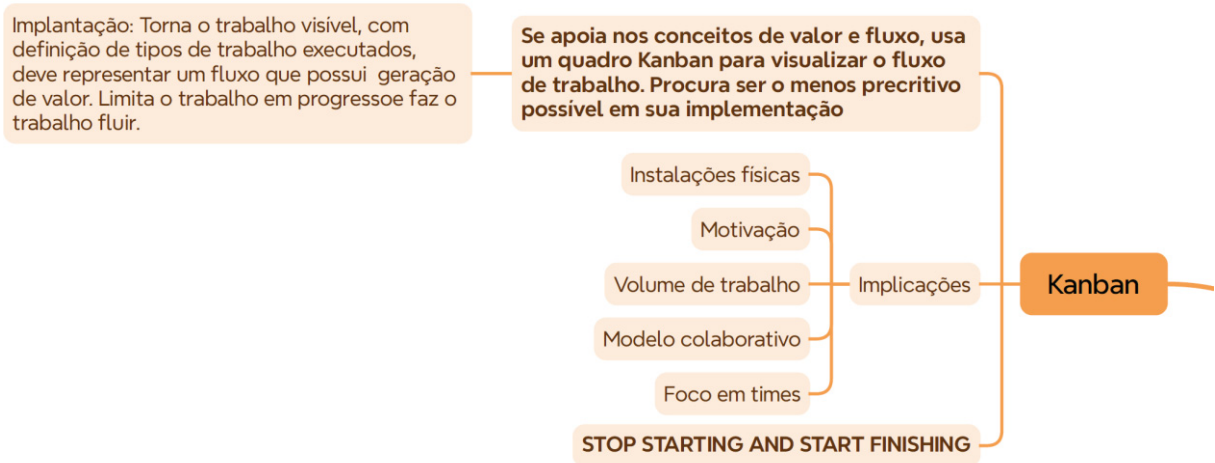
FONTE: (Autor, 2025)

FIGURA 2 - ENTREGA CONTÍNUA



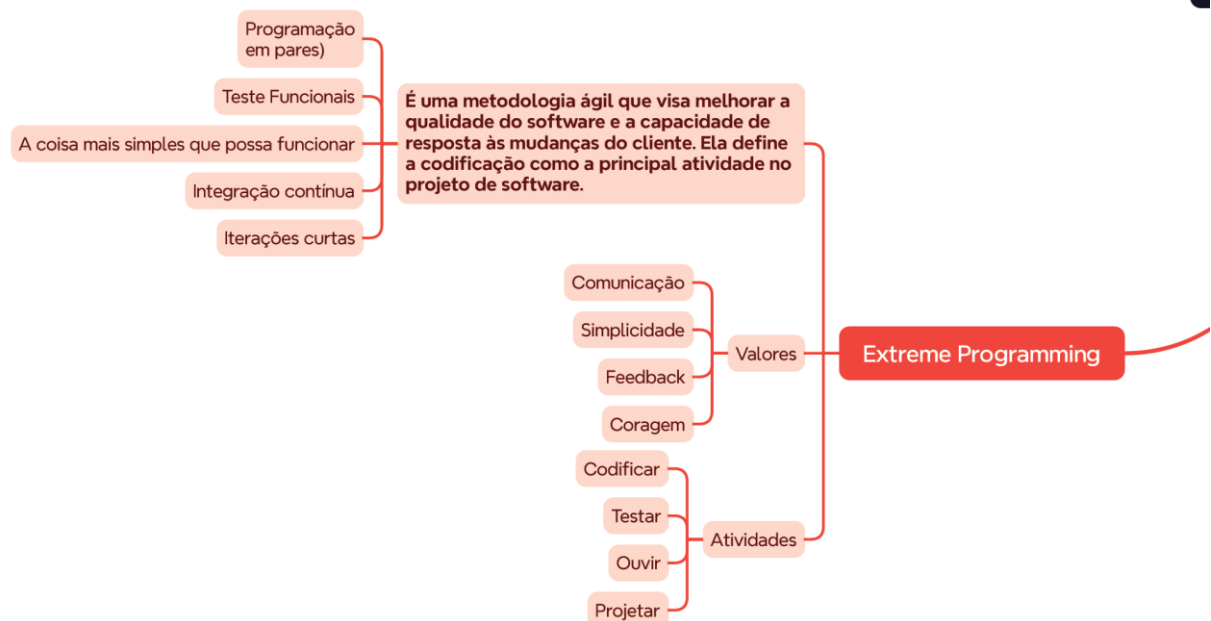
FONTE: (Autor, 2025)

FIGURA 3 - KANBAN



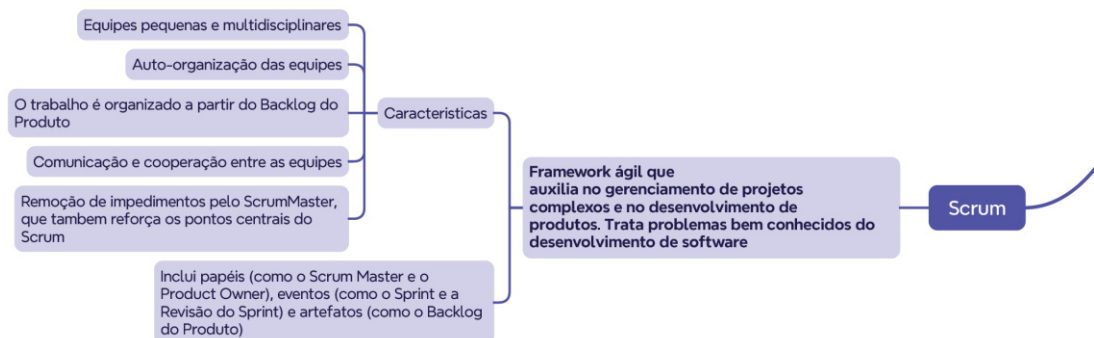
FONTE: (Autor, 2025)

FIGURA 4 - EXTREME PROGRAMMING



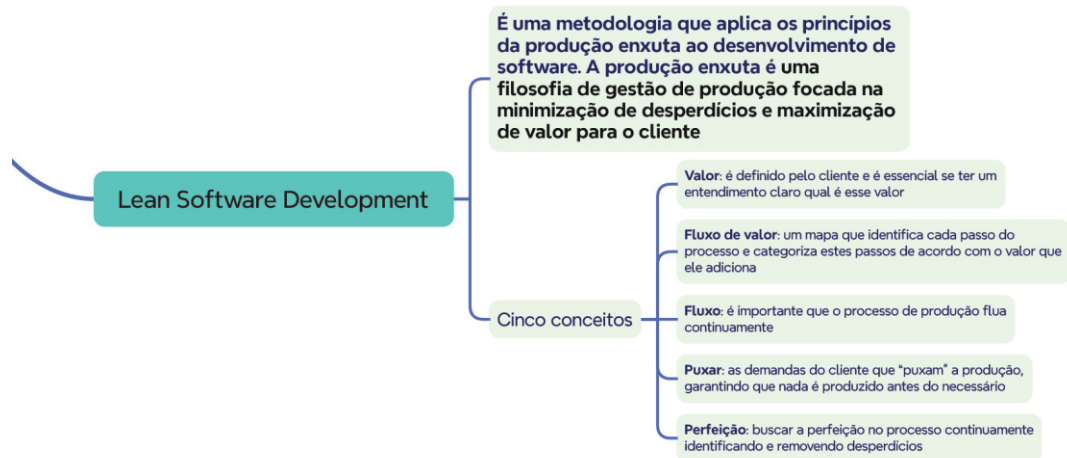
FONTE: (Autor, 2025)

FIGURA 5 - SCRUM



FONTE: (Autor, 2025)

FIGURA 6 - LEAN SOFTWARE DEVELOPMENT



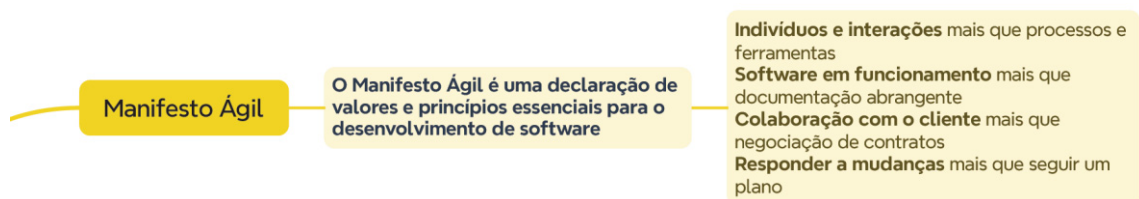
FONTE: (Autor, 2025)

FIGURA 7 - PRINCIPIOS ÁGEIS



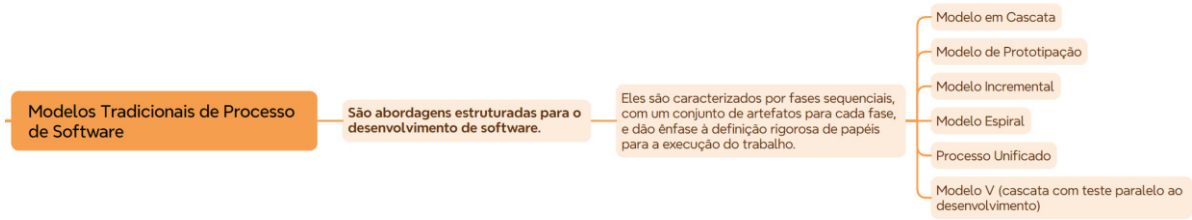
FONTE: (Autor, 2025)

FIGURA 8 - MANIFESTO ÁGIL



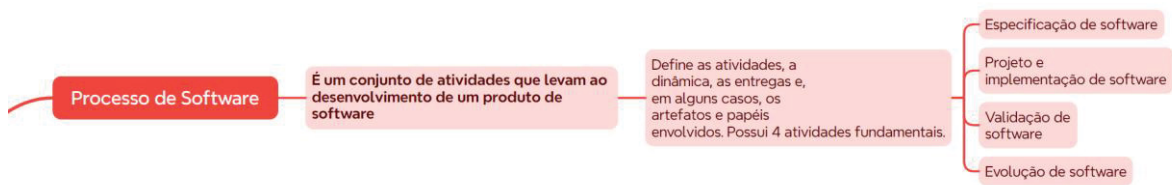
FONTE: (Autor, 2025)

FIGURA 9 - MODELOS TRADICIONAIS DE PROCESSO DE SOFTWARE



FONTE: (Autor, 2025)

FIGURA 10 - PROCESSO DE SOFTWARE



FONTE: (Autor, 2025)

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

O projeto do Sistema de Gestão de Condomínio, criado na matéria de Modelagem Ágil de *Software*, é uma aplicação real dos princípios das metodologias ágeis, mostrando a importância de uma modelagem bem estruturada no desenvolvimento de *Softwares*. Através da criação desse projeto pudemos aprofundar e fixar o conhecimento sobre a teoria de Especificação de Requisitos, diagrama de casos de uso e histórias de usuários.

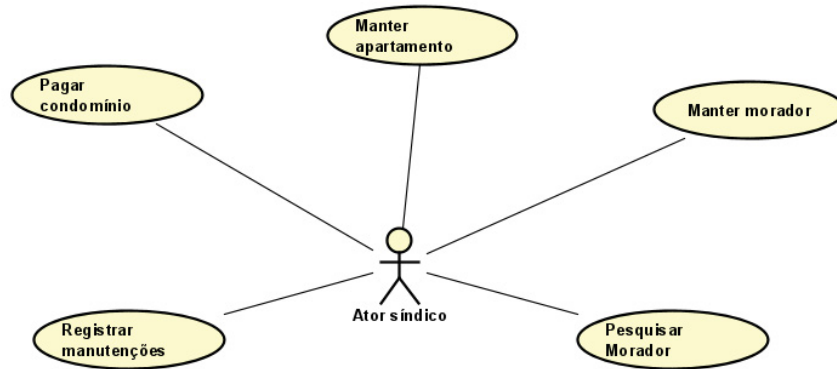
A aplicação das histórias de usuário (HU001 a HU005) mostra como os conceitos básicos do desenvolvimento ágil são usados na prática. Cada história seguiu a estrutura padrão "COMO... EU QUERO... PARA...", garantindo que as funções fossem definidas sempre do ponto de vista do usuário final, o síndico. Os critérios de aceitação detalhados definiram condições claras e que podem ser medidas para validar as funções.

A definição clara dessas regras facilita as etapas posteriores no desenvolvimento de um *software*, contextualizando a disciplina no desenvolvimento ágil. O modelo de dados presente nas histórias de usuário fornece informações diretas para uma modelagem de banco, onde as entidades Morador, Apartamento, Pagamento e Manutenção são organizadas em esquemas relacionais. Ao mesmo tempo, as especificações de interface definidas nas histórias guiam o desenvolvimento das camadas de apresentação nas matérias de Desenvolvimento Web, Mobile e as expectativas referente ao usuário que impactam em sua percepção do *software* (UX).

O projeto também estabelece bases para a matéria de Testes Automatizados por meio dos critérios de aceitação para testes especificados. Para estes cenários definidos, podem ser criados testes automatizados.

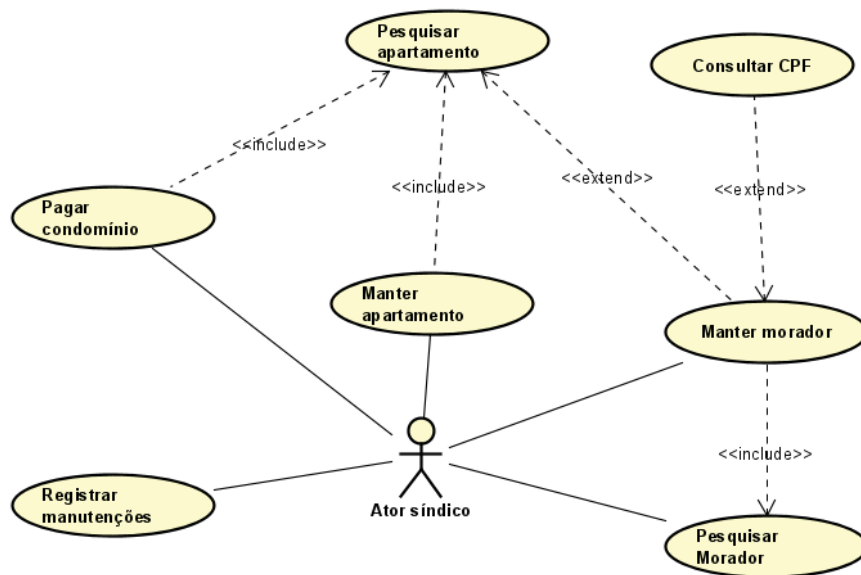
3.1 ARTEFATOS DO PROJETO

FIGURA 11 - CASO DE USO DE NÍVEL 1



FONTE: (Autor, 2025)

FIGURA 12 - CASO DE USO DE NÍVEL 2



FONTE: (Autor, 2025)

FIGURA 13 - HISTÓRIA DE USUÁRIOS – GESTÃO DE CONDOMÍNIO

HU001 – Gestão de Condomínio
SENDO O Síndico
QUERO Um sistema para gerenciamento de condomínio
PARA Automatizar os serviços

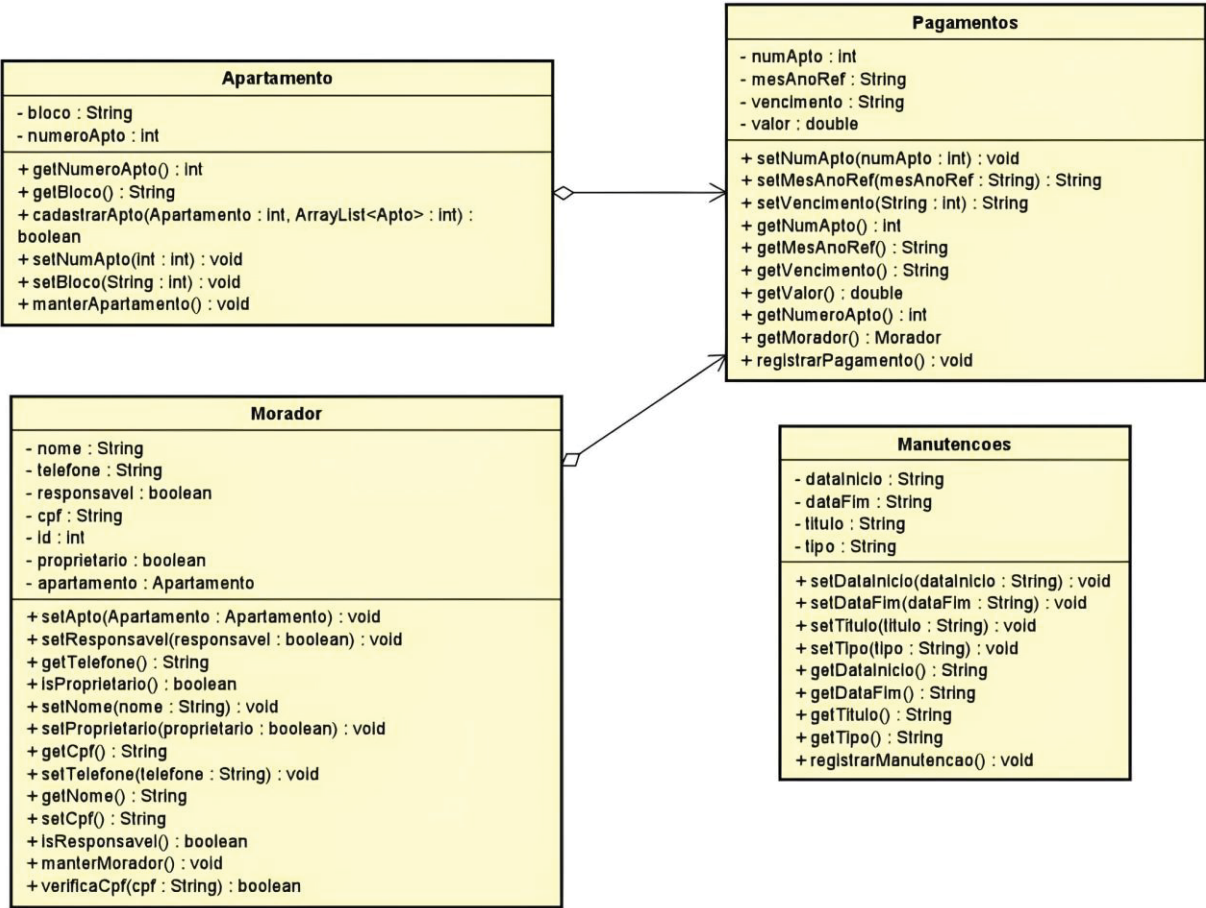
FONTE: (Autor, 2025)

FIGURA 14 - HISTÓRIA DE USUÁRIOS – MANTER MORADOR

HU002 – Manter Morador
SENDO o SÍNDICO
QUERO manter os dados dos moradores
PARA que seus dados fiquem atualizados

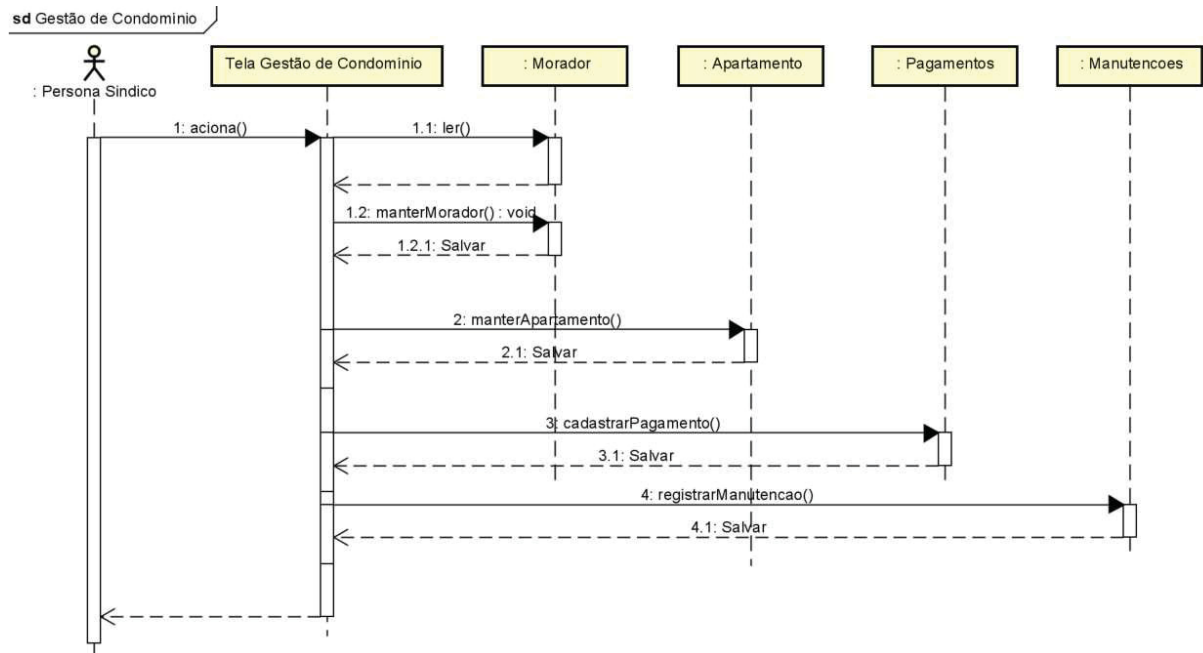
FONTE: (Autor, 2025)

FIGURA 15 - DIAGRAMA DE CLASSES



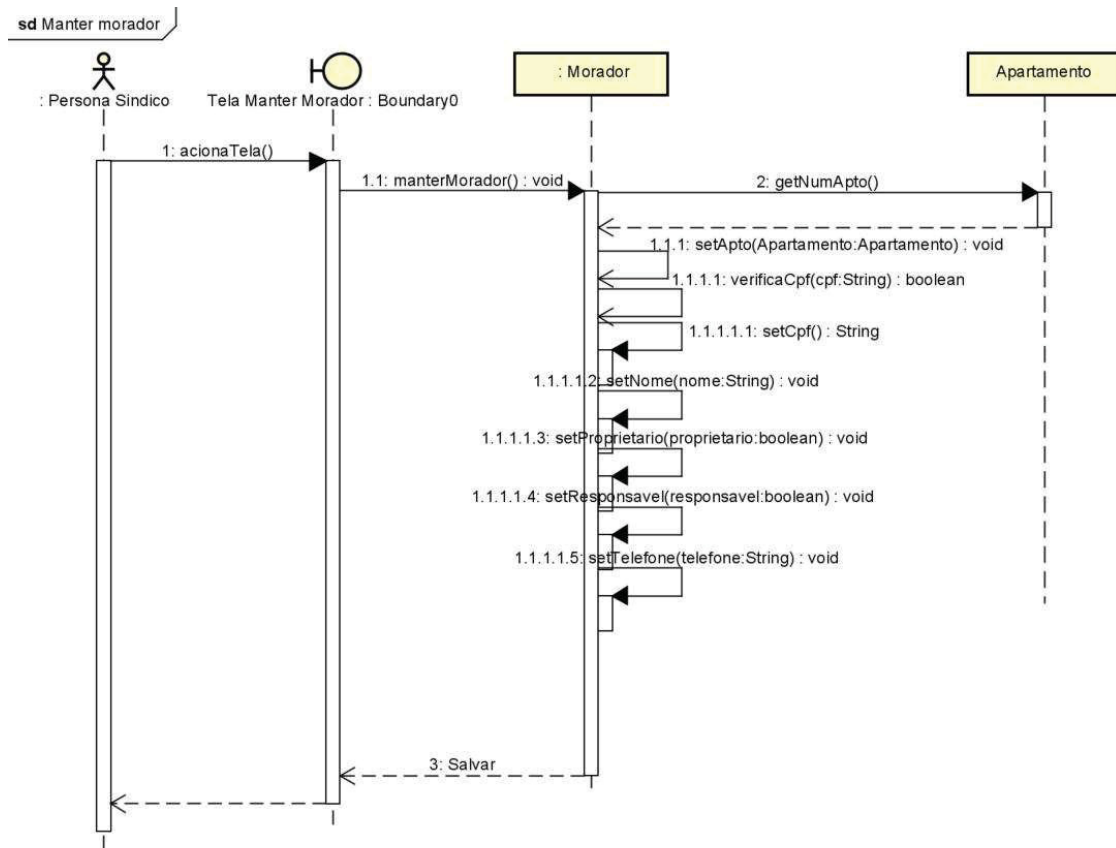
FONTE: (Autor, 2025)

FIGURA 16 - DIAGRAMA DE SEQUENCIA – GESTAO DE CONDOMINIO



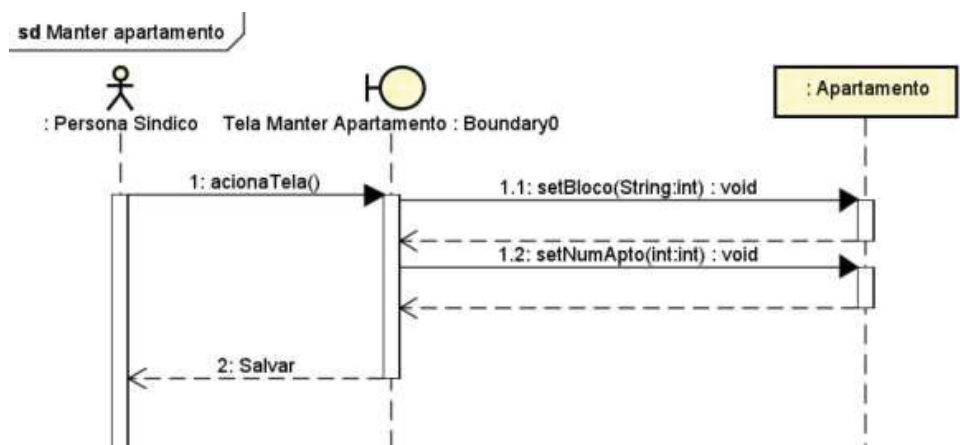
FONTE: (Autor, 2025)

FIGURA 17 - DIAGRAMA DE SEQUÊNCIA – MANTER MORADOR



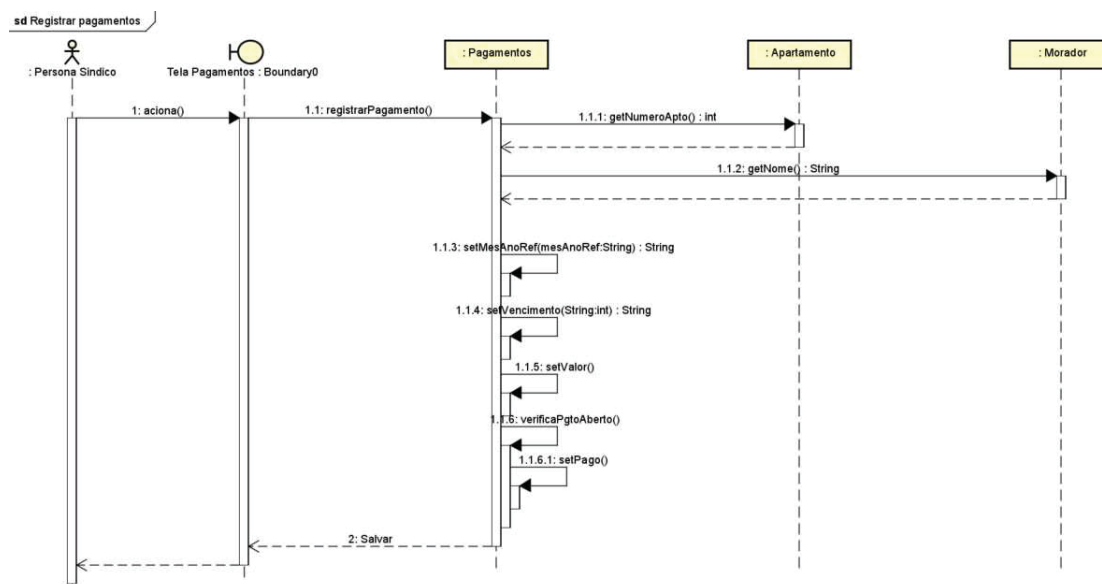
FONTE: (Autor, 2025)

FIGURA 18 - DIAGRAMA DE SEQUÊNCIA – MANTER APARTAMENTO



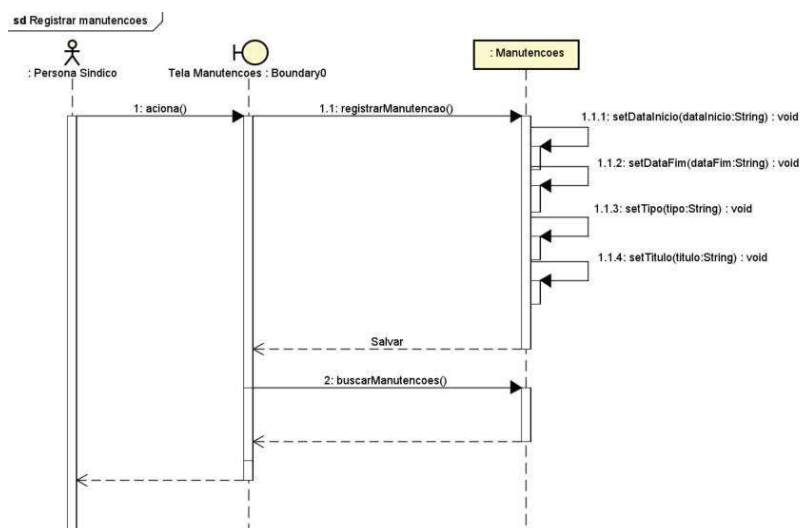
FONTE: (Autor, 2025)

FIGURA 19 - DIAGRAMA DE SEQUENCIA – PAGAR CONDOMINIO



FONTE: (Autor, 2025)

FIGURA 20 - DIAGRAMA DE SEQUÊNCIA – REGISTRAR MANUTENÇÕES



FONTE: (Autor, 2025)

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

Esta disciplina teve como objetivo conceituar e fixar o aprendizado sobre conceitos de gerenciamento de projetos, utilizando como referência o PMBOK em sua sexta edição (PMI, 2017). Sua contextualização envolveu o conceito de projetos de *software* baseados no Scrum. Scrum é um framework leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos (Schwaber; Sutherland, 2020). Com o planejamento na gestão ágil, abordaram-se temas como análise de pontos por função, contratação em projetos ágeis e planejamento de release. Também foram apresentadas técnicas como Design Thinking — “uma abordagem que utiliza métodos e ferramentas do design para gerar soluções inovadoras centradas no usuário” (Galvão; Schneider, 2023, p. 110), Design Sprint — “um processo estruturado de cinco dias para responder perguntas críticas de negócios através de design, prototipagem e testes com clientes” (Morozov, 2021, p. 8) — e Lean Inception — “um workshop colaborativo que visa alinhar pessoas para construir o produto certo” (Carolli, 2017, p. 22). Em sua segunda fase, a disciplina teve um foco em aspectos relacionados a métricas e gestão da qualidade. Os artefatos desenvolvidos compreendem um plano de release, bem como o uso do Kanban - “Kanban é um sistema que permite visualizar o fluxo de trabalho, limitar o trabalho em progresso e maximizar o fluxo” (Anderson, 2010, p. 15). Ambos permitem uma visibilidade e gerenciamento de expectativas, bem como tornam as etapas visíveis, como no caso do quadro Kanban demonstrando em tempo real a execução de atividades, permitindo saber em que passo estamos e reduzir tempos de espera e aumentar previsibilidade.

O plano foi planejado para *sprints* com duração de duas semanas, com histórias de usuários bem definidas. Utilizando estimativas para cálculo da velocidade e montagem de cada *sprint*.

Ela se integra as demais disciplinas, pois através do bom uso das informações geradas por essas ferramentas e métricas, é verificado possíveis gargalos, priorizações, alocações de recursos onde serão melhor utilizados. Em suma, é possível dizer que o planejamento afeta todo o ciclo de vida do desenvolvimento de *software*. Bem gerido, facilita as tarefas de todos os envolvidos.

4.1 ARTEFATOS DO PROJETO

QUADRO 1 – PLANO DE RELEASE

Cálculo da Velocidade:

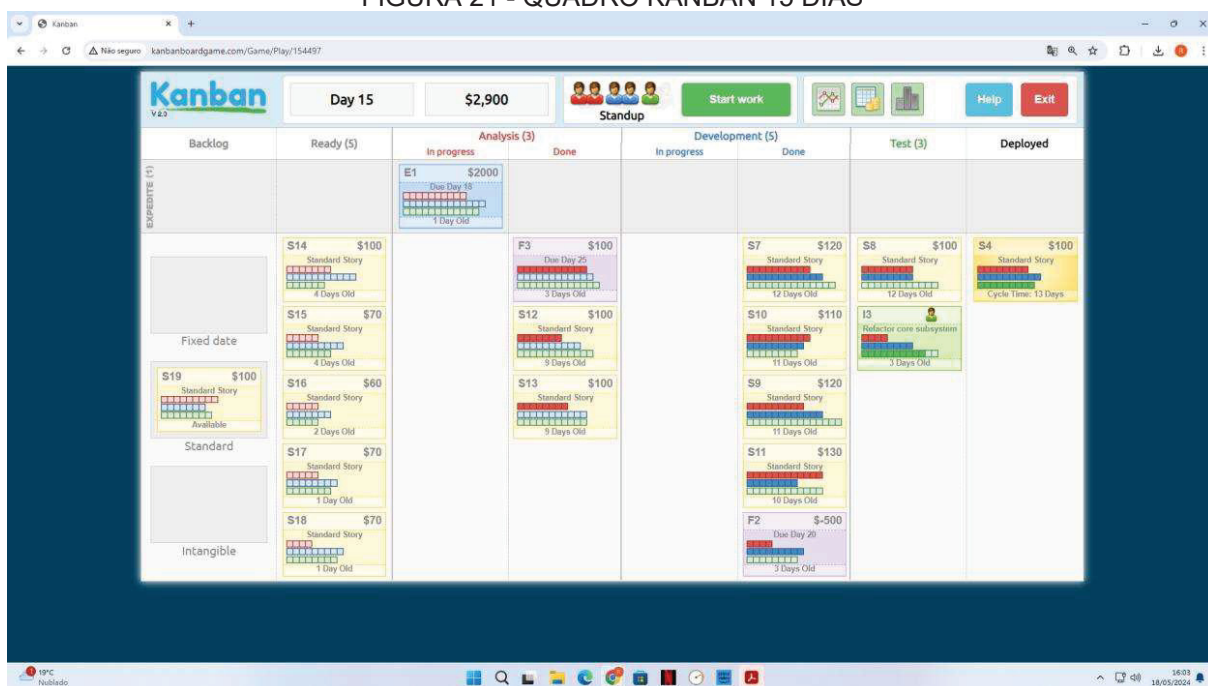
Horas disponíveis por dia:	8	Tamanho da Sprint:	Duas semanas
Horas disponíveis por Sprint:	80	Velocidade:	10

Plano de Release:

Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint N
Data Início: 01/01	Data Início: 15/01	Data Início: 29/02	Data Início: 13/03
Data Fim: 14/01	Data Fim: 28/01	Data Fim: 10/03	Data Fim: 24/03
<História de Usuário 1> SENDO um engenheiro QUERO fazer login PARA acessar o sistema de projetos ESTIMATIVA (2)	<História de Usuário 5> SENDO um engenheiro QUERO atribuir tarefas PARA delegar responsabilidades ESTIMATIVA (2)	<História de Usuário 9> SENDO um engenheiro QUERO fazer upload de arquivos PARA compartilhar documentos ESTIMATIVA (4)	<História de Usuário 13> SENDO um engenheiro QUERO definir metas PARA orientar o trabalho em equipe ESTIMATIVA (3)
<História de Usuário 2> SENDO um engenheiro QUERO criar novos trabalhos PARA iniciar novos trabalhos ESTIMATIVA (3)	<História de Usuário 6> SENDO um engenheiro QUERO acompanhar o progresso das tarefas PARA garantir o cumprimento dos prazos ESTIMATIVA (2)	<História de Usuário 10> SENDO um engenheiro QUERO fazer anotações PARA registrar ideia ou problemas ESTIMATIVA (2)	<História de Usuário 14> SENDO um engenheiro QUERO visualizar o histórico PARA entender mudanças ao longo do tempo ESTIMATIVA (2)
<História de Usuário 3> SENDO um engenheiro QUERO adicionar detalhes ao projeto PARA documentar o progresso ESTIMATIVA (2)	<História de Usuário 7> SENDO um engenheiro QUERO visualizar relatórios de progresso PARA avaliar a eficiência da equipe ESTIMATIVA (3)	<História de Usuário 11> SENDO um engenheiro QUERO programar reuniões PARA discutir o projeto ESTIMATIVA (2)	<História de Usuário ...> SENDO um engenheiro QUERO exportar o projeto PARA levar aos stakeholders ESTIMATIVA (3)
<História de Usuário 4> SENDO um engenheiro QUERO compartilhar informações PARA manter todos informados ESTIMATIVA (3)	<História de Usuário 8> SENDO um engenheiro QUERO receber notificações PARA estar ciente de atualizações e modificações ESTIMATIVA (3)	<História de Usuário 12> SENDO um engenheiro QUERO criar listas de validação PARA garantir o cumprimento das tarefas ESTIMATIVA (2)	<História de Usuário N> SENDO um engenheiro QUERO gerenciar permissões dos meus projetos PARA controlar o acesso ESTIMATIVA (2)

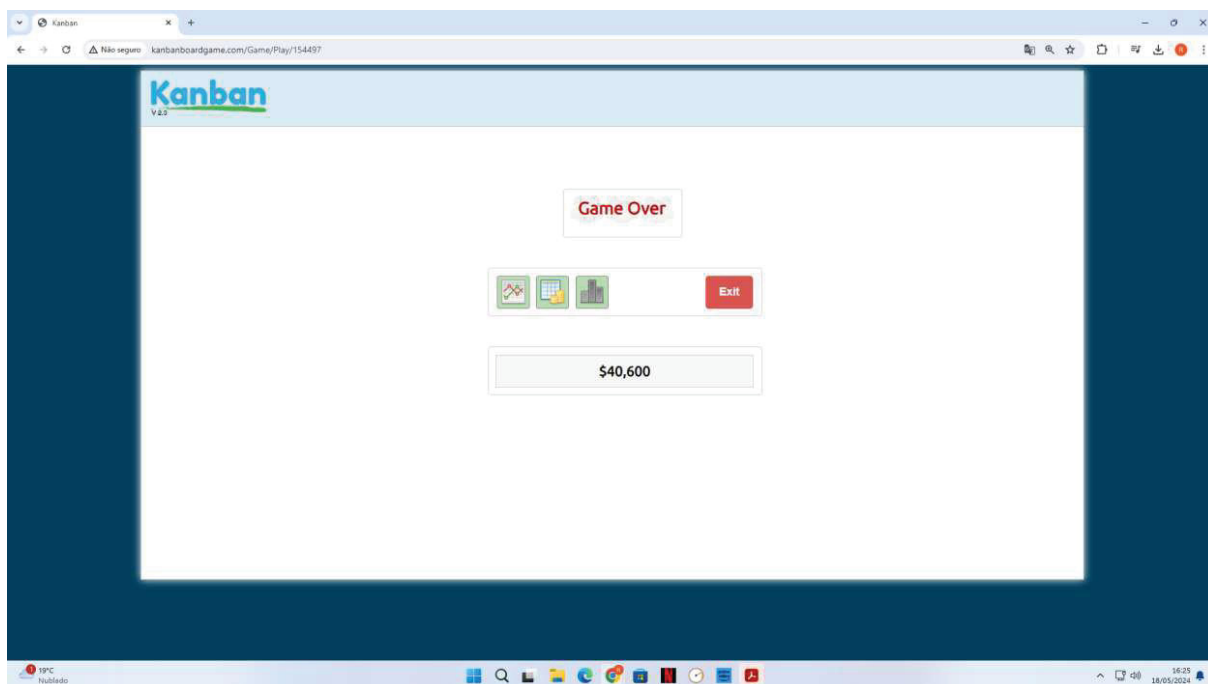
FONTE: (Autor, 2025)

FIGURA 21 - QUADRO KANBAN 15 DIAS



FONTE: (Autor, 2025)

FIGURA 22 - VALOR FINAL



FONTE: (Autor, 2025)

GRÁFICO 1 - CFD - Cumulative Flow Diagram



FONTE: (Autor, 2025)

5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

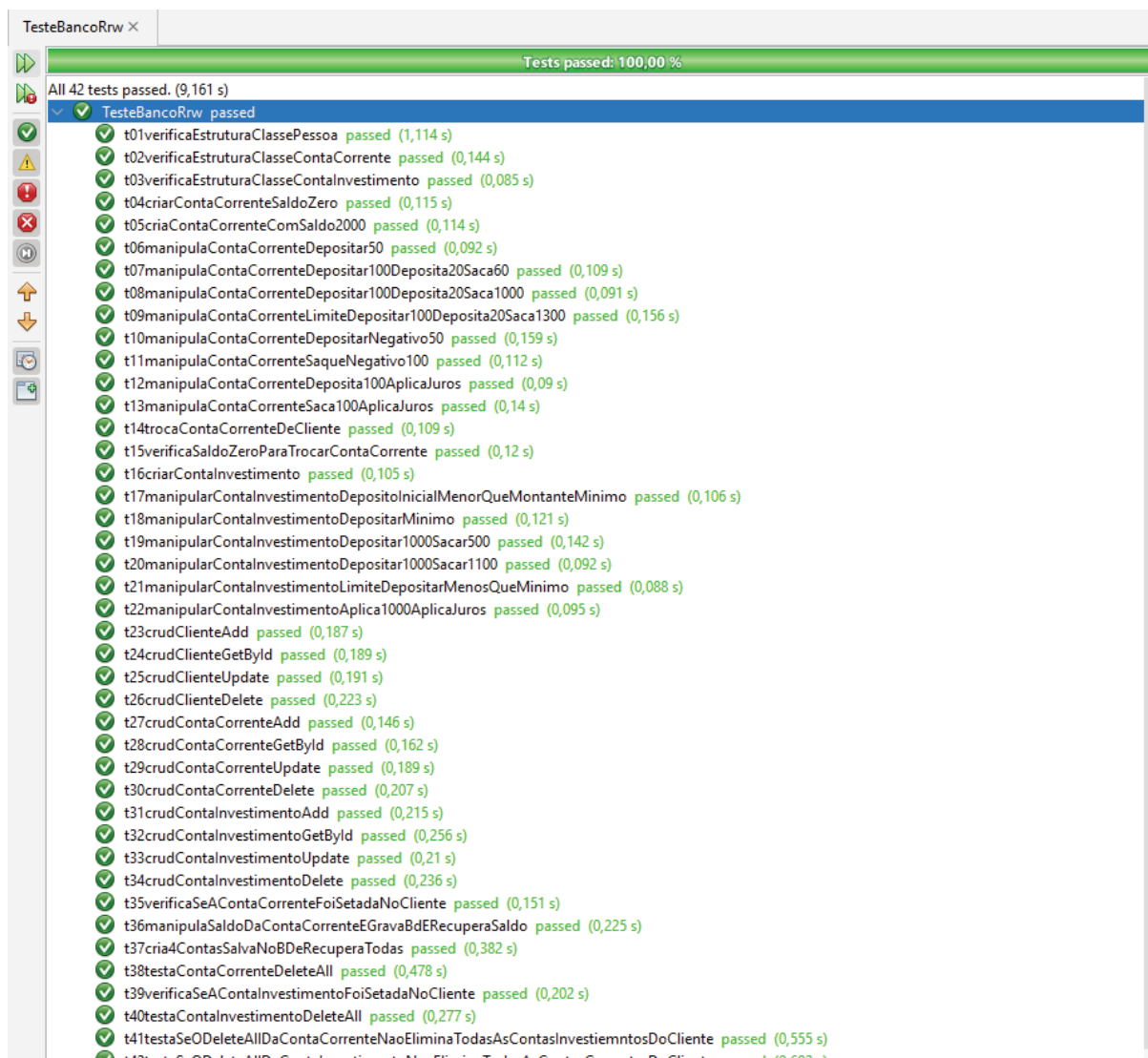
Nesta disciplina de Introdução à Programação, o projeto desenvolvido pela nossa equipe foi o *backend* de um sistema bancário simplificado de controle de cadastros de clientes, contas correntes e contas investimento. Para a implementação desse projeto, utilizou-se as tecnologias Java (Oracle, 2025), MySQL (Oracle, 2025), sendo o projeto centrado em construir classes de domínio e persistência de dados a partir de um diagrama de classes, validando as implementações através de testes unitários automatizados através do JUnit (JUnit team, 2025). Nesta disciplina, as práticas adotadas refletem práticas do desenvolvimento ágil, uma vez que a utilização de testes automatizados permite um rápido *feedback* no desenvolvimento de *software*.

Fomos encorajados, através desta didática, a corrigir todos os erros de maneira incremental ao longo do tempo, alinhar melhor os requisitos funcionais com a implementação pretendida, e, por fim, obter a cobertura funcional desejada. Este processo de codificação, teste, e correção é fundamental para as métricas de *software* da metodologia ágil. Além do referencial fundamental das competências de programação, a atividade como descrita também estabeleceu contatos diretos com as métricas de outras disciplinas do curso. O uso de um pool de conexões com banco de dados refletiu o paradigma da disciplina Banco de Dados na exigência de implementação de operações de persistência e manipulação de dados via SQL.

O uso em testes gerou subsídios para a disciplina de Testes de *Software*, estimulando a compreensão de práticas como TDD (*Test Driven Development* - “*TDD é uma prática em que os testes são escritos antes do código, guiando o design do software e garantindo cobertura de testes desde o início*” (Beck, 2010, p. 27). Por fim, a análise de requisitos através do diagrama de classes (Diagrama 8) e da cobertura dos testes contribuiu para uma abordagem mais madura e integrada com Engenharia de *Software* e Modelagem de Sistemas. Assim, o projeto serviu como base para compreensão da parte mais técnica relacionada ao desenvolvimento ágil como um todo, unindo teoria e prática em um ambiente didático e colaborativo.

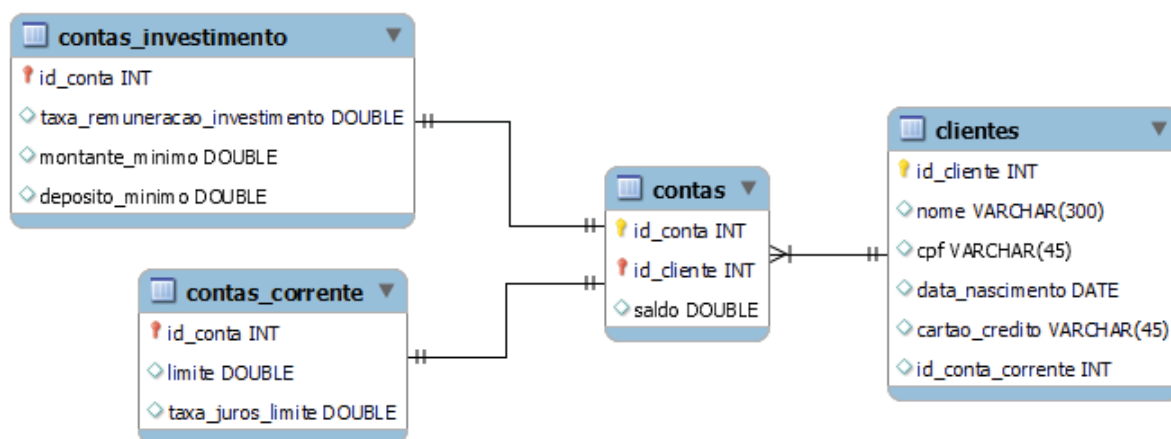
5.1 ARTEFATOS DO PROJETO

FIGURA 23 - TESTES COM JUNIT



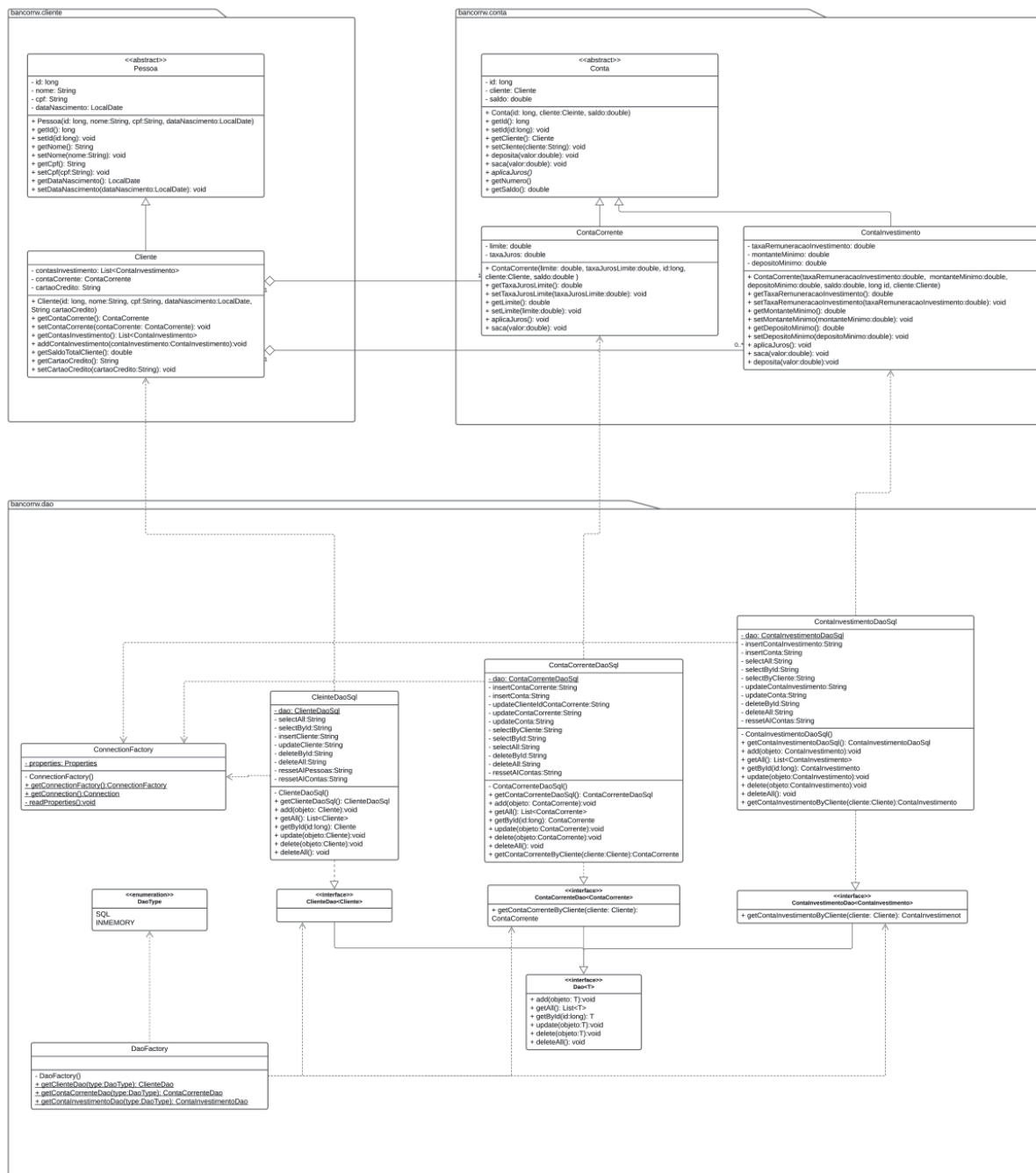
FONTE: (Autor, 2025)

DIAGRAMA 1 - ER BANCO CRIADO



FONTE: (Autor, 2025)

DIAGRAMA 2 - DIAGRAMA DE CLASSES



FONTE: (Autor, 2025)

6 DISCIPLINA: BD – BANCO DE DADOS

O projeto da disciplina de Banco de Dados é elemento fundamental na formação acadêmica de desenvolvimento de *software*, apresentando abordagem sistemática e prática ao gerenciamento de informações organizacionais. Os sistemas de gerenciamento de banco de dados estão cada vez mais presentes, superando antigos métodos que possuíam mais falhas e inconsistências (Silva; Nascimento; Santos, 2019). O trabalho realizado pela equipe contou com dois tópicos: um sistema de gerenciamento bibliotecário e um destinado ao controle de vacinação.

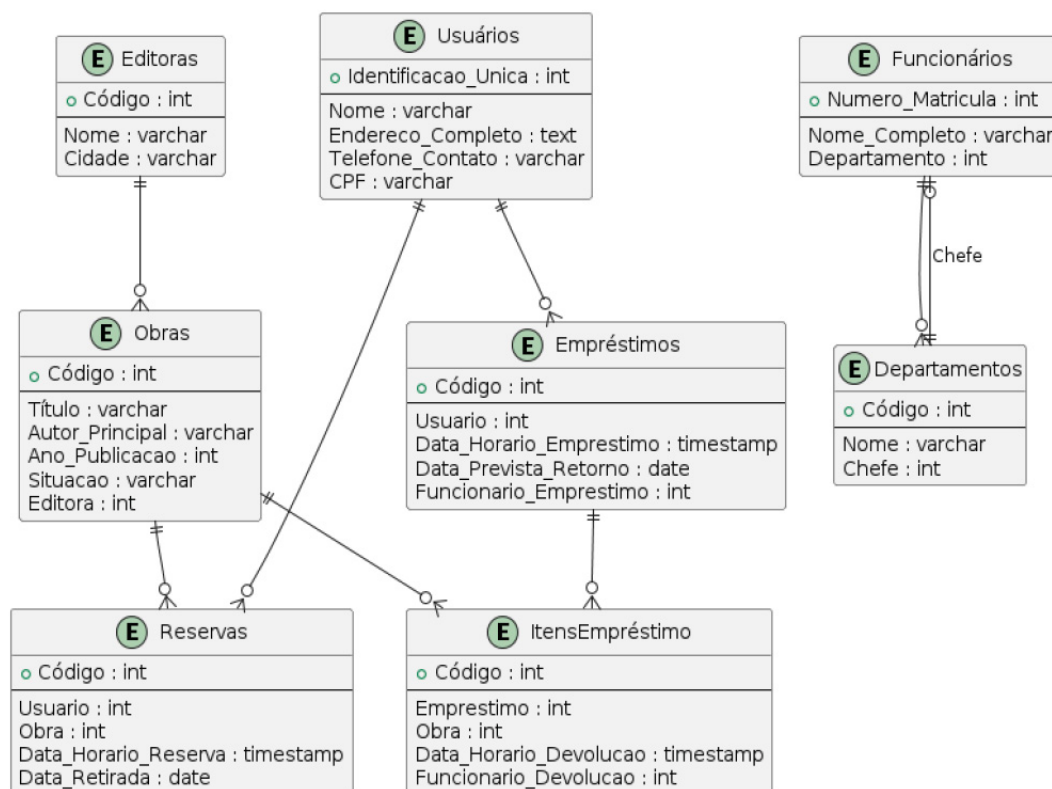
O desenvolvimento conduzido de modelagem de dados, representado nas relações complexas entre entidades como usuários, empréstimos, obras e funcionários, mostra a necessidade de estruturas flexíveis para desenvolvimento iterativo e incremental. A natureza colaborativa do projeto, desenvolvido em equipe, reflete as práticas contemporâneas onde comunicação efetiva e trabalho conjunto constituem elementos críticos para o sucesso.

O projeto estabelece relações sinérgicas com todo o conjunto de disciplinas. Através da modelagem de dados e alinhamento de requisitos, podemos verificar sua integração com a modelagem ágil de *software*. A disciplina de aspectos ágeis da programação está ligada pela produção de consultas SQL flexíveis e eficientes, enquanto aplicativos *web* e *mobile* são fundamentalmente desenvolvidos sobre bancos de dados otimizados para assegurar desempenho razoável. Quanto a experiência do usuário, usuários podem ter a percepção de uma interface mais lenta e menos reativa caso o banco seja mal projetado. A abordagem multidisciplinar está preparando profissionais para apreender interdependências entre componentes do sistema, competência essencial em equipes ágeis de hoje.

O projeto passa além de objetivos técnicos específicos, sendo experiência formativa completa que desenvolve habilidades fundamentais do mercado profissional. A complexidade das estruturas de dados oferece compreensão global dos processos de desenvolvimento, do trabalho de análise de requisitos até implantação e validação. A formação integrada é fundamental para profissionais que podem trabalhar efetivamente em ambientes ágeis, na qual a flexibilidade, comunicação e percepção sistêmica condicionam o sucesso dos projetos de desenvolvimento de *software*.

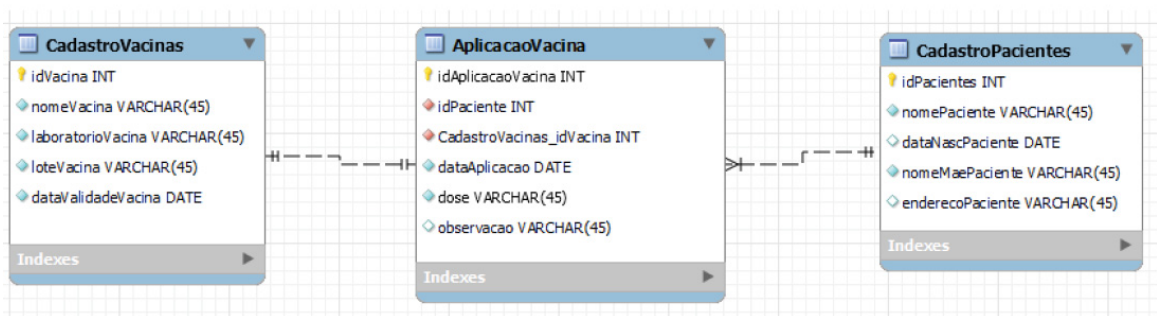
6.1 ARTEFATOS DO PROJETO

DIAGRAMA 3 – MODELO LOGICO (ER)



FONTE: (Autor, 2025)

DIAGRAMA 4 – MODELO EER



FONTE: (Autor, 2025)

FIGURA 24 - TABELA CADASTROPACIENTES

idPacientes	nomePaciente	dataNascPaciente	nomeMaePaciente	enderecoPaciente
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: (Autor, 2025)

FIGURA 25 - TABELA 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: (Autor, 2025)

FIGURA 26 - TABELA 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: (Autor, 2025)

7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

A disciplina de Aspectos Ágeis de Programação trouxe uma ampla gama de conhecimento voltados a boas práticas de desenvolvimento, como refinamento sucessivo e refatoração (Martin, 2009), trouxe a luz alguns princípios como TDD – *Test Driven Development* (Beck, 2010), BDD – *Behavior Driven Development* (North, 2006), arquitetura limpa (Martin, 2019), princípios de design como SRP (*Single Responsibility Principle*), OCP (*Open/Closed Principle*), LSP (*Liskov Substitution Principle*), ISP (*Interface Segregation Principle*), DIP (*Dependency Inversion Principle*), além do SOLID (conjunto dos cinco princípios mencionados) (Martin, 2009). Todos esses conhecimentos norteiam a codificação de um código limpo.

O projeto desenvolvido na disciplina de Aspectos Ágeis da Programação teve como foco principal a melhoria da legibilidade e manutenibilidade do código por meio da aplicação dos princípios do código limpo – *clean code* (Martin, 2009). A partir da padronização das declarações no estilo Java, foram realizados ajustes nos nomes de variáveis e métodos para garantir que fossem intuitivos e representassem claramente sua função e propósito. Além disso, métodos foram extraídos para promover a decomposição do código em partes menores, facilitando a compreensão e o entendimento do programa como um todo. A execução do sistema ocorre por meio de um método main, onde são instanciados os objetos necessários, e foi criada uma classe específica para a realização de testes do programa, assegurando sua funcionalidade.

Este projeto é fundamental para a prática do desenvolvimento ágil de *software*, pois enfatiza a importância da clareza e da simplicidade no código, princípios que reduzem a dívida técnica e facilitam alterações rápidas e seguras, características essenciais em ambientes ágeis. A habilidade de refatorar código continuamente, mantendo-o limpo e organizado, possibilita entregas frequentes e incrementais, alinhadas ao *feedback* do cliente e às adaptações constantes do processo.

A integração com outras disciplinas do curso é evidente, pois a modelagem clara e organizada do código serve de base para etapas futuras de desenvolvimento, incluindo a implementação, testes automatizados e integração contínua. E em situações mais complexas, fomentam o alinhamento entre análise de requisitos, programação, bancos de dados e testes, promovendo um resultado final que promove a qualidade final do *software*.

7.1 ARTEFATOS DO PROJETO

CÓDIGO-FONTE 1 – ORDENAÇÃO MÉTODO BOLHA

```

class Main {
    public static void main(String[] args) {
        int[] numeros = { 64, 34, 25, 12, 22, 11, 90 };
        int qtdeNumerosParaOrdenar = numeros.length;
        OrdenacaoBolhaMetodoObjeto.ordenar(numeros, qtdeNumerosParaOrdenar);
        OrdenacaoBolhaMetodoObjeto.imprimirArray(numeros, qtdeNumerosParaOrdenar);
    }
}

import java.util.stream.IntStream;

public class OrdenacaoBolhaMetodoObjeto {

    static void ordenar(int[] colecaoNumeros, int qtdeNumerosNaColecao) {
        for (int i = 0; i < qtdeNumerosNaColecao - 1; i++) {
            for (int j = 0; j < qtdeNumerosNaColecao - i - 1; j++) {
                aindaPrecisaOrdenacao(colecaoNumeros, j);
            }
        }
    }

    private static void aindaPrecisaOrdenacao(int[] colecaoNumeros, int j) {
        if (colecaoNumeros[j] > colecaoNumeros[j + 1]) {
            trocarElementos(colecaoNumeros, j, j + 1);
        }
    }

    static void trocarElementos(int[] colecaoNumeros, int indice1, int indice2) {
        int elementoTroca = colecaoNumeros[indice1];
        colecaoNumeros[indice1] = colecaoNumeros[indice2];
        colecaoNumeros[indice2] = elementoTroca;
    }

    static void imprimirArray( int[] arr, int tamanho){
        System.out.print("Array ordenado: ");
        IntStream.range(0, tamanho).mapToObj(i -> arr[i] + "
").forEach(System.out::print);
    }
}

```

FONTE: (Autor, 2025)

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

Os aplicativos *web* podem ser desenvolvidos com diversas tecnologias, apresentando funcionalidade e comportamento semelhantes aos aplicativos desenvolvidos localmente, mesmo estando vinculados à web, com a principal vantagem de poderem ser projetados independentemente do tipo de dispositivo móvel e plataforma a ser utilizada (Costa, 2018).

O projeto da disciplina consistiu na implementação de um sistema contendo 2 CRUDS, um para gerenciamento de Alunos e outro para Cursos, utilizando *Local Storage* para persistência dos dados e interfaces desenvolvidas com *Bootstrap*. Este projeto realizado de forma prática proporciona criação rápida e iterativa de funcionalidades essenciais, consolidando o conhecimento demonstrado durante as aulas em vídeo do curso.

Além disso, o projeto integra conhecimentos como arquitetura de software (na estruturação dos componentes Angular e serviços), programação (no uso de TypeScript e manipulação de dados), design UX (na utilização de Bootstrap para garantir usabilidade e estética).

O trabalho entregue, que inclui documentação em vídeo e organização da equipe, também enfatiza aspectos de apresentação e clareza que são essenciais para o sucesso em projetos de software colaborativos.

Portanto, este projeto não só consolida conhecimentos técnicos em Angular e desenvolvimento web, como também estimula habilidades sociais e organizacionais que são indispensáveis para o desenvolvimento ágil eficaz e para a integração harmoniosa com outras disciplinas do curso e práticas profissionais.

8.1 ARTEFATOS DO PROJETO

FIGURA 27 - CRUD CURSO

Curso		
Nome	Link	
UFRP	http://desenvolvimento.com	+ Novo ✎ Editar ✖ Remover

FONTE: (Autor, 2025)

FIGURA 28 - CREATE CURSO

INSIRA OS DADOS DOS CURSOS

Nome:

Link:

[Salvar](#)
[← Cancelar](#)

FONTE: (Autor, 2025)

FIGURA 29 - CREATE ALUNO

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 email do aluno.

Data de Nascimento:

[Salvar](#)
[← Cancelar](#)

FONTE: (Autor, 2025)

FIGURA 30 - CRUD ALUNO

Alunos				
Nome	CPF	E-mail	Data de Nascimento	
Ricardo M	276.664.680-91	muraoka@ufpr.br	30/01/1987	+ Novo ✎ Editar ✖ Remover

FONTE: (Autor, 2025)

9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

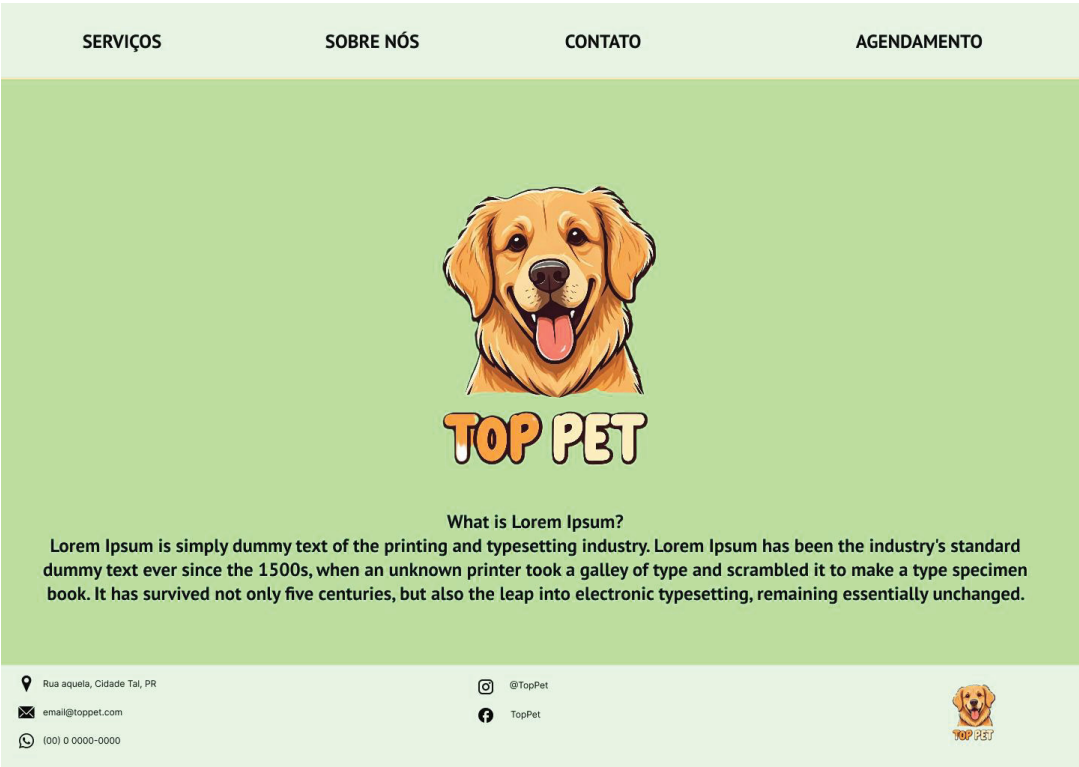
A experiência do usuário (UX) tem como principal objetivo criar interações significativas e satisfatórias para os usuários, considerando aspectos essenciais como usabilidade, acessibilidade, desempenho e estética (Reis, 2022). Um bom *design* de UX é fundamental para o sucesso de qualquer produto ou serviço digital, pois além de atrair usuários, promove sua retenção ao garantir uma experiência positiva, influenciando diretamente a reputação e resultados das organizações. Para isso, o UX reúne conhecimentos multidisciplinares de psicologia, design, tecnologia e negócios, visando desenvolver produtos que atendam às necessidades e desejos reais dos usuários, proporcionando experiências duradouras e impactantes (Reis, 2022).

O projeto da disciplina foi a criação de um protótipo de um produto de livre escolha, sendo meu projeto a de uma clínica veterinária. O processo do desenvolvimento do *design* das telas para a clínica veterinária resulta em um protótipo de cinco telas finais, todas com o mesmo traço de *design* voltado para fácil utilização, clareza visual e navegação intuitiva. A escolha das cores verdes, associadas à saúde e à natureza, assim como da tipografia, contribui para uma identidade visual mais moderna e acessível, sem perder a retomada do cuidado e da atenção. Além disso, a interface foi desenvolvida de modo a minimizar as quantidades de leitura necessárias, facilitando o entendimento rápido pelos usuários, fator crucial para casos de uso real. A imersão oferecida por projetos de UX ajuda a reforçar a interface com o usuário o mais rapidamente possível, e é importante para a usabilidade em cenários reais.

A visualização e a prototipagem rápida do modelo possibilitam a tomada de decisões constante e contínua com base no *feedback* recebido, propiciando entregas incrementais e contínuas para o alinhamento adequado entre a expectativa do usuário e a solução técnica. Este trabalho prático também permite realizar a integração com as outras disciplinas de Programação e Desenvolvimento Web, que possibilitam transformar os protótipos e as especificações em sistema efetivo; Banco de Dados, que realiza o armazenamento e estruturação das informações; e Testes de *Software*, permitindo validar as implementações das interfaces dos sistemas. Desse modo, a experiência do usuário não é um mero exercício de estética, é uma peça estratégica que define o direcionamento da construção do produto final, alinhando requisitos técnicos, funcionais e de experiência dos usuários em um ciclo iterativo de melhorias. Portanto, o *design* UX é indispensável.

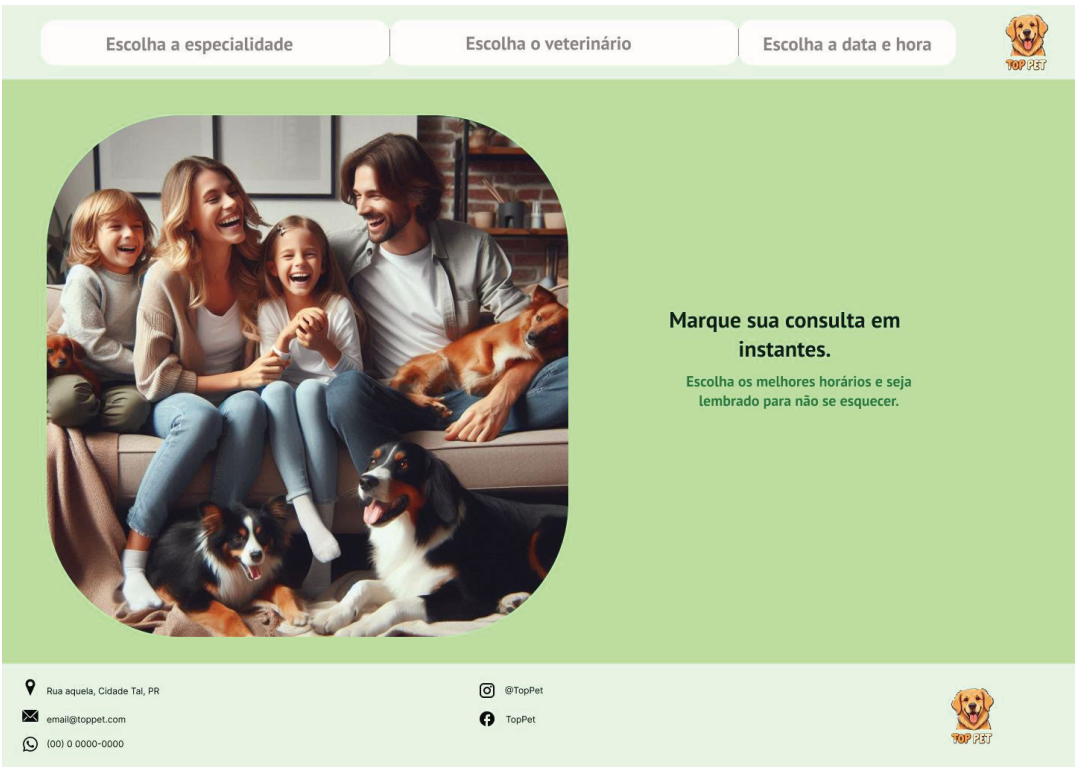
9.1 ARTEFATOS DO PROJETO

FIGURA 31 - PÁGINA INICIAL TOP PET



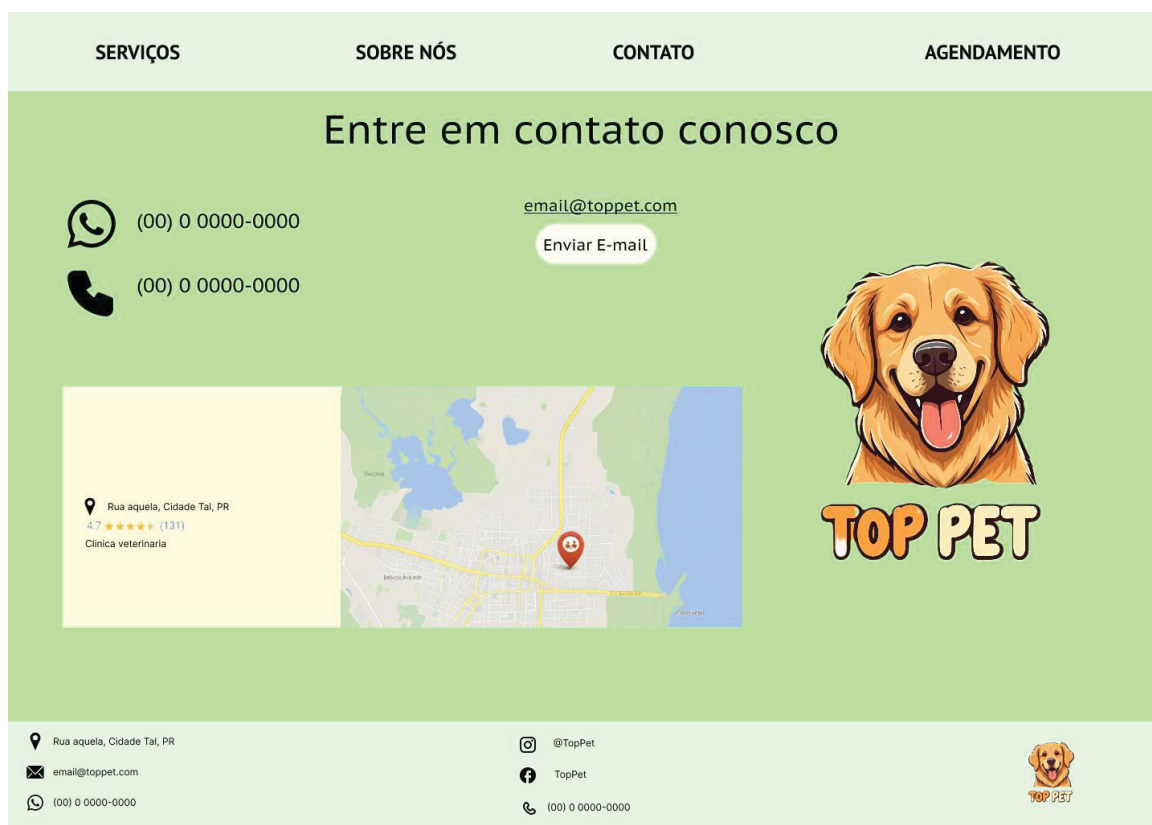
FONTE: (Autor, 2025)

FIGURA 32 - PÁGINA DE AGENDAMENTO TOP PET



FONTE: (Autor, 2025)

FIGURA 33 - CONTATO TOP PET



FONTE: (Autor, 2025)

FIGURA 34 - SOBRE A CLINICA TOP PET



FONTE: (Autor, 2025)

FIGURA 35 - SERVIÇOS OFERECIDOS TOP PET

SERVIÇOS

SOBRE NÓS

CONTATO

AGENDAMENTO



TOP PET

NOSSOS SERVIÇOS

Acupuntura

Veterinário Ipsum Lorem
Veterinária Ipsum Lorem

Clinica Geral

Veterinário Ipsum Lorem
Veterinária Ipsum Lorem

Dermatologia

Veterinário Ipsum Lorem
Veterinária Ipsum Lorem

Dermatologia

Veterinário Ipsum Lorem
Veterinária Ipsum Lorem

Rua aquela, Cidade Tal, PR
email@toppet.com
(00) 0 0000-0000

FONTE: (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 desenvolver aplicações móveis. Abordou os desafios do desenvolvimento considerando o número elevado de dispositivos existentes, com a personalização para diferentes idiomas e regiões, além do desafio de acessibilidade.

Demonstrou o processo passando por etapas como Desenvolvimento, teste e qualidade e a preparação para o lançamento. Nesta disciplina foi contextualizado sobre o Android, bem como o desenvolvimento prático era baseado para este sistema. Também trouxe a luz vantagens como desempenho, fluidez, acesso completo a APIs e funcionalidade, além da independência de *frameworks* de terceiros em apps nativos, bem como as desvantagens de custo elevado tempo no desenvolvimento e manutenções e atualizações, enquanto apps híbridos são desenvolvidos de forma rápida, com custos menores (Martins, 2020).

Assim como as disciplinas de desenvolvimento WEB, pode ser desenvolvimento dentro de um contexto ágil, com automatização das ferramentas de *deploy* e testes, com persistência em banco, com uma equipe colaborativa utilizando *frameworks* e metodologias ágeis em seu desenvolvimento, permeando toda a trilha percorrida nesta especialização.

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

A disciplina de Infraestrutura para Desenvolvimento e Implantação de *Software* (*DevOps*) é fundamental na formação dos profissionais de tecnologia na atualidade, especialmente em relação ao desenvolvimento ágil de *software*. Os profissionais de TI precisam se manter atualizados tecnicamente para garantir a qualidade técnica e obter melhores resultados para suas empresas, além de aumentar suas chances de empregabilidade, visto que quanto mais áreas de conhecimento o profissional tiver, mais preparado estará. (Gomes, 2023). Ela oferece conhecimento prático sobre como ferramentas de automação, containerização e integração contínua facilitam os ciclos iterativos e incrementais próprios da abordagem ágil. A experiência prática nos proporciona a chance de configurar e operar *pipelines* automatizados que aceleram os processos de desenvolvimento, teste e implantação, eliminando gargalos tradicionais que prejudicavam a agilidade das equipes.

Essa disciplina se integra de forma interdisciplinar ao currículo, estabelecendo conexões com outras áreas de conhecimento. A sinergia com métodos ágeis, modelagem de *software* e gerenciamento ágil de projetos é refletida na adoção de ferramentas e práticas que promovem entregas frequentes, feedback contínuo e versionamento colaborativo. Além disso, ela sustenta disciplinas focadas em programação, desenvolvimento web e mobile, experiência do usuário e testes automatizados, através da automação de processos, *deploy* eficiente e validação constante, qualidades essenciais para garantir qualidade e agilidade nos projetos.

O impacto formativo é uma melhor preparação para os desafios da indústria de *software* moderna. O domínio das práticas *DevOps* e das ferramentas do mercado potencializa a integração rápida em equipes reais, promovendo cultura colaborativa e orientada por processos. Dessa forma, a disciplina qualifica os futuros profissionais para liderar transformações organizacionais que aumentam a produtividade e a qualidade do desenvolvimento, resultando em projetos mais eficientes e alinhados às demandas do mercado atual (Gomes, 2023).

11.1 ARTEFATOS DO PROJETO

FIGURA 36 - COMANDO DOCKER

```

ricardo@Ricardo:~$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.48/containers/json": dial unix /var/run/docker.sock: connect: per
mission denied
ricardo@Ricardo:~$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
80ccdafdcf8d   dfandarti/gitlab-jenkins:3          "/assets/wrapper"       About a minute ago Up About a minute (health: starting)  0.0.0.0:22->22/tcp, [::]:22->22/tcp, 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:44
3->443/tcp, [::]:443->443/tcp, 0.0.0.0:9091->9091/tcp, [::]:9091->9091/tcp   muraoka
ricardo@Ricardo:~$

```

FONTE: (Autor, 2025)

FIGURA 37 - IMAGEM DO CONSOLE

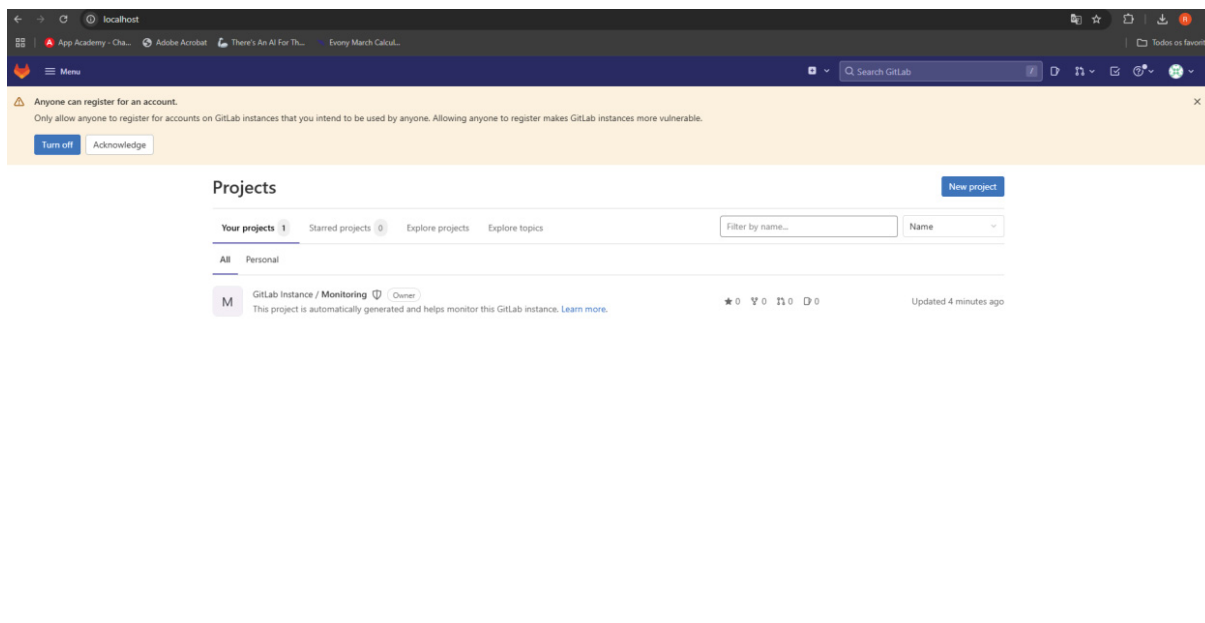
```

ricardo@Ricardo:~$ docker exec -it muraoka bash
root@80ccdafdcf8d:/# cat /etc/gitlab/initial_root_password
# WARNING: This value is valid only in the following conditions
# 1. If provided manually (either via 'GITLAB_ROOT_PASSWORD' environment variable or via 'gitlab_rails['initial
root_password']' setting in 'gitlab.rb', it was provided before database was seeded for the first time (usually, the fi
rst reconfigure run).
# 2. Password hasn't been changed manually, either via UI or via command line.
#
# If the password shown here doesn't work, you must reset the admin password following https://docs.gitlab.com/
ee/security/reset_user_password.html#reset-your-root-password.
Password: kp9ANabx3TnazlkdTV8R9GcVL1H72Gig+Iom4UY2xQY=
# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
root@80ccdafdcf8d:/#

```

FONTE: (Autor, 2025)

FIGURA 38 - IMAGEM DO BROWSER



FONTE: (Autor, 2025)

FIGURA 39 - COMMIT REALIZADO

```
ricardo@Ricardo: ~/Monitoring$ git log
commit ae1638acd7c571c703c4133a20e0f42d4591292c (HEAD -> main, origin/main)
Author: Ricardo Muraoka <muraoka@ufpr.br>
Date:   Sun Mar 9 15:58:29 2025 -0300

    Commit inicial
ricardo@Ricardo:~/Monitoring$ a|
```

FONTE: (Autor, 2025)

FIGURA 40 - IMAGEM DO BROWSER 2

GitLab Instance > Monitoring

M

Monitoring

Star 0 Fork 0

1 Commit 1 Branch 0 Tags 72 KB Project Storage

This project is automatically generated and helps monitor this GitLab instance. [Learn more.](#)

main

Monitoring / +

Find file

Web IDE

Clone

Commit inicial

Ricardo Muraoka authored 6 minutes ago

ae1638ac

Auto DevOps enabled

Add README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Add Kubernetes cluster

Configure Integrations

Name	Last commit	Last update
commit.txt	Commit inicial	6 minutes ago

FONTE: (Autor, 2025)

12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

Os testes automatizados permitem a verificação contínua e sistemática do comportamento do software, garantindo cobertura e qualidade ao longo de todo o ciclo de desenvolvimento (Aniche, 2020, p. 15). O projeto realizado utilizou o *framework* Selenium em um projeto Java utilizando Maven, me levando a codificar cenários de verificação que simulam ações reais de usuários em interfaces web. Essa abordagem permite identificar regressões e falhas precocemente, reduzindo retrabalhos e assegurando entregas mais confiáveis, alinhadas aos princípios de iteração e melhoria constante do desenvolvimento ágil.

O uso de ferramentas de testes, juntamente com práticas *DevOps* promove ciclos de feedback rápidos, nos quais a equipe recebe resultados de testes em tempo real, podendo corrigir defeitos antes que eles se propaguem para produção. Assim, o teste automatizado deixa de ser tarefa isolada e torna-se componente intrínseco ao fluxo de trabalho.

Por fim, a disciplina estimula mentalidade orientada à qualidade e colaborações interdisciplinares. Ao exigir que os testes sejam escritos de forma clara e mantidos atualizados conforme o sistema evolui, também é facilitada a criação da documentação e a comunicação de resultados, preparando futuros desenvolvedores e testadores para atuar em equipes ágeis, onde a responsabilidade compartilhada pela qualidade é um valor central (Aniche, 2020).

12.1 ARTEFATOS DO PROJETO

CÓDIGO-FONTE 2 – TESTE

```
package br.ufpr;

import org.junit.jupiter.api.Test;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.support.ui.ExpectedConditions;
import java.time.Duration;
import static org.junit.jupiter.api.Assertions.assertEquals;

public class EntregaTrabalhoTests {

    @Test
    public void AutomatePad() {
        System.setProperty("webdriver.chrome.driver",
            "src\\drivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        try {
            driver.get("https://pt.anotepad.com");

            String titulo = "Entrega trabalho TEST DAS 2024";
            String corpo = "Ricardo Muraoka";

            WebDriverWait wait = new WebDriverWait(driver,
                Duration.ofSeconds(2));
            WebElement tituloDaNota =
                wait.until(ExpectedConditions.elementToBeClickable(By.id("edit_
                    title"))));
            tituloDaNota.sendKeys(titulo);

            WebElement corpoDaNota =
                wait.until(ExpectedConditions.elementToBeClickable(By.id("edit_
                    textarea"))));
            corpoDaNota.sendKeys(corpo);

            assertEquals(titulo, tituloDaNota.getDomProperty("value"),
                "Título não corresponde ao esperado");
            assertEquals(corpo, corpoDaNota.getDomProperty("value"),
                "Conteúdo não corresponde ao esperado");

        } finally {
            driver.quit();
        }
    }
}
```

FONTE: (Autor, 2025)

13 CONCLUSÃO

O projeto memorial apresentado aqui evidencia o caminho trilhado na formação da especialização que soma a técnica à conceitos ágeis em contextos realistas de desenvolvimento de *software*. Os vários aprendizados adquiridos nas disciplinas se somam de modo a formar um fluxo de trabalho integrado, onde planejamento, ambientes, código e validações coexistem de forma coesa. Esse alinhamento demonstra ganhos de confiabilidade, repetibilidade e velocidade de entrega, pilares fundamentais para equipes que buscam responder rapidamente às demandas dos usuários e do mercado.

Entretanto, desafios ainda existem para se consolidar plenamente as práticas ágeis em ambientes corporativos. A gestão de dependências entre múltiplos containers, orquestração de pipelines complexos, codificação de grandes projetos, gerenciamento de grandes equipes multidisciplinares exigem disciplina e alto grau de conhecimento técnico, sob pena de comprometer a estabilidade e a qualidade dos processos e do resultado final. Todavia, a especialização nos trouxe uma base de conhecimentos teóricos, técnicos e a possibilidade da fixação destes conhecimentos obtidos através da experiência prática ao produzir os artefatos deste memorial.

Em suma, foi possível verificar que o processo de desenvolvimento ágil se torna plenamente viável quando disciplinas que, à primeira vista, podem parecer independentes convergem em um fluxo único de trabalho.

REFERÊNCIAS

AMBLER, S. W. **Agile Modeling: effective practices for eXtreme Programming and the Unified Process**. Upper Saddle River: Addison-Wesley, 2002.

ANICHE, M. **Testes automatizados de software: um guia prático**. São Paulo: Casa do Código, 2020.

ANDERSON, D. J. **Kanban: successful evolutionary change for your technology business**. Sequim: Blue Hole Press, 2010.

APACHE MAVEN PROJECT. **Maven – Welcome to Apache Maven**. Disponível em: <https://maven.apache.org/>. Acesso em: 12 out. 2025.

BECK, K. **TDD – Test Driven Development**. Tradução de Jean Felipe Patikowski Cheiran. Porto Alegre: Bookman, 2010.

CAROLLI, P. **Lean Inception: como alinhar pessoas e construir o produto certo**. 1. ed. São Paulo: Editora XYZ, 2017. Disponível em: <https://caroli.org/lean-inception-3/>. Acesso em: 12 out. 2025.

COSTA, M. S. **Sistemas web e mobile: uma visão geral para negócios empresariais**. Revista Científica Multidisciplinar Núcleo do Conhecimento, v. 9, n. 8, p. 82-99, 2018. Disponível em: <https://www.nucleodoconhecimento.com.br/ciencia-da-computacao/sistemas-web-e-mobile>. Acesso em: 27 out. 2025.

DOCKER. **Documentação oficial do Docker**. Disponível em: <https://docs.docker.com/>. Acesso em: 12 out. 2025.

DUVALL, P. M.; MATYAS, S.; GLOVER, A. **Continuous Integration: Improving Software Quality and Reducing Risk**. Boston: Addison-Wesley, 2007.

FOWLER, M. **Continuous Integration: Improving Software Quality and Reducing Risk**. Boston: Addison-Wesley, 2006.

GALVÃO, N. M. S.; SCHNEIDER, Henrique N. **Design Thinking na educação: um estudo bibliométrico em pesquisas internacionais**. Revista Ibero-Americana de Estudos em Educação, Araraquara, v. 18, n. 00, p. e023107, 2023. DOI: 10.21723/riaee.v18i00.17667.

GITLAB. **Documentação oficial do GitLab**. Disponível em: <https://docs.gitlab.com/>. Acesso em: 12 out. 2025.

GOMES, T. G. **Análise do processo de implantação da cultura DevOps em uma empresa brasileira**. 2023. Trabalho de Conclusão de Curso (Engenharia de Computação) – Centro de Informática, Universidade Federal da Paraíba, João Pessoa, 2023.

JENKINS. **Documentação oficial do Jenkins**. Disponível em: <https://www.jenkins.io/doc/>. Acesso em: 12 out. 2025.

JUNIT TEAM. **JUnit 5 User Guide**. Disponível em: <https://junit.org/junit5/docs/current/user-guide/>. Acesso em: 18 out. 2025.

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

MARTIN, R. C. **Arquitetura limpa: o guia do artesão para estrutura e design de software**. Rio de Janeiro: Alta Books, 2019.

MARTINS, A. R. et al. **Aplicativos híbridos: desenvolvimento de aplicativos utilizando tecnologias web**. Revista Ambiente Acadêmico, v. 6, n. 1, p. 48-59, 2020.

MOROZOV, I. **Design Sprint: passo a passo, exemplos e dicas**. Post no blog, 2021. Disponível em: <https://posdigital.pucpr.br/blog/design-sprint>. Acesso em: 12 out. 2025.

NEWMAN, S. **Building Microservices: Designing Fine-Grained Systems**. O'Reilly Media, 2015.

NORTH, D. **Introducing BDD**. *Better Software Magazine*, mar. 2006. Disponível em: <https://dannorth.net/introducing-bdd/>. Acesso em: 19 out. 2025.

ORACLE Corporation **Java Documentation**. Disponível em: <https://docs.oracle.com/en/java/>. Acesso em: 19 out. 2025.

ORACLE Corporation. **Manual de referência do MySQL 4.1**. Disponível em: <https://downloads.mysql.com/docs/refman-4.1-pt.a4.pdf>. Acesso em: 18 out. 2025.

REIS, G. **Fundamentos de UX: ensaio sobre a experiência do usuário**. São Paulo: Casa do Código, 2022.

SCHWABER, K.; SUTHERLAND, Jeff. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game**. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf>. Acesso em: 12 out. 2025.

SELENIUM. **Documentação oficial do Selenium**. Disponível em: <https://www.selenium.dev/documentation/>. Acesso em: 12 out. 2025.

SILVA, A. I.; NASCIMENTO, Emerson Adriano; SANTOS, Thiago Adriano Coleti dos. **A importância do banco de dados para o controle em uma indústria**. *Tekhne e Logos*, Botucatu, v. 10, n. 2, p. 94-106, 2019. Disponível em: <http://revista.fatecbt.edu.br/index.php/tl/article/view/590>. Acesso em: 27 out. 2025.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2018.

VOCKE, Ham. **The Practical Test Pyramid**. Disponível em: <https://martinfowler.com/articles/practical-test-pyramid.html>. Acesso em: 07 set. 2025.