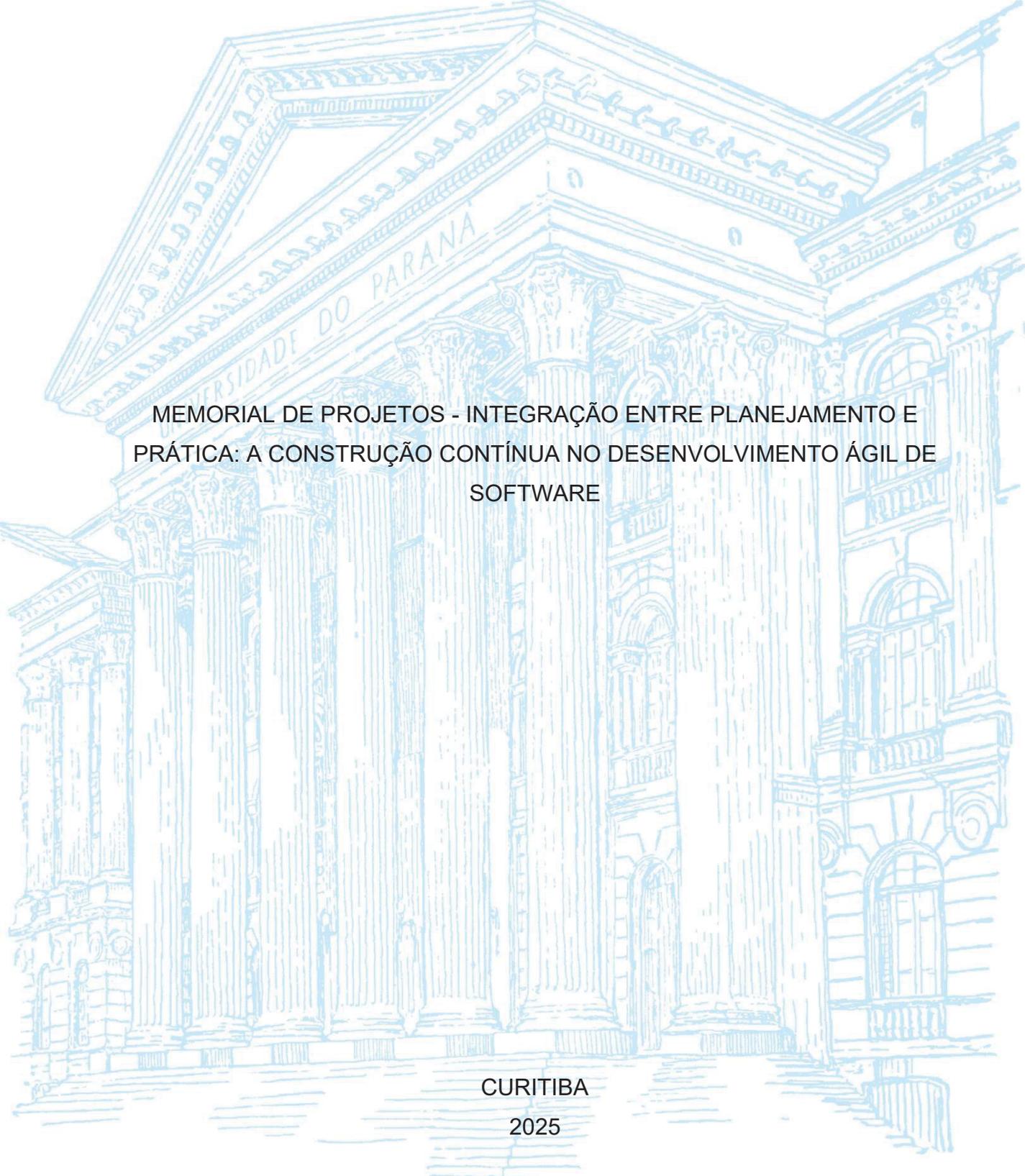


UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME EDUARDO RODRIGUES



MEMORIAL DE PROJETOS - INTEGRAÇÃO ENTRE PLANEJAMENTO E  
PRÁTICA: A CONSTRUÇÃO CONTÍNUA NO DESENVOLVIMENTO ÁGIL DE  
SOFTWARE

CURITIBA

2025

GUILHERME EDUARDO RODRIGUES

MEMORIAL DE PROJETOS - INTEGRAÇÃO ENTRE PLANEJAMENTO E  
PRÁTICA: A CONSTRUÇÃO CONTÍNUA NO DESENVOLVIMENTO ÁGIL DE  
SOFTWARE

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. Jaime Wojciechowski

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 GUILHERME EDUARDO RODRIGUES, intitulada: **MEMORIAL DE PROJETOS - INTEGRAÇÃO ENTRE PLANEJAMENTO E PRÁTICA: A CONSTRUÇÃO CONTÍNUA NO DESENVOLVIMENTO ÁGIL DE SOFTWARE**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 01 de Agosto de 2025.

JAIME WOJCIECHOWSKI  
Presidente da Banca Examinadora

RAFAELA MANTOVANI FONTANA  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

## RESUMO

O Memorial de Projetos – Integração entre Planejamento e Prática: A Construção Contínua no Desenvolvimento Ágil de Software tem como objetivo documentar e refletir sobre a trajetória percorrida ao longo do curso de Especialização em Desenvolvimento Ágil de Software da UFPR, destacando a integração entre teoria e prática sob a ótica do desenvolvimento ágil. Apresentado em ordem cronológica, o memorial apresenta como o planejamento, a análise de requisitos, a modelagem de processos e o desenvolvimento técnico se integram de forma incremental para atender às demandas de um projeto real. Os artefatos reunidos demonstram as entregas realizadas em cada fase. Cada disciplina é apresentada com seus objetivos, destacando sua relevância para o processo como um todo, e sua integração com as demais áreas do conhecimento. A abordagem ágil é refletida nas práticas de iteração contínua, adaptação ao feedback e colaboração entre os envolvidos. O memorial se encerra com a conclusão do parecer técnico, acompanhado de reflexões gerais sobre os aprendizados obtidos, dificuldades enfrentadas e o desenvolvimento das competências profissionais e técnicas adquiridas ao longo do curso.

Palavras-chave: Desenvolvimento de Software, Planejamento, Ágil, Parecer Técnico.

## **ABSTRACT**

The Project Report – Integrating Planning and Practice: Continuous Construction in Agile Software Development aims to document and reflect on the path taken throughout the Specialization Course in Agile Software Development at UFPR, highlighting the integration between theory and practice from the perspective of agile development. Presented in chronological order, the report shows how planning, requirements analysis, process modeling, and technical development are integrated in an incremental manner to meet the demands of a real project. The collected artifacts demonstrate the deliverables made in each phase. Each discipline is presented with its objectives, highlighting its relevance to the process as a whole, and its integration with other areas of knowledge. The agile approach is reflected in the practices of continuous iteration, adaptation to feedback, and collaboration among those involved. The report ends with the conclusion of the technical report, accompanied by general reflections on the lessons learned, difficulties faced, and the development of professional and technical skills acquired throughout the course.

Keywords: Software Development, Planning, Agile, Technical Report

## SUMÁRIO

<b>1 PARECER TÉCNICO.....</b>	<b>7</b>
<b>2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....</b>	<b>9</b>
2.1 ARTEFATOS DO PROJETO.....	10
<b>3 DISCIPLINA: MAG1 E MAG2 – MODELAGEM ÁGIL DE SOFTWARE 1 E 2.....</b>	<b>13</b>
3.1 ARTEFATOS DO PROJETO.....	14
<b>4 DISCIPLINA: GAP1 E GAP2 – GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2 .....</b>	<b>19</b>
4.1 ARTEFATOS DO PROJETO.....	20
<b>5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO .....</b>	<b>23</b>
5.1 ARTEFATOS DO PROJETO.....	24
<b>6 DISCIPLINA: BD – BANCO DE DADOS.....</b>	<b>25</b>
6.1 ARTEFATOS DO PROJETO.....	26
<b>7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO .....</b>	<b>30</b>
7.1 ARTEFATOS DO PROJETO.....	31
<b>8 DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2 .....</b>	<b>33</b>
<b>9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE.....</b>	<b>34</b>
9.1 ARTEFATOS DO PROJETO.....	35
<b>10 DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....</b>	<b>39</b>
<b>11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS) .....</b>	<b>40</b>
11.1 ARTEFATOS DO PROJETO.....	41
<b>12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS .....</b>	<b>43</b>
12.1 ARTEFATOS DO PROJETO.....	44
<b>13 CONCLUSÃO .....</b>	<b>46</b>
<b>14 REFERÊNCIAS.....</b>	<b>48</b>

## 1 PARECER TÉCNICO

Este parecer técnico apresenta uma análise integrada dos projetos desenvolvidos durante o curso de Especialização em Desenvolvimento Ágil de Software da UFPR, evidenciando a relação entre os conteúdos das disciplinas e a aplicação prática dos princípios ágeis. Ao longo da trajetória acadêmica, os projetos evoluíram da concepção inicial de ideias até a entrega de soluções completas, sempre orientados pela filosofia de colaboração, adaptação e entrega contínua de valor.

As primeiras disciplinas abordaram os fundamentos do planejamento e da análise de requisitos, estabelecendo a base para o entendimento do ciclo de vida de software ágil. O estudo do Manifesto Ágil (HIGHSMITH, 2002) e seus princípios possibilitou compreender que mais do que seguir um processo rígido, o foco está na entrega contínua e na capacidade de responder a mudanças (BECK et al., 2001). Os artefatos gerados nesta fase, como mapas mentais, épicos e histórias de usuário, podem ser encontrados nas seções iniciais deste memorial.

Seguindo, disciplinas como MAG1 e MAG2 (Modelagem Ágil de Software) possibilitaram a representação dos requisitos levantados por meio de diagramas de caso de uso, sequência e classes, reforçando a comunicação entre as equipes e facilitando o refinamento das funcionalidades. Como destaca Larman (2003), a modelagem ágil não busca a perfeição inicial, mas sim o suporte à tomada de decisões técnicas durante o desenvolvimento.

As disciplinas voltadas à gestão (GAP1 e GAP2) agregaram técnicas de planejamento incremental, como *release planning* e o uso de quadros Kanban, alinhadas aos valores da transparência e priorização contínua de tarefas com base no valor entregue ao cliente (POPPENDIECK; POPPENDIECK, 2003). Esses planos estão registrados nas seções correspondentes ao planejamento do projeto.

Com a base conceitual estabelecida, as disciplinas voltadas às práticas, como WEB1, WEB2 e MOB1 permitiram a construção de sistemas utilizando frameworks modernos, versionamento com Git, integração contínua e testes automatizados. Essa abordagem evidenciou como o conhecimento inicial em modelagem e planejamento se conecta diretamente ao desenvolvimento iterativo e incremental. A prática de TDD (*Test-Driven Development*), abordada em disciplinas como TEST, mostrou-se essencial para a manutenção da qualidade e da cobertura de testes ao longo do projeto, conforme recomendado por Martin (2008).

O curso também destacou a importância da experiência do usuário (UX) na entrega de valor. A criação de protótipos navegáveis no Figma, com base no feedback obtido em testes de usabilidade, reafirma o papel central do usuário no processo ágil. Este ciclo contínuo de prototipação, teste e melhoria encontra respaldo em autores como Norman (2013), que defende o design centrado no usuário como elemento-chave para a aceitação de sistemas.

Por fim, as disciplinas de infraestrutura e DevOps permitiram integrar pipelines de CI/CD, ferramentas de automação de testes e ambientes de homologação, reforçando a visão de que o desenvolvimento ágil também depende de uma cultura de entrega contínua e colaboração entre áreas técnicas e operacionais (HUMBLE; FARLEY, 2010).

Dessa forma, o memorial evidencia que os projetos não são etapas isoladas, mas partes de um processo fluido, iterativo e incremental, onde cada disciplina contribui com fundamentos essenciais para o desenvolvimento ágil de software. A coesão entre análise, modelagem, planejamento, codificação, testes e *deploy* contínuo demonstra uma aplicação concreta e integrada dos princípios ágeis no ambiente acadêmico.

O título “Integração entre Planejamento e Prática: A Construção Contínua no Desenvolvimento Ágil de Software” reflete a essência deste parecer, que valoriza a interdependência entre teoria e prática, reforçando a importância de um fluxo contínuo e adaptativo para o sucesso no desenvolvimento de sistemas.

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

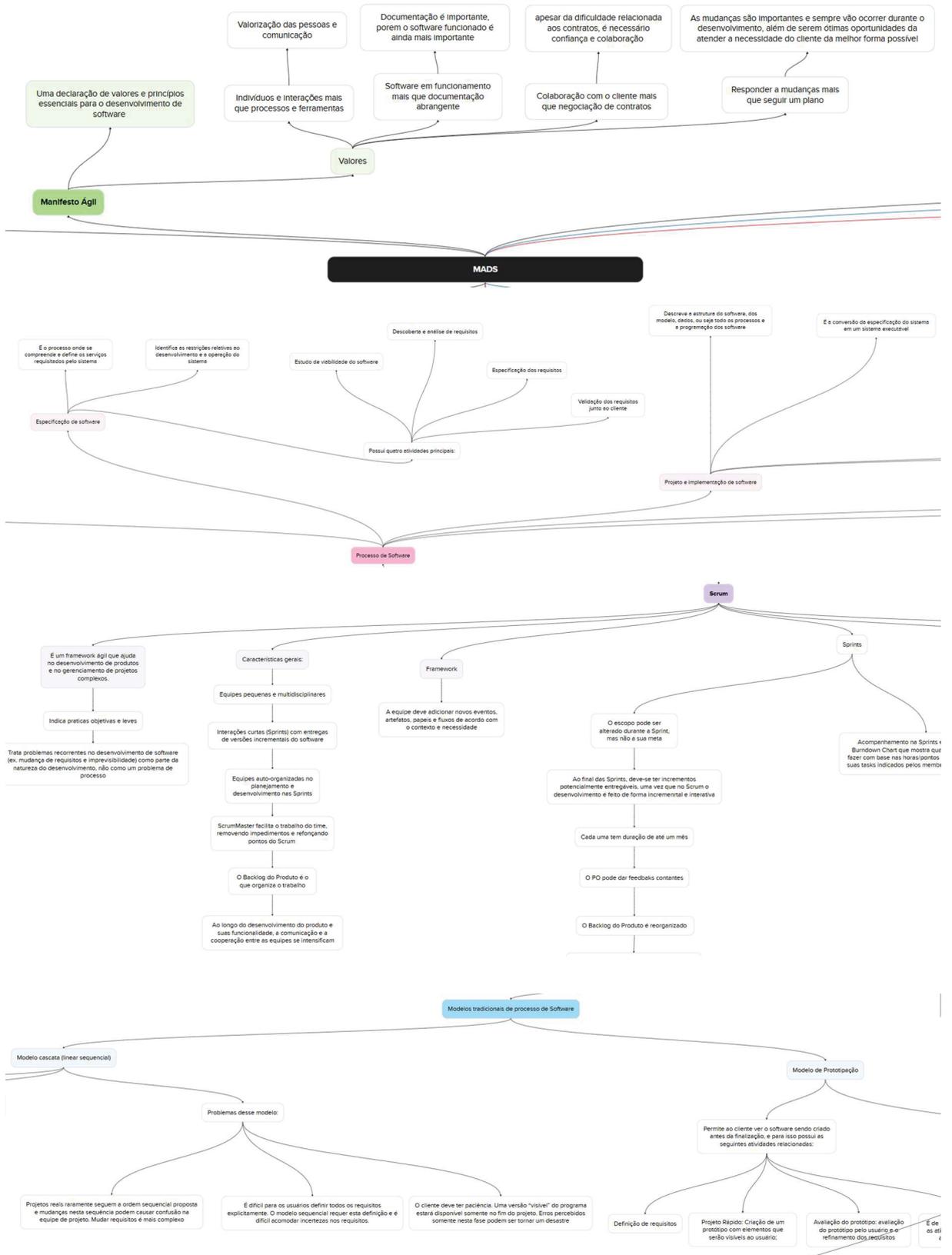
Para a disciplina de MADS (Métodos Ágeis para Desenvolvimento de Software), o projeto final teve como objetivo a construção de um mapa mental detalhado com os principais conceitos e práticas que sustentam o desenvolvimento ágil. O material foi organizado de forma hierárquica, com definição e aprofundamento de tópicos como: processo de software, modelos tradicionais de desenvolvimento, manifesto ágil, princípios ágeis, Scrum, Kanban, Extreme Programming, Lean Software Development e entrega contínua. Cada ramo do mapa foi elaborado com base no conteúdo apresentado na disciplina, permitindo uma visão abrangente e estruturada da metodologia ágil.

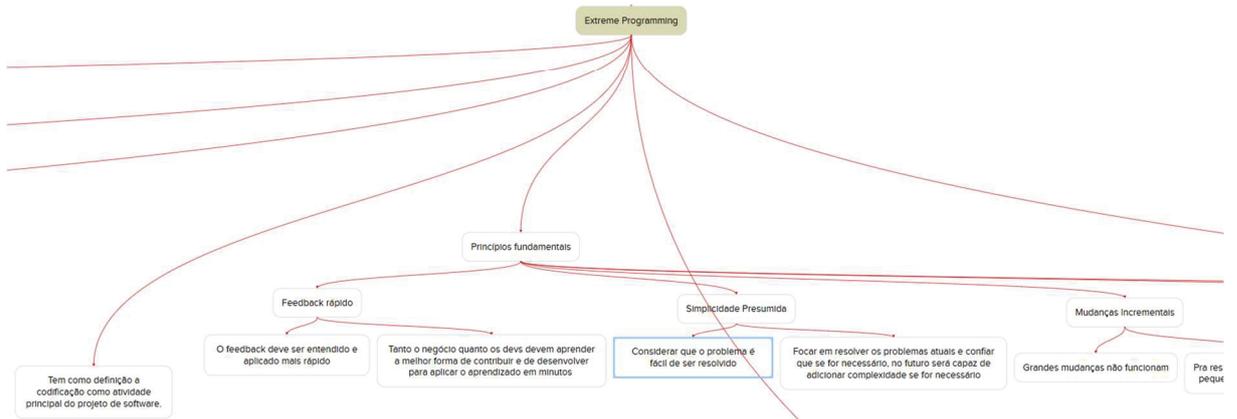
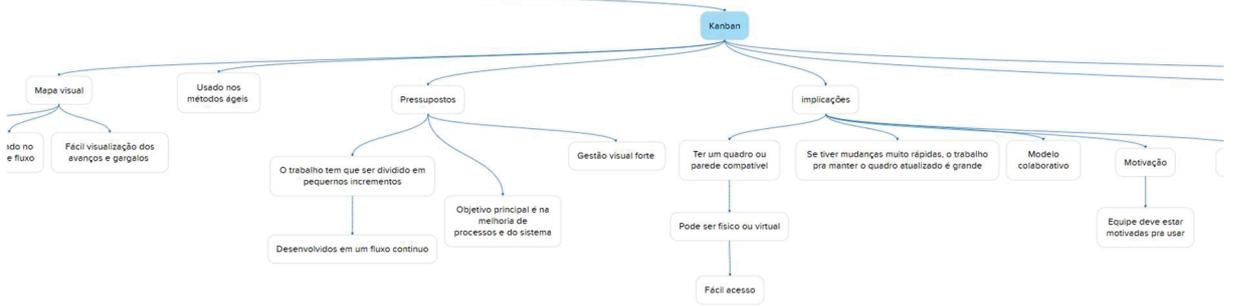
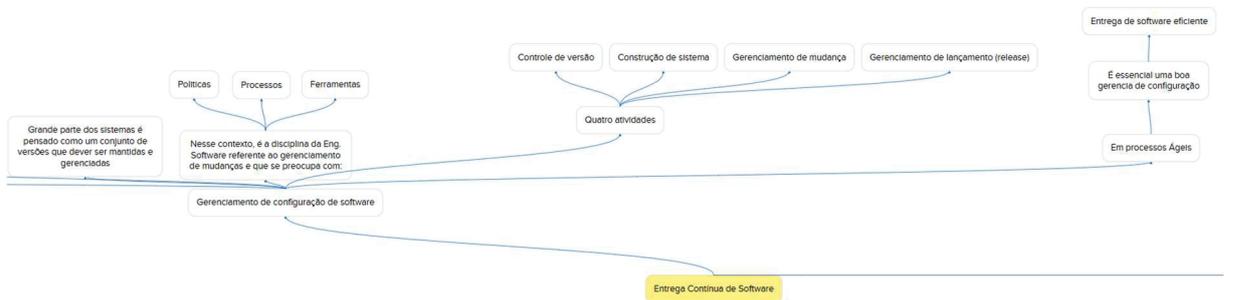
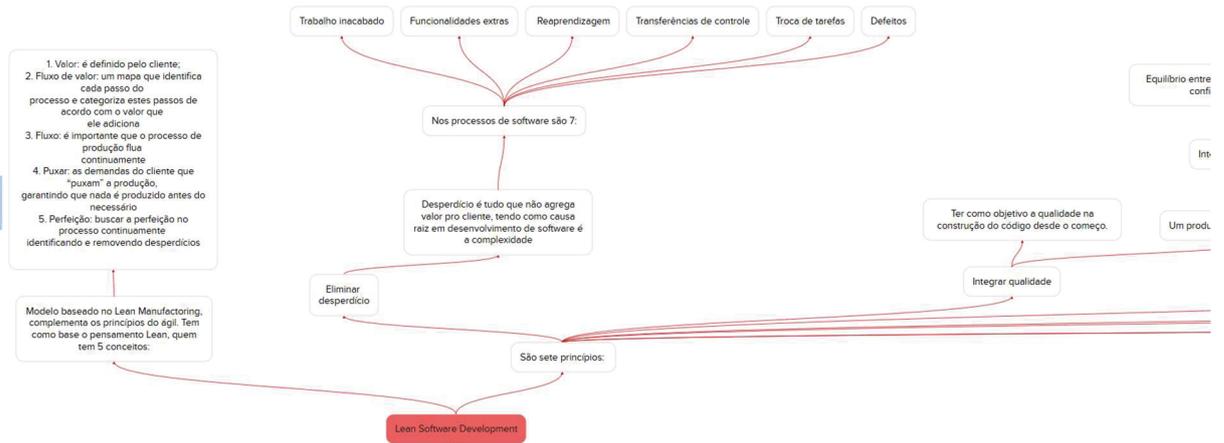
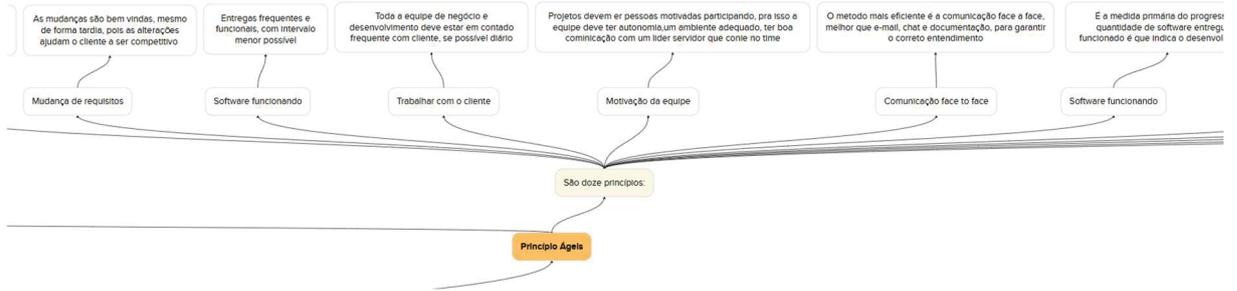
Esse projeto teve papel fundamental na formação da base conceitual do curso, proporcionando aos alunos uma compreensão sólida dos princípios que norteiam o desenvolvimento ágil de software, tendo a importância de fundamentar o conteúdo das demais disciplinas do curso.

Conceitos como iteração contínua, priorização de valor, feedback rápido e equipes auto-organizadas são elementos recorrentes nas disciplinas de gestão de projetos, modelagem ágil, testes e desenvolvimento de software. Assim, o conhecimento adquirido em MADS serviu de base para a aplicação prática das metodologias ágeis em projetos subsequentes, fortalecendo a articulação entre teoria e prática no decorrer da especialização.

Dessa forma, a disciplina de MADS foi essencial não apenas para a assimilação dos métodos ágeis em si, mas também para a construção de uma mentalidade voltada à flexibilidade, colaboração e entrega incremental, pilares que permeiam todas as etapas do desenvolvimento ágil de software.

## 2.1 ARTEFATOS DO PROJETO





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

Para as disciplinas de MAG1 e MAG2 (Modelagem Ágil de Software 1 e 2), os projetos finais tiveram como objetivo a modelagem de um sistema, a partir da aplicação prática dos princípios da modelagem ágil. A temática do projeto foi: sistema de gestão de condomínio, e sua modelagem envolveu a produção de diversos artefatos fundamentais, como diagramas de caso de uso (níveis 1 e 2), histórias de usuários, regras de negócio, diagramas de classes, e diagramas de sequência correspondentes a todas as funcionalidades mapeadas.

Os diagramas de classes, sequência e casos de uso ajudaram a esclarecer fluxos de trabalho, responsabilidades dos componentes e interações entre usuários e o sistema. Enquanto os diagramas UML desempenharam um papel importante na visualização e validação da arquitetura do sistema, atuando como ponto de apoio técnico para a equipe de desenvolvimento.

As histórias de usuário foram utilizadas como ferramenta central para a especificação dos requisitos, em conformidade com a abordagem ágil de documentação enxuta e orientada ao valor de negócio. Essas histórias incluíam critérios de aceitação, regras de negócio, observações técnicas e vínculos com outros artefatos, auxiliando durante o desenvolvimento. Essa prática não apenas facilitou o alinhamento de expectativas, mas também auxiliou na entrega incremental de valor ao cliente, conforme defendido por Beck et al. (2001) no Manifesto Ágil.

A importância dessas disciplinas no contexto do desenvolvimento ágil reside na sua capacidade de equilibrar planejamento e flexibilidade. A modelagem orientada por histórias de usuário e diagramas funcionais permite adaptar rapidamente o projeto a novas demandas, sem comprometer a estrutura e a integridade da solução. Além disso, a integração com disciplinas como MADS, GAP1/GAP2, TEST e desenvolvimento web/mobile é natural e necessária: enquanto MADS fornece os princípios, MAG1 e MAG2 materializam esses conceitos em artefatos que guiam o desenvolvimento técnico.

Assim, as disciplinas de Modelagem Ágil de Software 1 e 2 exercem um papel estratégico na especialização, pois conectam teoria e prática, garantindo que o projeto avance de forma coesa e alinhada com os pilares ágeis: colaboração, iteração, simplicidade e resposta rápida a mudanças.

3.1 ARTEFATOS DO PROJETO – MAG 1

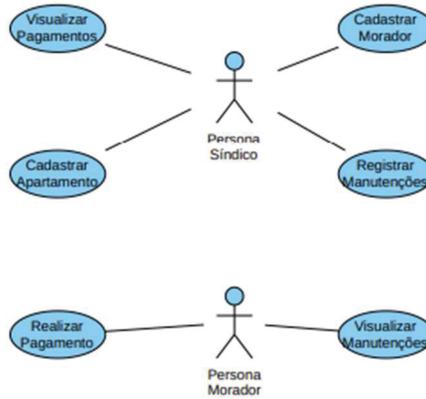


Diagrama de Caso de Uso Nível 1

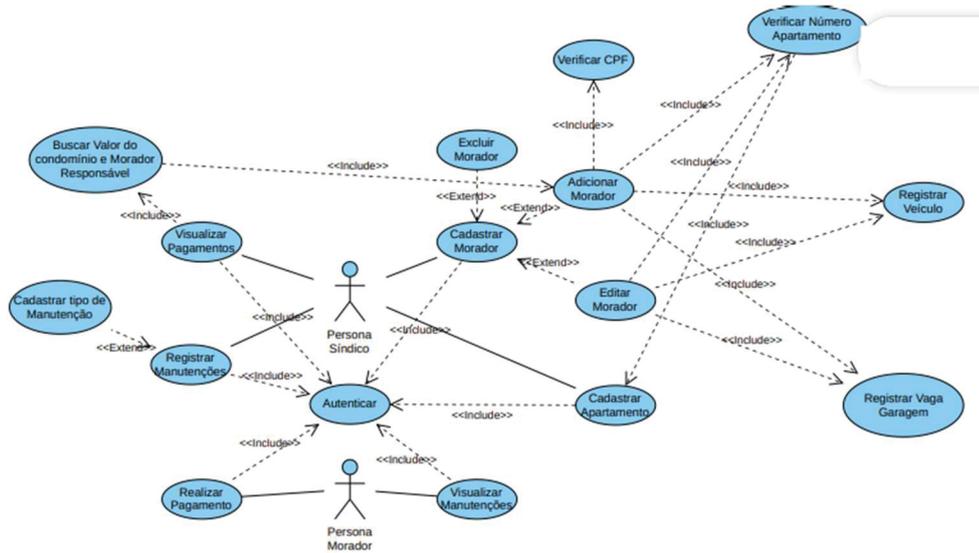


Diagrama de Caso de Uso Nível 2

## Histórias de Usuários

### Tela Pesquisar Apartamento

-Como um usuário do sistema, quero ser capaz de visualizar uma lista de todos os apartamentos cadastrados no condomínio para poder identificar rapidamente informações específicas sobre eles.

-Como um usuário do sistema, desejo ter a capacidade de pesquisar apartamentos por número ou bloco, a fim de encontrar facilmente o apartamento desejado na lista.

-Como um usuário do sistema, quero poder editar as informações de um apartamento existente para atualizar detalhes como número do apartamento ou bloco, caso haja alguma alteração.

-Como um usuário do sistema, desejo ter a opção de adicionar um novo apartamento ao sistema, inserindo informações como número do apartamento e bloco, garantindo que o cadastro esteja sempre atualizado com novas unidades.

-Como um usuário do sistema, quero poder acessar rapidamente as informações detalhadas de um apartamento clicando no botão de edição correspondente, facilitando a visualização e a modificação de seus detalhes.

Sistema gestão de condominio

Tela inicial

Login

#### Pesquisar Apartamentos

<b>Novo</b>	<b>Voltar</b>	<b>Pesquisar</b>	<input type="text"/>
-------------	---------------	------------------	----------------------

Numero	Bloco	
152	A	<b>Editar</b>
215	A	<b>Editar</b>

#### Apartamento

Numero

Bloco

**Salvar** **Voltar**

## Regras de Negócio

R1- O número do apartamento deve ser único.

R2- O número do apartamento e o bloco devem ser obrigatórios para o cadastro.

R3- O CPF de um morador deve ser único.

R4- O CPF, nome, telefone, número do apartamento e a informação se é responsável ou proprietário devem ser obrigatórios para o cadastro.

R5- Ao incluir um novo morador, o sistema deve verificar se o CPF já existe no cadastro.

R6- O sistema deve consistir se o número do apartamento informado já existe no cadastro.

R7- O sistema deve permitir o armazenamento dos dados dos veículos dos moradores, incluindo informações como modelo, placa e vaga de garagem.

R8- O número do apartamento, mês/ano de referência, data de vencimento, data de pagamento e valor do condomínio devem ser obrigatórios para o registro do pagamento.

R9- Ao digitar o número do apartamento, o sistema deve apresentar o nome do morador responsável e o valor do condomínio correspondente.

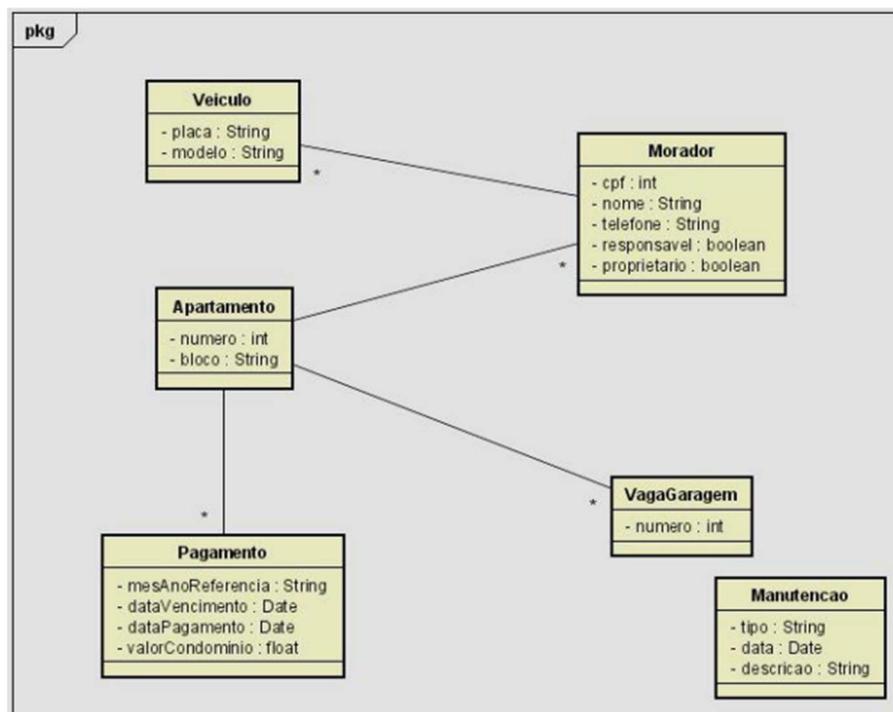
R10- Um pagamento só pode ser registrado se não houver valores anteriores vencidos para o mesmo apartamento.

R11- O sistema deve permitir o registro de diferentes tipos de manutenções, como pintura, limpeza de caixa d'água, jardinagem, etc.

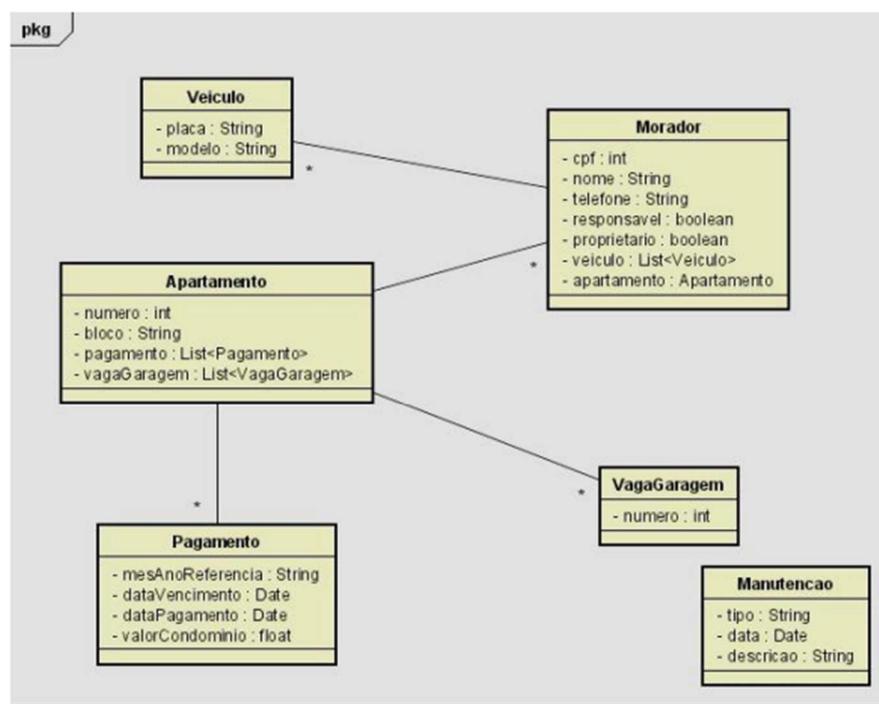
R12- Os detalhes da manutenção, como tipo, data de realização e descrição dos serviços realizados, devem ser registrados.

## 3.2 ARTEFATOS DO PROJETO – MAG 2

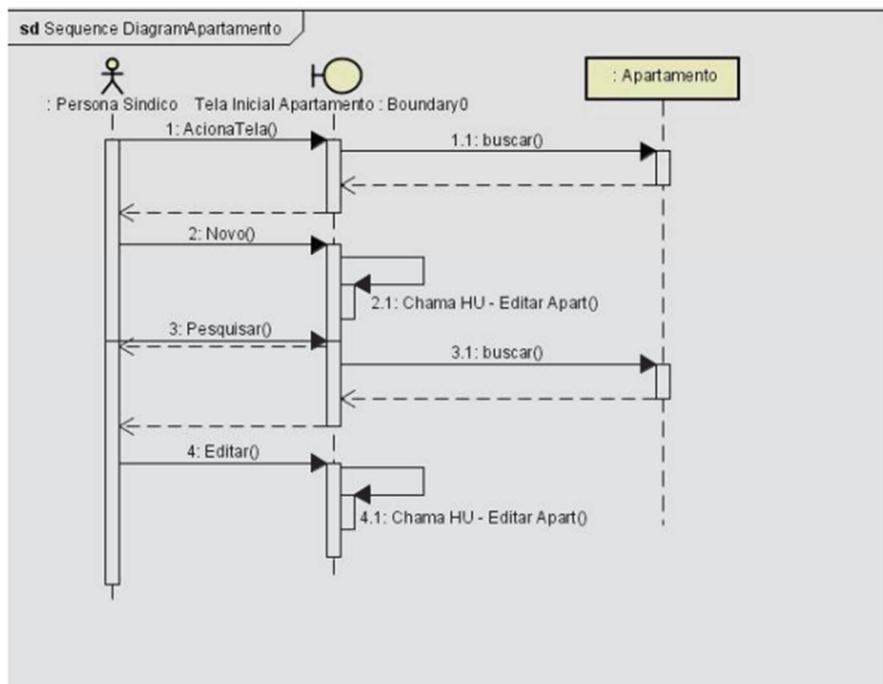
## Diagrama de Classes



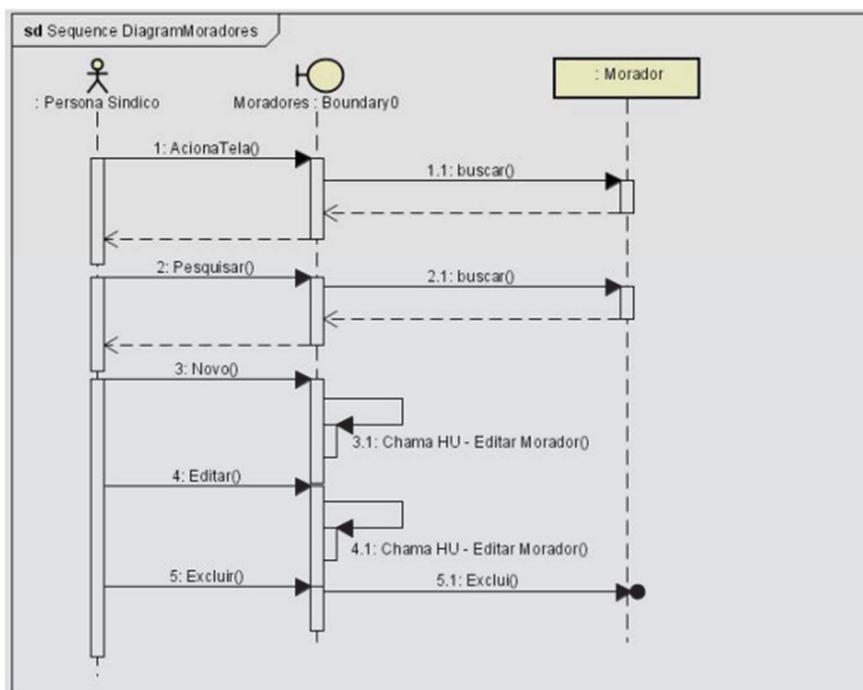
## Diagrama de Classes com atributos associados



## Diagrama de Sequência Tela Apartamento



## Diagrama de Sequência Tela Morador



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

Para as disciplinas de GAP1 e GAP2 (Gerenciamento Ágil de Projetos de Software 1 e 2), os projetos desenvolvidos tiveram como foco central a aplicação prática de técnicas de planejamento, acompanhamento e otimização de projetos ágeis, baseadas em frameworks como Scrum e Kanban. Em GAP1, o projeto final consistiu na elaboração de um plano de release para um sistema escolhido pelo aluno, contemplando o cálculo da velocidade da equipe, a estimativa de histórias de usuário, a divisão em sprints e a definição de datas previstas para cada entrega. Esse exercício permitiu estruturar as fases do projeto de forma realista e adaptável, seguindo os princípios ágeis de planejamento contínuo e entrega incremental.

Já em GAP2, a atividade prática envolveu a simulação de um ambiente de trabalho ágil por meio de um jogo online de Kanban, onde os alunos gerenciaram o fluxo de trabalho em um quadro visual, buscando otimizar o tempo de ciclo e reduzir gargalos. Ao final da simulação, foi realizada uma análise do Diagrama de Fluxo Cumulativo (CFD), permitindo avaliar o desempenho do time. Essa atividade reforçou a importância da visibilidade e da gestão do fluxo de valor, pilares fundamentais do método Kanban.

A relevância desses projetos no contexto do desenvolvimento ágil de software está na ênfase que ambos dão à organização, monitoramento e adaptação do projeto ao longo do tempo. Planejar de forma iterativa e monitorar o progresso por meio de métricas objetivas garante transparência e previsibilidade, essenciais para tomadas de decisão eficazes (POPPENDIECK; POPPENDIECK, 2003).

Além disso, as disciplinas de GAP1 e GAP2 se integram diretamente com outras disciplinas do curso, como MAG1, MAG2, WEB, MOB e TEST. O plano de release gerado em GAP1, por exemplo, fornece uma estrutura de entregas que orienta as atividades das demais disciplinas práticas, enquanto os conceitos de fluxo contínuo e gestão visual de GAP2 complementam o trabalho técnico com uma perspectiva de eficiência operacional.

Dessa forma, os projetos realizados nessas disciplinas foram fundamentais para consolidar uma visão estratégica e organizacional do desenvolvimento ágil, preparando os alunos para liderar projetos com foco em entregas frequentes, adaptabilidade e alto valor ao cliente.

## 4.1 ARTEFATOS DO PROJETO – GAP1

Gerenciamento Ágil de Projetos I  
 Prof.ª. Dra. Raíaela Mantovani Fontana  
 Plano de Release Software de Registro de Caminhões Produzidos – Aluno Guilherme Eduardo Rodrigues

**Cálculo da Velocidade:**

Horas disponíveis por dia:	6	Tamanho da Sprint:	2 Semanas
Horas disponíveis por Sprint:	6 horas dia * 5 dias por semana * 2 semanas = 60 horas por sprint	Velocidade:	60 horas por sprint/ 8 horas (1 ponto) = 7 pontos (arredondamento de 7,5)

**Plano de Release:**

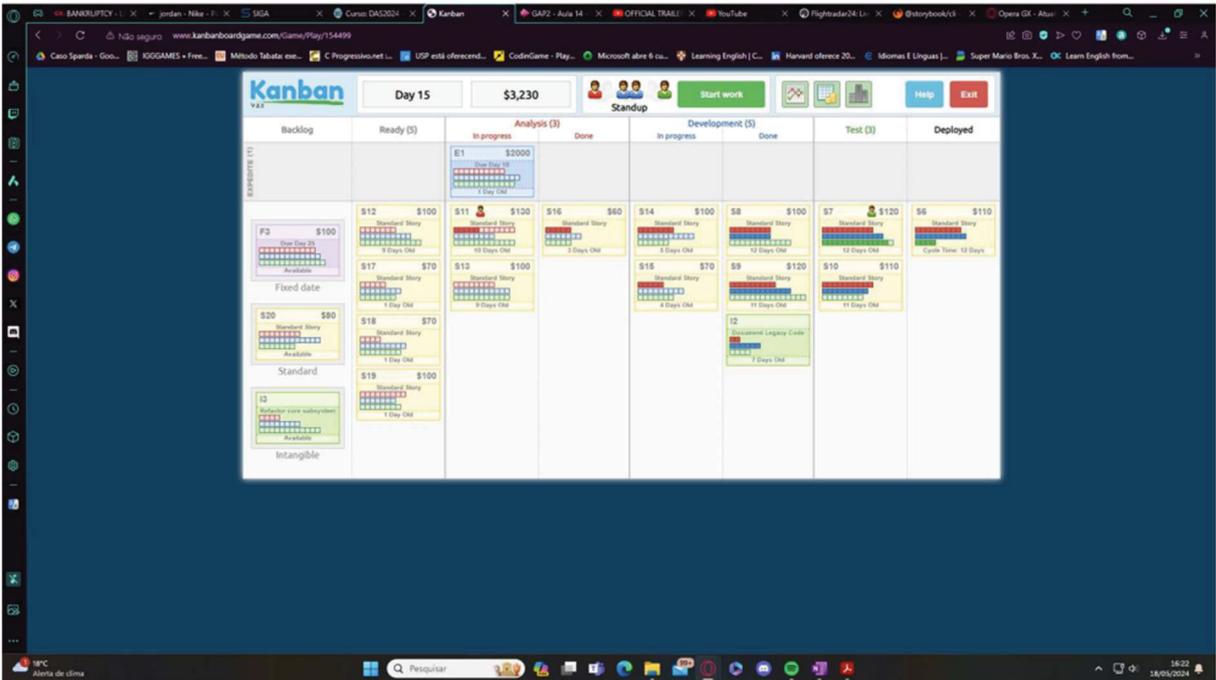
Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint 4
Data Início: 06/05/2024 Data Fim: 17/05/2024	Data Início: 20/05/2024 Data Fim: 31/05/2024	Data Início: 03/06/2024 Data Fim: 14/06/2024	Data Início: 17/06/2024 Data Fim: 28/06/2024
<HU001 – Fazer Login> SENDO um usuário cadastrado QUERO fazer login no sistema PARA acessar minhas informações ESTIMATIVA (2)	<HU005 – Visualizar Lista de Caminhões > SENDO um usuário cadastrado QUERO visualizar a lista de caminhões cadastrados PARA ter uma visão geral ESTIMATIVA (1)	<HU009 – Visualizar Histórico de vendas de Caminhões > SENDO um usuário cadastrado QUERO visualizar o histórico de vendas de caminhões PARA ter uma visão geral ESTIMATIVA (2)	<HU013 – Visualizar Gráfico de Caminhões Produzidos > SENDO um usuário não cadastrado QUERO poder ver o gráfico de produção diária de caminhões PARA identificar tendências e variações na produção ESTIMATIVA (3)
<HU002 – Cadastrar Usuário> SENDO um usuário cadastrado. QUERO cadastrar um novo cliente PARA manter seus dados registrados no sistema ESTIMATIVA (2)	<HU006 – Buscar Clientes> SENDO um usuário cadastrado QUERO buscar clientes pelo nome PARA encontrar informações específicas ESTIMATIVA (1)	<HU010 – Gerenciar permissões de acesso > SENDO um administrador do sistema QUERO gerenciar permissões de acesso dos usuários PARA controlar o nível de acesso às funcionalidades do sistema ESTIMATIVA (3)	<HU014 – Redefinir Senha > SENDO um usuário administrador QUERO poder redefinir senhas de usuário PARA ajudar os usuários a recuperar o acesso às suas contas ESTIMATIVA (1)
<HU003 – Cadastrar Caminhão> SENDO um usuário cadastrado QUERO cadastrar um novo caminhão PARA manter suas informações registradas no sistema ESTIMATIVA (2)	<HU007 – Editar Clientes e Caminhões > SENDO um usuário cadastrado QUERO editar as informações de um cliente e de um caminhão PARA mantê-las atualizadas ESTIMATIVA (4)	<HU011 – Visualizar Responsáveis pelos Cadastros dos Caminhões > SENDO um administrador do sistema QUERO visualizar a lista com os responsáveis pelos cadastros dos caminhões PARA ver o histórico dos cadastros realizados ESTIMATIVA (1)	<HU015 – Exportar Dados > SENDO um usuário administrador QUERO exportar dados gerais em PDF PARA utilizá-los em reuniões gerenciais ESTIMATIVA (1)

Gerenciamento Ágil de Projetos I  
 Prof.ª. Dra. Raíaela Mantovani Fontana  
 Plano de Release Software de Registro de Caminhões Produzidos – Aluno Guilherme Eduardo Rodrigues

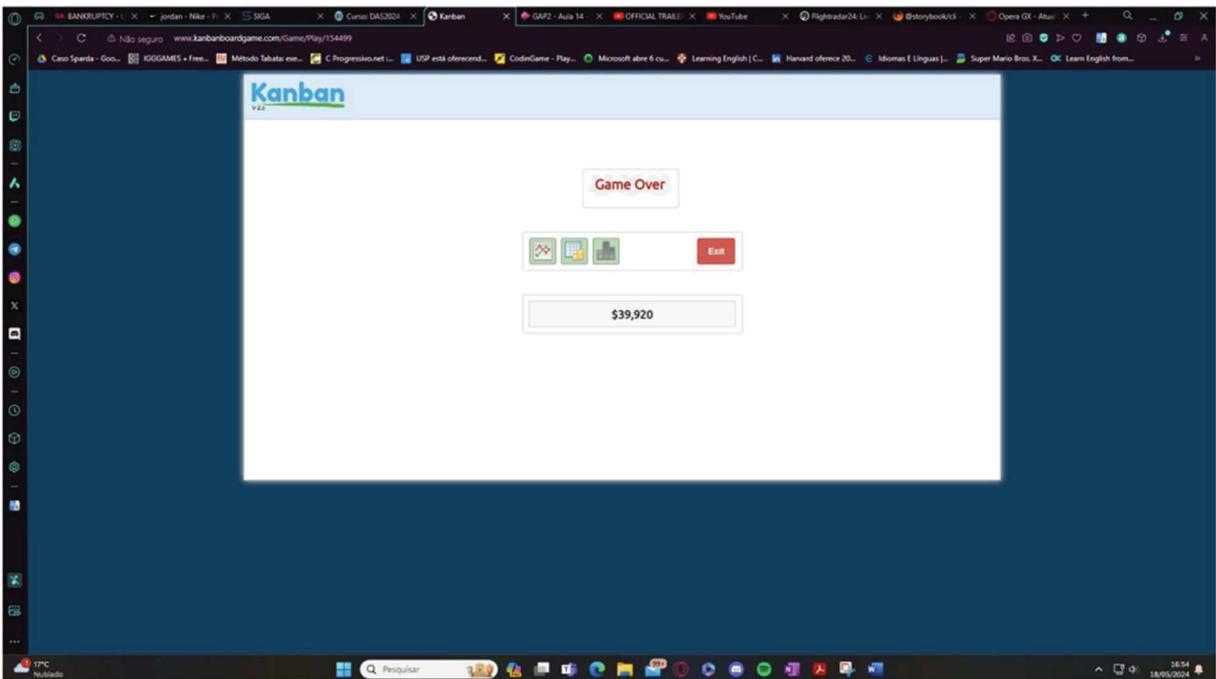
<HU004 – Visualizar Lista de Clientes> SENDO um usuário cadastrado QUERO visualizar a lista de clientes cadastrados PARA ter uma visão geral ESTIMATIVA (1)	<HU008 – Excluir Clientes e Caminhões> SENDO um usuário cadastrado QUERO excluir um cliente e um caminhão do sistema PARA remover registros desnecessários. ESTIMATIVA (1)	<HU012 – Visualizar Número de Caminhões Produzidos > SENDO um usuário não cadastrado QUERO poder visualizar o número total de caminhões produzidos PARA ter uma visão geral da produção ESTIMATIVA (1)	<HU016 – Selecionar Caminhão para Teste > SENDO um usuário administrador QUERO poder selecionar um caminhão para testes PARA garantir a qualidade ESTIMATIVA (2)
---	--	--	--

## 4.2 ARTEFATOS DO PROJETO – GAP2

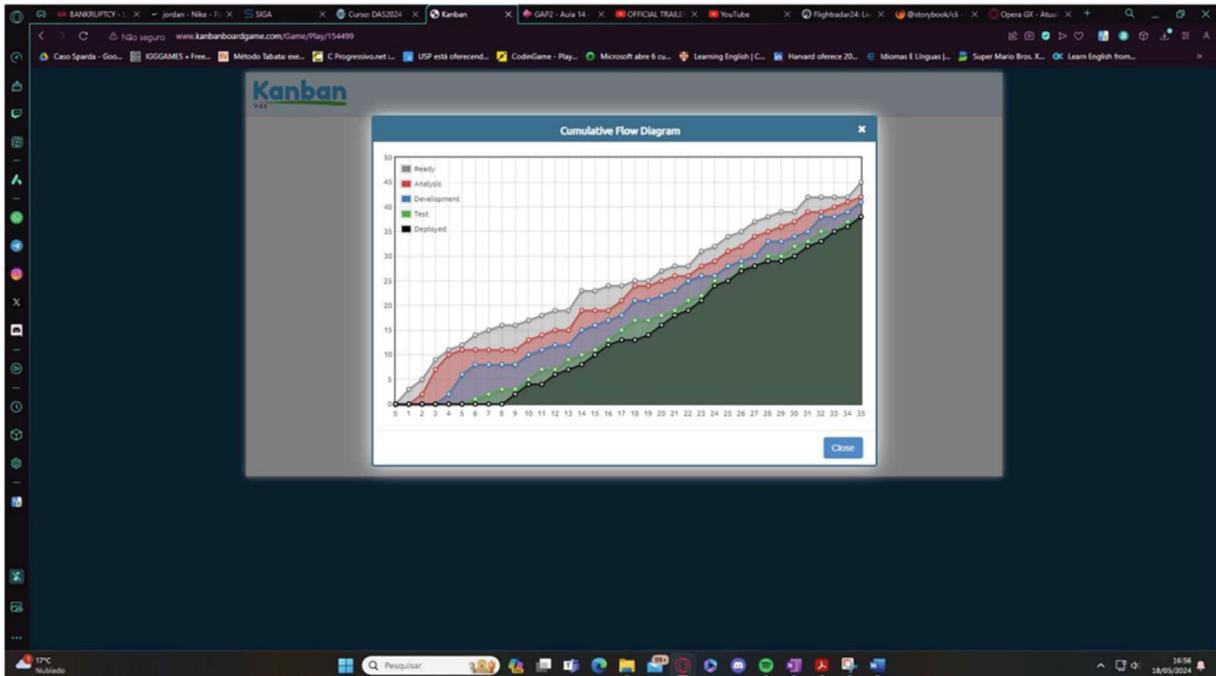
Print dia 15:



Print valor final:



Print CFD:



## 5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

Em INTRO (Introdução à Programação), o projeto final teve como objetivo a implementação do backend de um sistema bancário simplificado, utilizando a linguagem Java. O sistema tinha algumas funcionalidades como o cadastro de clientes, controle de contas-correntes e contas de investimento, permitindo ao aluno aplicar conceitos fundamentais de orientação a objetos. Foram disponibilizados um Diagrama de Classes, um projeto base no ambiente NetBeans com testes automatizados escritos em JUnit, e um script DDL para criação do banco de dados no MySQL para apoiar e direcionar no desenvolvimento.

A principal exigência do projeto era garantir que, ao final da implementação, pelo menos 40 dos 42 testes unitários fossem bem-sucedidos. Essa abordagem adotou os princípios do TDD (Test-Driven Development), técnica amplamente utilizada no desenvolvimento ágil de software. O TDD estimula a criação de código modular, coeso e fácil de manter, pois os testes são escritos antes da implementação, guiando o design do sistema.

A importância da disciplina no contexto do desenvolvimento ágil está justamente na introdução prática de conceitos como automação de testes, design orientado a testes e controle de versões, que são essenciais em projetos modernos com entregas incrementais e ciclos curtos de desenvolvimento.

Dessa forma, INTRO não apenas consolidou conhecimentos essenciais de programação, mas também preparou os alunos para aplicar boas práticas de engenharia de software em um contexto ágil, promovendo um desenvolvimento mais seguro, organizado e alinhado às necessidades do cliente.

## 5.1 ARTEFATOS DO PROJETO

The screenshot displays the Test Results window for a project named 'INTRO-TrabalhoFinal-BancoRRW\_'. The window shows a summary of test results and a detailed list of individual test cases.

**Summary:** Tests passed: 100,00 %  
All 42 tests passed. (9,725 s)

**Test Results:**

- TesteBancoRrw passed running...
- t01verificaEstruturaClassePessoa passed (0,004 s)
- t02verificaEstruturaClasseContaCorrente passed (0,001 s)
- t03verificaEstruturaClasseContaInvestimento passed (0,0 s)
- t04criarContaCorrenteSaldoZero passed (0,001 s)
- t05criaContaCorrenteComSaldo2000 passed (0,0 s)
- t06manipulaContaCorrenteDepositar50 passed (0,0 s)
- t07manipulaContaCorrenteDepositar100Deposita20Saca60 passed (0,0 s)
- t08manipulaContaCorrenteDepositar100Deposita20Saca1000 passed (0,001 s)
- t09manipulaContaCorrenteLimiteDepositar100Deposita20Saca1300 passed (0,009 s)
- t10manipulaContaCorrenteDepositarNegativo50 passed (0,002 s)
- t11manipulaContaCorrenteSaqueNegativo100 passed (0,0 s)
- t12manipulaContaCorrenteDeposita100AplicaJuros passed (0,0 s)
- t13manipulaContaCorrenteSaca100AplicaJuros passed (0,0 s)
- t14trocaContaCorrenteDeCliente passed (0,0 s)
- t15verificaSaldoZeroParaTrocarContaCorrente passed (0,0 s)
- t16criarContaInvestimento passed (0,0 s)
- t17manipularContaInvestimentoDepositoinicialMenorQueMontanteMinimo passed (0,0 s)
- t18manipularContaInvestimentoDepositarMinimo passed (0,0 s)
- t19manipularContaInvestimentoDepositar1000Sacar500 passed (0,0 s)
- t20manipularContaInvestimentoDepositar1000Sacar1100 passed (0,0 s)
- t21manipularContaInvestimentoLimiteDepositarMenosQueMinimo passed (0,001 s)
- t22manipularContaInvestimentoAplica1000AplicaJuros passed (0,0 s)
- t23crudClienteAdd passed (0,67 s)
- t24crudClienteGetById passed (0,4 s)
- t25crudClienteUpdate passed (0,691 s)
- t26crudClienteDelete passed (0,28 s)
- t27crudContaCorrenteAdd passed (0,271 s)
- t28crudContaCorrenteGetById passed (0,31 s)
- t29crudContaCorrenteUpdate passed (0,391 s)
- t30crudContaCorrenteDelete passed (0,314 s)
- t31crudContaInvestimentoAdd passed (0,293 s)
- t32crudContaInvestimentoGetById passed (0,328 s)
- t33crudContaInvestimentoUpdate passed (0,338 s)
- t34crudContaInvestimentoDelete passed (0,335 s)
- t35verificaSeAContaCorrenteFoiSetadaNoCliente passed (0,314 s)
- t36manipulaSaldoDaContaCorrenteEGravaBdERecuperaSaldo passed (0,3 s)
- t37cria4ContasSalvaNoBdERecuperaTodas passed (0,848 s)
- t38testaContaCorrenteDeleteAll passed (0,728 s)
- t39verificaSeAContaInvestimentoFoiSetadaNoCliente passed (0,318 s)
- t40testaContaInvestimentoDeleteAll passed (0,51 s)
- t41testaSeODEleteAllDaContaCorrenteNaoEliminaTodasAsContasInvestimentosDoCliente passed (1,023 s)
- t42testaSeODEleteAllDaContaInvestimentoNaoEliminaTodasAsContasCorrentesDoCliente passed (0,959 s)

## 6 DISCIPLINA: BD – BANCO DE DADOS

Para a disciplina de BD (Banco de Dados), o projeto final foi estruturado em duas etapas, com o objetivo de consolidar os conhecimentos teóricos e práticos sobre modelagem, criação e manipulação de bancos de dados relacionais. Na primeira etapa, foi proposto o desenvolvimento de um sistema de gerenciamento de biblioteca, exigindo a construção de um Modelo Entidade-Relacionamento (MER) conceitual, seguido por um Diagrama Entidade-Relacionamento (DER) lógico. Esse modelo detalhou entidades, atributos, chaves primárias e estrangeiras, bem como os relacionamentos entre tabelas, permitindo o mapeamento eficiente dos requisitos do sistema para uma estrutura de dados coerente.

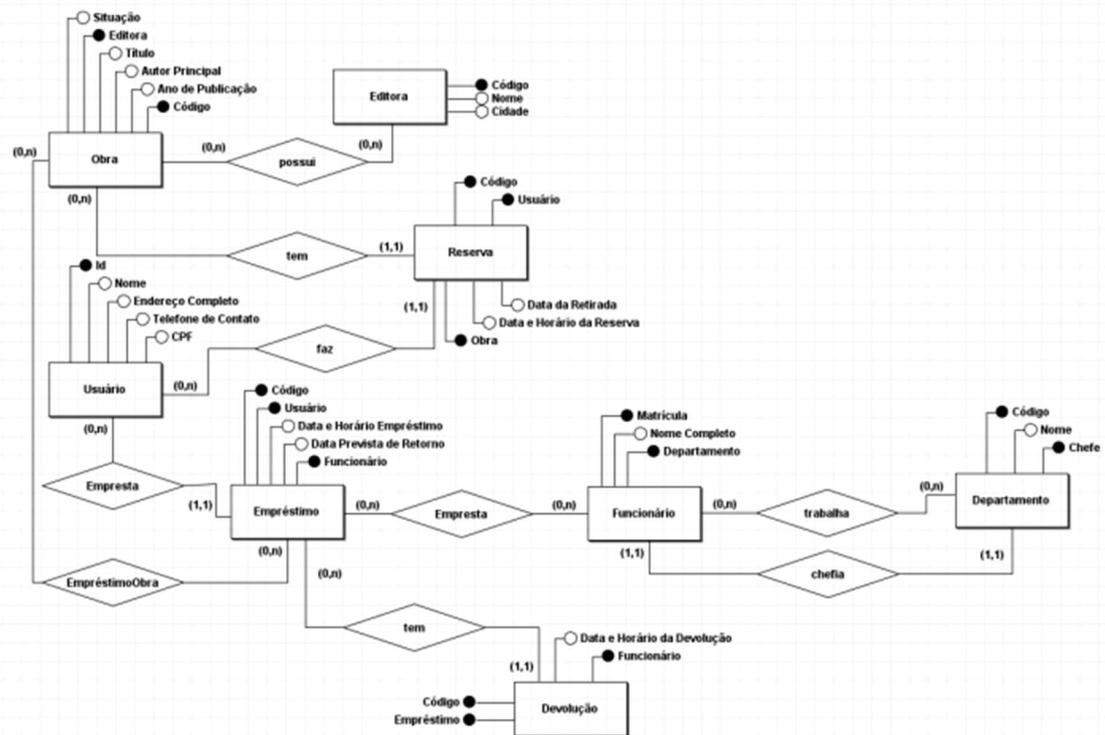
Na segunda etapa, o tema abordado foi Controle de veículos de uma concessionária, com foco na elaboração do modelo lógico, desenvolvimento de scripts SQL para criação do esquema no banco de dados, inserção de dados exemplares e formulação de consultas SQL que explorassem diferentes tipos de relacionamento (1:1, 1:N, N:N). Essa atividade reforçou a importância da normalização e da integridade referencial, princípios fundamentais para a consistência e escalabilidade dos dados.

A disciplina se mostrou essencial no contexto do desenvolvimento ágil de software, pois um banco de dados bem modelado e estruturado evita retrabalho e facilita a manutenção evolutiva do sistema. Em métodos ágeis, onde mudanças frequentes são comuns, uma base de dados resiliente e adaptável é vital para sustentar entregas contínuas com qualidade.

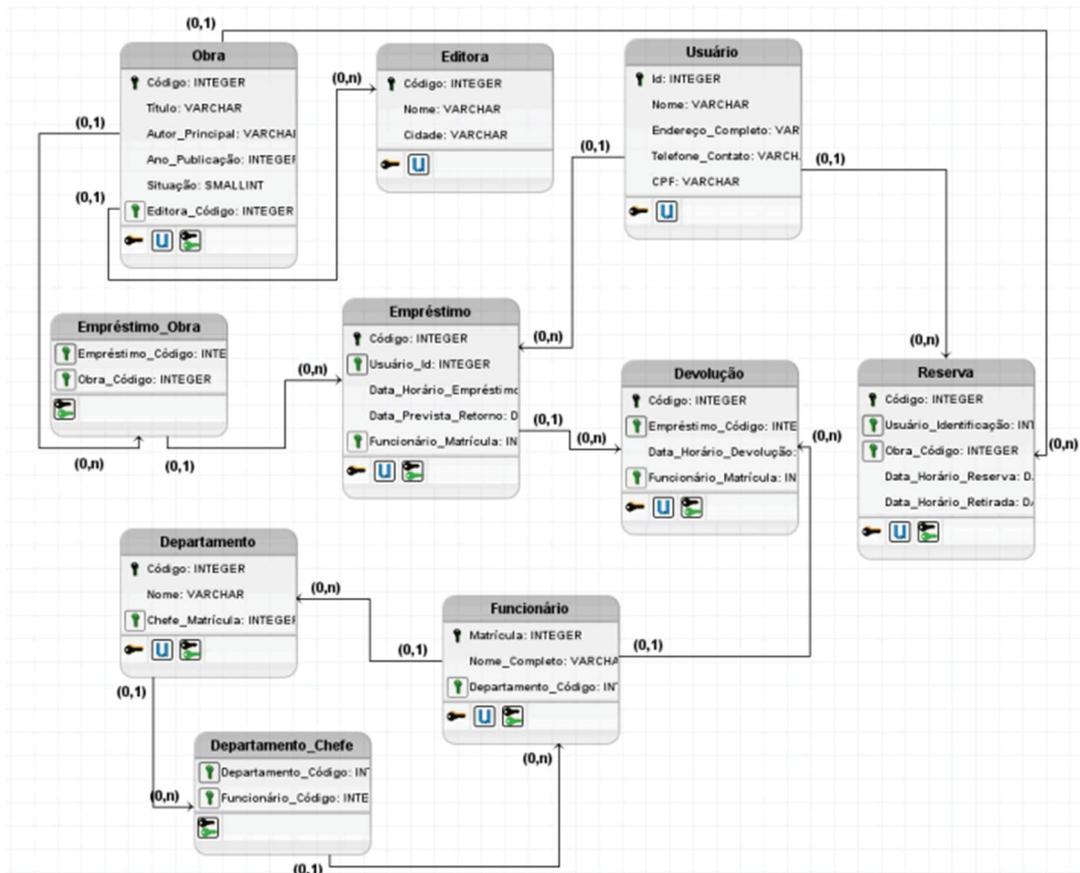
O conhecimento adquirido em Banco de Dados integra-se diretamente com as disciplinas de modelagem (MAG1/MAG2), programação (INTRO, WEB, MOB) e testes, permitindo uma abordagem mais robusta e orientada a dados durante a construção de sistemas. Dessa forma, o projeto desenvolvido em BD contribuiu significativamente para o domínio de boas práticas de persistência e consulta de dados, fortalecendo a base técnica necessária para a entrega de soluções ágeis, escaláveis e sustentáveis.

## 6.1 ARTEFATOS DO PROJETO

Modelo Entidade-Relacionamento Conceitual - Figura 1



Modelo Lógico - Diagrama de Entidade Relacionamento (tabelas) - Figura 2



O tema escolhido foi o controle de veículos de uma concessionária.

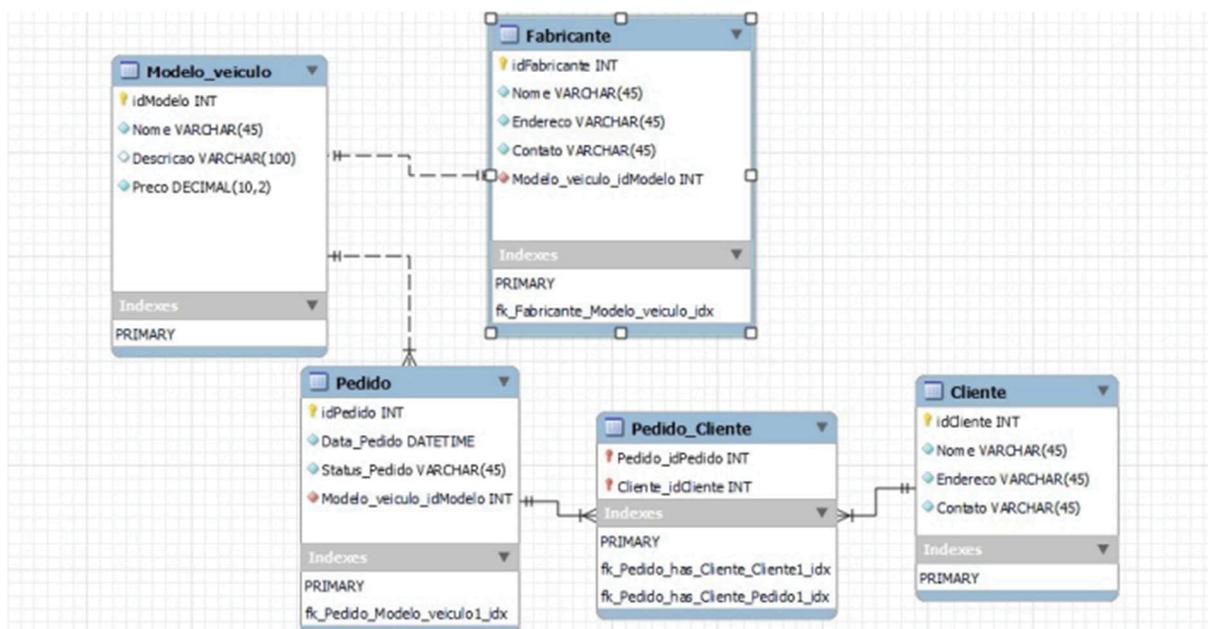
### Relacionamentos:

Relacionamento 1x1 (Modelo\_Veiculo para Fabricante): Um modelo de veículo pode ter apenas um fabricante, e um fabricante está associado a um único modelo. Portanto, é um relacionamento 1x1 entre Modelo\_Veiculo e Fabricante. O Modelo\_Veiculo seria referenciado pelo ID do Modelo na tabela de Fabricante.

Relacionamento 1xN (Pedido para Modelo\_Veiculo): Um pedido pode conter vários Modelos, mas um modelo está associado a apenas um pedido. Portanto, é um relacionamento 1xN entre Pedido e Modelo\_Veiculo. O Modelo\_Veiculo seria referenciado pelo ID do Modelo\_Veiculo na tabela de Pedido.

Relacionamento NxN (Pedido para Cliente): Um pedido pode pertencer a vários clientes, e um cliente pode fazer vários pedidos. Portanto, é um relacionamento NxN entre Pedido e Cliente. Seria necessário uma tabela adicional, por exemplo, "Pedido\_Cliente", que conteria os IDs de Pedido e Cliente para representar essa relação muitos-para-muitos.

### Modelo Lógico:



**Script de criação:**

```
-- MySQL Script generated by MySQL Workbench
```

```
-- Sun Jun 30 13:31:52 2024
```

```
-- Model: New Model  Version: 1.0
```

```
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
```

```
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
```

```
SET @OLD_SQL_MODE=@@SQL_MODE,
```

```
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,  
ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

**Script de inserção de registros:**

```
-- Inserção de registros na tabela Modelo_veiculo
```

```
INSERT INTO mydb.Modelo_veiculo (idModelo, Nome, Descricao, Preco) VALUES
```

```
(1, 'Civic', 'Sedan médio da Honda', 105000.00),
```

```
(2, 'Corolla', 'Sedan médio da Toyota', 110000.00),
```

```
(3, 'Mustang', 'Esportivo da Ford', 300000.00),
```

```
(4, 'Model S', 'Sedan elétrico da Tesla', 450000.00),
```

```
(5, 'A3', 'Sedan compacto da Audi', 120000.00);
```

```
-- Inserção de registros na tabela Fabricante
```

```
INSERT INTO mydb.Fabricante (idFabricante, Nome, Endereco, Contato,  
Modelo_veiculo_idModelo) VALUES
```

```
(1, 'Honda', 'Avenida Honda, 1000, São Paulo', '1111-1111', 1),
```

```
(2, 'Toyota', 'Avenida Toyota, 2000, São Bernardo do Campo', '2222-2222', 2),
```

```
(3, 'Ford', 'Avenida Ford, 3000, Camaçari', '3333-3333', 3),
```

```
(4, 'Tesla', '3500 Deer Creek Road, Palo Alto, CA', '4444-4444', 4),
```

```
(5, 'Audi', 'Avenida Audi, 4000, São José dos Pinhais', '5555-5555', 5);
```

```
1 • SELECT * FROM mydb.fabricante;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

	idFabricante	Nome	Endereco	Contato	Modelo_veiculo_idModelo
▶	1	Honda	Avenida Honda, 1000, São Paulo	1111-1111	1
	2	Toyota	Avenida Toyota, 2000, São Bernardo do Campo	2222-2222	2
	3	Ford	Avenida Ford, 3000, Camaçari	3333-3333	3
	4	Tesla	3500 Deer Creek Road, Palo Alto, CA	4444-4444	4
	5	Audi	Avenida Audi, 4000, São José dos Pinhais	5555-5555	5
•	NULL	NULL	NULL	NULL	NULL

```
1 • SELECT * FROM mydb.modelo_veiculo;
```

Result Grid | Filter Rows: | Edit:

	idModelo	Nome	Descricao	Preco
▶	1	Civic	Sedan médio da Honda	105000.00
	2	Corolla	Sedan médio da Toyota	110000.00
	3	Mustang	Esportivo da Ford	300000.00
	4	Model S	Sedan elétrico da Tesla	450000.00
	5	A3	Sedan compacto da Audi	120000.00
•	NULL	NULL	NULL	NULL

## 7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

Para a disciplina de AAP (Aspectos Ágeis de Programação), o projeto final teve como objetivo a refatoração do algoritmo de ordenação Bubble Sort com base nos princípios do Clean Code, promovendo maior legibilidade, simplicidade e facilidade de manutenção do código. A proposta exigia a realização de pelo menos três melhorias significativas, entre elas: a utilização de nomes claros e descritivos para variáveis e métodos, a remoção de comentários redundantes e trechos de código desnecessários, e a adoção do princípio da responsabilidade única, que define que cada método deve realizar uma tarefa específica de maneira clara.

Os princípios de Clean Code, são fundamentais no contexto do desenvolvimento ágil de software, pois favorecem um ambiente de trabalho mais colaborativo e produtivo, além de melhorar a manutenibilidade e compreensão. Em equipes ágeis, onde o código precisa ser constantemente revisto, aprimorado e expandido, a clareza e organização do código são essenciais para garantir entregas frequentes, revisão eficiente e integração contínua sem comprometer a qualidade do sistema.

A disciplina destacou que, ao adotar boas práticas de codificação, os desenvolvedores reduzem o tempo necessário para entendimento e modificação de código, mitigam riscos de introdução de erros e facilitam a evolução incremental do software. Essa abordagem encontra aplicação direta nas disciplinas práticas do curso, como INTRO, WEB, MOB e TEST, onde a qualidade do código influencia diretamente a eficiência da implementação e dos testes.

Além disso, os conceitos trabalhados em AAP se alinham aos pilares do Manifesto Ágil, como simplicidade e excelência técnica contínua, fortalecendo o ciclo de desenvolvimento iterativo e incremental. A compreensão e aplicação de Clean Code não apenas elevam o padrão de qualidade das soluções desenvolvidas, mas também promovem uma base sustentável para manutenção e escalabilidade dos sistemas.

Dessa forma, a disciplina de AAP contribuiu significativamente para o amadurecimento técnico dos alunos, reforçando práticas que, quando aplicadas de forma consistente, tornam o processo ágil mais fluido, confiável e eficiente.

## 7.1 ARTEFATOS DO PROJETO

```
J BubbleSort-refatorado-2-versao.java X
C: > Users > Guilherme > Downloads > J BubbleSort-refatorado-2-versao.java
1 // Implementação otimizada em Java do Bubble sort
2 // Código extraído de https://www.geeksforgeeks.org/bubble-sort/
3
4 import java.io.*;
5
6 class BubbleSort {
7
8     public static void main(String args[]) {
9         int arrayElementsToOrder[] = {64, 34, 25, 12, 22, 11, 90};
10        bubbleSort(arrayElementsToOrder);
11        System.out.println("Sorted array: ");
12        printArray(arrayElementsToOrder);
13    }
14
15    static void bubbleSort(int arrayElementsToOrder[]) {
16        for (int i = 0; i < arrayElementsToOrder.length - 1; i++) {
17            if (isSorted(arrayElementsToOrder, i)) break;
18        }
19    }
20
21    static boolean isSorted(int arrayElementsToOrder[], int index) {
22        boolean sorted = true;
23        for (int j = 0; j < arrayElementsToOrder.length - 1 - index; j++) {
24            if (arrayElementsToOrder[j] > arrayElementsToOrder[j + 1]) {
25                swap(arrayElementsToOrder, j, j + 1);
26                sorted = false;
27            }
28        }
29        return sorted;
30    }
31
32    static void swap(int arrayElementsToOrder[], int i, int j) {
33        int temp = arrayElementsToOrder[i];
34        arrayElementsToOrder[i] = arrayElementsToOrder[j];
35        arrayElementsToOrder[j] = temp;
36    }
37
38    static void printArray(int arrayElementsToOrder[]) {
39        System.out.println(Arrays.toString(arrayElementsToOrder));
40    }
41 }
42
43 // This code is contributed
44 // by Nikita Tiwari.
45
```

## Relatório de alterações/refatoração:

Nome de variáveis com intensão para auxiliar na compreensão:

arr - > arrayElementsToOrder

Extração de Métodos por meio da criação do método swap transferindo o trecho de código responsável pela ordenação dos números para dentro de um método específico que realiza somente essa alteração, evitando assim a duplicação e facilitando a leitura/entendimento.

Remoção da variável n, correspondente ao número de elementos do array a ser ordenado, substituindo dentro dos métodos pela utilização do arrayElementsToOrder.length para a extração da quantidade de elementos correspondentes ao total, minimizando assim os parâmetros a serem passados.

Padronização da linguagem geral do código para inglês (variáveis, métodos...).

Formatação por meio da remoção de espaçamentos desnecessários e a adição de espaços entre as funções a fim de auxiliar na compreensão do código.

Refatoração no for do método printArray para usar somente o (i : arrayElementsToOrder, para simplificação do código.

Reposicionamento dos métodos para que os métodos chamados fiquem abaixo dos chamadores.

Simplificação da função bubbleSort com a extração/criação do método isSorted para a redução da complexidade ciclomática facilitando assim a legibilidade.

Redução da quantidade de linhas do método bubbleSort por meio da extração dos métodos descritos anteriormente.

Redução/retirada dos comentários por conta da melhor organização do código, pois o código em si está mais auto-explicativo.

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

Para a disciplina de Desenvolvimento Web 1, o projeto proposto teve como foco a construção de dois CRUDs relacionados às entidades Aluno e Curso, utilizando o framework Angular para o frontend. A persistência dos dados foi realizada por meio do Local Storage, dispensando a necessidade de um backend. Essa abordagem possibilitou uma imersão mais profunda nos fundamentos do desenvolvimento de interfaces web modernas, e boas práticas de usabilidade. A estrutura do projeto, combinada aos questionários aplicados ao longo do curso, proporcionou uma base sólida sobre como soluções web podem ser construídas com agilidade e foco no cliente, reforçando princípios fundamentais do desenvolvimento ágil.

Já em Desenvolvimento Web 2, a proposta foi expandida com a implementação de um backend utilizando Spring Boot, integrado a um banco de dados PostgreSQL. Essa evolução permitiu compreender de forma mais clara como as camadas de um sistema web se comunicam. Embora a execução prática do projeto não tenha sido realizada diretamente, os conteúdos teóricos reforçaram a importância de arquiteturas bem definidas e da integração entre frontend e backend em ambientes de desenvolvimento colaborativo.

Essas disciplinas mostraram-se fortemente conectadas com outras áreas do curso, como Banco de Dados, ao exigirem planejamento, modelagem e entendimento das estruturas envolvidas em uma aplicação real. A combinação desses conhecimentos é essencial para o desenvolvimento ágil de software, pois permite entregas rápidas e iterativas com valor agregado, além de facilitar a adaptação a novas necessidades ao longo do projeto. Para essas disciplinas, não houve a entrega de projetos.

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

Para a disciplina de UX no Desenvolvimento Ágil de Software, o projeto desenvolvido teve como foco a criação de protótipos de interface para uma aplicação voltada ao gerenciamento de ponto de funcionários em uma indústria automobilística. A proposta era compreender e aplicar os princípios de UX (User Experience) dentro de um contexto ágil, priorizando as necessidades do usuário e a entrega contínua de valor. Durante o projeto, foi necessário construir personas, jornadas e fluxos, criar telas no Figma, alinhando a solução visual à experiência ideal mapeada, justificar a escolha das cores e elementos e por fim, realizar uma entrevista com um potencial usuário para obter feedbacks.

A importância dessa disciplina no contexto do desenvolvimento ágil ficou evidente na maneira como o design centrado no usuário foi incorporado desde as primeiras etapas do projeto. A prototipação rápida possibilita iterações frequentes, permitindo o refinamento contínuo do produto com base em testes e validações.

Além disso, o aprendizado adquirido nessa disciplina se conectou diretamente com outras áreas abordadas no curso. A integração com as disciplinas de Desenvolvimento Web e Mobile foi facilitada pelo uso do Figma, que gerou protótipos facilmente interpretáveis. Também houve alinhamento com práticas de engenharia de software e metodologias ágeis, especialmente no que diz respeito à priorização de funcionalidades e adaptação às mudanças. Assim, a disciplina de UX não só fortaleceu a visão de produto, mas também promoveu um entendimento mais profundo sobre a importância de colocar o usuário no centro do processo de desenvolvimento de software.

## 9.1 ARTEFATOS DO PROJETO

### 1 – Explicação do produto: App E-ponto

O E-ponto App é um aplicativo mobile que tem como objetivo o gerenciamento de ponto dos funcionários em uma indústria automobilística. Nele será possível realizar o registro do ponto em tempo real, verificar e ajustar os registros anteriores. Dessa forma, o controle dos registros pelos funcionários é facilitado e simplificado, sendo possível o acesso via smartphone. O acesso será realizado por meio da matrícula e senha dos funcionários.

#### Explicação do funcionamento e uso:

Para utilizar o aplicativo é necessário que os usuários, neste caso, os funcionários, realizem o login na tela inicial, que consiste em sua matrícula e sua senha. Na tela inicial é possível selecionar a opção de redefinição de senha que o redireciona a outra tela onde deve informar o seu e-mail corporativo cadastrado previamente no sistema e selecionar a opção Reset Password, que envia a nova senha para o usuário. Ao logar com sucesso no aplicativo, clicando no botão Logar, a próxima tela apresenta três opções, dois botões, um para registrar ponto, outro para abrir os registros anteriores e mais abaixo, um ícone para o logout, que redireciona a tela de login. Se escolhida a opção de registro de ponto, uma nova tela é carregada, com a data e hora atual, um botão voltar e um botão para o registro de ponto, onde se clicado, um pop up é carregado sinalizando o registro. Caso a opção selecionada na tela inicial seja registros anteriores, uma nova tela é apresentada, com todas os registros de um determinado dia. É possível selecionar a data em que se deseja carregar os registros correspondentes e editar os registros clicando no ícone a direita dos horários carregados, e selecionando um novo horário. Os registros podem ser as seguintes categorias: Ent. (entrada), Alm. (almoço) e Sai. (saída), essa informação está apresentada ao lado esquerdo do horário em cada linha de registro e é atribuída automaticamente pelo sistema, com base nos horários dos registros feitos pelo usuário ao longo de um dia. O aplicativo não permite remoção de registros. Ao editar, basta clicar no botão salvar edições, que um pop up é apresentado informado o sucesso dos ajustes.

### 2 -Telas desenvolvidas: (Telas também disponíveis em:

<https://www.figma.com/design/QfnuKtv1QIbQbL524XtoZZ/Trabalho-Ux?node-id=0-1&t=0GNk21EF0kw1mANC-1>)

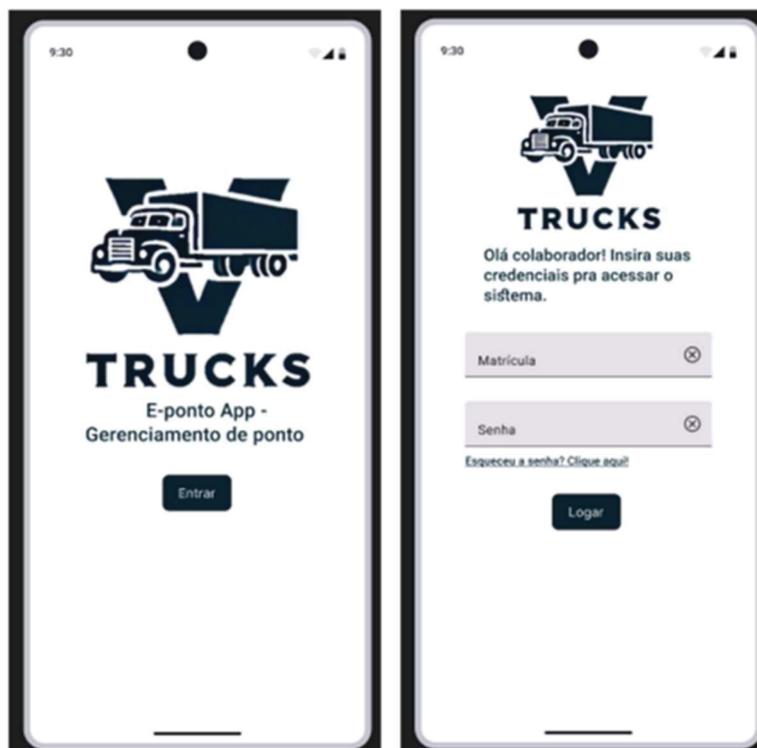


Figura 1 - Tela inicial

Figura 2- Tela Login

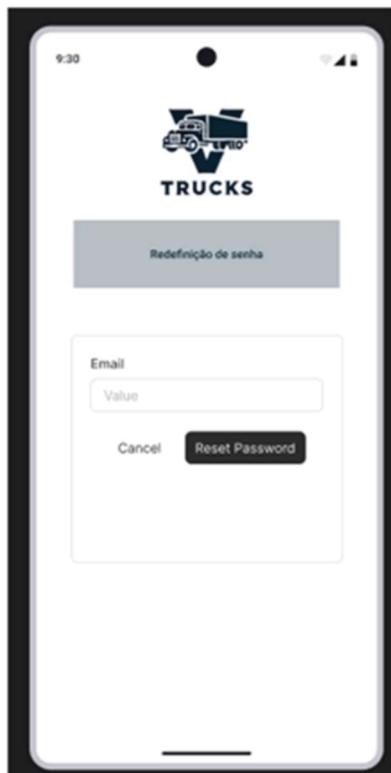


Figura 3- Tela de redefinição de senha



Figura 4 - Tela de seleção inicial

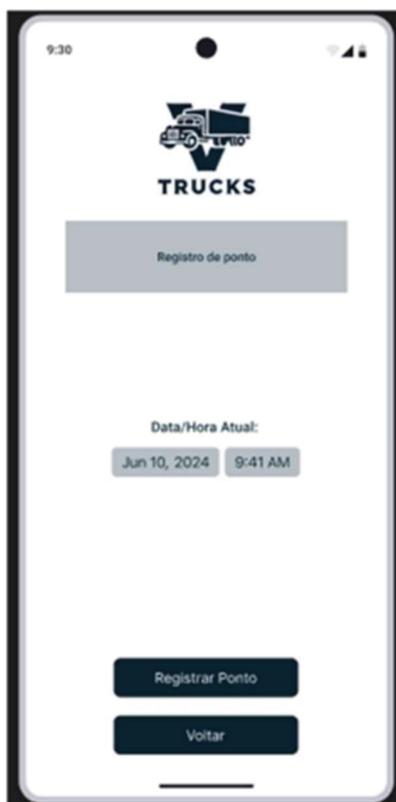


Figura 5 - Tela de Registro de ponto

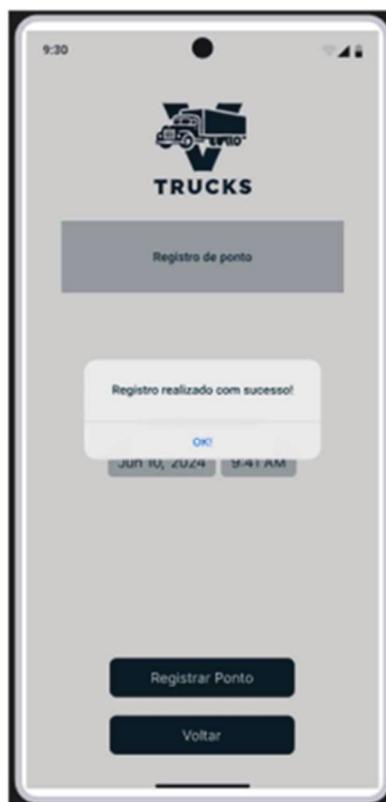


Figura 6 - Tela de Registro de ponto pós registro

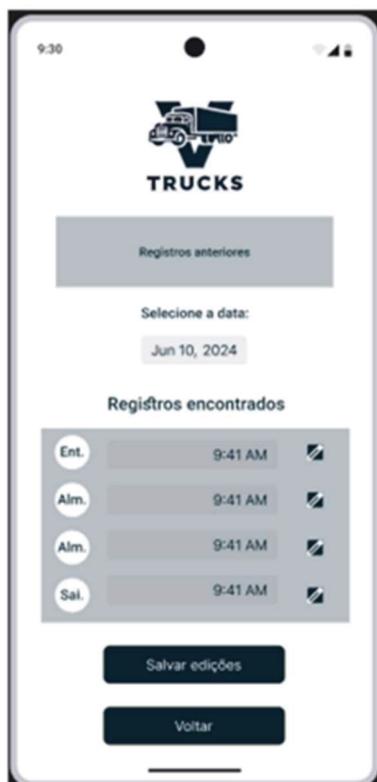


Figura 7 - Tela de Registros anteriores

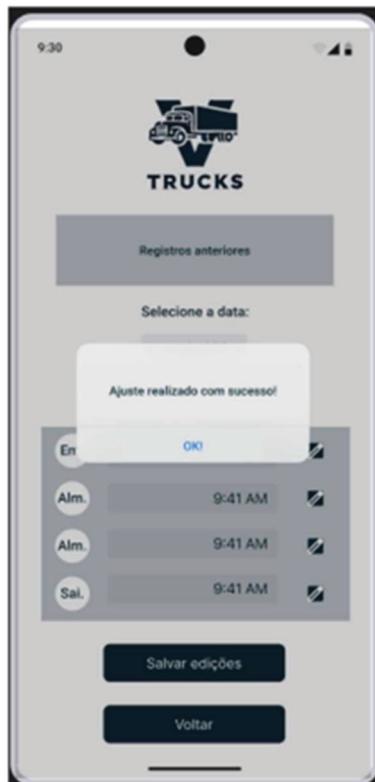


Figura 8 - Tela de Registro anteriores pós ajustes

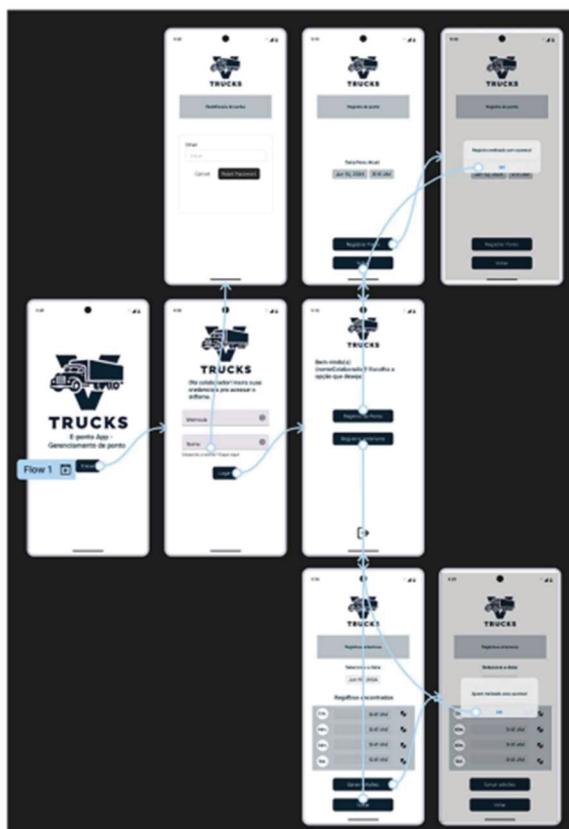


Figura 9 - Telas agrupadas e suas interações

### **3 - Explicar as escolhas feitas:**

As cores predominantes são o azul-escuro e branco, com algumas variações de tons de azul em algumas partes, essa escolha de cores faz referência a cor da empresa, predominantemente azul, como pode ser visto no ícone/logotipo presente em todas as telas, e a escolha do branco se dá pelo grande contraste com o azul utilizado, facilitando a acessibilidade e visibilidade das informações e elementos existentes na aplicação. Os tons utilizados permitem que uma gama de funcionários utilize de forma satisfatória a aplicação, por mais que tenha algum tipo de alteração de visão. As fontes foram escolhidas com base na sugestão do Material Design e dos elementos IOS, que já são otimizadas para as mais diversas aplicações. O layout utilizado também seguiu os modelos existentes no Material Design e dos elementos IOS pela padronização dos recursos, amplamente já utilizados nos mais diversos aplicativos.

### **4- Feedback de um possível usuário:**

Luiz, Analista de Dados em uma empresa automobilística. Perfil: 35 anos, trabalha no modelo híbrido, realiza ajustes de ponto com certa frequência por esquecimento.

Boa escolha de cores e tamanho de fontes, considerando que utiliza óculos, portanto fontes pequenas em aplicações dificultam o uso. Achou que seria fácil de utilizar por conta da navegação, das telas simplificadas e seria bem útil, principalmente nos dias de trabalho em home office. Podia contar com algum lembrete ou notificação no smartphone pra auxiliar a lembrar de registrar o ponto e permitir a criação de novos registros, não somente a edição dos já existentes. Disse que o sistema seria mais intuitivo e acessível do que o atual utilizado na empresa em que trabalha, via página Web.

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

Paras as disciplinas de MOB 1 e MOB 2 (Desenvolvimento Mobile 1 e 2), os projetos propostos buscaram consolidar os conhecimentos práticos no desenvolvimento de aplicativos Android, com forte ênfase na aplicação de princípios do desenvolvimento ágil. Em MOB 1, o foco esteve na criação de um aplicativo móvel funcional para controle de finanças pessoais, no qual o usuário poderia registrar receitas e despesas, promovendo maior controle e visibilidade de seus gastos. Os conteúdos visualizados permitiram o contato com os fundamentos da programação Android, como a manipulação de interfaces, persistência de dados e uso de componentes visuais interativos.

Já em MOB 2, a proposta evoluiu para a construção de um aplicativo que consumisse uma API externa baseada na temática de Harry Potter, enfatizando o uso de web services REST e a manipulação de dados JSON. O desafio era aprofundar o domínio das operações assíncronas, fundamentais para garantir uma boa experiência do usuário, mantendo o aplicativo fluido e responsivo. Embora os projetos tenham sido opcionais e não implementados na prática, as atividades teóricas e os questionários aplicados ao longo da disciplina foram essenciais para a assimilação dos conceitos-chave, como ciclo de vida das aplicações, arquitetura mobile e boas práticas de desenvolvimento.

A importância dessas disciplinas no contexto do desenvolvimento ágil de software se reflete na ênfase em entregas incrementais, na flexibilidade para lidar com mudanças e na valorização do feedback rápido dos usuários – princípios essenciais no universo mobile. A integração com outras disciplinas, como Introdução à Programação e UX, foi notável, pois os conhecimentos prévios em lógica de programação, prototipagem de interfaces e modelagem de dados forneceram base sólida para compreender os desafios e possibilidades do desenvolvimento mobile. Assim, MOB 1 e MOB 2 reforçaram a importância de uma abordagem multidisciplinar e iterativa na construção de soluções digitais modernas e centradas no usuário.

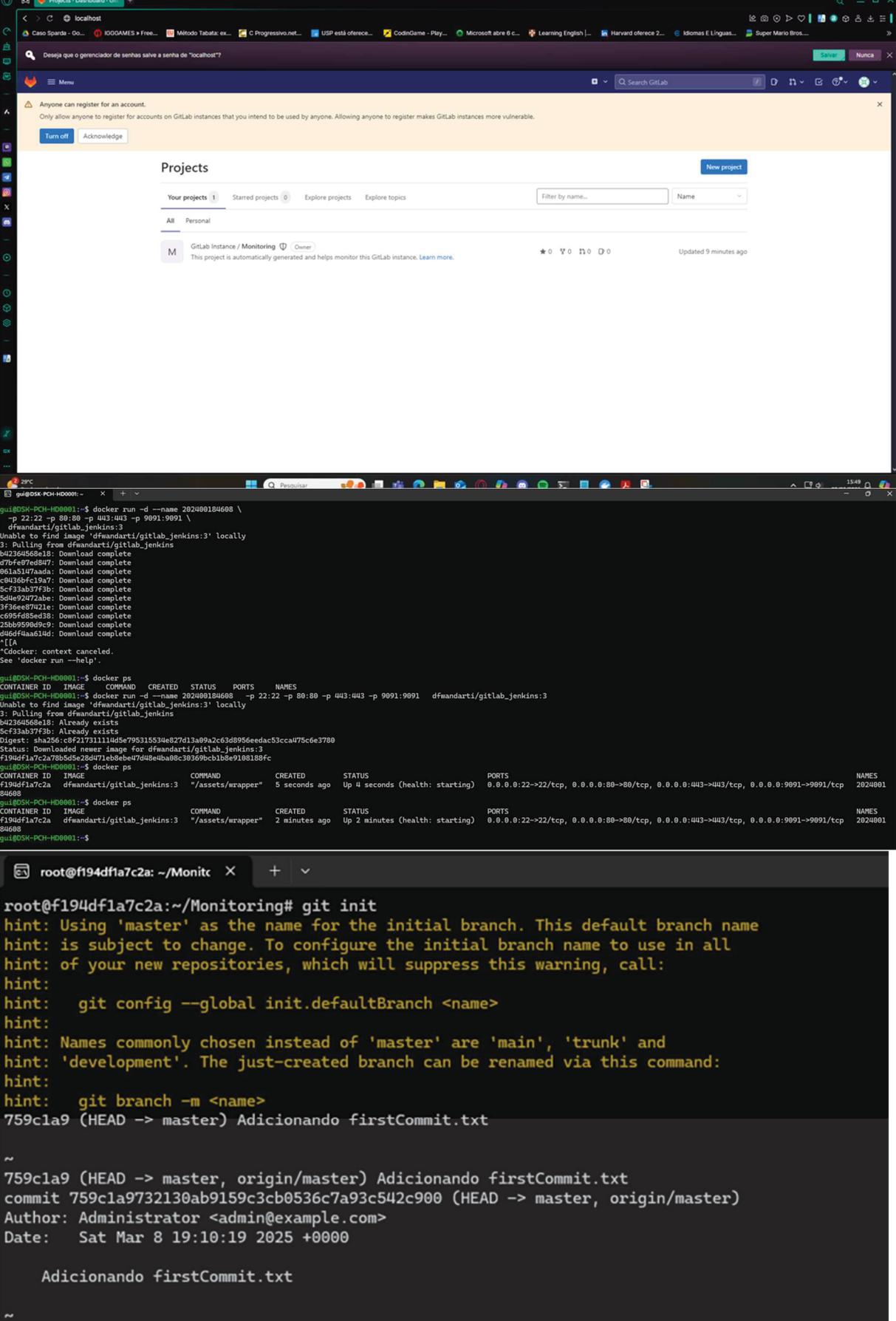
## 11 DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS)

Para a disciplina de Infraestrutura para Desenvolvimento e Implantação de Software (DevOps), o projeto final proporcionou uma vivência prática dos principais conceitos que sustentam a cultura DevOps, elemento central para o desenvolvimento ágil de software. O desafio envolveu a criação e configuração de um ambiente baseado em contêineres utilizando Docker, além da aplicação de práticas de versionamento e integração contínua com o uso do Git e GitLab. A atividade teve início com a execução de um contêiner a partir da imagem `dfwandarti/gitlab_jenkins:3`, utilizando como nome identificador a matrícula do aluno e expondo portas relevantes para o funcionamento de serviços (22, 80, 443 e 9091). O projeto demandou ainda a navegação interna no contêiner para recuperação de credenciais e posterior autenticação no GitLab, onde comandos Git como *commit* e *push* foram empregados para simular cenários reais de colaboração em equipe.

Um aspecto importante foi a possibilidade de utilizar tanto a linha de comando quanto interfaces gráficas para operar o Docker, o que ampliou a percepção sobre as diversas formas de automatizar processos de desenvolvimento e implantação. A disciplina se mostrou fundamental ao demonstrar como a automação, o versionamento e a integração contínua otimizam o ciclo de entrega de software e minimizam riscos operacionais.

Além disso, os conhecimentos adquiridos se mostraram complementares a outras disciplinas, como Web e Desenvolvimento Mobile, onde os conceitos de deploy contínuo e controle de versões são recorrentes. A disciplina de Infraestrutura reforçou o papel estratégico da colaboração entre times de desenvolvimento e operações, ressaltando como práticas DevOps aceleram entregas, melhoram a qualidade do produto e respondem de forma ágil às mudanças do mercado e às necessidades dos usuários.

## 11.1 ARTEFATOS DO PROJETO



The image shows a web browser window displaying the GitLab Projects dashboard. The dashboard includes a navigation menu, a search bar, and a list of projects. A notification banner at the top states: "Anyone can register for an account. Only allow anyone to register for accounts on GitLab instances that you intend to be used by anyone. Allowing anyone to register makes GitLab instances more vulnerable." Below this, there are buttons for "Turn off" and "Acknowledge".

The main content area shows the "Projects" section with a "New project" button. Under "Your projects", there is one project listed: "GitLab Instance / Monitoring" (Owner). A message below the project name states: "This project is automatically generated and helps monitor this GitLab instance. Learn more." The project has 0 stars, 0 forks, 0 issues, and 0 discussions, and was updated 9 minutes ago.

Below the browser window, a terminal window shows the following commands and output:

```

git@05sk-PCH-HD0001:~$ docker run -d --name 202400184608 \
-p 22:22 -p 80:80 -p 443:443 -p 9091:9091 \
dfwandarti/gitlab_jenkins:3
Unable to find image 'dfwandarti/gitlab_jenkins:3' locally
3: Pulling from dfwandarti/gitlab_jenkins
b42364568e18: Download complete
d7bfe97ed847: Download complete
061a5147aada: Download complete
e9426f312d7: Download complete
5cf33ab37f3b: Download complete
5d4e92472abe: Download complete
3f36ee87421e: Download complete
e69f468ed38: Download complete
25bb9590d9c9: Download complete
d46df4aa614d: Download complete
[[
*docker: context canceled.
See 'docker run --help'.

git@05sk-PCH-HD0001:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
git@05sk-PCH-HD0001:~$ docker run -d --name 202400184608 -p 22:22 -p 80:80 -p 443:443 -p 9091:9091 dfwandarti/gitlab_jenkins:3
Unable to find image 'dfwandarti/gitlab_jenkins:3' locally
3: Pulling from dfwandarti/gitlab_jenkins
b42364568e18: Already exists
5cf33ab37f3b: Already exists
Digest: sha256:c8f21731114d5e795315534e827d13a09a2c63d895eedac53cca475c6e3780
Status: Downloaded newer image for dfwandarti/gitlab_jenkins:3
f194df1a7c2a78b5d5e28d471eb8ebef7d40e4ba0c30369cb1b8e9108188fc
git@05sk-PCH-HD0001:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
f194df1a7c2a  dfwandarti/gitlab_jenkins:3  "/assets/wrapper"       5 seconds ago   Up 4 seconds (health: starting)  0.0.0.0:22->22/tcp, 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:9091->9091/tcp 202400184608
git@05sk-PCH-HD0001:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
f194df1a7c2a  dfwandarti/gitlab_jenkins:3  "/assets/wrapper"       2 minutes ago   Up 2 minutes (health: starting)  0.0.0.0:22->22/tcp, 0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:9091->9091/tcp 202400184608
git@05sk-PCH-HD0001:~$

```

Below the terminal window, another terminal window shows the following commands and output:

```

root@f194df1a7c2a: ~/Monitc
root@f194df1a7c2a:~/Monitoring# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
759c1a9 (HEAD -> master) Adicionando firstCommit.txt

~
759c1a9 (HEAD -> master, origin/master) Adicionando firstCommit.txt
commit 759c1a9732130ab9159c3cb0536c7a93c542c900 (HEAD -> master, origin/master)
Author: Administrator <admin@example.com>
Date: Sat Mar 8 19:10:19 2025 +0000

Adicionando firstCommit.txt

~

```

The image shows two parts of a GitLab installation process. The top part is a browser window displaying the GitLab sign-in page. The page has the GitLab logo and the text "GitLab A complete DevOps platform". Below this, it states "GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security. This is a self-managed instance of GitLab." To the right is a sign-in form with fields for "Username or email" and "Password", a "Remember me" checkbox, a "Forgot your password?" link, and a "Sign in" button. Below the form is a link for "Don't have an account yet? Register now".

The bottom part is a terminal window showing the command to view the initial root password:

```
gui@DSK-PCH-H00001:~$ docker exec -it 202480184688 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 first 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: lf/13UDmw6fiuFSUpjR/vBZOPcdx8c09ey2RvFILg=
# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
gui@DSK-PCH-H00001:~$
```

## 12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

Para a disciplina de Testes Automatizados, o projeto final consistiu no desenvolvimento de um teste funcional automatizado utilizando TypeScript em conjunto com o framework Selenium. A proposta prática do projeto envolveu o uso do navegador Google Chrome para acessar o endereço <https://pt.anotepad.com> e, de forma automatizada, preencher o campo de título com a mensagem “Entrega trabalho TEST DAS 2024”, além de registrar no corpo da nota o nome e a matrícula do aluno. A execução do teste simulou uma interação real de um usuário com a aplicação, evidenciando o domínio de técnicas de automação de ponta a ponta (end-to-end).

A disciplina foi fundamental para reforçar a importância da automação como um pilar do desenvolvimento ágil de software. Em ambientes ágeis, as entregas são incrementais e contínuas, exigindo testes frequentes para garantir que novas funcionalidades não impactem negativamente recursos já existentes. A automação garante rapidez, confiabilidade e rastreabilidade nos testes, permitindo identificar regressões com maior eficiência e promover uma entrega contínua de valor ao cliente.

Além disso, o projeto demonstrou como os testes automatizados podem se integrar com outras áreas do desenvolvimento, como DevOps (com a integração contínua de testes nas pipelines), desenvolvimento web (validando a interface e as funcionalidades implementadas), e levantamento de requisitos (assegurando que os critérios de aceitação estejam sempre cobertos). Assim, a disciplina de Testes Automatizados consolidou práticas essenciais para a qualidade e a robustez do software desenvolvido ao longo do curso, completando a formação técnica com foco na excelência e na agilidade.

## 12.1 ARTEFATOS DO PROJETO

```

pom.xml x
C: > Users > Guilherme > Downloads > pom.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>org.example</groupId>
8      <artifactId>trabalhoTEST</artifactId>
9      <version>1.0-SNAPSHOT</version>
10
11     <properties>
12         <maven.compiler.source>17</maven.compiler.source>
13         <maven.compiler.target>17</maven.compiler.target>
14         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15     </properties>
16
17     <dependencies>
18         <dependency>
19             <groupId>com.microsoft.playwright</groupId>
20             <artifactId>playwright</artifactId>
21             <version>1.51.0</version>
22         </dependency>
23     </dependencies>
24     <build>
25         <plugins>
26             <plugin>
27                 <groupId>org.apache.maven.plugins</groupId>
28                 <artifactId>maven-compiler-plugin</artifactId>
29                 <version>3.10.1</version>
30                 <!-- References to interface static methods are allowed only at source level 1.8 or above -->
31                 <configuration>
32                     <source>1.8</source>
33                     <target>1.8</target>
34                 </configuration>
35             </plugin>
36         </plugins>
37     </build>
38 </project>

```

The screenshot shows an IDE interface with the following components:

- Project Explorer:** Shows the project structure for 'trabalhoTEST', including 'src', 'resources', and 'test' directories.
- Code Editor:** Displays the 'TrabalhoTEST.java' file with the following code:

```

package org.example;
import com.microsoft.playwright.*;

public class TrabalhoTEST {
    public static void main(String[] args) {
        // Inicializando o Playwright e o navegador
        try (Playwright playwright = Playwright.create()) {
            Browser browser = playwright.chromium().launch(new BrowserType.LaunchOptions().setHeadless(false));
            BrowserContext context = browser.newContext();
            Page page = context.newPage();

            // Abrindo o site
            page.navigate(url: "https://pt.anotepad.com");

            // Usando o locator para o título e preenche-lo
            Locator titleField = page.locator(selector: "h1");
            titleField.fill(value: "Entrega trabalho TEST DAS 2024");

            // Usando o locator para o campo de texto e preenche-lo
            Locator bodyField = page.locator(selector: "input");
            bodyField.fill(value: "Guilherme Eduardo Rodrigues - Matrícula: 202400184608");

            // Adicionando um Timeout para ajudar na visualização de ação
            page.waitForTimeout(3000);

            // Fechando o navegador
            browser.close();
        }
    }
}

```
- Maven:** Shows the Maven dependency tree for 'trabalhoTEST', including 'Lifecycle', 'Plugins', 'Dependencies', and 'Repositories'.
- Run Console:** Shows the output of the 'TrabalhoTEST' run, indicating 'Process finished with exit code 0'.
- Project:** A dropdown menu is visible in the bottom right corner.

The image shows a screenshot of an IDE (IntelliJ IDEA) and a web browser. The IDE window displays a project named "trabalhoTEST" with a file structure including "src", "main", "java", "resources", "test", "target", ".gitignore", "pom.xml", "External Libraries", and "Scratches and Consoles". The "Run" tab shows the command prompt output: "\*C:\Program Files\Java\jdk-17\b". The browser window shows the "aNotepad" website (pt.anotepad.com) with the title "Entrega trabalho TEST DAS 2024". The page content includes the text "Guilherme Eduardo Rodrigues - Matricula: 202400184608". The browser also displays an advertisement for "THE FANTASTIC 4 JOIN MARVEL RIVALS!" with a "PLAY FREE NOW" button and a "Ver mais" button.

## 13 CONCLUSÃO

Foram apresentadas, neste memorial, todas as disciplinas cursadas ao longo dessa trajetória acadêmica e seus respectivos projetos que contribuíram com toda a construção de conhecimento obtida durante a graduação. Os temas aprendidos foram desde conhecimentos teóricos até conhecimento práticos, numa abordagem bem ampla de conhecimentos relacionados ao desenvolvimento ágil de software.

Ao longo do conteúdo, foram descritos os fundamentos teóricos que sustentam o desenvolvimento ágil, incluindo a modelagem de requisitos, o planejamento iterativo e incremental, e as práticas de documentação enxuta. A sequência lógica de estudos percorreu desde o desenho de sistemas com apoio de diagramas e narrativas de usuário até a aplicação prática com frameworks contemporâneos, como Angular e Spring Boot, evidenciando a relevância da integração entre teoria e prática.

As atividades práticas permitiram explorar tecnologias diversas e simular ambientes reais de projeto, nos quais a aplicação de métodos ágeis, como Scrum e Kanban, contribuiu diretamente para a organização, a entrega contínua e o foco no valor ao cliente. Foram abordadas ainda as disciplinas de Experiência do Usuário (UX), Desenvolvimento Web e Mobile, bem como práticas de DevOps, que reforçaram a importância da automação, integração contínua e versionamento como pilares da agilidade moderna.

Contudo, a aplicação do desenvolvimento ágil enfrenta alguns desafios práticos importantes, entre eles:

- Integração entre práticas e ferramentas: adotar metodologias ágeis exige uma harmonização entre diversas ferramentas de desenvolvimento, testes e colaboração, o que demanda aprendizado contínuo e alinhamento entre os membros da equipe.
- Cultura e resistência à mudança: a transição de modelos tradicionais para abordagens ágeis envolve mudança cultural, o que pode gerar resistência, especialmente em ambientes com estruturas rígidas.
- Manutenção da qualidade em ritmo acelerado: o desafio de manter padrões elevados de qualidade, mesmo em ciclos curtos de entrega, exige disciplina no uso de testes automatizados, revisão de código e boas práticas de programação.

Por fim, pode-se concluir que, o desenvolvimento ágil de software vai além de metodologias, ele representa um modo de pensar e agir colaborativamente. Os conteúdos apresentados neste documento reforçam a importância de um profissional versátil, capaz de planejar, implementar, testar e entregar soluções de forma eficiente, contínua e centrada no usuário. Os desafios observados não são obstáculos, mas pontos de atenção que, quando superados, fortalecem a prática ágil e sua aplicação no mercado real.

## 14 REFERÊNCIAS

BECK, Kent et al. Manifesto Ágil para Desenvolvimento de Software. 2001. Disponível em: <https://agilemanifesto.org>. Acesso em: 19 jun. 2025.

HIGHSMITH, Jim. **Agile Software Development Ecosystems**. Boston: Addison-Wesley, 2002.

HUMBLE, Jez; FARLEY, David. **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**. Boston: Addison-Wesley, 2010.

LARMAN, Craig. **Utilizando UML e Padrões**. 2. ed. Porto Alegre: Bookman, 2003.

MARTIN, Robert C. **Código limpo: habilidades práticas do Agile software**. Rio de Janeiro: Alta Books, 2008.

NORMAN, Donald. **O design do dia a dia**. Rio de Janeiro: Rocco, 2013.

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