

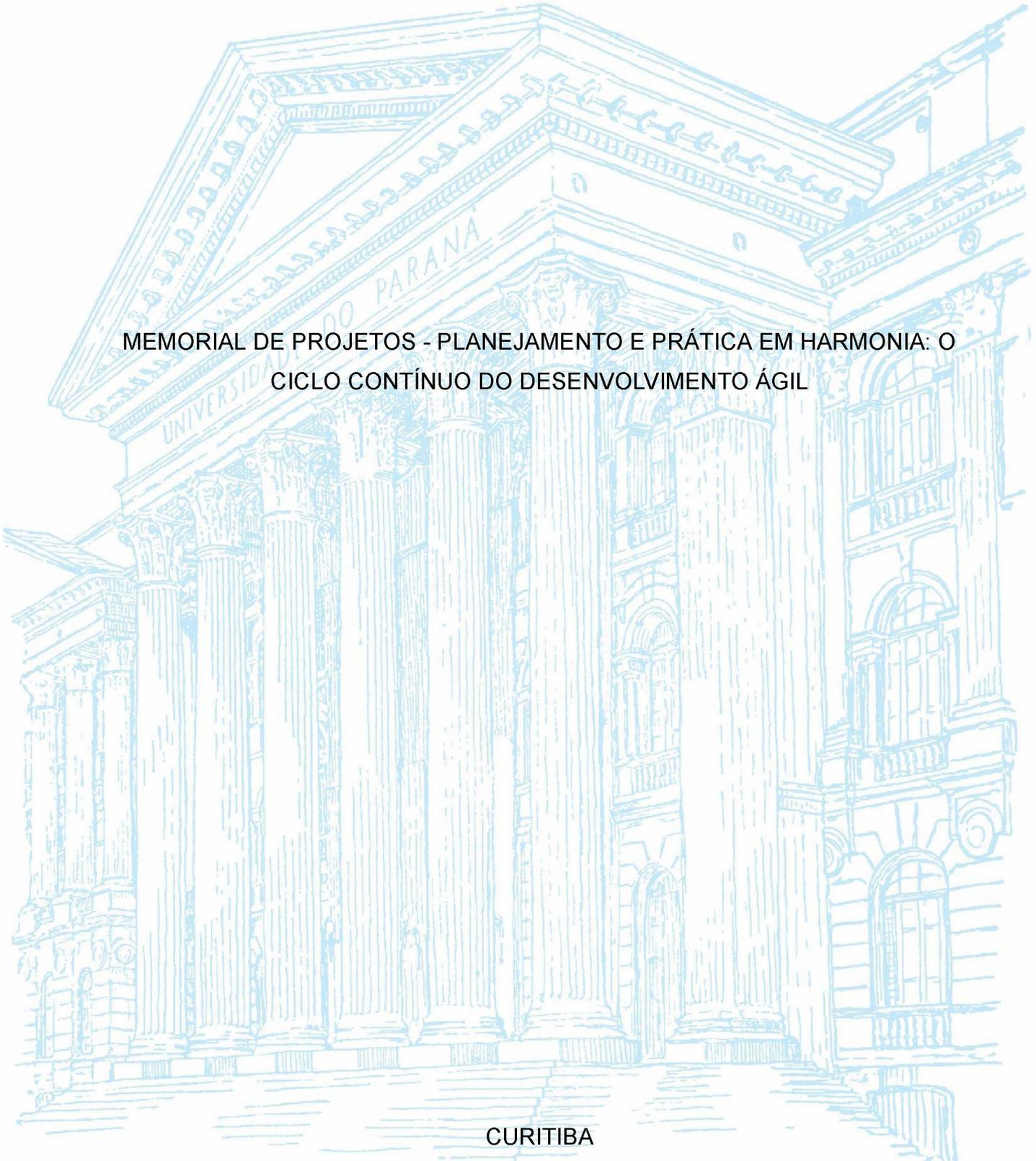
UNIVERSIDADE FEDERAL DO PARANÁ

JULIO CESAR BECKER VASCONCELOS

MEMORIAL DE PROJETOS - PLANEJAMENTO E PRÁTICA EM HARMONIA: O
CICLO CONTÍNUO DO DESENVOLVIMENTO ÁGIL

CURITIBA

2025



JULIO CESAR BECKER VASCONCELOS

MEMORIAL DE PROJETOS - PLANEJAMENTO E PRÁTICA EM HARMONIA: O
CICLO CONTÍNUO DO DESENVOLVIMENTO ÁGIL

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.

Orientadora: Profa. Dra. Rafaela Mantovani Fontana

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 **JULIO CESAR BECKER VASCONCELOS**, intitulada: **MEMORIAL DE PROJETOS - PLANEJAMENTO E PRÁTICA EM HARMONIA: O CICLO CONTÍNUO DO DESENVOLVIMENTO ÁGIL**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua **aprovação** no rito de defesa.

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

Curitiba, 28 de Outubro de 2025.


RAFAELA MANTOVANI FONTANA
Presidente da Banca Examinadora


JAIME WOSZCZOWSKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

O Memorial de Projetos – Planejamento e Prática em Harmonia: O Ciclo Contínuo do Desenvolvimento Ágil apresenta um registro reflexivo da experiência no curso de Especialização em Desenvolvimento Ágil de Software da UFPR. A proposta central é evidenciar como o equilíbrio entre planejamento e execução sustenta o fluxo contínuo característico da abordagem ágil. A narrativa percorre de maneira sequencial as etapas do projeto, destacando como atividades de planejamento, definição de requisitos, modelagem e implementação técnica se alinham para responder a demandas reais. Ao longo do memorial, são reunidos os principais artefatos produzidos, que ilustram as entregas realizadas em cada fase. Cada disciplina é apresentada em diálogo com as demais, ressaltando não apenas seus objetivos individuais, mas também sua contribuição para a construção integrada do conhecimento. O espírito ágil se materializa nas práticas de ciclos iterativos, na incorporação de feedback constante e na valorização da colaboração entre os envolvidos.

Palavras-chave: desenvolvimento de software; planejamento; ágil; parecer técnico; metodologias ágeis.

ABSTRACT

The Project Memorial – Planning and Practice in Harmony: The Continuous Cycle of Agile Development presents a reflective record of the experience in the UFPR's Graduate Program in Agile Software Development. Its central purpose is to highlight how the balance between planning and execution sustains the continuous flow that characterizes the agile approach. The narrative follows the project's stages sequentially, emphasizing how activities such as planning, requirements definition, modeling, and technical implementation align to address real demands. Throughout the memorial, the main artifacts produced are gathered, illustrating the deliverables achieved in each phase. Each course unit is presented in dialogue with the others, underscoring not only its individual objectives but also its contribution to the integrated construction of knowledge. The agile spirit becomes tangible in the practices of iterative cycles, the incorporation of constant feedback, and the emphasis on collaboration among those involved.

Keywords: software development; planning; agile; technical report; agile methodologies.

SUMÁRIO

1	PARECER TÉCNICO.....	7
2	DISCIPLINA: MADS - MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE.....	9
3	DISCIPLINA: MAG1 E MAG2 - MODELAGEM ÁGIL DE SOFTWARE (1 E 2).....	11
4	DISCIPLINA: GAP1 E GAP2 - GERENCIAMENTO ÁGIL DE PROJETOS DE SOFTWARE 1 E 2.....	17
5	DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO.....	20
6	DISCIPLINA: BD – BANCO DE DADOS.....	22
7	DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO.....	26
8	DISCIPLINA: WEB1 E WEB2 – DESENVOLVIMENTO WEB 1 E 2.....	30
9	DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE....	31
10	DISCIPLINA: MOB1 E MOB2 – DESENVOLVIMENTO MOBILE 1 E 2.....	38
11	DISCIPLINA: INFRA - INFRAESTRUTURA PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOFTWARE (DEVOPS).....	39
12	DISCIPLINA: TEST – TESTES AUTOMATIZADOS.....	42
13	CONCLUSÃO.....	44
	REFERÊNCIAS.....	46

1 PARECER TÉCNICO

Este parecer técnico apresenta uma análise articulada dos projetos realizados durante o curso de Especialização em Desenvolvimento Ágil de Software da UFPR, destacando como o conteúdo das disciplinas dialoga com a aplicação prática dos princípios ágeis. A trajetória desenvolvida ao longo do curso evidencia uma evolução que parte da concepção inicial de ideias e culmina na entrega de soluções completas, sempre guiadas por valores como colaboração, adaptação e entrega contínua de valor.

Nas etapas iniciais, disciplinas voltadas ao planejamento e à análise de requisitos ofereceram a base necessária para compreender o ciclo de vida do software sob a ótica ágil. O estudo do Manifesto Ágil (Highsmith, 2002) e de seus princípios reforçou que mais importante do que seguir processos rígidos é garantir entregas constantes e manter flexibilidade diante de mudanças (Beck et al., 2001). Os primeiros artefatos, como mapas mentais, épicos e histórias de usuário, estão documentados nas seções iniciais deste memorial.

Na sequência, disciplinas como MAG1 e MAG2 (Modelagem Ágil de Software) possibilitaram traduzir requisitos em representações gráficas, por meio de diagramas de caso de uso, sequência e classes. Esses recursos facilitaram a comunicação entre as equipes e apoiaram o refinamento incremental das funcionalidades. Conforme aponta Larman (2003), a modelagem ágil não busca a perfeição absoluta, mas sim oferecer suporte à tomada de decisões técnicas durante o desenvolvimento.

A dimensão da gestão foi abordada em disciplinas como GAP1 e GAP2 (Gerenciamento Ágil de Projetos de Software 1 e 2), que introduziram práticas de planejamento incremental, como release planning e o uso de quadros Kanban. Essas técnicas, fundamentadas nos valores de transparência e priorização de valor ao cliente (Poppendieck; Poppendieck, 2003), estão registradas nas seções destinadas ao planejamento do projeto.

Com a base conceitual consolidada, disciplinas práticas como WEB1, WEB2 (Desenvolvimento Web 1 e 2) e MOB1 (Desenvolvimento Mobile 1) possibilitaram o desenvolvimento de sistemas utilizando frameworks modernos, versionamento com Git, integração contínua e testes automatizados. Esse conjunto de práticas demonstrou como a modelagem e o planejamento se conectam diretamente ao ciclo

iterativo e incremental do desenvolvimento. O TDD (Test-Driven Development), abordado em TEST, reforçou a importância da qualidade e da cobertura de testes ao longo do projeto, conforme defendido por Martin (2008).

Outro ponto central foi a experiência do usuário (UX). A elaboração de protótipos navegáveis no Figma e a realização de testes de usabilidade evidenciaram a relevância do usuário no processo de criação. Esse ciclo contínuo de prototipação, avaliação e refinamento encontra respaldo em Norman (2013), que ressalta o design centrado no usuário como elemento essencial para a aceitação de sistemas.

As disciplinas de Infraestrutura para Desenvolvimento e Implantação de Software (DEVOPS) complementaram a formação ao integrar pipelines de CI/CD, automação de testes e ambientes de homologação. Esse conjunto de práticas reforça a ideia de que o ágil não se limita ao código, mas depende de uma cultura de entrega contínua e colaboração entre áreas técnicas e operacionais (Humble; Farley, 2010).

Assim, o memorial evidencia que os projetos desenvolvidos não foram etapas isoladas, mas sim partes de um processo fluido, iterativo e incremental, no qual cada disciplina contribuiu de forma essencial para a construção coletiva do conhecimento. A integração entre análise, modelagem, planejamento, codificação, testes e deploy contínuo demonstra a aplicação concreta dos princípios ágeis no ambiente acadêmico.

O título “Planejamento e Prática em Harmonia: O Ciclo Contínuo do Desenvolvimento Ágil” traduz de forma fiel a essência deste parecer, reforçando a ideia de que teoria e prática não são dimensões separadas, mas sim complementares, sustentando um fluxo adaptativo e contínuo como condição para o sucesso no desenvolvimento de software.

2 DISCIPLINA: MADS – MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

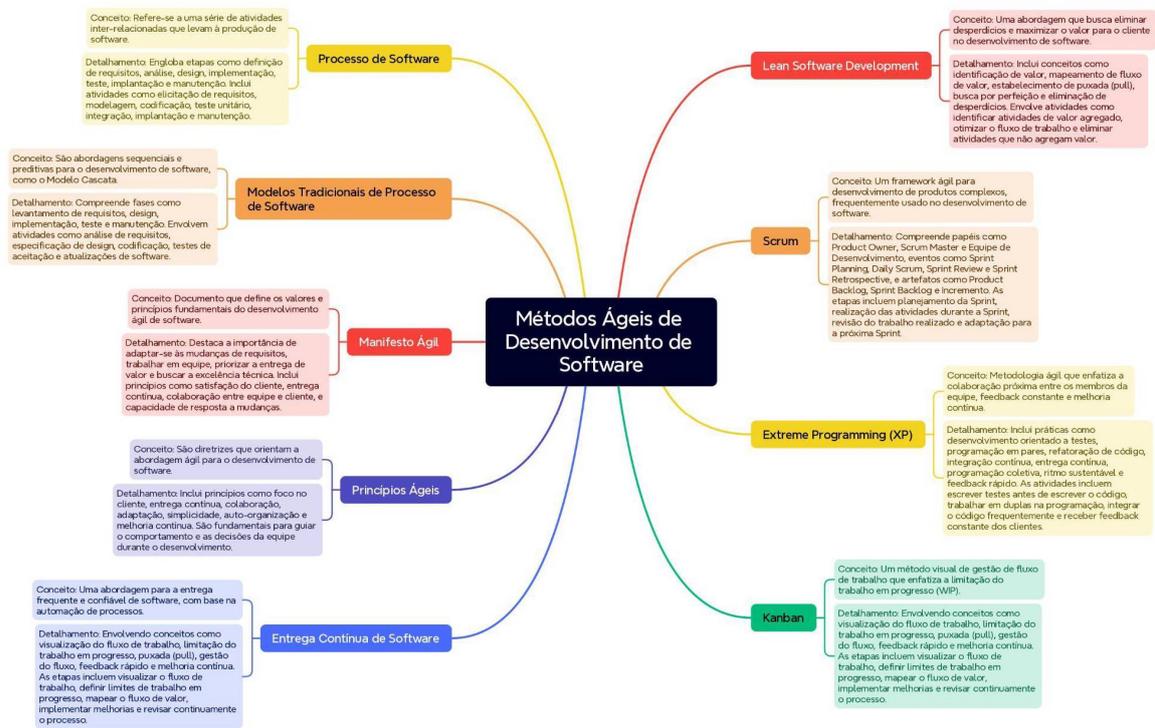
Na disciplina de MADS (Métodos Ágeis para Desenvolvimento de Software), o projeto final consistiu na criação de um mapa mental abrangente que reunia os principais conceitos e práticas relacionados ao desenvolvimento ágil. Esse material foi organizado em estrutura hierárquica, permitindo explorar e aprofundar tópicos como processo de software, modelos tradicionais, Manifesto Ágil, princípios fundamentais, além de frameworks e abordagens como Scrum, Kanban, Extreme Programming, Lean Software Development e entrega contínua. Cada ramificação do mapa foi construída a partir do conteúdo trabalhado em aula, oferecendo uma visão clara e integrada da metodologia ágil.

Esse trabalho teve papel decisivo na consolidação da base conceitual do curso, garantindo aos alunos uma compreensão sólida dos fundamentos que orientam o desenvolvimento ágil de software. Além disso, serviu de alicerce para a compreensão das demais disciplinas, nas quais os conceitos de iteração contínua, priorização de valor, feedback rápido e equipes auto-organizadas se manifestaram de forma recorrente em atividades de gestão, modelagem, testes e desenvolvimento.

Assim, a disciplina de MADS não apenas apresentou os métodos ágeis, mas também promoveu a construção de uma mentalidade voltada para a flexibilidade, a colaboração e a entrega incremental — valores que sustentam todo o ciclo de desenvolvimento ágil e reforçam a harmonia entre planejamento e prática ao longo da especialização oferecendo uma visão clara e integrada da metodologia ágil, conforme ilustra a FIGURA 1.

2.1 ARTEFATOS DO PROJETO

FIGURA 1 - ARTEFATO DA DISCIPLINA DE MADS



Presented with xmind

FONTE: O autor (2025)

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

Nas disciplinas de MAG1 e MAG2 (Modelagem Ágil de Software 1 e 2), o foco esteve na modelagem de um sistema a partir da aplicação prática dos princípios da modelagem ágil. O tema escolhido foi um sistema de gestão de condomínio, cuja representação envolveu a criação de diversos artefatos essenciais, como diagramas de caso de uso (níveis 1 e 2), ilustrados na FIGURA 2, histórias de usuário, regras de negócio, detalhadas na FIGURA 3, além de diagramas de classes (FIGURA 4) e de sequência (FIGURA 5) que detalharam todas as funcionalidades mapeadas.

Os diagramas de classes, sequência e casos de uso contribuíram para esclarecer fluxos de trabalho, responsabilidades dos componentes e interações entre os usuários e o sistema. Já os diagramas UML desempenharam papel importante na validação da arquitetura e no suporte técnico à equipe, funcionando como um guia visual para o desenvolvimento.

As histórias de usuário assumiram papel central na definição de requisitos, alinhadas à filosofia ágil de documentação enxuta e orientada ao valor de negócio. Cada história foi elaborada com critérios de aceitação, regras de negócio, observações técnicas e vínculos com outros artefatos, favorecendo tanto o alinhamento de expectativas quanto a entrega incremental de valor ao cliente — em consonância com os princípios do Manifesto Ágil (Beck et al., 2001).

O valor dessas disciplinas para o desenvolvimento ágil está na capacidade de equilibrar planejamento e flexibilidade. A modelagem orientada por histórias e diagramas possibilita adaptar o sistema a novas demandas sem comprometer a estrutura ou a consistência da solução. Além disso, o vínculo natural com outras disciplinas — como MADS, GAP1/GAP2, TEST e desenvolvimento web/mobile — demonstra a integração entre fundamentos conceituais e práticas técnicas.

Assim, MAG1 e MAG2 desempenham papel estratégico na especialização, pois transformam princípios em artefatos concretos que norteiam o avanço do projeto. Ao conectar teoria e prática, essas disciplinas asseguram que o desenvolvimento se mantenha coeso e alinhado aos pilares ágeis de colaboração, iteração, simplicidade e adaptação contínua.

3.1 ARTEFATOS DO PROJETO MAG 1

FIGURA 2 - ARTEFATO 1 DA DISCIPLINA DE MAG 1

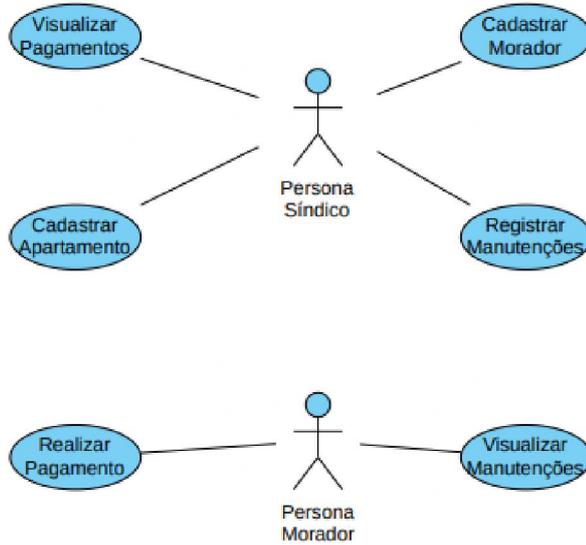


Diagrama de Caso de Uso Nível 1

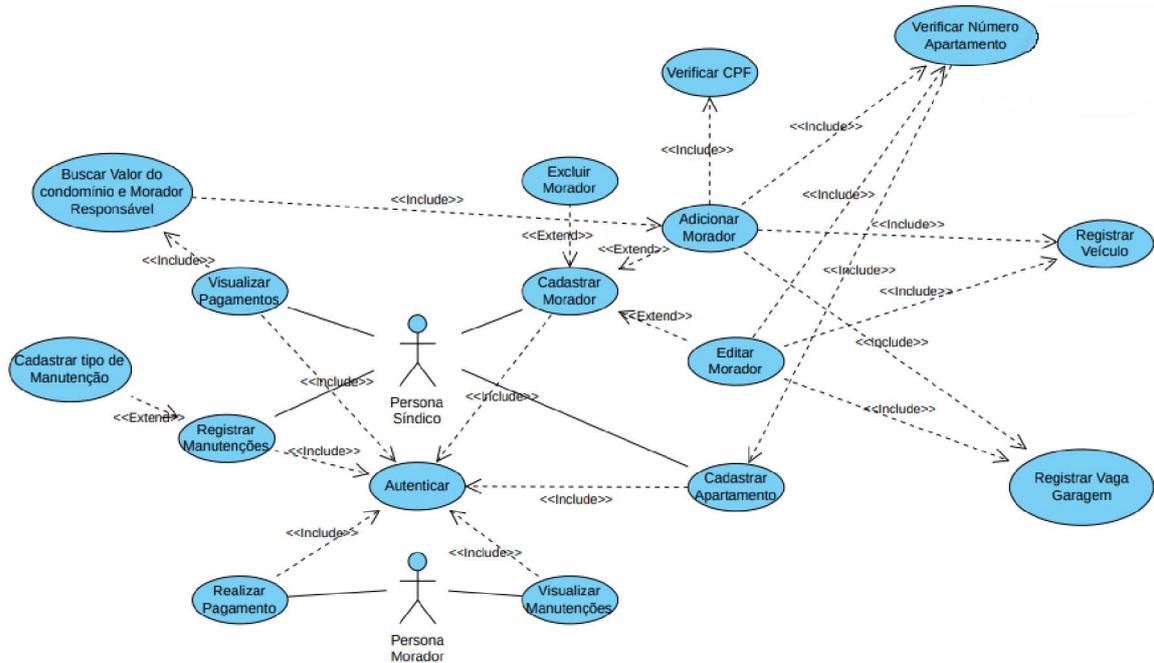


Diagrama de Caso de Uso Nível 2

FONTE: O autor (2025)

FIGURA 3 - ARTEFATO 2 DA DISCIPLINA DE MAG 1

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.

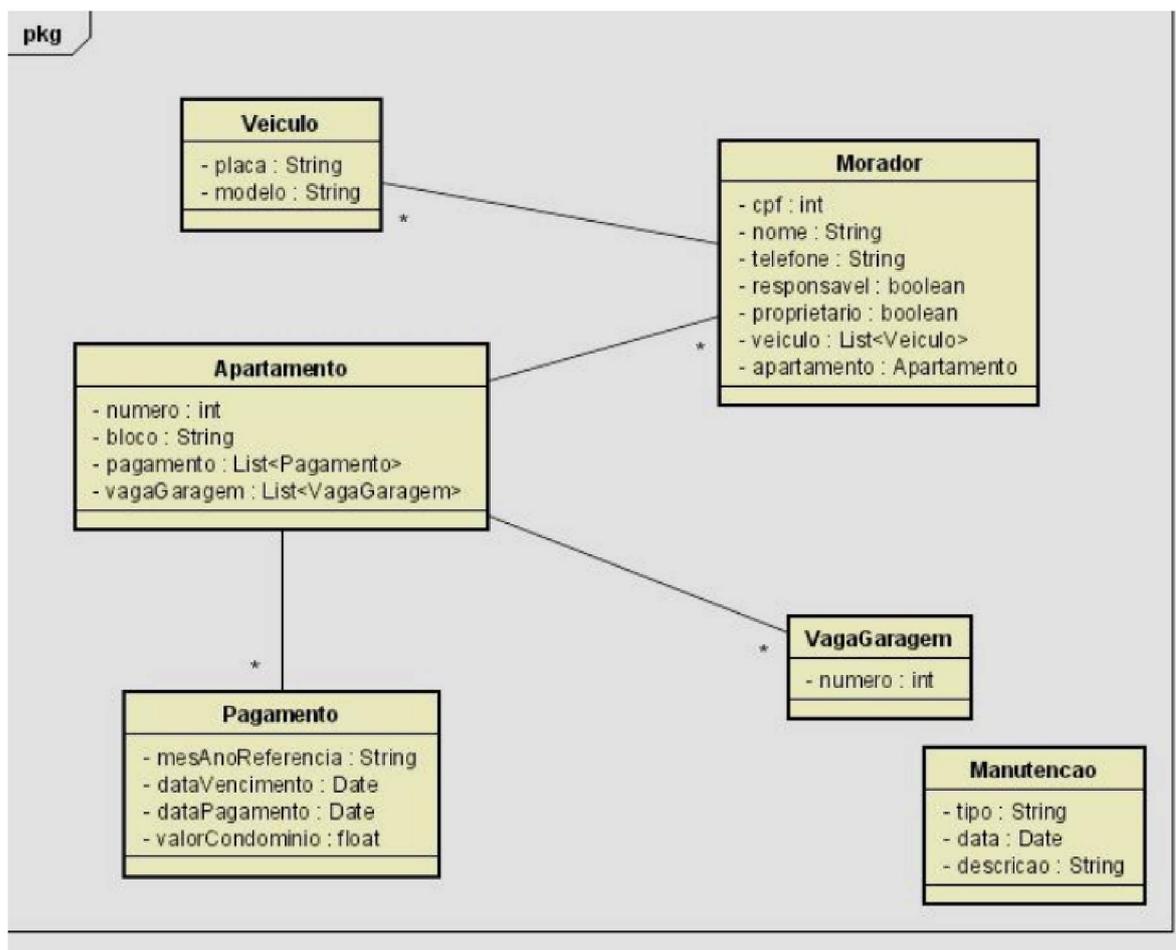
R11- O sistema deve permitir o registro de diferentes tipos de manutenções, como pintura, limpeza de caixa d'água, jardinagem, etc.

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

FONTE: O autor (2025)

3.2 ARTEFATOS DO PROJETO MAG 2

FIGURA 4 - ARTEFATO 1 DA DISCIPLINA DE MAG 2

Diagrama de Classes com atributos associados

FONTE: O autor (2025)

FIGURA 5 - ARTEFATO 2 DA DISCIPLINA DE MAG 2

Diagrama de Sequência Tela Morador

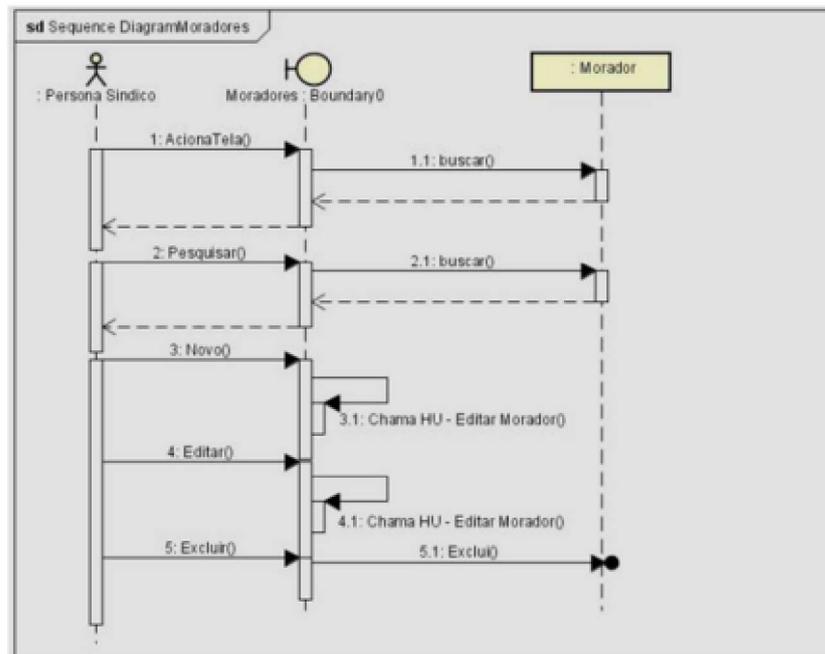
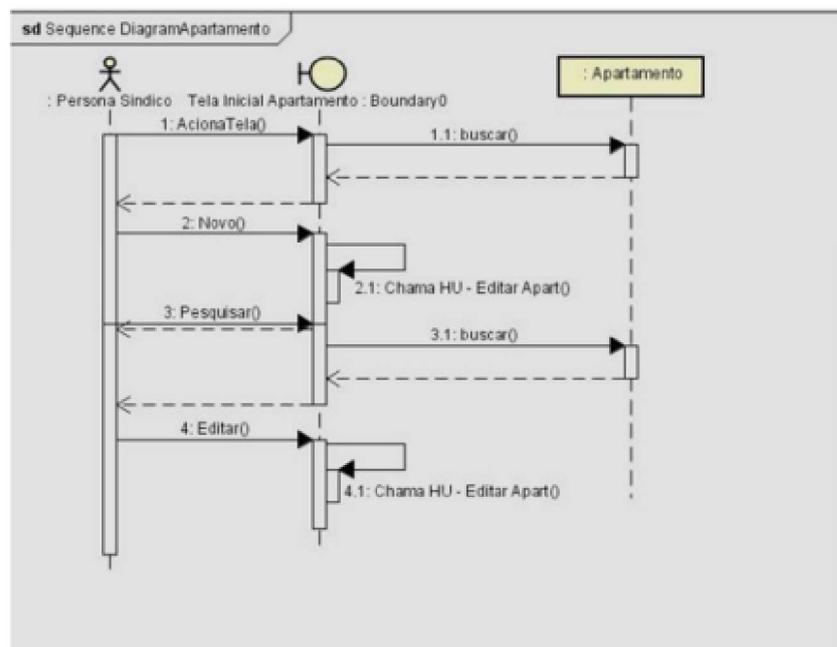


Diagrama de Sequência Tela Apartamento



FONTE: O autor (2025)

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

Nas disciplinas de GAP1 e GAP2 (Gerenciamento Ágil de Projetos de Software 1 e 2), o foco esteve na aplicação prática de técnicas de planejamento, acompanhamento e otimização de projetos com base em frameworks como Scrum e Kanban.

Em GAP1, o projeto final consistiu na elaboração de um plano de release para um sistema definido pelos alunos, cujo artefato pode ser visto na FIGURA 6, contemplando cálculo da velocidade da equipe, estimativa de histórias de usuário, divisão em sprints e definição de prazos para cada entrega. Esse exercício possibilitou estruturar o trabalho de forma realista e adaptável, em consonância com os princípios ágeis de planejamento contínuo e entregas incrementais.

Já em GAP2, a atividade prática envolveu uma simulação de ambiente ágil por meio de um jogo online de Kanban. Nessa dinâmica, os alunos gerenciaram o fluxo de trabalho em um quadro visual (FIGURA 7), buscando otimizar o tempo de ciclo e reduzir gargalos. A tela final da simulação (FIGURA 8) registrou o desempenho da equipe. A análise posterior do Diagrama de Fluxo Cumulativo (CFD), apresentado na FIGURA 9, permitiu avaliar métricas de desempenho, reforçando a importância da transparência e da gestão de fluxo de valor — fundamentos essenciais do método Kanban.

A relevância dessas disciplinas está na ênfase em organização, monitoramento e adaptação contínua, elementos que asseguram previsibilidade e embasam tomadas de decisão eficazes (Poppendieck; Poppendieck, 2003). Além disso, sua integração com outras áreas do curso é evidente: o plano de release produzido em GAP1 serviu como guia para atividades práticas de disciplinas como MAG, WEB, MOB e TEST, enquanto os conceitos de fluxo contínuo e gestão visual de GAP2 complementaram os trabalhos técnicos com uma perspectiva de eficiência operacional.

Assim, GAP1 e GAP2 consolidam uma visão estratégica do desenvolvimento ágil, preparando os alunos não apenas para executar, mas também para liderar projetos com foco em entregas frequentes, adaptabilidade e geração de valor ao cliente.

4.1 ARTEFATOS DO PROJETO GAP 1

FIGURA 6 - ARTEFATO DA DISCIPLINA DE GAP 1

Gerenciamento Ágil de Projetos I
 Profa. Dra. Raíscia Mentovani Fontana
 Template para o Plano de Release

Cálculo da Velocidade:

Horas disponíveis por dia:	5 horas	Tamanho da Sprint:	1 semana (5 dias úteis)
Horas disponíveis por Sprint:	25 horas (Calculado: 5 dias * 5 horas/dia)	Velocidade:	3 Pontos

Plano de Release:

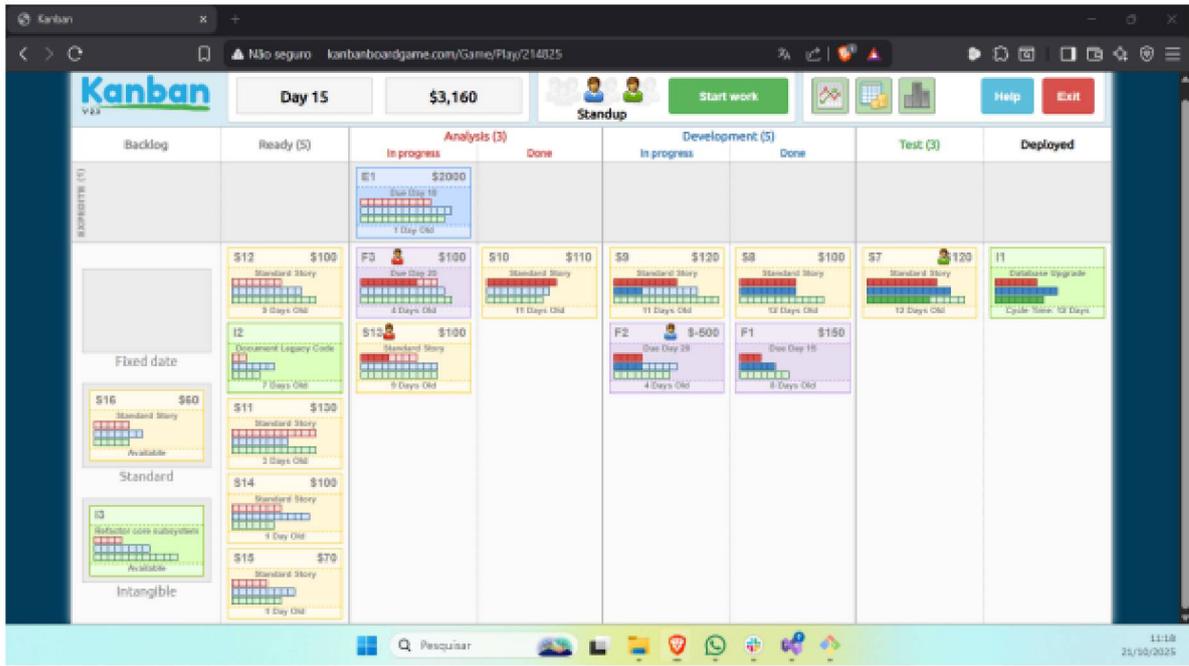
Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint N
Data Início: 03/06/2025	Data Início: 10/06/2025	Data Início: 17/06/2025	Data Início: 24/06/2024
Data Fim: 07/06/2025	Data Fim: 14/06/2025	Data Fim: 21/06/2025	Data Fim: 28/06/2024
<História de Usuário 1> SENDO um usuário, QUERO registrar uma refeição (café da manhã, almoço, jantar, lanche) com descrição e data/hora. PARA acompanhar minha alimentação. ESTIMATIVA (2 Pontos)	<História de Usuário 3> SENDO um usuário, QUERO visualizar as refeições que registrei em um dia específico. PARA revisar o que comi. ESTIMATIVA (1 Ponto)	<História de Usuário 5> SENDO um usuário, QUERO poder editar uma refeição registrada. PARA corrigir erros ou adicionar detalhes. ESTIMATIVA (1.5 Pontos)	<História de Usuário 7> SENDO um usuário, QUERO ver um resumo calórico estimado para o dia (baseado em valores que eu insiro manualmente por refeição). PARA ter uma noção do meu consumo. ESTIMATIVA (2 Pontos)
<História de Usuário 2> SENDO um usuário, QUERO poder excluir um registro de refeição. PARA remover entradas incorretas. ESTIMATIVA (0.5 Ponto)	<História de Usuário 4> SENDO um usuário, QUERO poder navegar entre os dias (dia anterior/próximo). PARA ver meu histórico facilmente. ESTIMATIVA (1 Ponto)	<História de Usuário 6> SENDO um usuário, QUERO poder marcar refeições como "saudáveis" ou "não saudáveis". PARA análise posterior. ESTIMATIVA (1 Ponto)	

FONTE: O autor (2025)

4.2 ARTEFATOS DO PROJETO GAP 2

FIGURA 7 - ARTEFATO 1 DA DISCIPLINA DE GAP 2

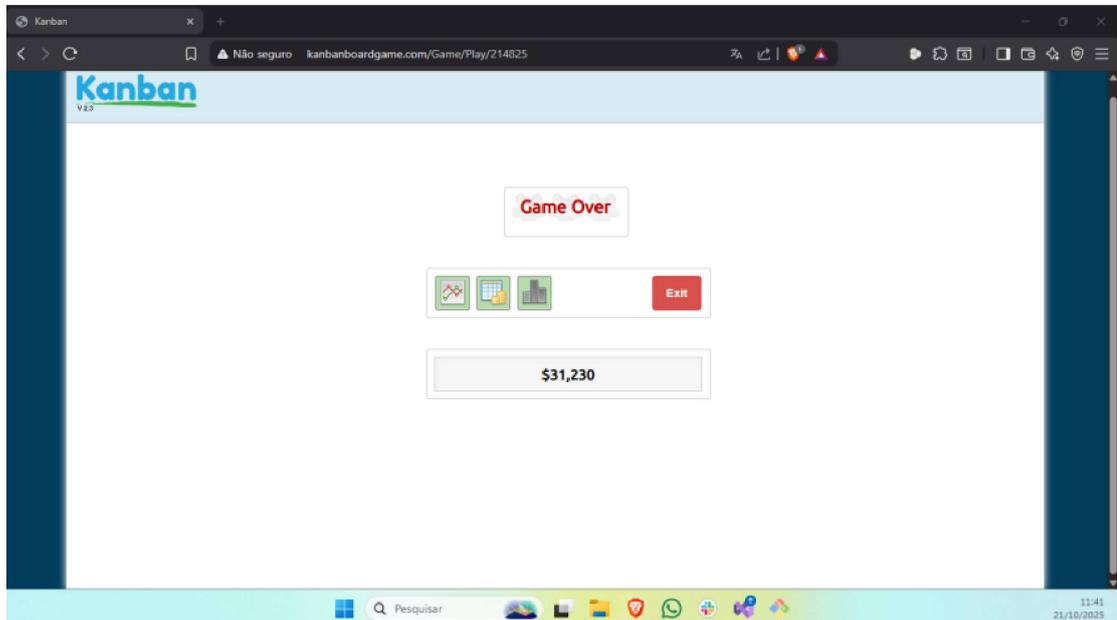
DIA 15



FONTE: O autor (2025)

FIGURA 8 - ARTEFATO 2 DA DISCIPLINA DE GAP 2

VALOR FINAL



FONTE: O autor (2025)

FIGURA 9 - ARTEFATO 3 DA DISCIPLINA DE GAP 2

CFD



FONTE: O autor (2025)

5 DISCIPLINA: INTRO – INTRODUÇÃO À PROGRAMAÇÃO

Na disciplina de INTRO (Introdução à Programação), o projeto final teve como proposta a implementação do backend de um sistema bancário simplificado, desenvolvido em Java. O sistema contemplava funcionalidades como cadastro de clientes, gerenciamento de contas-correntes e contas de investimento, permitindo a aplicação prática de conceitos fundamentais de orientação a objetos. Para apoiar o trabalho, foram fornecidos um Diagrama de Classes, um projeto base no NetBeans com testes automatizados em JUnit e um script DDL para criação do banco de dados no MySQL.

O principal requisito do projeto era garantir que, ao final da implementação, pelo menos 40 dos 42 testes unitários fossem executados com sucesso, resultado evidenciado na FIGURA 10. Essa exigência alinhava-se diretamente ao uso do TDD (Test-Driven Development), prática característica do desenvolvimento ágil que incentiva a escrita de testes antes da implementação. Essa abordagem favorece a criação de código modular, coeso e de fácil manutenção, além de orientar o design do sistema de forma incremental.

A relevância da disciplina no contexto ágil está em introduzir de maneira prática conceitos como automação de testes, design orientado a testes e controle de versões, que se tornaram essenciais em projetos modernos com ciclos curtos e entregas contínuas.

Dessa forma, INTRO não apenas consolidou as bases da programação, mas também preparou os alunos para adotar boas práticas de engenharia de software em um ambiente ágil, promovendo um desenvolvimento mais estruturado, confiável e alinhado às necessidades do cliente.

5.1 ARTEFATOS DO PROJETO

FIGURA 10 - ARTEFATO DA DISCIPLINA DE INTRO

```

Tests passed: 95,24 %
40 tests passed, 1 test failed, 1 test caused an error. (5,973 s)
  - TesteBancoRrw Failed
    - t01verificaEstruturaClassePessoa passed (0,007 s)
    - t02verificaEstruturaClasseContaCorrente passed (0,002 s)
    - t03verificaEstruturaClasseContaInvestimento passed (0,001 s)
    - t04criarContaCorrenteSaldoZero passed (0,003 s)
    - t05criarContaCorrenteComSaldo2000 passed (0,0 s)
    - t06manipulaContaCorrenteDepositar50 passed (0,0 s)
    - t07manipulaContaCorrenteDepositar100Deposita20Saca60 passed (0,0 s)
    - t08manipulaContaCorrenteDepositar100Deposita20Saca1000 passed (0,0 s)
    - t09manipulaContaCorrenteLimiteDepositar100Deposita20Saca1300 passed (0,017 s)
    - t10manipulaContaCorrenteDepositarNegativo50 passed (0,006 s)
    - t11manipulaContaCorrenteSaqueNegativo100 passed (0,001 s)
    - t12manipulaContaCorrenteDeposita100AplicaJuros passed (0,0 s)
    - t13manipulaContaCorrenteSaca100AplicaJuros passed (0,0 s)
    - t14trocaContaCorrenteDeCliente passed (0,001 s)
    - t15verificaSaldoZeroParaTrocarContaCorrente passed (0,001 s)
    - t16criarContaInvestimento passed (0,001 s)
    - t17manipularContaInvestimentoDepositoinicialMenorQueMontanteMinimo passed (0,001 s)
    - t18manipularContaInvestimentoDepositarMinimo passed (0,001 s)
    - t19manipularContaInvestimentoDepositar1000Sacar500 passed (0,001 s)
    - t20manipularContaInvestimentoDepositar1000Sacar1100 passed (0,001 s)
    - t21manipularContaInvestimentoLimiteDepositarMenosQueMinimo passed (0,0 s)
    - t22manipularContaInvestimentoAplica1000AplicaJuros passed (0,0 s)
    - t23crudClienteAdd passed (1,077 s)
    - t24crudClienteGetByld passed (0,214 s)
    - t25crudClienteUpdate passed (0,216 s)
    - t26crudClienteDelete passed (0,216 s)
    - t27crudContaCorrenteAdd passed (0,168 s)
    - t28crudContaCorrenteGetByld passed (0,181 s)
    - t29crudContaCorrenteUpdate passed (0,204 s)
    - t30crudContaCorrenteDelete passed (0,207 s)
    - t31crudContaInvestimentoAdd passed (0,182 s)
    - t32crudContaInvestimentoGetByld passed (0,231 s)
    - t33crudContaInvestimentoUpdate passed (0,219 s)
    - t34crudContaInvestimentoDelete passed (0,23 s)
    - t35verificaSeAContaCorrenteFoiSetadaNoCliente passed (0,185 s)
    - t36manipulaSaldoDaContaCorrenteEGravaBdERecuperaSaldo caused an ERROR: Não pode modificar a conta corrente, pois saldo da original não está zerado. Para fazer isso primeiro zere o saldo da conta
    - t37cria4ContasSalvaNoBdERecuperaTodas passed (0,299 s)
    - t38testaContaCorrenteDeleteAll passed (0,409 s)
    - t39verificaSeAContaInvestimentoFoiSetadaNoCliente Failed: expected:<2> but was:<4>
    - t40testaContaInvestimentoDeleteAll passed (0,275 s)
    - t41testaSeODeleteAllDaContaCorrenteNaoEliminaTodasAsContasInvestimentosDoCliente passed (0,444 s)
  
```

FONTE: O autor (2025)

6 DISCIPLINA: BD – BANCO DE DADOS

Na disciplina de BD (Banco de Dados), o projeto final foi desenvolvido em duas etapas, com o propósito de consolidar tanto a teoria quanto a prática sobre modelagem e manipulação de bancos de dados relacionais. Na primeira etapa, trabalhou-se com um sistema de gerenciamento de biblioteca, no qual foi necessário construir o Modelo Entidade-Relacionamento (MER) e, em seguida, o Diagrama Entidade-Relacionamento (DER), conforme demonstrado na FIGURA 11. Essa atividade exigiu a definição de entidades, atributos, chaves primárias e estrangeiras, além dos relacionamentos entre tabelas, traduzindo requisitos do sistema em uma estrutura de dados consistente.

Na segunda etapa, o desafio passou a ser o controle de veículos de uma concessionária, abordando desde a elaboração do modelo lógico (FIGURA 12) até a implementação de scripts SQL para criação do esquema no banco de dados (FIGURA 14). Foram realizadas também a inserção de dados e a construção de consultas (FIGURA 13) que exploravam relacionamentos do tipo 1:1, 1:N e N:N, reforçando a importância de conceitos como normalização e integridade referencial para garantir consistência e escalabilidade.

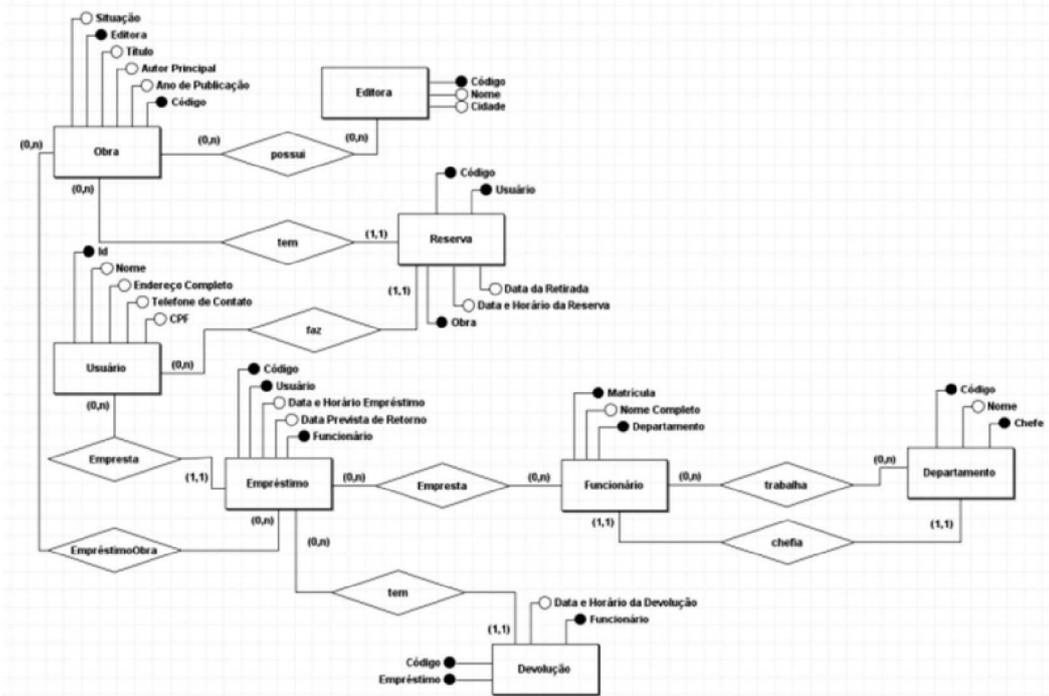
O conteúdo da disciplina mostrou-se essencial no contexto do desenvolvimento ágil, já que um banco de dados bem planejado reduz retrabalho e facilita a evolução do sistema em cenários de mudanças frequentes. A integração desse conhecimento com áreas de modelagem (MAG1/MAG2), programação (INTRO, WEB, MOB) e testes fortaleceu a compreensão sobre persistência de dados como um pilar central da engenharia de software.

Assim, a experiência em BD não apenas consolidou boas práticas de modelagem e consulta, mas também preparou os alunos para lidar com a base técnica necessária a soluções ágeis, escaláveis e sustentáveis.

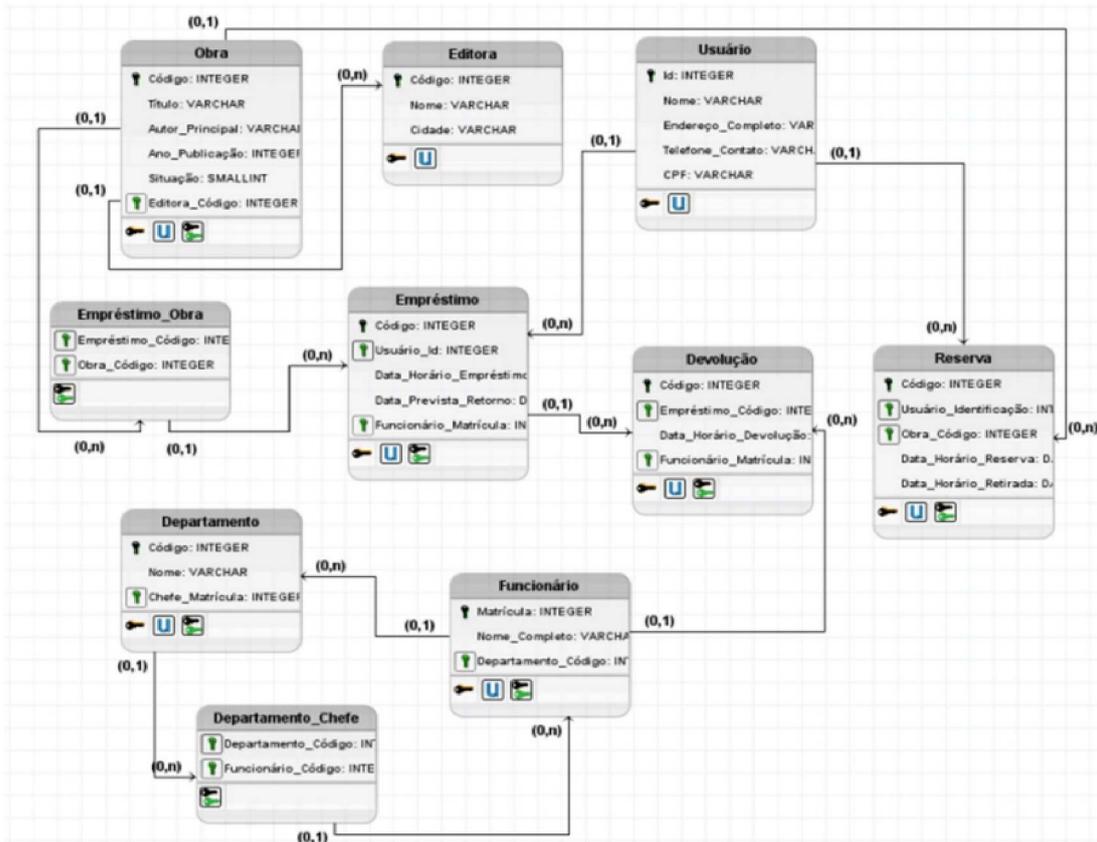
6.1 ARTEFATOS DO PROJETO

FIGURA 11 - ARTEFATO 1 DA DISCIPLINA DE BD

Modelo Entidade-Relacionamento Conceitual - Figura 1



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



FONTE: O autor (2025)

FIGURA 12 - ARTEFATO 2 DA DISCIPLINA DE BD

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.

FONTE: O autor (2025)

FIGURA 13 - ARTEFATO 3 DA DISCIPLINA DE BD

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

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

FONTE: O autor (2025)

FIGURA 14 - ARTEFATO 4 DA DISCIPLINA DE BD

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);
```

FONTE: O autor (2025)

7 DISCIPLINA: AAP – ASPECTOS ÁGEIS DE PROGRAMAÇÃO

Na disciplina de AAP (Aspectos Ágeis de Programação), o projeto final teve como foco a refatoração do algoritmo Bubble Sort à luz dos princípios de Clean Code, priorizando clareza, simplicidade e facilidade de manutenção. A proposta exigia pelo menos três melhorias relevantes, como a escolha de nomes descritivos para variáveis e métodos, a eliminação de comentários redundantes e trechos desnecessários, além da aplicação do princípio da responsabilidade única, segundo o qual cada método deve cumprir uma função bem definida. O código refatorado (FIGURA 15) e as explicações das melhorias (FIGURA 16 e FIGURA 17) ilustram esse processo.

Os conceitos de Clean Code assumem papel central no desenvolvimento ágil, já que equipes precisam lidar com código em constante evolução. Um código limpo e organizado favorece revisões mais rápidas, integração contínua e entregas frequentes sem comprometer a qualidade. Dessa forma, a disciplina destacou que boas práticas de codificação reduzem o tempo de compreensão e alteração, mitigam erros e criam condições para a evolução incremental do sistema.

A relevância de AAP também se reflete nas disciplinas práticas do curso — INTRO, WEB, MOB e TEST — onde a qualidade do código impacta diretamente a eficiência da implementação e a confiabilidade dos testes. Além disso, os princípios abordados dialogam com valores do Manifesto Ágil, como simplicidade e busca pela excelência técnica, reforçando a importância de um ciclo de desenvolvimento iterativo e sustentável.

Assim, a experiência em AAP contribuiu para o amadurecimento técnico dos alunos, evidenciando que a aplicação consistente de Clean Code fortalece a fluidez do processo ágil e assegura soluções mais confiáveis, escaláveis e duradouras.

7.1 ARTEFATOS DO PROJETO

FIGURA 15 - ARTEFATO 1 DA DISCIPLINA DE AAP

```
J BubbleSort - Refatorado - Reenvio.java X
C: > Users > julio > Downloads > J BubbleSort - Refatorado - Reenvio.java
1  import java.io.*;
2
3  public class BubbleSort {
4
5      // Driver program
6      static void main(String[] args) {
7          int[] arr = { 64, 34, 25, 12, 22, 11, 90 };
8          bubbleSort(arr);
9          System.out.println("Sorted Array: ");
10         printArray(arr);
11     }
12
13     // Swaps two elements in an array
14     static void swap(int[] arr, int i, int j) {
15         int temp = arr[i];
16         arr[i] = arr[j];
17         arr[j] = temp;
18     }
19
20     // An optimized version of Bubble Sort
21     static void bubbleSort(int[] arr) {
22         for (int i = 0; i < arr.length - 1; i++) {
23             if (!bubble(arr, arr.length - i - 1)) break;
24         }
25     }
26
27     // Performs one pass of the Bubble Sort algorithm
28     static boolean bubble(int[] arr, int end) {
29         boolean swapped = false;
30         for (int j = 0; j < end; j++) {
31             if (arr[j] > arr[j + 1]) {
32                 swap(arr, j, j + 1);
33                 swapped = true;
34             }
35         }
36         return swapped;
37     }
38
39     // Function to print an array
40     static void printArray(int[] arr) {
41         for (int value : arr) {
42             System.out.print(value + " ");
43         }
44         System.out.println();
45     }
46
47 }
```

FONTE: O autor (2025)

FIGURA 16 - ARTEFATO 2 DA DISCIPLINA DE AAP

Explicação das Alterações:

- 1. Nomeação Consistente em Inglês:**
 - **Consistência:** Utilizar uma única língua (inglês, neste caso) para nomes de métodos e variáveis melhora a legibilidade e a manutenção do código. Isso evita confusão e garante que todos os desenvolvedores, independentemente de sua língua nativa, possam entender o código facilmente.
 - **Alterações Específicas:**
 - **trocado para swapped:** A variável booleana `trocado` foi renomeada para `swapped`, que é o termo apropriado em inglês para indicar se houve troca de elementos.
- 2. Remoção do Parâmetro `n` em `bubbleSort` e `printArray`:**
 - **Original:** Ambos os métodos `bubbleSort` e `printArray` recebiam o parâmetro `n` (o comprimento do array), o que não era necessário, pois o comprimento do array pode ser obtido diretamente através da propriedade `length`.
 - **Refatorado:** O parâmetro `n` foi removido de ambos os métodos. O método `bubbleSort` agora usa `arr.length` diretamente, e o método `printArray` também opera diretamente com o array, sem necessidade de passar o comprimento como parâmetro. Essa mudança simplifica a assinatura dos métodos e melhora a clareza do código.
- 3. Extração do Código de Troca para um Método Separado:**
 - **Original:** O código de troca estava dentro do método `bubbleSort`, tornando o método mais complexo e menos modular.
 - **Refatorado:** A lógica de troca foi extraída para um método `swap` separado. Isso melhora a legibilidade e segue o princípio de responsabilidade única, facilitando a manutenção e o entendimento do código.
- 4. Não Declaração Inicial das Variáveis `i` e `j`:**
 - **Original:** As variáveis `i` e `j` eram declaradas fora dos loops onde eram usadas.
 - **Refatorado:** As variáveis `i` e `j` agora são declaradas diretamente dentro dos loops onde são usadas. Isso reduz o escopo das variáveis e melhora a clareza do código.
- 5. Troca da Validação Explícita `swapped == false`:**
 - **Original:** A verificação para encerrar o loop externo usava a comparação explícita `if (swapped == false)`.
 - **Refatorado:** A verificação foi simplificada para `if (!swapped)`, tornando o código mais conciso e fácil de ler.
- 6. Uso de Enhanced For Loop em `printArray`:**
 - **Original:** Um loop `for` tradicional era usado para iterar sobre os elementos do array.
 - **Refatorado:** Um loop `enhanced for` (ou `for-each`) foi utilizado para melhorar a legibilidade e evitar a manipulação direta do índice. Isso é mais limpo e menos propenso a erros.

FONTE: O autor (2025)

FIGURA 17 - ARTEFATO 2 DA DISCIPLINA DE AAP

Mudanças realizadas segunda entrega:

- **Método `swap`:** Mantido acima do `bubbleSort` para seguir a ordem de chamada.
- **Método `bubbleSort`:** Agora, ele delega a lógica de uma passagem do Bubble Sort para um método separado chamado `bubble`, o que reduz a complexidade e melhora a clareza.
- **Método `bubble`:** Isola a lógica de uma única passagem do Bubble Sort, retornando um booleano para indicar se houve uma troca. Isso simplifica o loop principal.

FONTE: O autor (2025)

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

Na disciplina de Desenvolvimento Web 1, o projeto teve como foco a implementação de dois CRUDs, referentes às entidades Aluno e Curso, utilizando Angular no frontend e persistência de dados via Local Storage, sem necessidade de backend. Essa escolha permitiu aos alunos aprofundar-se nos fundamentos do desenvolvimento de interfaces modernas e em boas práticas de usabilidade, ao mesmo tempo em que reforçava princípios ágeis de entregas rápidas e foco no usuário. A estrutura prática do projeto, associada aos questionários realizados durante a disciplina, consolidou uma base sólida para a criação de aplicações web de forma ágil e organizada.

Em Desenvolvimento Web 2, o aprendizado foi ampliado com a introdução de um backend em Spring Boot, integrado a um banco de dados PostgreSQL. Embora a atividade prática não tenha sido concluída integralmente, os conteúdos teóricos evidenciaram a importância de arquiteturas bem estruturadas e da integração entre frontend e backend, aspectos essenciais em equipes que trabalham de forma colaborativa em ambientes ágeis.

Essas disciplinas mantêm forte conexão com outras áreas do curso, especialmente Banco de Dados, ao demandarem planejamento, modelagem e entendimento das camadas de uma aplicação real. Esse conjunto de conhecimentos é decisivo no contexto do desenvolvimento ágil de software, pois garante entregas iterativas com valor agregado e maior capacidade de adaptação a mudanças durante o ciclo do projeto. Ainda que não tenha havido entrega formal de projetos, o aprendizado construído em Desenvolvimento Web 1 e 2 foi fundamental para integrar teoria e prática no percurso da especialização.

9 DISCIPLINA: UX – UX NO DESENVOLVIMENTO ÁGIL DE SOFTWARE

Na disciplina de UX no Desenvolvimento Ágil de Software, o projeto teve como objetivo a criação de protótipos de interface para uma aplicação de gestão de ponto de funcionários em uma indústria automobilística. O exercício buscou aplicar os princípios de UX (User Experience) dentro de um contexto ágil, priorizando as necessidades do usuário e a entrega contínua de valor. Para isso, foram construídas personas, jornadas e fluxos de uso (FIGURA 18), desenvolvidas telas no Figma, – como as de login (FIGURA 19), home (FIGURA 20), exercícios (FIGURA 21) e perfil (FIGURA 22) –, justificadas as escolhas visuais (FIGURA 23) e, por fim, conduzida uma entrevista com um potencial usuário para coleta de feedbacks (FIGURA 24).

A disciplina evidenciou a importância do design centrado no usuário desde as etapas iniciais do desenvolvimento. A prototipação rápida e os ciclos de validação contínua permitiram aprimorar a solução de forma iterativa, em alinhamento direto com os princípios ágeis de adaptação e foco no valor entregue.

O aprendizado também se conectou de maneira prática com outras disciplinas: em Desenvolvimento Web e Mobile, os protótipos em Figma serviram como referência clara para implementação; já em áreas de engenharia de software e gerenciamento ágil de projetos, destacou-se a relevância da priorização de funcionalidades e da adaptação a mudanças.

Dessa forma, a disciplina de UX não apenas fortaleceu a visão de produto dos alunos, mas também consolidou a compreensão de que colocar o usuário no centro do processo é um dos pilares para o sucesso no desenvolvimento ágil de software.

9.1 ARTEFATOS DO PROJETO

FIGURA 18 - ARTEFATO 1 DA DISCIPLINA DE UX

Explicação do Produto - App/Site

Nome do Produto: RelaxTime

Função Principal:

O **RelaxTime** é um aplicativo de relaxamento e mindfulness que ajuda os usuários a melhorar a qualidade do sono e reduzir o estresse. Ele oferece trilhas de meditação, sons relaxantes e exercícios de respiração guiados. A proposta é simples: criar uma experiência de bem-estar diária por meio de sessões rápidas e intuitivas.

Justificativa para o Desenvolvimento:

Com o aumento do estresse cotidiano e dificuldades para dormir, muitas pessoas procuram ferramentas acessíveis que possam ajudar no bem-estar mental. Estudos mostram que meditações e sons relaxantes podem ajudar na saúde mental, mas a maioria dos apps disponíveis exige assinatura ou possui uma interface complexa. O **RelaxTime** foi criado para ser acessível, intuitivo e gratuito, focado em usuários que desejam uma solução prática para melhorar sua qualidade de vida.

Funções Principais:

- **Trilhas de Relaxamento:** Sessões de meditação, som de chuva e sons da natureza.
- **Exercícios de Respiração:** Orientações simples para ajudar na respiração e alívio da ansiedade.
- **Monitoramento do Humor:** Os usuários podem registrar seus sentimentos para acompanhar o progresso emocional.
- **Relatórios Semanais:** Comportamentos de humor e tempo de uso são apresentados para autoconsciência e progresso.
- **Configurações de Notificação:** Alertas diários para lembrar os usuários de reservar um tempo para si.

FONTE: O autor (2025)

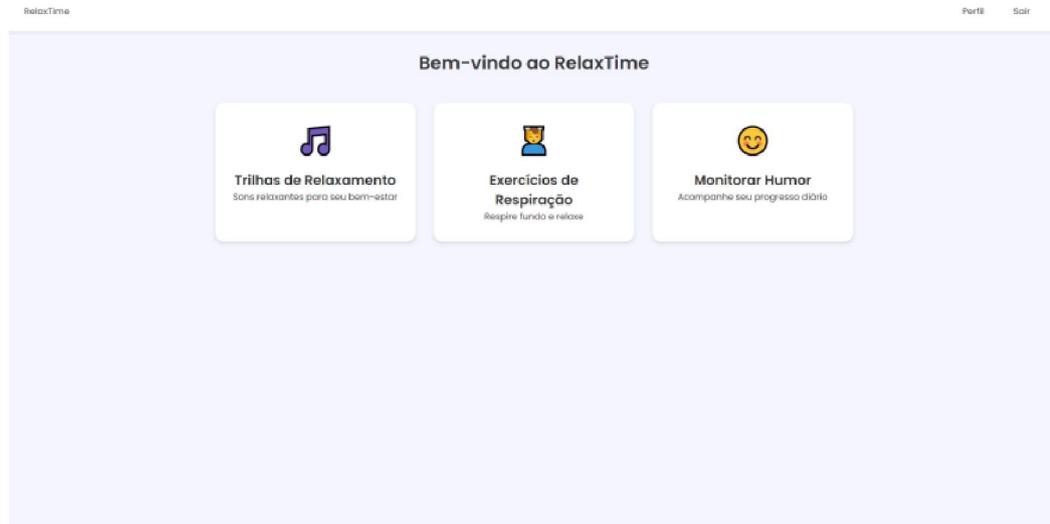
FIGURA 19 - ARTEFATO 2 DA DISCIPLINA DE UX

Tela de Boas-Vindas / Login

The image shows a login screen for 'RelaxTime'. The screen has a light blue background. In the center, there is a white rounded rectangle containing the following elements: the title 'RelaxTime' in bold black text; an input field for 'E-mail'; an input field for 'Senha'; a blue button labeled 'Entrar'; a white button labeled 'Entrar com Google'; and two links at the bottom: 'Criar conta' and 'Esqueci minha senha'.

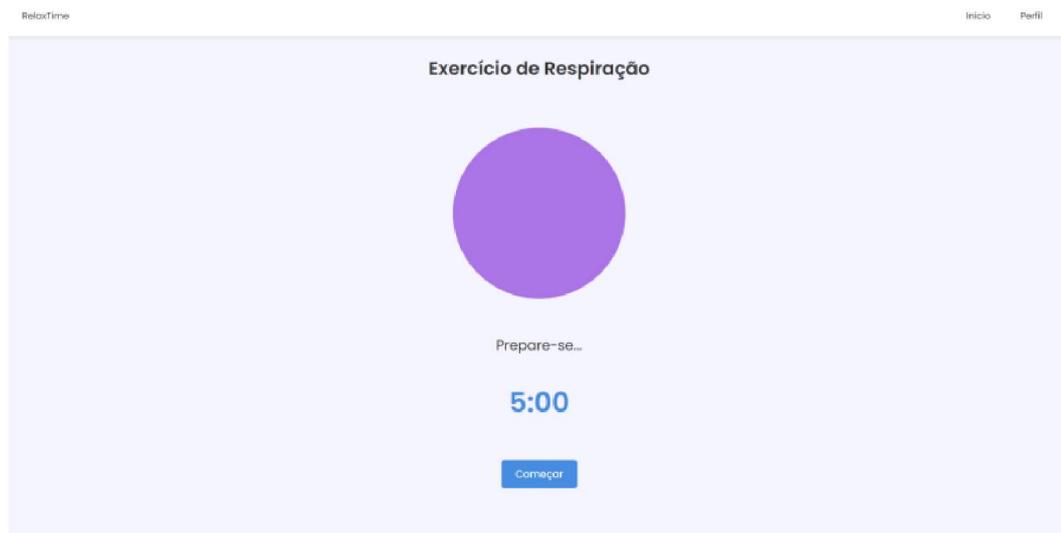
FONTE: O autor (2025)

FIGURA 20 - ARTEFATO 3 DA DISCIPLINA DE UX

Tela Inicial (Home)

FONTE: O autor (2025)

FIGURA 21 - ARTEFATO 4 DA DISCIPLINA DE UX

Tela de Exercícios de Respiração

FONTE: O autor (2025)

FIGURA 22 - ARTEFATO 5 DA DISCIPLINA DE UX

Tela de Monitoramento do Humor

RelaxTime Início Sair

Meu Perfil



[Alterar Foto](#)

Nome

E-mail

Nova Senha

[Salvar Alterações](#)

Minhas Estatísticas

Minutos Meditados	Exercícios Realizados	Dias Consecutivos
120	15	7

FONTE: O autor (2025)

FIGURA 23 - ARTEFATO 6 DA DISCIPLINA DE UX

Explicações das Escolhas de Design

Paleta de Cores:

- **Tons Calmos (Azul Claro e Verde):** Cores que transmitem tranquilidade e relaxamento, ideais para um app de bem-estar.
- **Acentos em Lilás:** Para pequenos detalhes e botões, o lilás adiciona um toque suave e moderno, sem quebrar o clima de relaxamento.

Tipografia:

- **Fonte Principal (Sans-Serif, ex: Poppins ou Montserrat):** Limpa e moderna, ajuda na leitura e cria um visual amigável e acessível.
- **Fonte Secundária (Serif para títulos):** Se usada, uma serif leve para contrastar suavemente com a fonte principal.

Layout:

- **Navegação Simples:** Menu principal com ícones e poucas opções para não sobrecarregar visualmente o usuário.
- **Elementos Visuais:** Ícones e botões grandes para facilitar o uso em telas menores e atender a usuários que buscam uma experiência simplificada.

Animações:

- **Respiração Guiada:** Animações leves (círculo que expande e contrai) que guiam o usuário na respiração, mantendo um ritmo calmo e constante.

FONTE: O autor (2025)

FIGURA 24 - ARTEFATO 7 DA DISCIPLINA DE UX

Feedback do Usuário – (Protótipos)

Usuário: "O fundo azul claro na tela de login ficou muito bom, já passa uma sensação de calma só de olhar o protótipo. Acho que a ideia de relaxamento já começa aí, sabe? Mas, olhando assim, será que o botão de 'Criar Conta' poderia se destacar um pouco mais? Pra quem está entrando pela primeira vez, ele quase passa despercebido."

Usuário: "Na tela inicial, a distribuição dos ícones tá bem organizada, dá pra entender direitinho onde cada coisa está. As cores em tons de azul ajudam a manter um visual leve. Só... me parece que os ícones poderiam ter um pouco mais de contraste com o fundo. Acho que em algumas condições de luz, isso ajudaria."

Usuário: "Essas trilhas de relaxamento parecem bem atraentes. Os ícones que representam cada som estão bem intuitivos; olhando o protótipo, já dá pra entender o que é cada som. O botão de play/pause tá claro também. Só fiquei pensando... e se a barra de progresso fosse de uma cor diferente? Talvez ajudasse a ver o andamento melhor."

FONTE: O autor (2025)

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

Nas disciplinas de MOB 1 e MOB 2 (Desenvolvimento Mobile 1 e 2), os projetos foram direcionados ao fortalecimento das habilidades práticas no desenvolvimento de aplicativos Android, sempre em sintonia com os princípios do desenvolvimento ágil.

Em MOB 1, a atividade principal consistiu na criação de um aplicativo de controle de finanças pessoais, permitindo ao usuário registrar receitas e despesas para acompanhar melhor sua vida financeira. Essa proposta introduziu os alunos aos fundamentos do ecossistema Android, como manipulação de interfaces, persistência de dados e uso de componentes visuais interativos.

Na sequência, em MOB 2, o desafio foi ampliado com a construção de um aplicativo que consumia uma API externa, utilizando como temática o universo de Harry Potter. O projeto enfatizou o uso de web services REST, a manipulação de dados em JSON e o domínio de operações assíncronas, fundamentais para garantir fluidez e responsividade em aplicativos móveis. Ainda que os projetos fossem opcionais e não tenham sido implementados, as atividades teóricas e questionários ao longo da disciplina foram decisivos para consolidar conceitos como ciclo de vida das aplicações, arquitetura mobile e boas práticas de desenvolvimento.

O valor dessas disciplinas para o contexto ágil se manifesta na ênfase em entregas incrementais, na flexibilidade diante de mudanças e no destaque dado ao feedback rápido do usuário – aspectos centrais no desenvolvimento mobile. A integração com outras áreas, como Introdução à Programação e UX, também foi evidente, já que conhecimentos prévios em lógica, modelagem de dados e prototipação serviram como base para enfrentar os desafios da programação mobile.

Assim, MOB 1 e MOB 2 reforçaram a necessidade de uma abordagem iterativa e multidisciplinar na criação de soluções digitais, destacando a importância de construir aplicativos modernos, responsivos e centrados no usuário.

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

Na 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, hoje indispensável para o desenvolvimento ágil. O desafio envolveu a criação e configuração de um ambiente baseado em contêineres com Docker, aliado à aplicação de práticas de versionamento e integração contínua utilizando Git e GitLab.

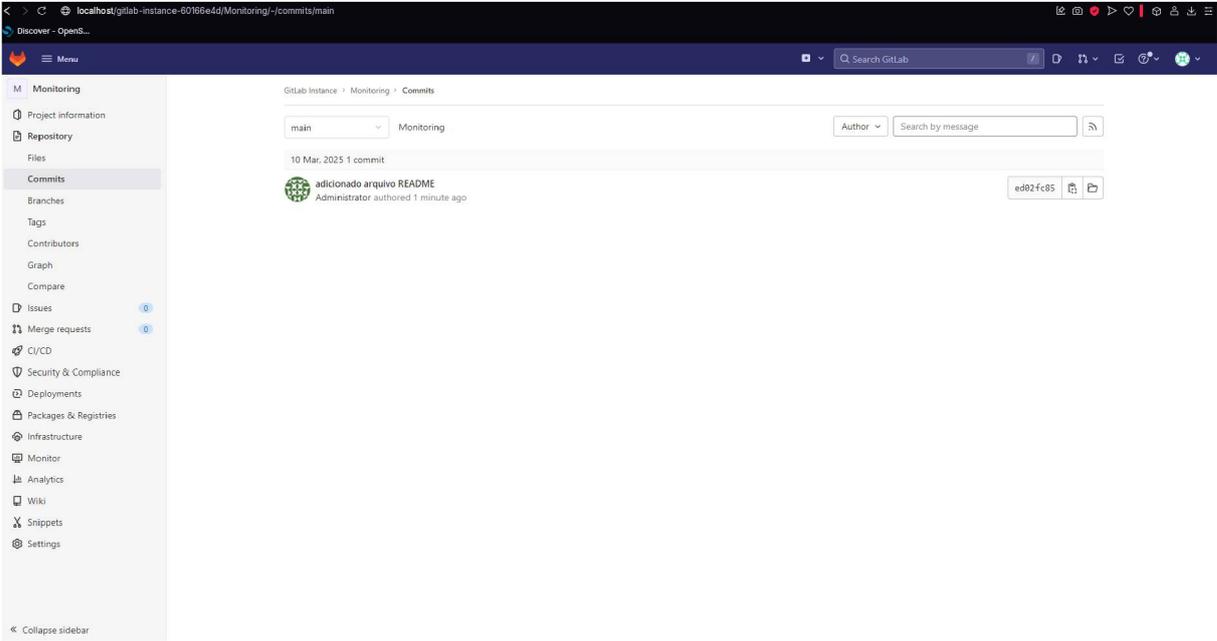
A atividade iniciou-se com a execução de um contêiner a partir da imagem *dfwandarti/gitlab_jenkins:3*, como mostra o comando na FIGURA 25, configurado com a matrícula do aluno como identificador e exposição de portas essenciais (22, 80, 443 e 9091). Em seguida, foi necessária a navegação interna no contêiner para recuperar credenciais e realizar a autenticação no GitLab, onde comandos Git como *commit* e *push* simularam cenários reais de colaboração entre equipes, conforme detalhado na FIGURA 26.

Um ponto de destaque foi a flexibilidade em poder utilizar tanto linha de comando quanto interfaces gráficas para operar o Docker, ampliando a compreensão sobre como diferentes abordagens podem automatizar processos de desenvolvimento e implantação. Essa experiência evidenciou como automação, versionamento e integração contínua otimizam o ciclo de entrega e reduzem riscos operacionais.

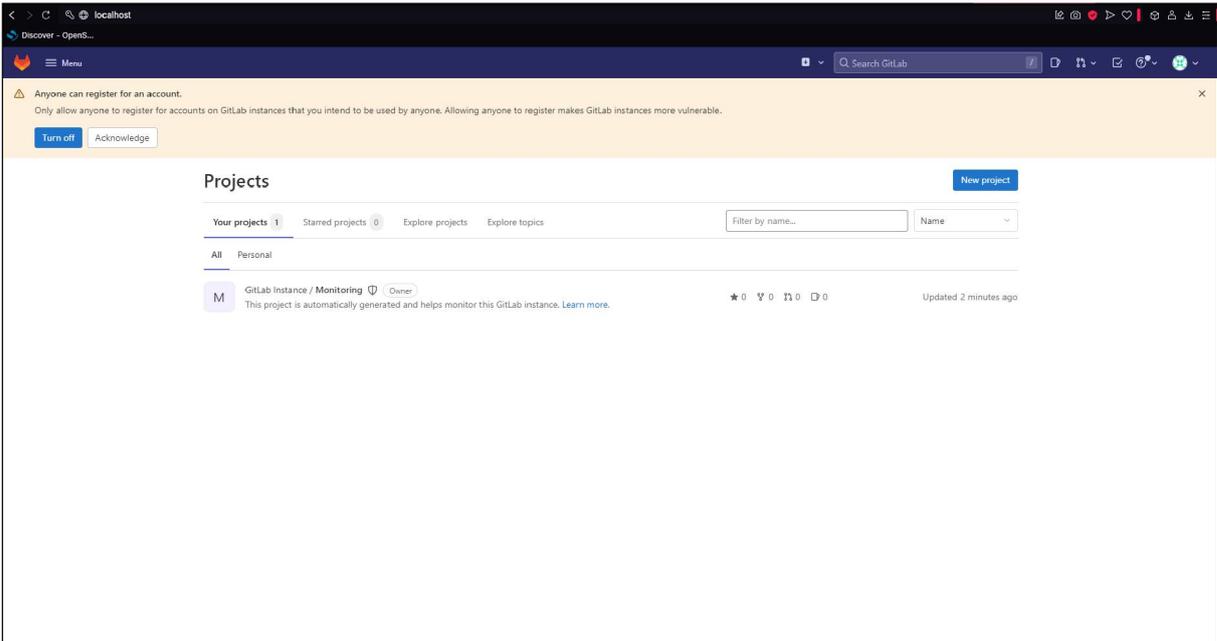
O aprendizado da disciplina mostrou-se complementar a áreas como Desenvolvimento Web e Mobile, em que práticas de deploy contínuo e controle de versões são frequentemente aplicadas. Mais do que ferramentas, a disciplina destacou a importância da colaboração entre equipes de desenvolvimento e operações, pilar fundamental para acelerar entregas, garantir qualidade e responder com agilidade às mudanças do mercado e às necessidades dos usuários.

11.1 ARTEFATOS DO PROJETO

FIGURA 25 - ARTEFATO 1 DA DISCIPLINA DE INFRA



The screenshot shows the GitLab Monitoring interface. The left sidebar contains navigation options: Monitoring, Project information, Repository, Files, Commits, Branches, Tags, Contributors, Graph, Compare, Issues (0), Merge requests (0), CI/CD, Security & Compliance, Deployments, Packages & Registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings. The main content area displays the commit history for the 'main' branch, showing a single commit: 'adicionado arquivo README' by 'Administrator' on '10 Mar. 2025'.

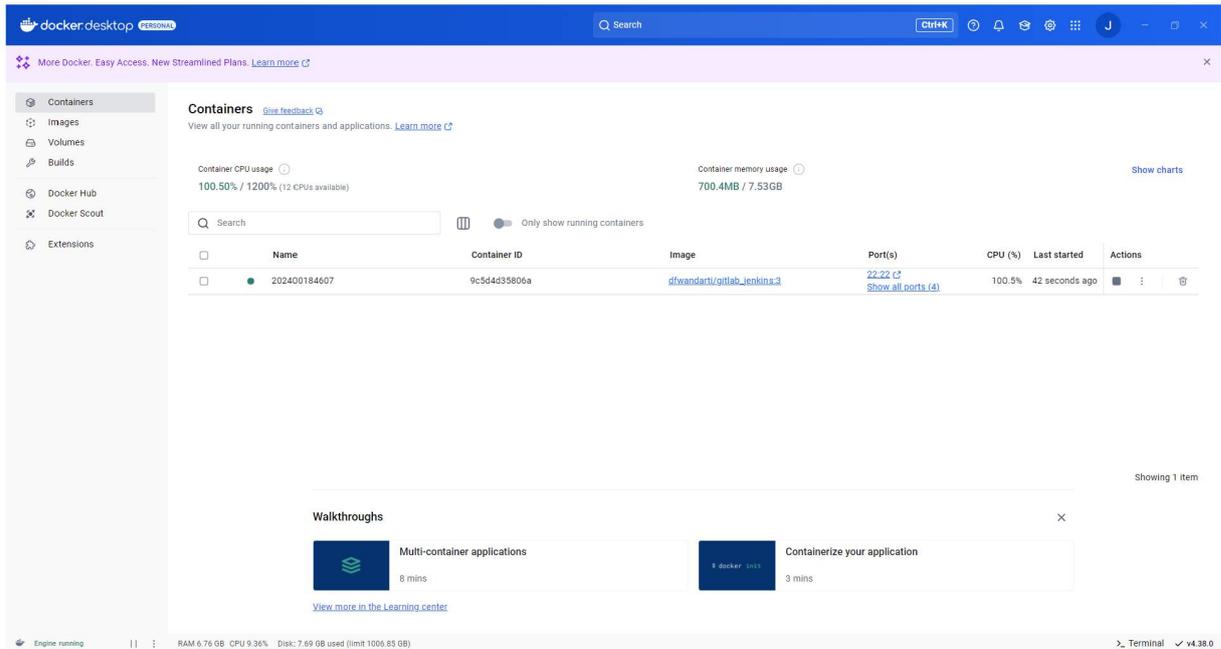


The screenshot shows the GitLab Projects page. A warning banner at the top states: 'Anyone can register for an account. Only allow anyone to register for accounts on GitLab instances that you intend to be used by anyone. Allowing anyone to register makes GitLab instances more vulnerable.' Below this, the 'Projects' section is displayed, showing a list of projects. The 'GitLab Instance / Monitoring' project is highlighted, with a description: 'This project is automatically generated and helps monitor this GitLab instance. Learn more.' and statistics: 0 stars, 0 forks, 0 issues, 0 pull requests.

```
C:\Users\julio>docker run -d -p 22:22 -p 80:80 -p 443:443 -p 9091:9091 --name 202400184607 dfwandarti/gitlab_jenkins:39c5d4d35806ae01e16cca938382439b51a653e9ab77fc26463c85569738966a2
C:\Users\julio>
```

FONTE: O autor (2025)

FIGURA 26 - ARTEFATO 2 DA DISCIPLINA DE INFRA



```

root@9c5d4d35806a:/Monitoring
root@9c5d4d35806a:/# git config --global user.name "Administrator"
root@9c5d4d35806a:/# git config --global user.email "teste@teste.com"
root@9c5d4d35806a:/# git clone http://9c5d4d35806a/gitlab-instance-60166e4d/Monitoring.git
Cloning into 'Monitoring'...
Username for 'http://9c5d4d35806a': root
Password for 'http://root@9c5d4d35806a':
warning: You appear to have cloned an empty repository.
root@9c5d4d35806a:/# cd Monitoring/
root@9c5d4d35806a:/Monitoring# git switch -c main
Switched to a new branch 'main'
root@9c5d4d35806a:/Monitoring# touch README.md
root@9c5d4d35806a:/Monitoring# git add README.md
root@9c5d4d35806a:/Monitoring# git commit -m "adicionado arquivo README"
[main (root-commit) ed02fc8] adicionado arquivo README
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
root@9c5d4d35806a:/Monitoring# git push -u origin main
Username for 'http://9c5d4d35806a': root
Password for 'http://root@9c5d4d35806a':
remote: HTTP Basic: Access denied
fatal: Authentication failed for 'http://9c5d4d35806a/gitlab-instance-60166e4d/Monitoring.git/'
root@9c5d4d35806a:/Monitoring# git push -u origin main
Username for 'http://9c5d4d35806a': root
Password for 'http://root@9c5d4d35806a':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://9c5d4d35806a/gitlab-instance-60166e4d/Monitoring.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
root@9c5d4d35806a:/Monitoring#

```

FONTE: O autor (2025)

12 DISCIPLINA: TEST – TESTES AUTOMATIZADOS

Na disciplina de Testes Automatizados, o projeto final consistiu na criação de um teste funcional automatizado utilizando TypeScript em conjunto com o framework Selenium. O exercício envolveu o navegador Google Chrome, acessando o site <https://pt.anotepad.com>, preenchendo automaticamente o campo de título com a mensagem “Entrega trabalho TEST DAS 2024” e registrando no corpo da nota o nome e a matrícula do aluno. Essa atividade simulou uma interação real do usuário, demonstrando domínio de técnicas de teste ponta a ponta (end-to-end), através do script apresentado na FIGURA 27.

A disciplina evidenciou a importância da automação como pilar do desenvolvimento ágil, uma vez que ambientes ágeis dependem de entregas contínuas e incrementais. A automação de testes garante rapidez, confiabilidade e rastreabilidade, permitindo identificar regressões e validar funcionalidades de forma eficiente, promovendo entregas de valor consistentes ao cliente.

Além disso, o projeto mostrou como os testes automatizados se conectam a outras áreas do curso, como DevOps, integrando-se em pipelines de integração contínua, e Desenvolvimento Web, validando tanto a interface quanto as funcionalidades implementadas. A disciplina também reforçou a importância de alinhar testes aos critérios de aceitação definidos nos requisitos, assegurando a qualidade e a robustez do software.

Assim, Testes Automatizados consolidou práticas essenciais para manter a excelência técnica e a confiabilidade dos sistemas, complementando a formação dos alunos com foco em agilidade, qualidade e entregas contínuas.

12.1 ARTEFATOS DO PROJETO

FIGURA 27 - ARTEFATO DA DISCIPLINA DE TEST

```

main.py
C: > Users > julio > Downloads > main.py
1  import time
2  from selenium import webdriver
3  from selenium.webdriver.common.by import By
4  from selenium.webdriver.support.ui import WebDriverWait
5  from selenium.webdriver.support import expected_conditions as EC
6
7  options = webdriver.ChromeOptions()
8  options.add_argument("--start-maximized") # Inicia o navegador maximizado
9  driver = webdriver.Chrome(options=options)
10 driver.get("https://pt.anotepad.com")
11
12 titulo_nota = "Entrega trabalho TEST DAS 2024"
13 participantes = [
14     {"nome": "Julio Cesar Becker Vasconcelos", "matricula": "202400184607"}
15 ]
16
17 wait = WebDriverWait(driver, 15) # ajuda para esperar elementos carregarem
18
19 # Preenche o título
20 campo_titulo = wait.until(EC.visibility_of_element_located((By.ID, "edit_title")))
21 campo_titulo.clear()
22 campo_titulo.send_keys(titulo_nota)
23
24 conteudo_nota = ""
25 for participante in participantes:
26     conteudo_nota += f"Nome: {participante['nome']}\nMatrícula: {participante['matricula']}\n\n"
27
28 # Preenche a nota
29 campo_texto = wait.until(EC.visibility_of_element_located((By.ID, "edit_textarea")))
30 campo_texto.clear()
31 campo_texto.send_keys(conteudo_nota)
32
33 # Salva a nota
34 botao_salvar = wait.until(EC.presence_of_element_located((By.ID, "btnSaveNote")))
35 print("Botão Salvar localizado. Tentando clicar...")
36
37 # Executa o clique
38 driver.execute_script("arguments[0].scrollIntoView(true);", botao_salvar)
39 time.sleep(0.5) # Espera um pouco para garantir que o botão está visível
40 driver.execute_script("arguments[0].click();", botao_salvar)
41 print("Clique executado no botão Salvar.")
42 print("Nota criada com sucesso!")
43
44 # Mantém o navegador aberto por 20 segundos para visualização
45 print("Mantendo o navegador aberto por 20 segundos...")
46 time.sleep(20)
47 driver.quit()
48 print("Navegador fechado.")

```

FONTE: O autor (2025)

13 CONCLUSÃO

Neste memorial, foram apresentadas todas as disciplinas cursadas ao longo da especialização, destacando os projetos desenvolvidos e sua contribuição para a construção de conhecimento em desenvolvimento ágil de software. O percurso acadêmico contemplou desde fundamentos teóricos até experiências práticas, proporcionando uma visão ampla e integrada sobre as metodologias ágeis.

Ao longo do memorial, evidenciou-se a importância de modelagem de requisitos, planejamento iterativo e incremental e documentação enxuta, bem como a aplicação prática de frameworks modernos como Angular e Spring Boot. Essa abordagem permitiu compreender como a teoria e a prática se conectam, mostrando que o desenvolvimento ágil exige planejamento contínuo aliado à execução eficiente.

As atividades práticas possibilitaram explorar tecnologias variadas e simular cenários reais de projeto, nos quais metodologias como Scrum e Kanban favoreceram a organização, a entrega contínua e o foco no valor entregue ao usuário. Disciplinas como UX, Desenvolvimento Web e Mobile e DevOps reforçaram a importância de automação, integração contínua e versionamento como pilares da agilidade moderna, evidenciando a necessidade de colaboração entre equipes multidisciplinares.

Entretanto, a aplicação prática do desenvolvimento ágil apresenta desafios relevantes:

- **Integração entre práticas e ferramentas:** utilizar metodologias ágeis exige harmonização entre ferramentas de desenvolvimento, testes e colaboração, demandando aprendizado constante e alinhamento entre os membros da equipe.
- **Cultura e resistência à mudança:** a transição de modelos tradicionais para abordagens ágeis envolve adaptação cultural, o que pode gerar resistência, especialmente em estruturas mais rígidas.

- Manutenção da qualidade em ciclos curtos: preservar altos padrões de qualidade mesmo em entregas rápidas requer disciplina no uso de testes automatizados, revisão de código e boas práticas de programação.

Conclui-se, portanto, que o desenvolvimento ágil de software vai além de metodologias; trata-se de uma forma de pensar e agir colaborativamente, integrando planejamento e prática em harmonia. O memorial evidencia a importância de profissionais capazes de planejar, implementar, testar e entregar soluções de forma eficiente, contínua e centrada no usuário, mostrando que os desafios encontrados são oportunidades de fortalecimento da prática ágil e de sua aplicação no mercado real.

REFERÊNCIAS

- BECK, K. et al. **Manifesto Ágil para Desenvolvimento de Software**. 2001. Disponível em: <https://agilemanifesto.org>. Acesso em: 19 jun. 2025.
- HIGHSMITH, J. **Agile Software Development Ecosystems**. Boston: Addison-Wesley, 2002.
- HUMBLE, J.; FARLEY, D. **Continuous Delivery**: Reliable Software Releases through Build, Test, and Deployment Automation. Boston: Addison-Wesley, 2010.
- LARMAN, C. **Utilizando UML e Padrões**. 2. ed. Porto Alegre: Bookman, 2003.
- MARTIN, R. C. **Código limpo**: habilidades práticas do Agile software. Rio de Janeiro: Alta Books, 2008.
- NORMAN, D. **O design do dia a dia**. Rio de Janeiro: Rocco, 2013.
- POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development**: An Agile Toolkit. Boston: Addison-Wesley, 2003.