

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
ESCOLA POLITÉCNICA  
CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE  
AGES – AGÊNCIA EXPERIMENTAL DE ENGENHARIA DE SOFTWARE

FELIPE ROQUE TASONIERO

**MEMORIAL DE ATUAÇÃO NA AGÊNCIA EXPERIMENTAL DE ENGENHARIA DE  
SOFTWARE - PERÍODO 2018 A 2025  
AGES I, II, III E IV**

Porto Alegre, Rio Grande do Sul  
2025

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pela força, sabedoria e amparo em todos os momentos desta caminhada. Agradeço à minha mãe, Tania Regina Vieira Roque, pelo apoio, carinho e presença constante ao longo desta jornada acadêmica e pessoal. Registro também minha profunda gratidão ao meu pai, Luis Augusto Ruschel Tasoniero (in memoriam), cujo exemplo, valores e ensinamentos permanecem como inspiração e força em minha vida. Agradeço ainda à PUCRS, pela construção e manutenção do curso de Bacharelado em Engenharia de Software, que me proporcionou uma formação sólida e experiências transformadoras. Por fim, agradeço aos meus professores, que, com dedicação e incentivo, contribuíram de maneira fundamental para minha formação e para meu crescimento como engenheiro de software.

*“Quem tem um porquê enfrenta  
quase qualquer como.”*

**Friedrich Nietzsche**

## **RESUMO**

O presente documento tem como objetivo descrever o percurso do aluno Felipe Roque Tasoniero durante sua participação na disciplina AGES, destacando experiências acadêmicas, responsabilidades assumidas e aprendizados conquistados. A AGES integra o curso de Bacharelado em Engenharia de Software da PUCRS e é estruturada em quatro níveis, cada um correspondendo a funções distintas em um projeto de desenvolvimento de software. Essa proposta pedagógica proporciona um ambiente de colaboração entre alunos de diferentes semestres, que trabalham em conjunto com o objetivo comum de planejar, desenvolver e entregar uma solução tecnológica.

A dinâmica da disciplina permite que os estudantes experimentem papéis progressivos ao longo da graduação: no nível AGES I, o foco recai sobre tarefas de programação; no AGES II, na análise e administração de banco de dados; no AGES III, na definição e implementação da arquitetura de software; e no AGES IV, na gestão de equipes e de projetos. Esse formato possibilita a vivência prática de diferentes etapas do ciclo de desenvolvimento de software, promovendo não apenas o aprimoramento técnico, mas também o desenvolvimento de competências interpessoais, como comunicação, liderança, organização e colaboração.

**PALAVRAS CHAVES:** AGES, Engenharia de Software, Projeto, Experiência, Aprendizado, Colaboração, Gestão.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Foto da equipe de desenvolvimento e orientadores do projeto Simulação de Desastres . . . . .	12
Figura 2 – Proposta enviada pelo stakeholder para o fluxo de funcionamento do simulador. . . . .	16
Figura 3 – Foto da equipe de desenvolvimento, stakeholders e professor orientador do projeto FICAI 4.0. . . . .	22
Figura 4 – Mockup da tela de Login - FICAI 4.0 . . . . .	25
Figura 5 – Modelo conceitual do banco de dados - FICAI 4.0. . . . .	26
Figura 6 – Modelo lógico do banco de dados - FICAI 4.0 . . . . .	26
Figura 7 – Tela Home (ou Tela Inicial) desenvolvida durante a Sprint 1 - FICAI 4.0. . . . .	28
Figura 8 – Tela Home completa desenvolvida durante a Sprint 4 - FICAI 4.0. . . . .	31
Figura 9 – Tela completa de Edição do Aluno desenvolvida durante a Sprint 4 - FICAI 4.0. . . . .	32
Figura 10 – Equipe de desenvolvimento e Stakeholder - Polymathech . . . . .	36
Figura 11 – Modelo ER do banco de dados - Polymathech . . . . .	37
Figura 12 – Modelo Lógico do banco de dados - Polymathech . . . . .	38
Figura 13 – Arquitetura de deploy da aplicação na AWS - Polymathech . . . . .	39
Figura 14 – Tela inicial - Polymathech . . . . .	40
Figura 15 – Tela de login - Polymathech . . . . .	40
Figura 16 – Tela de cadastro - Polymathech . . . . .	41
Figura 17 – Tela home - Polymathech . . . . .	41
Figura 18 – Tela de teste vocacional - Polymathech . . . . .	42
Figura 19 – Tela de resultados do teste vocacional - Polymathech . . . . .	42
Figura 20 – Equipe de desenvolvimento do projeto - Ai Produtor . . . . .	50
Figura 21 – Modelo lógico do banco de dados - Ai Produtor . . . . .	51
Figura 22 – Arquitetura Back-End do projeto - Ai Produtor . . . . .	52
Figura 23 – Tela Home - Ai Produtor . . . . .	53
Figura 24 – Tela de Novo Plantio - Ai Produtor . . . . .	54
Figura 25 – Painel de Controle de Safra - Ai Produtor . . . . .	54

Figura 26 – Lista de Safras no Painel de Controle - Ai Produtor . . . . .	55
Figura 27 – Histórico de Safras - Ai Produtor . . . . .	55
Figura 28 – Tela de Edição de Safra - Ai Produtor . . . . .	56
Figura 29 – Tela de Listagem de Safras - Ai Produtor . . . . .	56
Figura 30 – Tela de Criação de Safra - Ai Produtor . . . . .	57
Figura 31 – Tela de Desenho de Área de Plantio - Ai Produtor . . . . .	57
Figura 32 – Tela de Listagem de Áreas de Plantio - Ai Produtor . . . . .	58
Figura 33 – Tela de Cadastro de Produtor - Ai Produtor . . . . .	58

## LISTA DE SIGLAS

<b>AGES</b>	Agência Experimental de Engenharia de Software . . . . .	9
<b>API</b>	Application Programming Interface . . . . .	13
<b>AWS</b>	Amazon Web Services . . . . .	10
<b>CDM</b>	Centro de Gerenciamento de Desastres . . . . .	11
<b>CI/CD</b>	Integração Contínua e Entrega Contínua . . . . .	10
<b>CLI</b>	Command Line Interface . . . . .	14
<b>CRUD</b>	Create, Read, Update and Delete . . . . .	62
<b>DDD</b>	Domain Driven Design . . . . .	10
<b>DTO</b>	Data Transfer Object . . . . .	52
<b>EC2</b>	Elastic Compute Cloud . . . . .	38
<b>ER</b>	Entidade Relacionamento . . . . .	37
<b>EAP</b>	Estrutura Analítica do Projeto . . . . .	60
<b>FICAI 4.0</b>	Ficha de Comunicação de Aluno Infrequente 4.0 . . . . .	10
<b>GPIN</b>	Grupo de Pesquisa em Inteligência de Negócio . . . . .	9
<b>HP</b>	Hewlett-Packard . . . . .	9
<b>HTML</b>	Linguagem de Marcação de Hipertexto . . . . .	30
<b>HTTP</b>	Hypertext Transfer Protocol . . . . .	27
<b>IA</b>	Inteligência Artificial . . . . .	63
<b>IJCNN</b>	International Joint Conference on Neural Networks . . . . .	9
<b>INEP</b>	O Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira . . . . .	35
<b>JWT</b>	JSON Web Token . . . . .	44
<b>LIS</b>	Laboratório de Inovação em Software . . . . .	9
<b>MAPC</b>	Multi-Agent Programming Contest . . . . .	11
<b>MVC</b>	Model-View-Controller . . . . .	23

<b>NPM</b>	Node Package Manager . . . . .	24
<b>PET</b>	Programa de Educação Tutorial . . . . .	9
<b>PROCEMPA</b>	Companhia de Processamento de Dados de Porto Alegre . . . . .	21
<b>PUCRS</b>	Pontifícia Universidade Católica do Rio Grande do Sul . . . . .	9
<b>REST</b>	Representational State Transfer . . . . .	14
<b>SGBD</b>	Sistemas de Gerenciamento de Banco de Dados . . . . .	51
<b>SSH</b>	Secure Shell . . . . .	38
<b>SQL</b>	Structured Query Language . . . . .	30
<b>TI</b>	Tecnologia da Informação . . . . .	9
<b>UFRGS</b>	Universidade Federal do Rio Grande do Sul . . . . .	9
<b>UI</b>	User Interface . . . . .	10
<b>US</b>	User Stories . . . . .	28
<b>UX</b>	User Experience . . . . .	10
<b>VR</b>	Virtual Reality . . . . .	9
<b>XML</b>	Extensible Markup Language . . . . .	14
<b>XSS</b>	Cross-Site Scripting . . . . .	29

## SUMÁRIO

1	<b>APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO . . . . .</b>	9
2	<b>AGES I — “SIMULAÇÃO DE DESASTRES 2018” . . . . .</b>	11
2.1	<b>Introdução . . . . .</b>	11
2.2	<b>Desenvolvimento do Projeto . . . . .</b>	12
2.3	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	15
2.4	<b>Conclusão . . . . .</b>	19
3	<b>AGES II — “FICAI 4.0 – NENHUMA CRIANÇA OU ADOLESCENTE FORA DA ESCOLA 2022” . . . . .</b>	21
3.1	<b>Introdução . . . . .</b>	21
3.2	<b>Desenvolvimento do Projeto . . . . .</b>	22
3.3	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	24
3.4	<b>Conclusão . . . . .</b>	33
4	<b>AGES III — “POLYMATHECH 2024” . . . . .</b>	35
4.1	<b>Introdução . . . . .</b>	35
4.2	<b>Desenvolvimento do Projeto . . . . .</b>	36
4.3	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	43
4.4	<b>Conclusão . . . . .</b>	47
5	<b>AGES IV — “AI PRODUTOR 2025” . . . . .</b>	49
5.1	<b>Introdução . . . . .</b>	49
5.2	<b>Desenvolvimento do Projeto . . . . .</b>	50
5.3	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	59
5.4	<b>Conclusão . . . . .</b>	66
6	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	68
	<b>Referências . . . . .</b>	70

## 1 APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

Em 2018, iniciei minha carreira na área de Tecnologia da Informação (TI) através do curso de Engenharia de Software na Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Porém, esse não foi exatamente o início da minha carreira profissional, mas sim uma transição. Minha primeira formação foi no curso de Bacharelado em Engenharia Mecânica pela Universidade Federal do Rio Grande do Sul (UFRGS), onde tive meu primeiro contato com programação de computadores. Ao longo da graduação, realizei trabalhos acadêmicos que integravam conceitos de Mecânica com programação, o que me permitiu enxergar o potencial transformador da área de TI. Em 2017, após concluir a graduação em Engenharia Mecânica, decidi seguir minha verdadeira vocação: tornar-me engenheiro de software. Como primeiro passo, ingressei no curso de Engenharia de Software na PUCRS.

Já no primeiro semestre da graduação, percebi que havia tomado a decisão certa. O ambiente acadêmico, o contato com professores e colegas e a possibilidade de atuar como bolsista de Iniciação Científica no grupo Programa de Educação Tutorial (PET) e no laboratório Grupo de Pesquisa em Inteligência de Negócio (GPIN), além da experiência como Agência Experimental de Engenharia de Software (AGES) I no projeto Simulação de Desastres, foram determinantes para consolidar essa escolha. Entre 2018 e 2019, no GPIN, tive contato intenso com a área de Deep Learning, o que despertou meu interesse em continuar na pesquisa acadêmica. Como já possuía uma graduação, decidi candidatar-me a uma vaga de Mestrado em Ciência da Computação na PUCRS. Fui aprovado tanto no curso quanto para atuar como bolsista no laboratório Laboratório de Inovação em Software (LIS). Esse período foi marcado por pesquisas em Reconstrução 3D com modelos de Deep Learning, resultando na publicação de um artigo no International Joint Conference on Neural Networks (IJCNN), além de trabalhos relacionados a aplicações de realidade Virtual Reality (VR) no contexto de Aprendizado Profundo.

Ao concluir o Mestrado, iniciei minha carreira profissional como Engenheiro de Software na Hewlett-Packard (HP) Inc., consolidando minha transição de carreira. Após alguns meses de experiência prática no mercado, retomei o curso de Engenharia de Software em 2022, conciliando as atividades acadêmicas com minha atuação profis-

sional. Nesse período, eu já acumulava experiência em Aprendizado de Máquina e Computação Gráfica, mas percebia a importância de consolidar conhecimentos mais amplos e aprofundados em Engenharia de Software.

Como Engenheiro de Software, pude me desenvolver em áreas como desenvolvimento web (Back-End e Front-End) utilizando tecnologias .NET, containerização de aplicações, segurança da informação, processos de Integração Contínua e Entrega Contínua (CI/CD) e práticas de gerenciamento de projetos. Ao mesmo tempo, como AGES II no projeto Ficha de Comunicação de Aluno Infrequente 4.0 (FICAI 4.0), em 2022, tive contato com tecnologias como Angular e Spring Boot, além de aprimorar noções de User Experience (UX)/User Interface (UI). Nas disciplinas da graduação, consolidei aprendizados em Testes de Software, Processamento Paralelo, Gerenciamento de Configuração e Gerenciamento de Projetos de Software.

O ano de 2023 foi marcado pelo aprofundamento em diferentes tecnologias, tanto em projetos profissionais quanto acadêmicos, incluindo um maior envolvimento com Cloud Computing. Já em 2024, participei como AGES III no projeto Polymathech, atuando como arquiteto de software. Nessa função, pude auxiliar na definição da estrutura do projeto utilizando NodeJS, NestJS e ReactJS, além de configurar a aplicação na Amazon Web Services (AWS). Também contribuí para a organização das arquiteturas de Front-End, baseadas em componentização, e de Back-End, fundamentadas na Clean Architecture e Domain Driven Design (DDD).

Ao longo de 2025, concluí o último semestre do curso de Engenharia de Software na PUCRS enquanto seguia atuando como Engenheiro de Software na HP Inc. Esse período marcou a convergência entre prática profissional e formação acadêmica, permitindo aplicar no trabalho conceitos avançados de arquitetura, desenvolvimento e CI/CD, ao mesmo tempo em que aprofundava fundamentos e refletia sobre decisões de engenharia na universidade.

Minha participação na AGES IV, no projeto Ai Produtor, consolidou essa maturidade. Nesse papel, pude exercer liderança técnica, apoiar squads, estruturar processos e conduzir decisões arquiteturais envolvendo contêineres e serviços em AWS. A experiência reforçou a importância de organização, comunicação clara e orientação ao time, encerrando minha trajetória na AGES de forma integrada à realidade do mercado e alinhada à minha atuação profissional.

## 2 AGES I — “SIMULAÇÃO DE DESASTRES 2018”

Esta seção busca apresentar a passagem do aluno como AGES I pelo projeto Simulação de Desastres. Aqui estão descritos os artefatos entregues, a atuação ao longo das sprints e os pontos de melhoria identificados pelo aluno.

### 2.1 Introdução

O projeto *Simulação de Desastres* foi desenvolvido entre agosto e novembro de 2018, sob a supervisão do professor Dr. Marcelo Viegas Neves, posteriormente substituído pelo professor Dr. Edson Moreno em virtude de um afastamento. A equipe de desenvolvimento foi composta por 18 alunos, conforme ilustrado na Figura 1, e contou com o apoio do arquiteto de software da AGES, Cássio Trindade. Os stakeholders do projeto foram o mestrande Tabajara Krausburg Rodrigues e o professor Dr. Rafael Heitor Bordini, ambos vinculados a um estudo sobre a formação de coalizões restritas entre agentes heterogêneos.

O principal objetivo do projeto foi desenvolver um simulador estocástico de desastres naturais, com foco em enchentes, no qual agentes inteligentes seriam capazes de atuar de forma autônoma diante de situações simuladas como fotografar áreas específicas, coletar amostras de água e resgatar vítimas. Cada agente era representado por um veículo autônomo, podendo ser terrestre (carro), aquático (barco) ou aéreo (drone), possuindo conjuntos distintos de habilidades e limitações de acordo com o ambiente em que operava.

A simulação foi executada sobre mapas reais, exportados diretamente do *OpenStreetMap* (OPENSTREETMAP FOUNDATION, 2025), o que conferiu ao projeto um caráter mais realista e dinâmico. Os agentes interagiam tanto com o ambiente quanto entre si, tomando decisões com base em percepções geradas a cada turno da simulação. Além disso, o sistema contava com um ponto central fixo, o *Centro de Gerenciamento de Desastres* (Centro de Gerenciamento de Desastres (CDM)), onde os agentes podiam entregar vítimas, recarregar baterias e realizar trocas de informações e recursos.

O software desenvolvido foi baseado na ferramenta *MASSIM* (KÖSTER, MICHAEL AND MATTHES, FLORIAN AND SEITZ, FLORIAN, 2017), utilizada no Multi-Agent

Programming Contest (MAPC) para pesquisas envolvendo sistemas multiagente. O simulador seguia uma estrutura de turnos, em que a cada rodada os agentes recebiam percepções, tomavam decisões e executavam uma única ação, alterando o estado global da simulação.

O ciclo completo da simulação iniciava-se com a conexão dos agentes ao servidor, a geração de eventos aleatórios e a atualização do estado global. A partir daí, as ações dos agentes eram processadas e o ambiente reagia a elas, permitindo a observação do comportamento coletivo e a avaliação das estratégias adotadas por cada agente. Essa abordagem proporcionou uma experiência de aprendizado prática e interdisciplinar, unindo conceitos de programação, inteligência artificial e modelagem de sistemas complexos.

**Figura 1** – Foto da equipe de desenvolvimento e orientadores do projeto Simulação de Desastres.



Fonte: Wiki do projeto.

## 2.2 Desenvolvimento do Projeto

Nesta seção são apresentados os artefatos desenvolvidos ao longo do semestre no qual se situou o projeto. Também estão descritas as etapas realizadas para as entregas destes artefatos, bem como ilustrações e seus respectivos links para a wiki do projeto.

### 2.2.1 Repositório do Código Fonte do Projeto

Foi criado um único repositório, intitulado “Desastres”, que concentrava tanto o código-fonte quanto a documentação completa do projeto. A wiki do repositório continha instruções detalhadas sobre a execução do simulador, configuração do ambiente e explicações sobre o funcionamento interno do sistema. O repositório pode ser acessado através do seguinte link: <https://tools.ages.pucrs.br/desastres>.

### 2.2.2 Banco de Dados Utilizado

Como o simulador mantinha seu estado diretamente em memória e realizava leitura e escrita em arquivos locais, não houve necessidade de integração com um sistema de gerenciamento de banco de dados. Dessa forma, a persistência de informações era feita através de arquivos de configuração e logs, garantindo desempenho satisfatório e simplicidade para os testes realizados durante o desenvolvimento.

### 2.2.3 Arquitetura Utilizada

A arquitetura do sistema foi desenvolvida com base no modelo cliente-servidor. O servidor, responsável pela execução da simulação e pelo controle do ambiente, disponibilizava uma Application Programming Interface (API) de comunicação com os agentes, que representavam os clientes. Como o objetivo principal do projeto era construir o núcleo do simulador, o foco do desenvolvimento concentrou-se inteiramente na implementação do servidor. A estrutura interna do simulador foi dividida em duas camadas principais:

- **Camada de comunicação:** responsável por gerenciar o envio e o recebimento de informações entre os agentes e o servidor;
- **Camada de simulação:** núcleo lógico do sistema, responsável pela geração de eventos, controle de estado e processamento das ações dos agentes.

Dentro da camada de simulação, dois módulos desempenhavam papéis específicos: o módulo *IO*, responsável pela leitura e escrita de arquivos, e o módulo *Core*, que continha a lógica principal da simulação. Essa divisão facilitou a manutenção e a evolução do código, garantindo uma separação clara entre as responsabilidades.

O fluxo geral do sistema seguia o seguinte processo: os agentes se conectavam ao servidor, recebiam identificadores únicos e aguardavam o início da simulação. Após o registro de todos os agentes, o servidor iniciava a execução dos turnos, processando ações, atualizando estados e retornando as percepções aos agentes de forma cíclica. Essa estrutura refletia bem os princípios de um sistema multiagente distribuído e reforçou minha compreensão sobre interação entre entidades autônomas.

#### 2.2.4 Protótipos das Telas Desenvolvidas

Como o foco do projeto estava voltado ao desenvolvimento de um simulador com Command Line Interface (CLI), não foram criados protótipos gráficos ou interfaces visuais para aprovação dos stakeholders. O sistema operava inteiramente por meio de comandos e respostas textuais, o que simplificou a execução e o teste das funcionalidades durante as sprints.

#### 2.2.5 Tecnologias Utilizadas

O desenvolvimento foi realizado utilizando a linguagem *Python* (PYTHON SOFTWARE FOUNDATION, 2025), aplicando conceitos de orientação a objetos para representar agentes, eventos e entidades do ambiente de forma modular e reutilizável. Cada agente e evento era estruturado como uma classe, com atributos e métodos próprios para refletir seu comportamento no ambiente simulado.

A comunicação entre o simulador e os agentes foi implementada através de uma API Representational State Transfer (REST), construída com o framework *Flask* (GRINBERG, MIGUEL, 2025). Inicialmente, essa API era utilizada apenas para o registro dos agentes na simulação, mas foi posteriormente expandida para permitir trocas de informações mais complexas. Para possibilitar comunicação em tempo real, utilizou-se a extensão *Flask-SocketIO*, que permitiu a troca bidirecional de mensagens via HTTP e WebSocket, garantindo sincronização durante os turnos da simulação.

O cenário do simulador foi construído a partir de arquivos Extensible Markup Language (XML) exportados do *OpenStreetMap*, contendo pontos de interesse e rotas reais, o que conferia ao ambiente características autênticas e dinâmicas. Essa integração foi fundamental para que os agentes pudessem se mover em um espaço realista, reconhecendo terrenos, estradas e regiões alagadas durante os testes.

## 2.3 Atividades Desempenhadas Pelo Aluno no Projeto

### 2.3.1 Sprint 0

A Sprint 0 marcou o primeiro contato da equipe com o projeto *Simulação de Desastres*. Nessa etapa inicial, ocorreu a apresentação do projeto pelo stakeholder, que compartilhou a proposta, os objetivos gerais e as expectativas em relação ao produto final. A partir desse encontro, iniciou-se um processo de discussão coletiva para compreender as necessidades do projeto, identificar possíveis abordagens de implementação e definir tecnologias adequadas para o seu desenvolvimento.

O stakeholder forneceu uma base sólida para o início dos trabalhos, disponibilizando fluxogramas Figura 2, imagens explicativas e algumas *User Stories* já definidas, o que auxiliou a equipe a visualizar de maneira mais clara os requisitos e funcionalidades esperadas do simulador. A partir dessas informações, o grupo deu início ao processo de organização e estruturação do projeto, estabelecendo o ambiente de desenvolvimento e discutindo as primeiras estratégias de implementação.

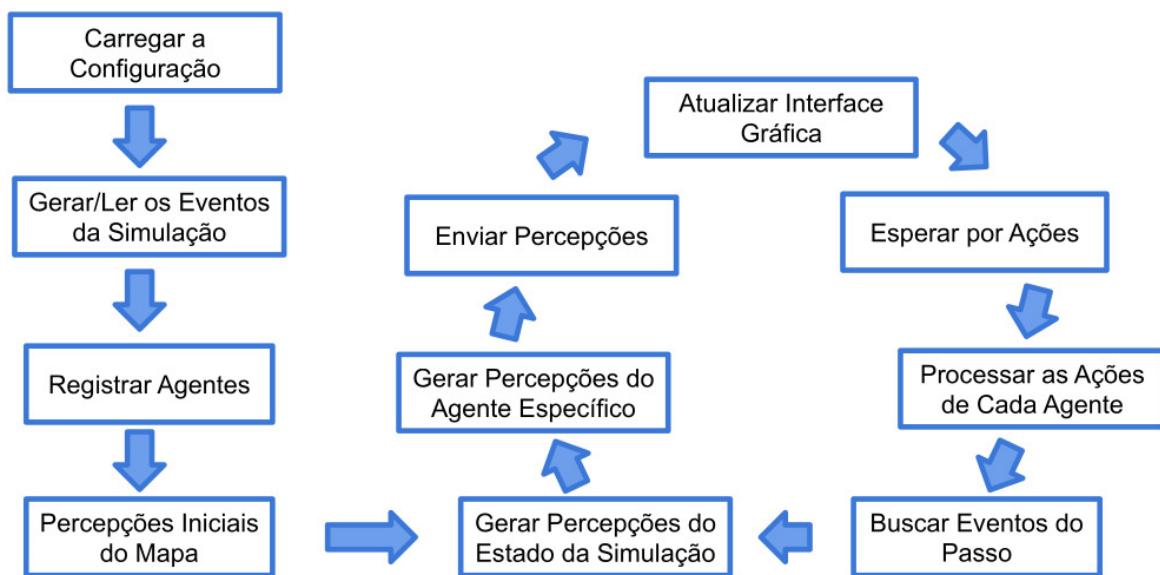
Como integrante do nível AGES I, minha principal responsabilidade era atuar como desenvolvedor e auxiliar na documentação do sistema. Embora eu já tivesse tido algum contato prévio com projetos relacionados à área de Simulação Computacional, essa experiência representou meu primeiro envolvimento em um desenvolvimento colaborativo com papéis e responsabilidades formalmente definidos. Logo percebi que o sucesso do projeto dependia fortemente de uma comunicação eficaz e de uma boa integração entre os membros da equipe.

Durante essa sprint, tive também meu primeiro contato com a linguagem de programação Python e com o framework Flask, que seriam amplamente utilizados no projeto. Com o apoio de colegas mais experientes, utilizei esse período para estudar os fundamentos dessas tecnologias e compreender suas principais estruturas e padrões de uso. Além disso, busquei contribuir com o time na análise do comportamento esperado do simulador. No entanto, essa atividade revelou-se desafiadora, pois quanto mais detalhávamos o funcionamento do sistema, mais dúvidas surgiam sobre os processos internos e a interação entre seus componentes.

Foi também nesse momento que entrei em contato com diversos conceitos que até então eram novos para mim, como o funcionamento de uma API, os protocolos de

comunicação, os tipos de dados trocados entre os agentes e o servidor, e o processamento realizado pelo núcleo do simulador. Apesar das incertezas, essa sprint foi fundamental para o amadurecimento técnico e conceitual da equipe. Ao final dessa fase, ainda não havia uma estrutura completamente definida do sistema, mas a reunião de encerramento com o stakeholder foi essencial para esclarecer dúvidas e alinhar expectativas, consolidando a base para as etapas seguintes do projeto.

**Figura 2** – Proposta enviada pelo stakeholder para o fluxo de funcionamento do simulador.



Fonte: Wiki do projeto.

### 2.3.2 Sprint 1

Durante a Sprint 1, os integrantes de nível AGES IV decidiram dividir a equipe em dois grupos, a fim de otimizar o processo de desenvolvimento. O primeiro grupo, do qual eu fiz parte, ficou responsável pela criação da estrutura de comunicação entre os agentes e a API. O segundo grupo dedicou-se à implementação do núcleo do simulador, que seria o componente central do sistema.

No início dessa sprint, a equipe ainda não possuía uma definição concreta da arquitetura do simulador nem dos mecanismos específicos de comunicação que seriam utilizados. Diante dessa incerteza, o grupo responsável pela comunicação iniciou um processo de análise e planejamento, buscando compreender quais informações precisariam ser trocadas entre os agentes e o servidor. A partir dessas discussões, começamos também a estruturar a documentação técnica do projeto na Wiki, descrevendo

as decisões tomadas e os formatos de mensagens previstos para a interação entre as partes do sistema.

Durante essa etapa, enfrentei algumas dificuldades relacionadas à comunicação com os colegas dos níveis AGES II e III, especialmente quando necessitava de apoio técnico sobre as tecnologias adotadas no projeto. Essa limitação reforçou a importância de uma comunicação mais integrada entre os diferentes níveis da equipe, um aspecto essencial em projetos de desenvolvimento colaborativo.

Paralelamente, mantive o foco nos estudos dirigidos sobre Python e Flask, aprofundando meus conhecimentos nas ferramentas que seriam fundamentais para o andamento do projeto. Também contribuí na organização e atualização da Wiki, registrando informações sobre as primeiras decisões arquiteturais e o comportamento esperado do sistema. Além disso, da mesma forma que na Sprint 0, não tivemos nenhuma entrega concreta da aplicação, concentrando nossos esforços em alinhar com o stakeholder a forma como estávamos estruturando o simulador e o modelo de comunicação. Esta sprint foi, portanto, um momento de consolidação do aprendizado técnico e de amadurecimento na dinâmica de trabalho em equipe, servindo de base para o desenvolvimento mais estruturado das etapas seguintes.

### 2.3.3 Sprint 2

Durante a Sprint 2, foram iniciados o desenvolvimento do motor de simulação e a implementação da estrutura dos agentes. No motor da simulação foram incorporados os geradores de eventos, responsáveis por criar situações dinâmicas durante os turnos de execução. Paralelamente, minha equipe ficou encarregada de implementar as classes de agentes, definindo a estrutura necessária para a execução das ações disponíveis, como locomoção, coleta de imagens, resgate de vítimas, entre outras.

Nessa etapa, eu ainda enfrentava certa dificuldade de comunicação com alguns colegas mais experientes, principalmente no esclarecimento de dúvidas técnicas relacionadas à arquitetura do sistema e às tecnologias utilizadas. Apesar disso, recebi apoio constante dos alunos de nível AGES IV, que auxiliaram no entendimento do funcionamento geral da aplicação e na organização das tarefas.

Como ainda havia algumas barreiras técnicas para que eu pudesse contribuir diretamente com o desenvolvimento das principais funcionalidades, assumi a responsa-

bilidade pela implementação da estrutura de testes unitários. Juntamente com outro colega de nível AGES I, dedicamos boa parte do tempo dessa sprint à construção de testes para validar as ações realizadas pelos agentes, como movimentação e coleta de dados. Essa experiência foi fundamental para compreender a importância da verificação automática de código e do papel dos testes na garantia da qualidade e confiabilidade do sistema.

#### 2.3.4 Sprint 3

Durante a Sprint 3, a equipe demonstrou um certo desânimo em relação ao andamento do projeto, o que impactou o ritmo de desenvolvimento. Nessa etapa, iniciamos a implementação do processo de comunicação entre os agentes e o simulador. Para essa finalidade, foi adotada a comunicação via *WebSockets*, utilizando os recursos oferecidos pelo *Flask-SocketIO*.

A escolha por essa tecnologia trouxe diversas vantagens ao projeto. Diferentemente do modelo tradicional de requisições *HTTP*, os *WebSockets* permitem uma comunicação bidirecional e em tempo real entre cliente e servidor. Essa característica foi fundamental para o simulador, uma vez que os agentes precisavam enviar e receber informações continuamente a cada turno de simulação, sem a necessidade de estabelecer novas conexões a cada interação. Além disso, o uso do *Flask-SocketIO* simplificou a integração com o framework já utilizado, garantindo maior desempenho e escalabilidade à aplicação.

Com o avanço do desenvolvimento, surgiram diversos erros de execução e inconsistências na troca de mensagens, o que levou a equipe a concentrar seus esforços na realização de testes e na correção dos problemas de comunicação. Ao final da sprint, constatamos que nenhuma *User Story* havia sido concluída integralmente, uma vez que o foco se voltou para a estabilização do sistema.

Apesar das dificuldades enfrentadas, percebi uma melhora significativa na minha comunicação com os colegas mais experientes, o que me permitiu contribuir de forma mais efetiva nas discussões técnicas. Pude propor ideias e possíveis soluções para os problemas de sincronização e envio de mensagens entre os agentes e o simulador, o que representou um importante avanço na minha participação e no meu desenvolvimento dentro do projeto.

### 2.3.5 Sprint 4

Na Sprint 4, a equipe decidiu interromper o desenvolvimento de novos testes para concentrar os esforços na correção de erros e na estabilização do funcionamento do simulador. O objetivo principal dessa etapa foi garantir uma comunicação consistente entre o simulador e os agentes, uma vez que ainda enfrentávamos diversos problemas relacionados à manipulação das informações transmitidas e recebidas durante a execução. Para simplificar o processo e assegurar resultados mais controlados, optamos por implementar apenas algumas ações básicas para os agentes, como locomoção e coleta de água.

Durante essa última sprint, busquei auxiliar os colegas em diferentes tarefas, especialmente na identificação e correção de falhas no código. No entanto, ainda percebi dificuldades de comunicação e um menor nível de engajamento por parte de alguns membros mais experientes, o que impactou diretamente o ritmo de desenvolvimento.

Ao término da sprint, não conseguimos concluir integralmente as entregas propostas pelo stakeholder. O simulador ainda apresentava comportamentos inconsistentes e carecia da implementação de diversas ações adicionais para os agentes, bem como de uma transmissão completamente estável das informações entre os componentes do sistema. Apesar disso, essa etapa final foi importante para consolidar aprendizados sobre trabalho colaborativo, gestão de prioridades e resolução de problemas em ambientes de desenvolvimento de software, encerrando a experiência da AGES I com uma visão mais realista dos desafios presentes em projetos complexos.

## 2.4 Conclusão

O projeto *Simulação de Desastres*, desenvolvido na disciplina AGES I, marcou meu primeiro contato com um ambiente de desenvolvimento colaborativo real e estruturado. Essa experiência representou um ponto de virada na minha formação, pois me permitiu vivenciar de forma prática os desafios de um projeto de software conduzido por uma equipe multidisciplinar, com diferentes níveis de experiência e responsabilidades. Ao longo do semestre, comprehendi que o sucesso de um projeto não depende apenas do domínio técnico, mas também da capacidade de comunicação, da clareza de papéis e da organização coletiva do grupo.

O desenvolvimento do simulador exigiu o aprendizado de novas tecnologias e paradigmas. Foi meu primeiro contato com a linguagem *Python*, com o framework *Flask* e com o uso de *WebSockets* por meio do *Flask-SocketIO*, que possibilitaram a comunicação em tempo real entre agentes e servidor. Além do aprendizado técnico, o projeto me proporcionou uma visão mais ampla sobre arquitetura de sistemas distribuídos, testes automatizados e a importância da documentação como instrumento essencial para o alinhamento e continuidade do trabalho em equipe.

Nos primeiros momentos, predominava um sentimento de incerteza quanto à complexidade do simulador e às responsabilidades individuais. Entretanto, conforme as tarefas foram sendo divididas, ficou evidente o quanto a integração entre os membros era determinante para o avanço do projeto. Mesmo diante das dificuldades, pude contribuir com a documentação técnica, a criação de testes unitários e a proposição de soluções para os problemas de comunicação entre os agentes e o simulador.

Apesar dos progressos obtidos, o projeto enfrentou entraves significativos relacionados à gestão e à comunicação da equipe. A ausência de uma liderança constante, o distanciamento de alguns integrantes mais experientes e a centralização das decisões em poucos membros acabaram impactando o ritmo e a qualidade das entregas. Além disso, a falta de acompanhamento efetivo das tarefas dificultou a visualização do progresso geral.

Para futuras edições de projetos semelhantes, considero essencial a adoção de práticas que reforcem a colaboração e a transparência entre os membros da equipe. Seria recomendável definir papéis de liderança técnica e organizacional mais bem delimitados, garantindo uma melhor distribuição de responsabilidades e o acompanhamento das atividades. Por fim, o incentivo à cultura de feedback e à troca de conhecimento entre os níveis contribuiria para uma equipe mais engajada, coesa e preparada para lidar com desafios complexos.

Encerrar o AGES I evidenciou que meu maior aprendizado não esteve nas tecnologias, mas na percepção de como falhas de organização, liderança e comunicação comprometem um projeto. Essa experiência reforçou que a maturidade do processo e a colaboração são tão importantes quanto a técnica. Saio do AGES I consciente de que trabalhar em equipe é tão desafiador quanto indispensável para atuar de forma profissional.

### **3 AGES II — “FICAI 4.0 – NENHUMA CRIANÇA OU ADOLESCENTE FORA DA ESCOLA 2022”**

Esta seção apresenta a atuação do aluno como integrante do nível AGES II no projeto *FICAI 4.0*. São descritos os artefatos desenvolvidos, as atividades desempenhadas ao longo das sprints e os principais pontos de melhoria observados durante a execução do projeto.

#### **3.1 Introdução**

O projeto *FICAI 4.0* surgiu da necessidade de aprimorar uma ferramenta já existente, denominada *FICAI*, que tem como objetivo tornar mais eficiente o processo de identificação das causas de abandono e infrequência escolar. A iniciativa visa enfrentar esses problemas de forma estruturada e colaborativa, especialmente considerando o agravamento desses índices durante o período da pandemia de Covid-19.

O principal propósito do projeto foi desenvolver uma aplicação web responsiva capaz de armazenar e gerenciar dados de alunos em situação de infrequência ou abandono escolar, permitindo o acompanhamento e a intervenção de forma mais ágil e integrada. A nova ferramenta buscou definir fluxos de comunicação entre os diferentes atores envolvidos — como escolas, conselhos tutelares, secretarias e o Ministério Público — de modo que as informações fossem compartilhadas de forma contínua entre todas as etapas do processo. Além disso, o sistema foi projetado para incluir novas funcionalidades, tais como: (1) uma interface mais intuitiva e acessível para o registro de dados; (2) geração de relatórios que possibilitam a análise das causas da infrequência escolar em larga escala; e (3) implementação de um modelo de acesso ampliado, permitindo que mais usuários possam utilizar a plataforma de maneira segura e eficiente.

O projeto ocorreu entre os meses de agosto a dezembro de 2022, sob a orientação do professor Dr. Daniel Callegari. A equipe de desenvolvimento foi composta por 14 alunos do curso de Engenharia de Software da PUCRS e contou com a participação dos stakeholders Dra. Luciana Cano Casarotto, representante do Ministério Público do Rio Grande do Sul, e Renata Bauerie, representante da Companhia de Processamento de Dados de Porto Alegre (PROCEMPA), conforme ilustrado na Figura 3.

**Figura 3** – Foto da equipe de desenvolvimento, stakeholders e professor orientador do projeto FICAI 4.0.



Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/home>

### 3.2 Desenvolvimento do Projeto

Nesta seção são apresentados os artefatos desenvolvidos ao longo do semestre de execução do projeto, bem como as etapas percorridas para a entrega desses artefatos. Também são incluídas ilustrações e links de acesso à wiki do projeto, onde se encontram registros técnicos detalhados sobre cada componente desenvolvido.

#### 3.2.1 Repositório do Código Fonte do Projeto

O código-fonte do projeto *FICAI 4.0* foi organizado em dois repositórios distintos, com o objetivo de separar de forma clara as responsabilidades entre o Back-End e o Front-End. O primeiro repositório contém a implementação da API do sistema, responsável pela lógica de negócios e pela integração com o banco de dados. O segundo abriga o código da interface de usuário, responsável pela camada de apresentação e interação com o sistema.

Os repositórios podem ser acessados por meio dos seguintes links:

- <https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-frontend>
- <https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-backend>

### 3.2.2 Banco de Dados Utilizado

No projeto FICAI 4.0 foi necessário utilizar o banco de dados relacional PostgreSQL (POSTGRESQL, 2025), pois estava contido no conjunto de tecnologias especificadas pelos stakeholders para o desenvolvimento do projeto.

Informações detalhadas sobre a modelagem e a implementação do banco de dados podem ser consultadas na wiki do projeto, por meio do seguinte link:

- [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/banco\\_dados](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/banco_dados)

### 3.2.3 Arquitetura Utilizada

A arquitetura adotada para o desenvolvimento do sistema foi baseada no padrão Model-View-Controller (MVC) (BUSCHMANN et al., 1996), que visa separar as responsabilidades da aplicação em três camadas principais: *Model*, *View* e *Controller*.

As camadas *Model* e *Controller* foram implementadas no módulo Back-End, responsáveis pelo acesso e manipulação dos dados, pela lógica de negócios e pela comunicação com a camada de apresentação. Já a camada *View* foi desenvolvida no módulo Front-End, sendo responsável pela exibição das informações e pela interação do usuário com a aplicação.

Mais detalhes sobre a arquitetura do projeto estão disponíveis na wiki, no endereço:

- <https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/arquitetura>

### 3.2.4 Protótipos das Telas Desenvolvidas

O processo de design da *FICAI 4.0* foi dividido em duas etapas principais. Na primeira, a equipe desenvolveu protótipos de baixa fidelidade com o objetivo de discutir funcionalidades, fluxos de navegação e integração entre diferentes telas da aplicação. Na segunda etapa, foram elaborados protótipos de alta fidelidade na ferramenta *Figma* (FIGMA, 2025), permitindo visualizar o fluxo completo de interação do usuário e definir o conjunto de elementos visuais a serem aplicados no desenvolvimento.

Os protótipos e o material de design estão disponíveis no seguinte link:

- [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design\\_Mockups](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design_Mockups)

### 3.2.5 Tecnologias Utilizadas

O projeto *FICAI 4.0* contou com um conjunto de tecnologias pré-definidas pelos stakeholders, as quais deveriam compor a base do desenvolvimento: PostgreSQL (POSTGRESQL, 2025), Spring Boot (SPRING BOOT, 2025) e Angular (ANGULAR, 2025). Além dessas, outras ferramentas foram incorporadas para otimizar o processo de desenvolvimento e garantir maior padronização, como Docker (DOCKER, 2025) e Swagger (SWAGGER, 2025).

As linguagens de programação empregadas foram Java (JAVA, 2025), utilizada no módulo Back-End, e *TypeScript* (TYPESCRIPT, 2025), empregada no módulo Front-End. Para o gerenciamento de dependências e pacotes, utilizaram-se as ferramentas Maven (MAVEN, 2025) no Back-End e Node Package Manager (NPM) (NPM, 2025) no Front-End. Essa combinação de tecnologias garantiu uma base sólida para o desenvolvimento, permitindo a integração entre os módulos e a manutenção facilitada do código.

## 3.3 Atividades Desempenhadas Pelo Aluno no Projeto

### 3.3.1 Sprint 0

Durante a Sprint 0 do projeto *FICAI 4.0*, foram desenvolvidas as seguintes atividades: estudos dirigidos, criação dos *mockups*, modelagem do banco de dados, configuração inicial das tecnologias utilizadas e suporte aos alunos de nível AGES I.

Como etapa inicial, foi necessário adquirir uma compreensão básica das principais tecnologias adotadas no projeto, a fim de poder contribuir de forma mais efetiva nas etapas seguintes. Esse aprendizado ocorreu por meio de estudos dirigidos, com base em vídeos e leituras em blogs técnicos. Em seguida, após uma série de reuniões com o time de desenvolvimento e com as stakeholders, foi possível mapear as necessidades da ferramenta e traduzi-las em um modelo visual e funcional — os *mockups* — que representavam de maneira próxima o comportamento esperado da aplicação *FICAI 4.0*, conforme ilustrado na Figura 4.

Os *mockups* desenvolvidos contemplaram tanto aspectos visuais, como paleta de cores, imagens e logotipo, quanto elementos funcionais, como botões interativos, menus do tipo *dropdown* e tabelas expansíveis. Esses protótipos serviram como referê-

cia para a equipe de desenvolvimento, permitindo uma visão mais clara da interface e da experiência de uso do sistema.

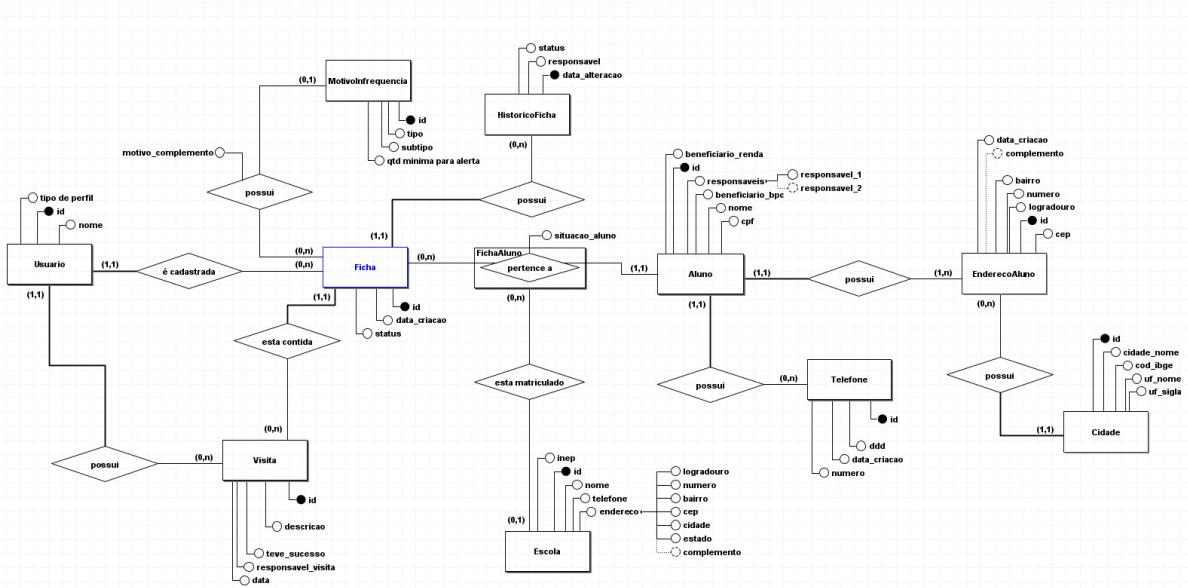
Paralelamente, foram elaborados os modelos conceitual e lógico do banco de dados a ser utilizado no projeto. O modelo conceitual (Figura 5) possibilitou identificar as entidades principais — como usuários, fichas, alunos e escolas —, bem como os relacionamentos entre elas e os atributos correspondentes de cada entidade.

**Figura 4 – Mockup da tela de Login - FICAI 4.0**



Fonte: [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design_mockups)

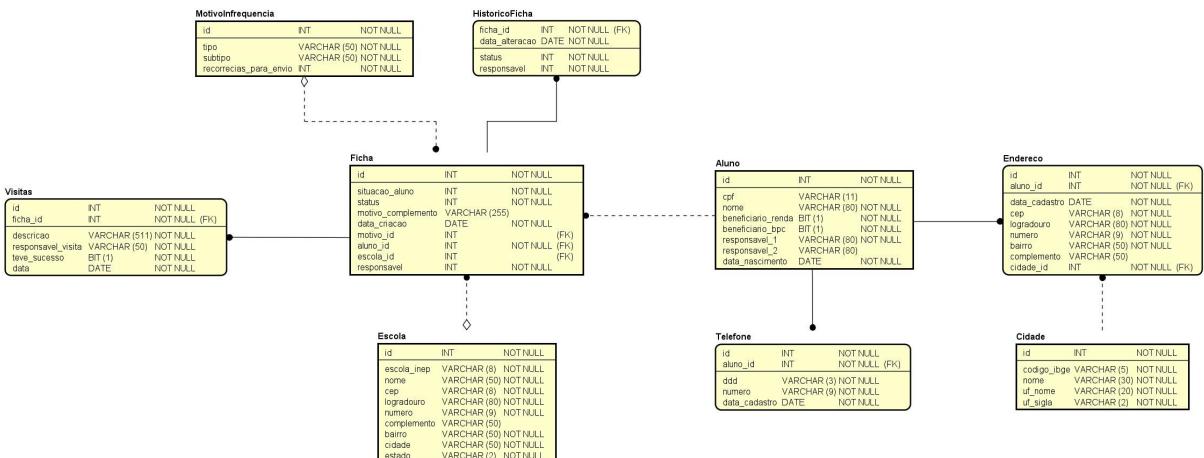
**Figura 5 – Modelo conceitual do banco de dados - FICAI 4.0.**



Fonte: [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/bancoz\\_dados](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/bancoz_dados)

A partir do modelo conceitual, foi desenvolvido o modelo lógico (Figura 6), que detalhou as entidades que seriam efetivamente criadas no banco de dados, seus relacionamentos e os tipos de dados correspondentes a cada atributo.

**Figura 6 – Modelo lógico do banco de dados - FICAI 4.0**



Fonte: [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/bancoz\\_dados](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/bancoz_dados)

Por fim, foram realizadas as configurações iniciais do ambiente de desenvolvimento, permitindo o início efetivo das atividades da Sprint 1. Durante esse processo, também foi prestado suporte aos alunos de nível AGES I, contribuindo para sua integração às atividades da equipe e para o alinhamento das práticas de desenvolvimento adotadas pelo grupo.

### 3.3.2 Sprint 1

Durante a Sprint 1 do projeto *FICAI 4.0*, foram desenvolvidas as telas *Home*, *Buscar Aluno* e *Aluno Não Encontrado*. Além disso, também foi possível contribuir para a implementação da funcionalidade de busca de alunos no módulo Back-End da aplicação.

O desenvolvimento das telas foi conduzido com base no princípio de componentização, visando garantir modularidade e reutilização de código. Como estratégia, optou-se por criar um conjunto fixo de componentes compartilhados entre as telas, incluindo o cabeçalho (*Header*), a barra de busca, o botão de pesquisa, o botão de parametrização e o botão de cadastro de novos alunos. A partir dessa base comum, foi possível desenvolver de forma independente as funcionalidades específicas de cada tela e, posteriormente, integrá-las ao sistema, assegurando consistência na interface e fluidez na navegação.

Na tela *Home* (ou Tela Inicial), ilustrada na Figura 7, foram inseridas imagens que simulam um painel de controle (*dashboard*) contendo informações que futuramente seriam obtidas do banco de dados, além do botão “Exibir Mais”. Esses elementos foram implementados com o propósito de indicar como os componentes seriam dispostos visualmente nas próximas sprints, quando o sistema estivesse integrado às fontes reais de dados.

Na tela *Buscar Aluno*, foi implementado um sistema de tabela com paginação, conectado ao módulo Back-End por meio de requisições Hypertext Transfer Protocol (HTTP). Dessa forma, ao digitar uma informação na barra de pesquisa, o módulo Back-End retorna os dados correspondentes, enquanto o módulo Front-End organiza os resultados em formato de tabela, exibindo até oito registros por página. Nessa tela, também foi incluído o botão “Cadastrar Novo Aluno”, que redireciona o usuário para o formulário de cadastro.

Já na tela *Aluno Não Encontrado*, foram implementados apenas os elementos essenciais: uma mensagem informando a ausência de resultados e o botão “Cadastrar Novo Aluno”, permitindo ao usuário registrar um novo aluno caso o termo buscado não seja encontrado.

**Figura 7 – Tela Home (ou Tela Inicial) desenvolvida durante a Sprint 1 - FICAI 4.0.**



Fonte: [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design_mockups)

Durante essa sprint, percebi um bom nível de entrosamento entre os membros da equipe. Recebi grande apoio dos colegas de nível AGES IV, especialmente em questões relacionadas às regras de negócio da aplicação, e também pude colaborar com os alunos de nível AGES I em suas primeiras atividades. Essa troca de conhecimento foi fundamental para o avanço coletivo e contribuiu para a integração do grupo como um todo.

### 3.3.3 Sprint 2

Durante a Sprint 2, a equipe de desenvolvimento optou por trabalhar em User Stories (USs) que contemplavam tanto novas funcionalidades a serem incluídas na aplicação quanto atividades classificadas como “débito técnico”. Considerando que, nas Sprints 0 e 1, minha atuação esteve mais voltada ao desenvolvimento dos *mockups* e à implementação de telas no módulo Front-End, nesta sprint decidi contribuir de forma mais direta com o módulo Back-End da aplicação.

Logo no início da sprint, identifiquei um problema relacionado à criação de registros do tipo *Aluno* no banco de dados. Ao inserir um novo aluno, os atributos “aluno\_id” das entidades *Endereço* e *Telefone* estavam sendo armazenados com valores nulos. Diante dessa situação, criamos uma US específica para mapear o problema, definir os critérios de aceite e desenvolver a solução. Para isso, formamos uma *squad* composta por quatro integrantes da equipe e realizamos as correções necessárias no módulo Back-End. Essa solução foi essencial para viabilizar o desenvolvimento da tela de criação de alunos, permitindo o correto relacionamento entre os dados de endereço, telefone e aluno.

Durante o andamento da sprint, também foram identificados outros pontos de melhoria na aplicação. Um deles consistia na adição de um campo de observação na tela de “Histórico de Endereços”, implementado como um componente do tipo *Text Area* (do framework Angular (ANGULAR, 2025)), a fim de possibilitar a visualização consolidada de todas as informações referentes ao endereço de um aluno em um único campo da UI.

Outro ajuste relevante foi apontado pelo professor orientador, que sugeriu a implementação de um processo de sanitização dos dados no módulo Front-End, com o objetivo de evitar vulnerabilidades, como ataques de Cross-Site Scripting (XSS). Atuei diretamente nessa atividade, implementando validações e sanitizações em todos os campos de inserção de texto, incluindo os campos de busca e os formulários de cadastro de alunos. O processo envolveu a remoção de espaços em branco desnecessários, a exclusão de símbolos inválidos e a verificação do tipo de dado inserido, garantindo maior segurança e integridade das informações.

Apesar dos avanços alcançados, observei que o ritmo de desenvolvimento da equipe diminuiu em comparação com a sprint anterior. Essa redução de produtividade resultou no acúmulo de tarefas classificadas como débito técnico, que precisariam ser retomadas nas sprints seguintes. Ainda assim, a sprint representou um avanço importante na consolidação da estrutura do sistema e no amadurecimento técnico do grupo.

### 3.3.4 Sprint 3

Durante a Sprint 3, o desempenho geral da equipe permaneceu abaixo do esperado, o que impactou o cumprimento integral das atividades planejadas. Ainda assim, foi possível avançar no desenvolvimento de funcionalidades de grande relevância para a aplicação, consolidando pontos importantes da estrutura do sistema.

Nesta sprint, dei continuidade ao processo de sanitização de dados iniciado anteriormente. Com as implementações do módulo Front-End já concluídas na Sprint 2, iniciei a etapa de sanitização no módulo Back-End. Esse processo envolveu a criação de validações para os dados recebidos pelo servidor, incluindo a remoção de espaços em branco desnecessários e a verificação da presença de tags Linguagem de Marcação de Hipertexto (HTML) ou comandos Structured Query Language (SQL) em campos de entrada. Essa etapa me proporcionou um aprendizado significativo sobre segurança em aplicações web, especialmente sobre práticas de prevenção a ataques XSS, tentativas de injeção de código e manipulação indevida de informações sensíveis.

Paralelamente, também participei do processo de revisão e ajuste do modelo lógico do banco de dados, em colaboração com outros colegas de nível AGES II. Essa atividade reforçou a importância dos conceitos aprendidos na disciplina de Modelagem de Banco de Dados, pois permitiu aplicar na prática os conhecimentos adquiridos em sala de aula. Durante essa revisão, foram definidos os tipos de dados de cada atributo das entidades e realizada a exclusão da entidade “Usuário”, que havia se tornado desnecessária diante das mudanças de requisitos solicitadas pelas stakeholders.

Durante esta sprint, contei com o apoio constante dos colegas de níveis AGES II e III, o que facilitou o desenvolvimento das minhas atividades. Apesar do ritmo de trabalho ainda estar aquém do ideal, foi possível notar uma leve melhora em comparação à Sprint 2. Nesse contexto, a atuação dos alunos de nível AGES IV foi essencial para estimular o engajamento do grupo, promovendo uma comunicação mais efetiva e um ambiente de colaboração que preparou a equipe para retomar o projeto com mais motivação nas etapas seguintes.

### 3.3.5 Sprint 4

Na última sprint do projeto, foram implementadas diversas funcionalidades de grande relevância para o sistema, consolidando o trabalho desenvolvido ao longo do semestre.

Inicialmente, atuei no desenvolvimento do *dashboard* da tela inicial (*Home*), conforme ilustrado na Figura 8. Nessa etapa, participei tanto da implementação dos componentes gráficos no módulo Front-End, utilizando o framework Angular (ANGULAR, 2025), quanto da integração com o módulo Back-End para a consulta e exibição dos dados armazenados. Essa atividade foi importante para unir as partes visuais e lógicas da aplicação, resultando em uma tela inicial mais dinâmica e informativa.

**Figura 8** – Tela Home completa desenvolvida durante a Sprint 4 - FICAI 4.0.



Fonte: [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design_mockups)

Também atuei nas atividades relacionadas ao fluxo de ações na FICAI, ou seja, ações que seriam realizadas nas fichas de cada aluno cadastrado na aplicação. Essas ações seriam realizadas através de botões específicos incluídos na ficha de um determinado aluno cadastrado, como podem ser visualizados na Figura 9. Pude auxiliar no desenvolvimento das ações Arquivar, Adicionar Contato, Atuar na Ficai, Compartilhar com Conselho Tutelar, Compartilhar com Ministério Público e Compartilhar com rede. Durante o desenvolvimento dessas atividades, pude contar com o auxílio de muitos colegas, principalmente os colegas AGES I e IV. Nessas atividades pude desenvolver tanto no módulo Front-End quanto no módulo Back-End da aplicação.

**Figura 9 – Tela completa de Edição do Aluno desenvolvida durante a Sprint 4 - FICAI 4.0.**

The screenshot displays the 'FICAI' application interface. At the top, there's a header with the 'FICAI' logo and navigation links for 'Voltar' (Back), 'Cadastrar FICAI', and 'Sair' (Logout). Below the header, the first panel is titled 'Dados do Aluno' (Student Data) and contains fields for Name, CPF, Date of Birth, Phone, Responsible by phone, Address, and Beneficiaries. The second panel is titled 'FICAI do Aluno #1000' (Student FICAI #1000) and shows details like Motivation of Situation (Suspected domestic violence), Responsible (Education Council), and Status (Waiting for Visit). Buttons for saving changes and sharing the record are also present.

Fonte: [https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/ficai-4.0/ficai-4.0-wiki/-/wikis/design_mockups)

Próximo ao término da sprint, solicitei autorização aos colegas de nível AGES IV para auxiliar o grupo de AGES III nas atividades de versionamento e integração de novas funcionalidades ao sistema. Essa colaboração foi motivada pela alta velocidade de desenvolvimento do time, que estava gerando conflitos frequentes nas versões do código. Nessa etapa, atuei na resolução de conflitos entre *branches*, na validação de novas alterações e, quando necessário, na reversão de modificações que poderiam comprometer a estabilidade da aplicação. Essa experiência reforçou a importância da disciplina de Gerenciamento de Configuração de Software, permitindo aplicar na prática os conceitos de controle de versões e integração contínua de forma eficiente.

Ao final da Sprint 4, percebi um ambiente de grande motivação e satisfação entre os membros da equipe. A entrega das funcionalidades planejadas representou o amadurecimento coletivo do grupo e consolidou o aprendizado técnico adquirido ao longo do projeto. Pude também reconhecer o quanto evoluí na comunicação e colaboração com os colegas, especialmente com os alunos de níveis AGES I, II, III e IV, o que tornou essa sprint um marco positivo no encerramento da AGES II.

### 3.4 Conclusão

Ao concluir o projeto *FICAI 4.0*, pude refletir sobre minha trajetória ao longo do semestre e perceber uma evolução significativa, tanto técnica quanto pessoal. Meu desempenho variou em diferentes momentos do projeto, mas de forma geral consegui desempenhar com comprometimento e qualidade as atividades atribuídas ao meu papel como aluno de nível AGES II. Desde o início, busquei compreender profundamente o escopo do sistema, o que contribuiu para o desenvolvimento dos modelos conceitual e lógico do banco de dados e para a criação dos primeiros *mockups*. Os estudos dirigidos realizados na Sprint 0 também foram fundamentais para minha adaptação às tecnologias utilizadas, permitindo que eu participasse de forma mais ativa na construção das telas iniciais da aplicação durante a Sprint 1.

Durante o desenvolvimento das atividades, pude aplicar conhecimentos adquiridos em disciplinas essenciais do curso, especialmente Modelagem de Banco de Dados e Gerenciamento de Configuração de Software. Esses conteúdos foram diretamente utilizados na modelagem do banco de dados, na organização das entidades e relacionamentos e, posteriormente, no controle de versionamento e na manutenção do código-fonte. Além disso, a experiência prática proporcionada pelo projeto permitiu consolidar conceitos sobre arquitetura de sistemas e boas práticas de desenvolvimento.

Nas Sprints 2 e 3, percebi uma queda momentânea no meu rendimento devido à limitação de tempo, causada por outras demandas externas à universidade. Apesar disso, essas etapas representaram um período de aprendizado intenso, especialmente no que diz respeito à segurança de aplicações web. Ao implementar processos de sanitização de dados tanto no Front-End quanto no Back-End, aprendi a importância de práticas voltadas à prevenção de vulnerabilidades, como ataques de XSS e injeções de código. Essa vivência reforçou a relevância dos testes, da verificação de entradas e da validação de dados para o desenvolvimento seguro e confiável de sistemas. A partir dessas dificuldades, desenvolvi uma melhor capacidade de organização pessoal e gestão do tempo, o que foi essencial para retomar um desempenho consistente na última sprint.

Na Sprint 4, consegui aplicar de forma integrada diversos conhecimentos adquiridos ao longo do curso e das sprints anteriores. Atuei no desenvolvimento do *dashboard* principal da aplicação, bem como nas ações vinculadas às fichas de alunos,

envolvendo a comunicação entre os módulos Front-End e Back-End. Além disso, participei ativamente no processo de versionamento e resolução de conflitos de código, aplicando os conceitos aprendidos na disciplina de Gerenciamento de Configuração de Software. Essa experiência prática consolidou minha compreensão sobre integração contínua, controle de versões e boas práticas de trabalho colaborativo em equipes de desenvolvimento.

Comparando este projeto com o anterior, percebo avanços importantes em aspectos que fazem com que este projeto seja considerado por mim melhor que o anterior, sendo alguns deles: 1) comprometimento da equipe; 2) comunicação; 3) foco no resultado. Ainda assim, reconheço que alguns pontos, como a constância nas entregas e a organização das atividades, poderiam ter sido aprimorados, o que resultou em uma baixa qualidade no desenvolvimento da aplicação em alguns momentos. Um aprendizado marcante foi compreender a importância de equilibrar o ritmo de trabalho entre todos os módulos da aplicação, garantindo que todos os integrantes da equipe possam contribuir de forma efetiva. Percebeu-se que as atividades relacionadas ao módulo Front-End tiveram um avanço muito maior do que as tarefas do módulo Back-End nas primeiras Sprints, e que alguns colegas acabaram não conseguindo participar de forma adequada de algumas atividades, seja por indisponibilidade de tempo, seja por falta de conhecimento de tecnologias.. Essa percepção reforçou a necessidade de estimular maior integração entre os membros, especialmente nos estágios iniciais do projeto.

Por fim, o resultado entregue às stakeholders superou nossas expectativas. A aplicação final apresentou um nível de qualidade e usabilidade elevado, refletindo o empenho coletivo do grupo. O feedback constante das stakeholders foi fundamental para orientar decisões técnicas e reforçar o propósito social do sistema, voltado ao combate à evasão escolar. Sinto-me extremamente satisfeito com meu desempenho e orgulhoso de ter participado do projeto *FICAI 4.0*, que não apenas ampliou minhas habilidades técnicas, mas também fortaleceu minha capacidade de comunicação, colaboração e liderança em equipes de desenvolvimento.

## 4 AGES III — “POLYMATHECH 2024”

Esta seção apresenta a atuação do aluno como integrante de nível AGES III no projeto *Polymathech*. São descritos os artefatos desenvolvidos, as atividades realizadas ao longo das sprints e os principais pontos de melhoria identificados durante o projeto.

### 4.1 Introdução

O projeto *Polymathech* surgiu da necessidade de enfrentar uma problemática que afeta milhares de jovens em todo o Brasil: a dificuldade na escolha da carreira profissional a seguir. Essa decisão, muitas vezes tomada sem orientação adequada, tem impactado diretamente os índices de permanência e conclusão no Ensino Superior.

A partir da análise de dados disponibilizados pelo O Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), foi possível identificar que, na última década, apenas cerca de 30% dos estudantes matriculados em Instituições de Ensino Superior concluíram sua graduação, enquanto mais da metade desistiu do curso ou migrou para outra área. Esses dados evidenciam a importância de oferecer uma orientação vocacional mais acessível e eficaz, especialmente voltada ao público jovem, para auxiliá-lo na definição de caminhos profissionais de acordo com suas aptidões e interesses.

Com base nesse contexto, o projeto *Polymathech* foi idealizado com o objetivo de desenvolver uma plataforma web voltada à orientação profissional de jovens de forma atrativa, realista e interativa. A aplicação busca combinar uma linguagem clara e próxima da juventude com recursos tecnológicos que permitam a realização de testes vocacionais, o acesso a conteúdos educacionais e um ambiente de estudo dinâmico e acolhedor. Dessa forma, o sistema pretende contribuir para a redução da evasão universitária, promovendo uma escolha de carreira mais consciente e assertiva.

O projeto teve seu período de execução entre os meses de março e julho de 2024 e contou com a participação dos stakeholders Vinícius da Rosa Soares e Arthur Battistel Ilha, da professora orientadora Cristina Nunes e de uma equipe composta por 15 alunos dos cursos de Engenharia de Software e Sistemas de Informação da PUCRS, conforme ilustrado na Figura 10.

**Figura 10 – Equipe de desenvolvimento e Stakeholder - Polymathech**



Fonte: <https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/home>

## 4.2 Desenvolvimento do Projeto

Nesta seção são apresentados os principais artefatos desenvolvidos ao longo do semestre de execução do projeto *Polymathech*. São descritas as etapas realizadas para as entregas, acompanhadas de ilustrações e links para a Wiki do projeto, onde é possível consultar informações mais detalhadas sobre cada item.

### 4.2.1 Repositório do Código Fonte do Projeto

O código-fonte do projeto *Polymathech* foi organizado em dois repositórios principais, sendo um destinado ao módulo Back-End, responsável pela API da aplicação, e outro ao módulo Front-End, responsável pela UI. Além disso, a Wiki do projeto contém informações adicionais sobre a estrutura, dependências e procedimentos de configuração utilizados. Os repositórios podem ser acessados nos links abaixo:

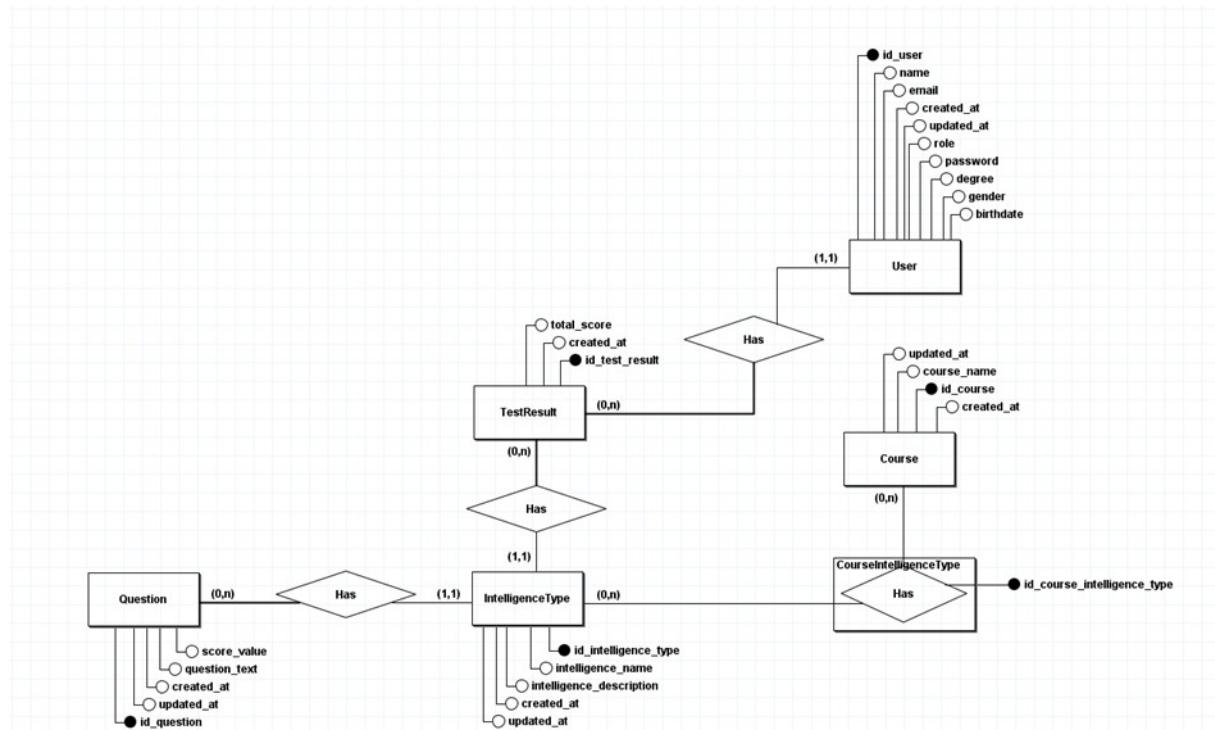
- <https://tools.ages.pucrs.br/polymathech/polymathech-backend>
- <https://tools.ages.pucrs.br/polymathech/polymathech-frontend>
- <https://tools.ages.pucrs.br/polymathech/wiki>

#### 4.2.2 Banco de Dados Utilizado

Neste projeto, optamos por utilizar uma estrutura de banco de dados relacional SQL, pois foi identificado entre os membros da equipe de desenvolvimento que esse tipo de estrutura seria mais adequado ao gerenciamento dos dados que precisariam ser relacionados na aplicação. Foram levantados os requisitos necessários para o desenvolvimento da estrutura de dados da aplicação e, assim, foi possível realizar a modelagem Entidade Relacionamento (ER), como indicado na Figura 11, e a modelagem lógica, indicada na Figura 12. As informações sobre a modelagem do banco de dados e sobre sua implementação podem ser acessadas pelo seguinte link:

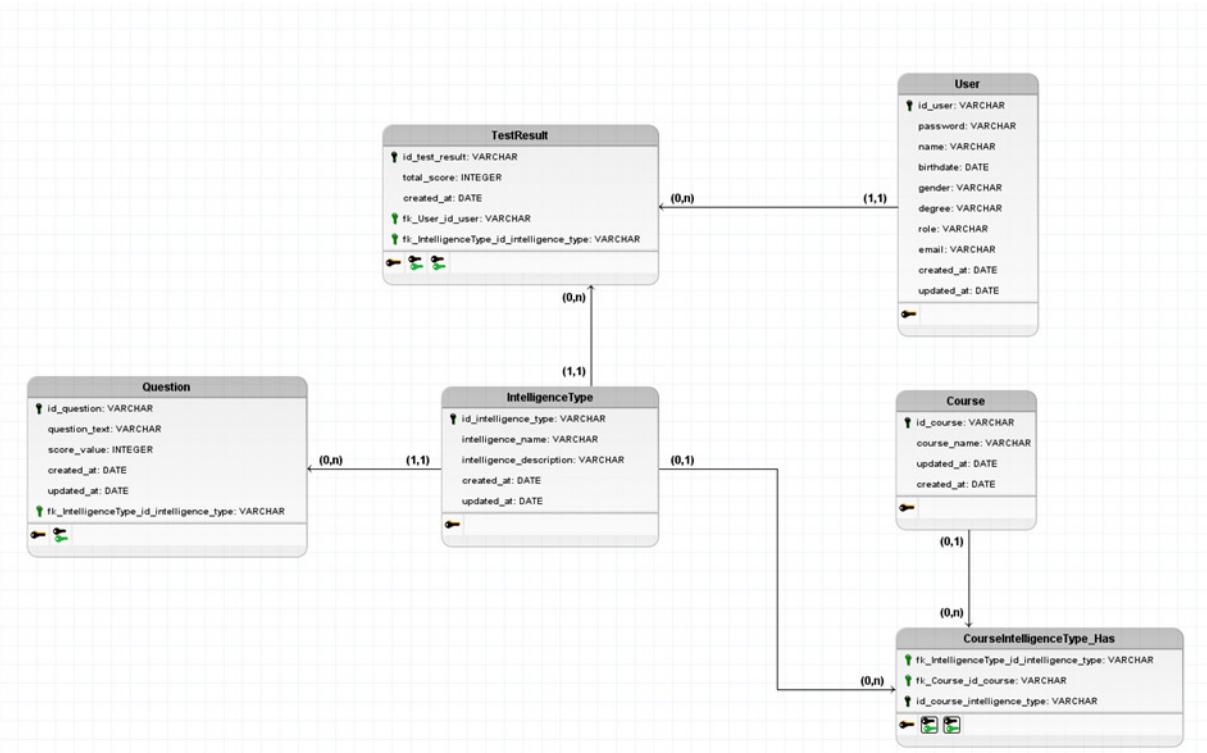
- <https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/Banco%20de%20Dados>

**Figura 11 – Modelo ER do banco de dados - Polymathech**



Fonte: <https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/Banco%20de%20Dados>

**Figura 12 – Modelo Lógico do banco de dados - Polymathech**



Fonte: <https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/Banco%20de%20Dados>

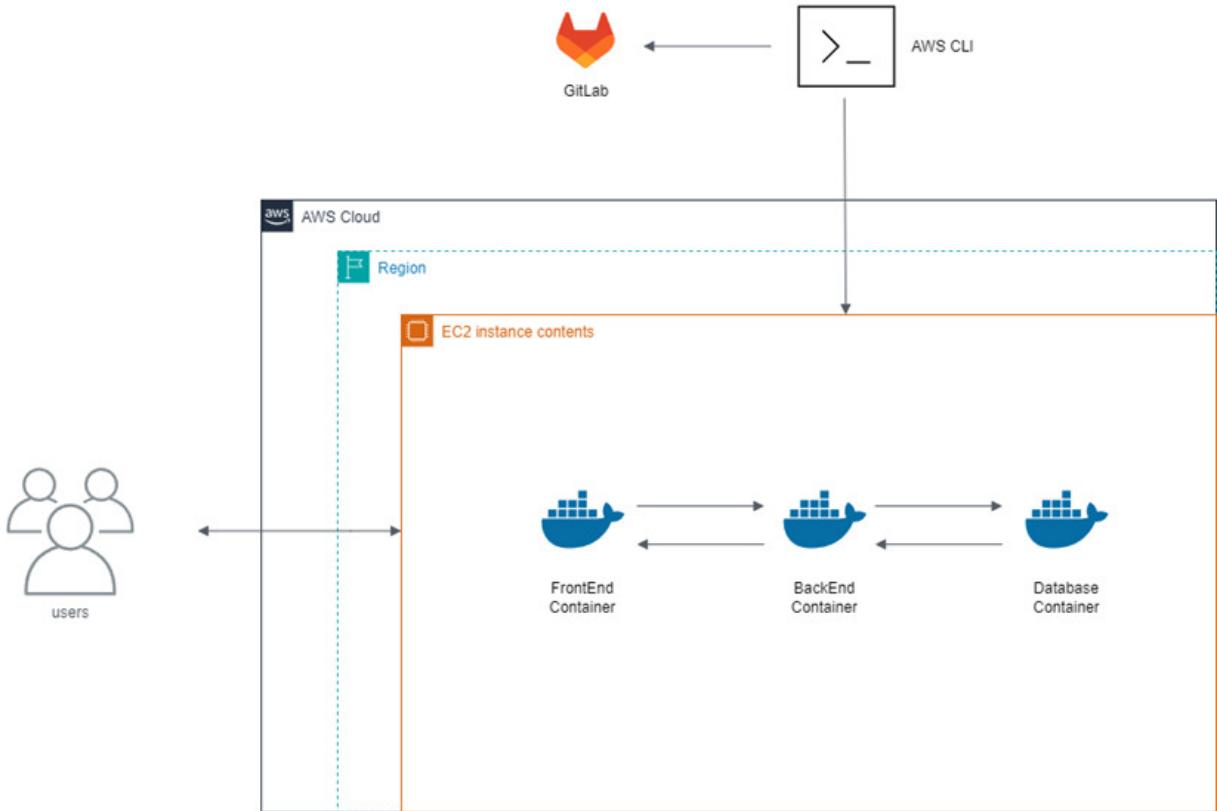
#### 4.2.3 Arquitetura Utilizada

A arquitetura do projeto foi definida de forma a garantir modularidade, escalabilidade e facilidade de manutenção. Foram estruturados três componentes principais: o módulo Front-End, o módulo Back-End e o processo de deploy na AWS.

No módulo Front-End, adotou-se uma organização baseada em componentização e páginas, seguindo boas práticas de desenvolvimento com React. Já o Back-End foi estruturado em camadas, conforme os princípios da *Clean Architecture* (MARTIN, 2017) e do DDD, a fim de separar regras de negócio, lógica de infraestrutura e controladores da aplicação. O deploy foi realizado manualmente na AWS, por meio da configuração de uma instância Elastic Compute Cloud (EC2). A aplicação foi implantada em containers Docker (DOCKER, 2025), configurados via CLI e Secure Shell (SSH), conforme ilustrado na Figura 13. Essas decisões foram tomadas considerando a complexidade do sistema e o tempo disponível para desenvolvimento. As informações completas sobre a arquitetura estão disponíveis na Wiki do projeto:

- <https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/arquitetura>

**Figura 13 – Arquitetura de deploy da aplicação na AWS - Polymathech**



Fonte: <https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/arquitetura>

#### 4.2.4 Protótipos das Telas Desenvolvidas

No projeto *Polymathech*, optou-se por iniciar diretamente o desenvolvimento dos protótipos em alto nível. Essa decisão surgiu a partir de discussões entre os stakeholders e a equipe de desenvolvimento, que definiram como diretriz a adoção de um padrão visual semelhante ao de outras aplicações web existentes. A partir desse ponto, foram elaboradas as telas e fluxos da aplicação com base em referências modernas, adicionando funcionalidades específicas do projeto e um design próprio, atrativo e acessível.

Os principais protótipos, que representam as etapas de interação do usuário, estão apresentados nas Figuras 14 a 19. As versões completas podem ser consultadas na Wiki do projeto ou no Apêndice A deste relatório:

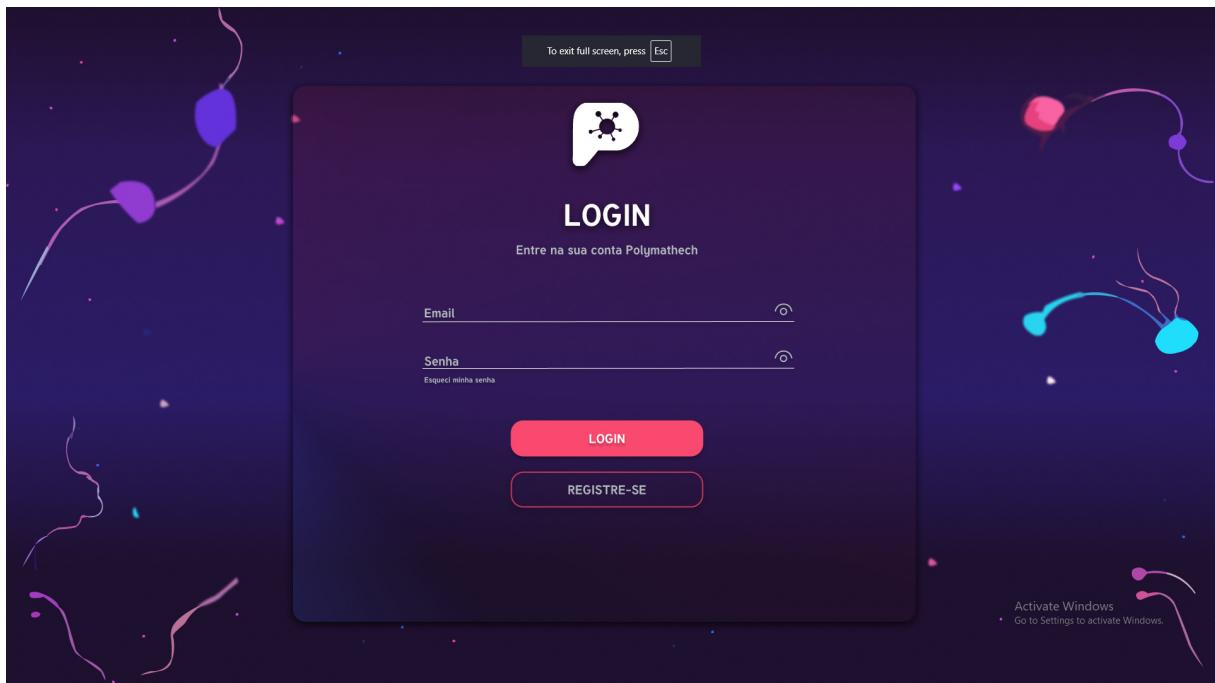
- [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

**Figura 14 – Tela inicial - Polymathech**



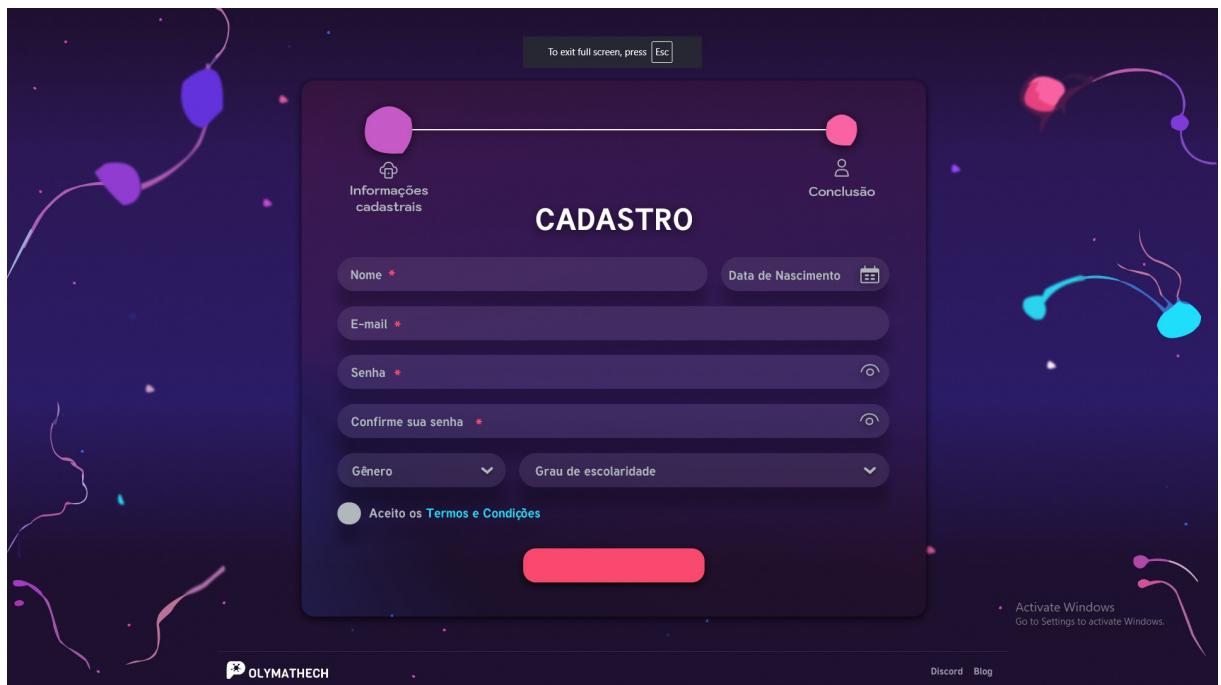
Fonte: [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

**Figura 15 – Tela de login - Polymathech**



Fonte: [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

**Figura 16 – Tela de cadastro - Polymathech**



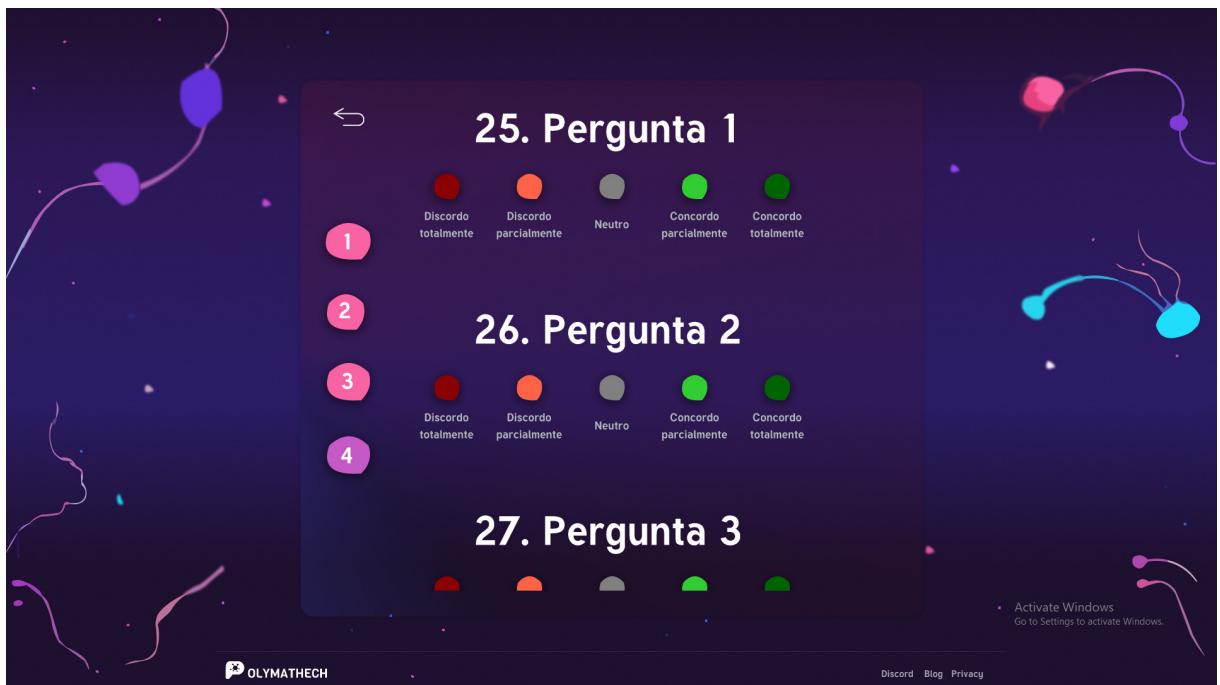
Fonte: [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

**Figura 17 – Tela home - Polymathech**



Fonte: [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

**Figura 18 – Tela de teste vocacional - Polymathech**



Fonte: [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

**Figura 19 – Tela de resultados do teste vocacional - Polymathech**



Fonte: [https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design\\_mockups](https://tools.ages.pucrs.br/polymathech/wiki/-/wikis/design_mockups)

#### 4.2.5 Tecnologias Utilizadas

As tecnologias utilizadas no projeto foram selecionadas de modo a otimizar o processo de desenvolvimento, reduzir a curva de aprendizado e garantir um ambiente seguro e bem estruturado.

No módulo Front-End, foi utilizado o framework ReactJS (REACTJS, 2025), em conjunto com a linguagem TypeScript (TYPESCRIPT, 2025), devido à ampla documentação, suporte da comunidade e facilidade de componentização. Para o módulo Back-End, adotou-se o framework NestJS (NESTJS, 2025), também desenvolvido em TypeScript, por ser uma ferramenta opinativa que favorece a organização do código e a aplicação de boas práticas arquiteturais.

O banco de dados escolhido foi o PostgreSQL (POSTGRESQL, 2025), por sua robustez, confiabilidade e custo computacional reduzido. A camada de acesso e manipulação de dados foi implementada utilizando o Prisma ORM, que facilitou o controle de versões do esquema, a execução de migrações e a realização de consultas de forma tipada e eficiente.

### 4.3 Atividades Desempenhadas Pelo Aluno no Projeto

#### 4.3.1 Sprint 0

Durante a primeira sprint do projeto *Polymathech*, atuei no desenvolvimento de diversas atividades fundamentais para a estruturação inicial da aplicação, entre elas: estudos dirigidos, organização dos repositórios, documentação e configuração dos módulos Front-End e Back-End, definição dos processos de versionamento de código, elaboração das arquiteturas de cada módulo, apoio na criação das primeiras US e suporte aos colegas dos níveis AGES I e AGES II.

Na etapa inicial, realizei estudos dirigidos com o objetivo de compreender quais modelos de arquitetura e tecnologias seriam mais adequados ao desenvolvimento do projeto. Esses estudos foram realizados por meio de vídeos, cursos rápidos e materiais disponibilizados nas disciplinas do curso de Engenharia de Software da PUCRS. Com base nesse aprendizado, iniciei a estruturação dos módulos Front-End e Back-End, configurando seus diretórios e dependências principais.

Paralelamente, organizei os repositórios do projeto, concedendo acesso aos mem-

bros da equipe, definindo o fluxo de versionamento, papéis e responsabilidades, além de documentar os procedimentos de configuração de ambiente. Essa documentação teve como objetivo facilitar a integração dos integrantes e garantir que todos pudessem configurar e executar os módulos localmente de forma padronizada e eficiente.

Como etapa complementar da sprint, e considerando que parte das minhas atribuições como AGES III já havia sido concluída, pude contribuir também no desenvolvimento dos protótipos da aplicação. Atuei na padronização de componentes e dimensões de tela, além de incluir fluxos automatizados de navegação entre as páginas no *Figma* (FIGMA, 2025). Essa funcionalidade foi importante para proporcionar aos stakeholders e aos demais membros da equipe uma visão mais realista e interativa da aplicação em sua versão final.

#### 4.3.2 Sprint 1

Durante a Sprint 1, apresentei ao time a estrutura inicial dos módulos Front-End e Back-End do projeto, em uma reunião remota que teve como objetivo alinhar o entendimento sobre a arquitetura proposta e esclarecer dúvidas técnicas. Esse momento foi essencial para que a equipe pudesse propor melhorias e identificar oportunidades de otimização no desenvolvimento da aplicação.

Com a divisão da equipe em *Squads* — pequenos grupos responsáveis por funcionalidades específicas — passei a atuar na implementação das rotas de cadastro de usuário no módulo Back-End. Nessa etapa, contribuí na criação das rotas da API, na configuração da conexão com o banco de dados PostgreSQL (POSTGRESQL, 2025), na execução das migrações e na aplicação de mecanismos de validação e segurança dos dados, como a utilização de senhas criptografadas.

O rápido progresso desse módulo permitiu validar as entregas junto aos alunos AGES IV. Após a validação, a equipe foi redistribuída para apoiar outras *Squads*. Nessa nova etapa, atuei novamente no Back-End, desta vez no desenvolvimento das funcionalidades de login e autorização de usuários. Implementamos o sistema de autenticação com tokens JSON Web Token (JWT), que garante maior segurança ao permitir a troca de informações criptografadas entre os módulos Front-End e Back-End, utilizando chaves públicas e privadas para validação. Além disso, foi implementada a verificação de permissões de acesso com base no papel do usuário, restringindo

determinadas rotas conforme o tipo de perfil (usuário comum ou administrador).

Com a conclusão dessas atividades, realizamos uma nova rodada de validações junto aos AGES IV. A partir daí, passei a apoiar o desenvolvimento no módulo Front-End, colaborando na correção de falhas e na integração entre os módulos. Com os dois ambientes devidamente integrados, executamos testes de ponta a ponta (*End-to-End*) para identificar pendências e garantir o correto funcionamento das funcionalidades implementadas.

#### 4.3.3 Sprint 2

Durante a Sprint 2, após a reorganização dos alunos em suas respectivas *Squads*, meu primeiro objetivo foi repassar conhecimentos técnicos aos colegas que passaram a ser responsáveis pelo desenvolvimento de novas funcionalidades no módulo Back-End da aplicação. Essa etapa inicial foi importante para alinhar práticas de desenvolvimento e garantir que todos tivessem domínio sobre as ferramentas e padrões utilizados.

Concluída essa fase, concentrei meus esforços nas atividades diretamente sob minha responsabilidade, que consistiram em: (1) desenvolver novas rotas no Back-End para a geração dos resultados dos testes vocacionais, e (2) realizar a integração dessas funcionalidades com o módulo Front-End.

Como primeiro passo, optamos por atualizar a modelagem do banco de dados, com o intuito de otimizar as consultas necessárias para o processamento dos resultados dos testes. Assim, elaboramos um novo modelo ER e seu correspondente modelo lógico, incluindo as entidades, atributos e relacionamentos essenciais para o funcionamento completo da aplicação. Nessa etapa, também foram atualizados os scripts de migração de dados no Prisma ORM (PRISMA ORM, 2025), garantindo a criação adequada das estruturas no banco de dados PostgreSQL (POSTGRESQL, 2025).

Por fim, realizei a integração das novas rotas do Back-End com o Front-End, permitindo a geração e exibição dinâmica dos resultados dos testes vocacionais. Embora a sprint tenha resultado em entregas relevantes, observou-se que a comunicação entre alguns membros da equipe foi limitada em determinados momentos, o que ocasionou um acúmulo de tarefas próximo ao encerramento da sprint.

#### 4.3.4 Sprint 3

Durante a Sprint 3, muitas das funcionalidades planejadas para o projeto já se encontravam praticamente concluídas, resultado do elevado desempenho da equipe na sprint anterior. Assim, as principais demandas concentraram-se no desenvolvimento e ajuste de novas páginas no módulo Front-End e na configuração do processo de deploy da aplicação na AWS.

Meu principal objetivo nesta sprint foi iniciar o desenvolvimento das configurações necessárias para o deploy da aplicação. Juntamente com os membros da minha *Squad*, configurei o uso de containers Docker (DOCKER, 2025) tanto para o módulo API quanto para o módulo UI, preparando o ambiente para execução integrada dos serviços.

Durante o desenvolvimento dessas configurações, enfrentamos alguns desafios técnicos, como a comunicação entre containers e o gerenciamento adequado das variáveis de ambiente. Além disso, dediquei parte significativa do tempo à revisão de *Merge Requests* criados para os módulos Front-End e Back-End, o que acabou reduzindo o tempo disponível para o avanço das configurações de deploy.

Outro ponto relevante desta sprint foi o avanço na criação de testes unitários para as rotas do Back-End, bem como a configuração de um *pipeline* no GitLab, responsável por executar o *build* do módulo Back-End a cada novo *Merge Request*. Essa automação contribuiu para o aumento da qualidade do código e para a detecção precoce de erros durante o desenvolvimento.

Como reflexão final, identifiquei que um dos principais pontos a melhorar nesta sprint foi a baixa delegação de tarefas. Dediquei uma parcela considerável do tempo a revisões de código que poderiam ter sido redistribuídas entre os demais membros da equipe, o que teria permitido maior foco no avanço do processo de deploy e no refinamento das configurações da aplicação.

#### 4.3.5 Sprint 4

Ao iniciar a última sprint do projeto, a aplicação já se encontrava praticamente concluída em relação às funcionalidades previstas para entrega ao final do semestre. Dessa forma, o foco concentrou-se na finalização das funcionalidades em andamento e na melhoria de outras já implementadas, como a geração dos resultados dos testes

vocationais, o envio desses resultados por e-mail aos usuários e a funcionalidade de recuperação de senha.

Com um número reduzido de atividades restantes, foi possível redistribuir as tarefas de forma mais equilibrada entre os membros do time. Assim, pude dedicar-me integralmente à configuração do processo de deploy da aplicação na AWS, uma vez que as etapas básicas de conteinerização já haviam sido concluídas na sprint anterior.

O primeiro passo consistiu na análise dos serviços necessários para a execução da aplicação em ambiente de computação em nuvem. Com o apoio do arquiteto de software da AGES, escolhemos um serviço simples de configurar, mas capaz de atender a todos os requisitos técnicos do sistema. Optamos pela utilização de uma instância EC2 da AWS, com recursos computacionais adequados para garantir o desempenho e a estabilidade da aplicação.

A partir dessa definição, elaboramos o diagrama de deploy e iniciamos as configurações necessárias. O processo envolveu a configuração de rede da instância EC2, com IPs públicos e privados (realizada pelo arquiteto de software), o acesso remoto via SSH, a clonagem dos repositórios do projeto e a execução manual dos containers Docker (DOCKER, 2025). Essa abordagem foi escolhida devido ao tempo limitado disponível para a conclusão do projeto, garantindo que, mesmo com a execução manual das etapas, a aplicação permanecesse estável e acessível a qualquer usuário.

Ao final da sprint, foi possível entregar todas as funcionalidades planejadas, bem como disponibilizar a aplicação completamente configurada e operacional no ambiente da AWS. Observou-se também uma significativa melhora na organização e na comunicação entre os membros da equipe, o que resultou na conclusão do projeto antes mesmo do prazo final estabelecido.

#### 4.4 Conclusão

Ao refletir sobre minha participação neste projeto, percebo que pude contribuir de maneira significativa para a estruturação da aplicação, atuando tanto nas etapas iniciais quanto nas fases finais do desenvolvimento — o que estava diretamente alinhado às minhas responsabilidades como arquiteto de software do projeto. Também participeiativamente da construção do módulo Back-End, da configuração do processo de conteinerização da aplicação e das revisões de código realizadas pelos demais

membros da equipe.

Além das contribuições técnicas, destaco o papel relevante que desempenhei na passagem de conhecimento aos colegas de equipe. Essa troca constante de informações e experiências foi essencial para o avanço coletivo e para a resolução de bloqueios técnicos durante o desenvolvimento. Nesse sentido, percebo que evoluí consideravelmente em competências interpessoais, aprimorando minha capacidade de comunicação, liderança técnica e colaboração em equipe.

Assim como em projetos anteriores, notei a aplicação direta de conhecimentos adquiridos em diversas disciplinas do curso de Engenharia de Software, como Construção de Software, Arquitetura e Desenvolvimento de Software e Fundamentos de Redes de Computadores. Somam-se a isso as experiências adquiridas em minha atuação profissional como engenheiro de software, que contribuíram para uma visão mais prática e assertiva das decisões técnicas tomadas ao longo do projeto.

Apesar de termos concluído com sucesso todas as funcionalidades propostas para o Polymathech, identifiquei um ponto relevante de melhoria relacionado à arquitetura de serviços da AWS. Em alguns momentos, precisei oferecer suporte a outras atividades, o que reduziu o tempo que eu poderia dedicar à implementação completa dos serviços em nuvem. Assim, embora tenhamos conseguido disponibilizar a aplicação funcionalmente na AWS, acredito que poderíamos ter aprimorado a solução ao empregar outros serviços que otimizassem o desempenho geral do sistema — por exemplo, no processamento e renderização de imagens, na gestão eficiente das informações de banco de dados e na melhoria do tempo de resposta da aplicação.

Finalizar o Polymathech consolidou minha compreensão de que assumir um papel técnico mais central — especialmente como arquiteto — envolve, inevitavelmente, equilibrar profundidade técnica com apoio às necessidades do time. Embora eu tenha conseguido garantir estabilidade na arquitetura e auxiliar na evolução do Back-End e do processo de conteinerização, ficou claro que o envolvimento em múltiplas frentes impõe limites reais à dedicação em temas mais complexos, como a arquitetura em nuvem da AWS. Essa percepção reforçou a importância de priorização e de delegação inteligente, competências essenciais para cargos de liderança técnica. No AGES III, mais do que codificar, aprendi a orientar, ouvir e tomar decisões que influenciam diretamente o fluxo de trabalho — um amadurecimento significativo em minha trajetória.

## 5 AGES IV — “AI PRODUTOR 2025”

Esta seção apresenta a atuação do aluno como AGES IV no projeto *Ai Produtor*, descrevendo os artefatos desenvolvidos, as atividades realizadas ao longo das sprints e os principais pontos de melhoria identificados durante o processo.

### 5.1 Introdução

O projeto *Ai Produtor* consiste em um sistema web responsivo desenvolvido com o propósito de otimizar o processo de cadastro e a gestão de produtores de hortifrúti. A solução foi concebida para enfrentar desafios recorrentes do setor agrícola, que ainda depende fortemente de métodos manuais ou de ferramentas pouco eficientes para o registro de informações, o acompanhamento de produtividade e o planejamento de safras.

A plataforma busca oferecer um ambiente centralizado, intuitivo e acessível, capaz de integrar diferentes dimensões do processo produtivo. Entre suas principais funcionalidades destacam-se: o mapeamento georreferenciado de propriedades e áreas de plantio, por meio da integração com a Google Maps API; o controle de safras em cenários de monocultura e policultura; e a geração de relatórios gerenciais que possibilitam não apenas a análise de produtividade, mas também a elaboração de estratégias de planejamento baseadas em dados reais. Dessa forma, o *Ai Produtor* se propõe a ser mais do que um simples sistema de cadastro — tornando-se uma ferramenta de apoio à tomada de decisão para produtores rurais, técnicos agrícolas e gestores do setor.

O time de stakeholders é composto por Cássio, David, Lucca e Guilherme, responsáveis pelo levantamento de requisitos e pela validação das funcionalidades. A condução acadêmica e técnica do projeto é realizada pelo professor orientador Daniel Dalalana, que supervisiona o desenvolvimento e assegura o alinhamento com os objetivos pedagógicos da disciplina. O período de execução do projeto comprehende o intervalo entre agosto de 2025 e novembro de 2025, de acordo com o calendário acadêmico da disciplina AGES IV.

A equipe multidisciplinar de desenvolvimento envolvida no projeto pode ser visualizada na Figura 20, que apresenta o grupo responsável por transformar as demandas do setor hortifrúti em uma solução tecnológica inovadora.

**Figura 20 – Equipe de desenvolvimento do projeto - Ai Produtor**



Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/home>

## 5.2 Desenvolvimento do Projeto

### 5.2.1 Repositório do Código Fonte do Projeto

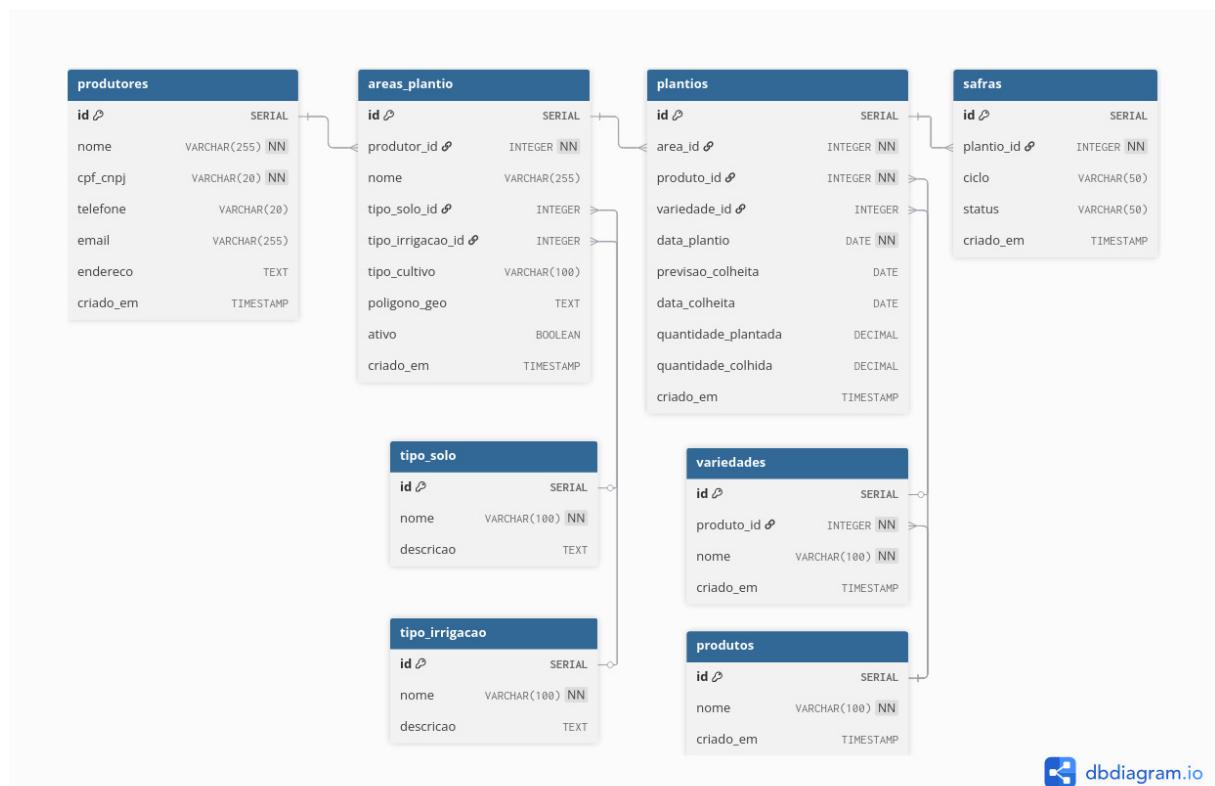
O código-fonte do *Ai Produtor* está organizado em repositórios distintos para Front-End, Back-End e documentação, todos hospedados na plataforma de ferramentas da AGES. Essa separação de responsabilidades favorece um ciclo de desenvolvimento organizado, facilita a manutenção e viabiliza expansões futuras do sistema. Os principais repositórios são:

- **Wiki:** <https://tools.ages.pucrs.br/.../aiprodutor-wiki>
- **Front-End:** <https://tools.ages.pucrs.br/.../AiProdutor-frontend>
- **Back-End:** <https://tools.ages.pucrs.br/.../aiprodutor-backend>

### 5.2.2 Banco de Dados Utilizado

Para o armazenamento e a gestão das informações do *Ai Produtor* utilizou-se o Sistemas de Gerenciamento de Banco de Dados (SGBD) relacional PostgreSQL (POSTGRESQL, 2025), estendido com a extensão PostGIS (POSTGIS, 2025) para suporte a dados geoespaciais — requisito essencial para o mapeamento de áreas de plantio. A modelagem lógica (Figura 21) define os relacionamentos necessários e dá suporte a operações com dados SQL georreferenciados.

**Figura 21 – Modelo lógico do banco de dados - Ai Produtor**



Fonte: [https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/banco\\_dados](https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/banco_dados)

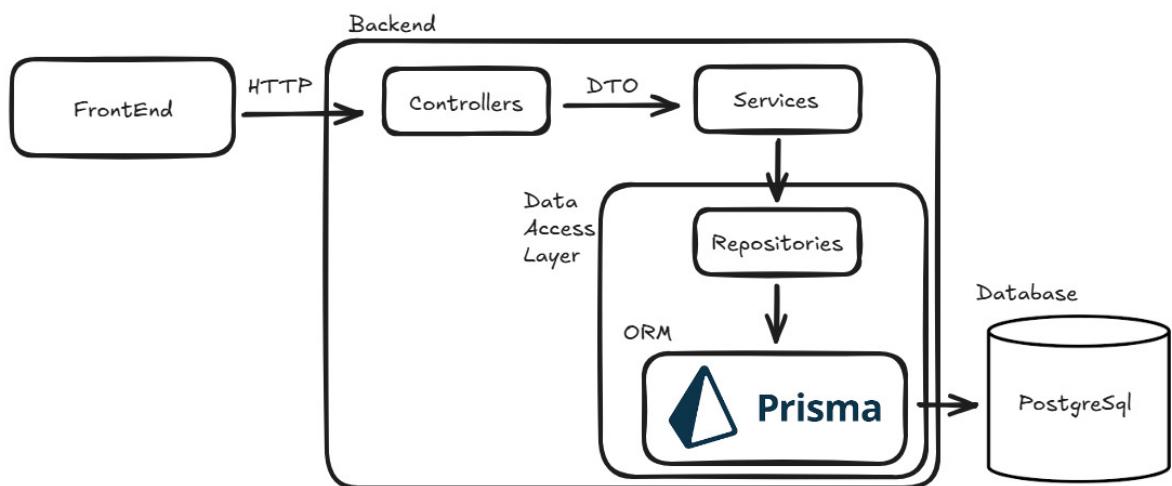
Essa abordagem segue boas práticas tipicamente encontradas em sistemas de informação geográfica e agricultura de precisão, que exigem integridade, escalabilidade e suporte para consultas espaciais eficientes. O PostgreSQL, de acordo com sua documentação oficial, oferece robustez, confiabilidade e uma comunidade ativa de suporte, além de documentação extensa que regula como utilizar índices, tipos de dados especiais e integrações geoespaciais via PostGIS.

### 5.2.3 Arquitetura Utilizada

A arquitetura do Front-End do Ai Produtor foi desenvolvida usando Next.js (VERCEL, 2025) com React, adotando uma estrutura componentizada que facilita a manutenção, o teste e a escalabilidade da interface. Cada componente é isolado em termos de UI, estados de loading e tratamento de erros, o que permite clareza no fluxo visual e modularidade para possíveis expansões. Essa escolha permite reaproveitamento de blocos visuais, facilita debug e prepara terreno para testes unitários ou de integração mais precisos.

No Back-End, utilizou-se NestJS (NESTJS, 2025) sobre Node.js (NODEJS, 2025), em um monólito modular orientado a domínios (por exemplo, *producers*, *areas*, *harvests*). Cada domínio é implementado como um módulo com camadas explícitas — *controller*, *service* e *repository* — e uso de Data Transfer Objects (DTOs) com validação na borda. Essa organização promove *Separation of Concerns*, manutenção facilitada, testabilidade e desenvolvimento paralelo. O esquema geral é apresentado na Figura 22.

**Figura 22 – Arquitetura Back-End do projeto - Ai Produtor**



Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/arquitetura>

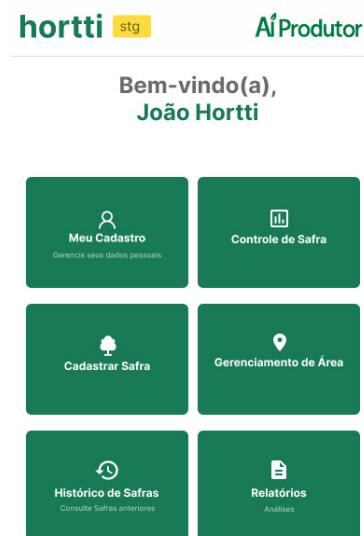
Artigos e comunidades técnicas têm discutido as vantagens de arquiteturas moduladas com NestJS para monólitos limpos, escaláveis e bem organizados, ressaltando que essa abordagem possibilita crescimento do sistema com menor acoplamento e facilidade para evolução futura, sem os custos iniciais elevados de microsserviços.

#### 5.2.4 Protótipos das Telas Desenvolvidas

Os protótipos de interface desenvolvidos no Figma (FIGMA, 2025) foram etapas essenciais do processo de design da aplicação, inseridos no ciclo iterativo de protótipação e validação com os stakeholders. Essa abordagem permitiu avaliar fluxos de navegação, validar requisitos de usabilidade e alinhar a concepção visual à identidade do projeto antes da implementação efetiva. Além disso, contribuiu para reduzir re-trabalho e garantir uma visão compartilhada entre equipe técnica e clientes sobre a experiência do usuário.

A tela inicial (Figura 23) foi pensada para oferecer uma visão geral do sistema, funcionando como ponto de entrada para as demais funcionalidades. O processo de cadastro e acompanhamento de plantios foi estruturado em diferentes etapas: a tela de Novo Plantio (Figura 24) facilita o registro inicial, enquanto o Painel de Controle de Safra (Figura 25) centraliza informações de monitoramento e produtividade.

**Figura 23 – Tela Home - Ai Produtor**



Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifruti/aiprodutor-wiki/-/wikis/mockups>

**Figura 24 – Tela de Novo Plantio - Ai Produtor**

**Novo Plantio**

Áreas

Área total de produção: 0,00 ha

**Selecionar Áreas**

Área Total  
25,00 ha

Data Início \*  
01/08/2025

Data Final \*  
01/08/2026

Insumo  
Insumo

Nome da Operação de Plantio \*  
1º plantio de laranja

Observações:

**Criar**

**Figura 25 – Painel de Controle de Safra - Ai Produtor**

**Painel de Controle**

Informações Gerais

Nº de Áreas  
3

Área Total  
25,00 ha

Cultivar Laranja

Produtividade Esperada  
16.000 kg

Previsão de Colheita: 25/02/2026

Plantado em Sementes  
5.000 un

Preço - Kg  
R\$ 12,40

Despesa Gerada:  
R\$ 62.000,00 ↓

Preço - kg  
R\$ 25,40

Produtividade Realizada  
15.560 kg

Receita esperada:  
R\$ 395.224,00 ↑

**Salvar**

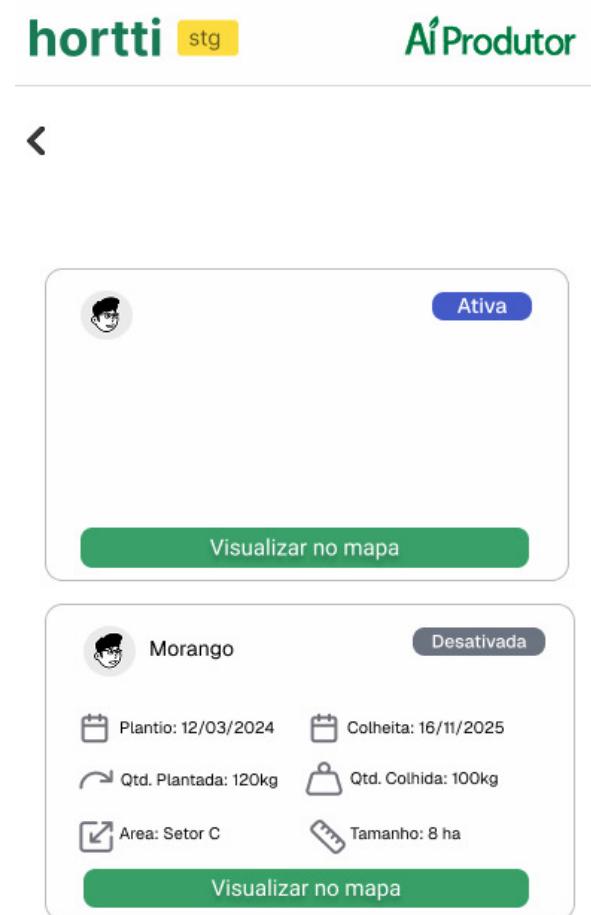
Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/mockups>

Outras telas complementam esse fluxo, como a lista consolidada de safras (Figura 26) e o Histórico de Safras (Figura 27), que permitem a rastreabilidade de ciclos anteriores. A interface também contempla a edição (Figura 28) e a listagem de safras (Figura 29), oferecendo flexibilidade no gerenciamento.

**Figura 26 – Lista de Safras no Painel de Controle - Ai Produtor**



**Figura 27 – Histórico de Safras - Ai Produtor**



Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/mockups>

**Figura 28 – Tela de Edição de Safra - Ai Produtor**

hortti stg

AíProdutor

## Editar Safra

Selecione o tipo de cultivar:

- Monocultura \*
- Policultura \*

Select an item

Laranja do céu

Data Início \* 01/08/2025

Data Final \* 01/08/2026

Nome \* Safra de Laranja 25/26

Áreas  
10,0 ha

Selecionar Áreas

Cancelar Salvar

**Figura 29 – Tela de Listagem de Safras - Ai Produtor**

hortti stg

AíProdutor

## Safras

Nova Safra

Editar Safra

Selecionar uma Safra

Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/mockups>

Complementando a experiência do usuário, as telas de Criação de Safra (Figura 30) e Desenho de Área (Figura 31) permitem representar graficamente as áreas de plantio com o apoio de dados geoespaciais, além de armazenarem informações importantes sobre o tamanho das áreas e sua localização, facilitando a navegação do usuário. A listagem de áreas (Figura 32) e o cadastro de produtores (Figura 33) reforçam a rastreabilidade entre produtores, áreas e safras.

Esses protótipos, no conjunto, demonstram a preocupação do time em oferecer uma interface intuitiva, escalável e aderente às necessidades de gestão agrícola. Eles serviram como guia visual e funcional para os desenvolvedores, além de instrumento de comunicação entre equipe e stakeholders durante todo o ciclo de desenvolvimento.

**Figura 30 – Tela de Criação de Safra - Ai Produtor**

hortti stg

Ai Produtor

< Nova Safra

Selecione o tipo de cultivar:

Monocultura \*

Select an item

Policultura \*

Select an item

Data Início \* 01/08/2025

Data Final \* 01/08/2026

Nome \* Safra de Laranja 25/26

Áreas

Área total de produção: 0,00 ha

Selecionar Áreas

Criar

Cancelar

**Figura 31 – Tela de Desenho de Área de Plantio - Ai Produtor**



Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/mockups>

**Figura 32 – Tela de Listagem de Áreas de Plantio - Ai Produtor**

The screenshot shows a list of planting areas. At the top, there's a header with the Hortti logo and a dropdown menu. Below the header, a green button says '+ Criar nova área' (Create new area). The list contains four items, each in a card-like box:

- Área Sul**: Type of soil: Arenoso, Irrigation type: Superficial, Size: 1 ha (10000m<sup>2</sup>). Buttons: 'Editar' (Edit) and 'Excluir' (Delete).
- Área Norte**: Type of soil: Argiloso, Irrigation type: Aspersão, Size: 2 ha (20000m<sup>2</sup>). Buttons: 'Editar' (Edit) and 'Excluir' (Delete).
- Área Leste**: Type of soil: Terra Roxa, Irrigation type: Aspersão, Size: 1.5 ha (15000m<sup>2</sup>). Buttons: 'Editar' (Edit) and 'Excluir' (Delete).
- Área Oeste**: (No details shown)

**Figura 33 – Tela de Cadastro de Produtor - Ai Produtor**

The screenshot shows a registration form. At the top, there's a header with the Hortti logo and 'Ai Produtor'. Below the header, the form fields are:

- Nome: João Horttirfruti
- CPF/CNPJ: 00.000.000/0000-00
- Telefone: (51) 9 9999-9999
- CEP: 76422-028
- Cidade - UF: Porto Alegre - RS
- CEP: São Caetano
- Rua: Rua João Silva
- Número: 99
- Complemento: Campo

At the bottom, there are two buttons: 'Cancelar' (Cancel) and a large green 'Salvar' (Save) button.

Fonte: <https://tools.ages.pucrs.br/ai-produtor-sistema-de-cadastro-e-gestao-de-produtores-de-hortifrutie/aiprodutor-wiki/-/wikis/mockups>

### 5.2.5 Tecnologias Utilizadas

O projeto Ai Produtor faz uso de um conjunto de tecnologias modernas tanto para o Back-End quanto para o Front-End, a fim de garantir qualidade, desempenho, manutenção e produtividade da equipe. No Back-End, utiliza-se o framework NestJS (NESTJS, 2025) com TypeScript (TYPESCRIPT, 2025), que entrega tipagem estática e modularidade ao código; Prisma como ORM para abstração de persistência e mapeamento objeto-relacional; PostgreSQL (POSTGRESQL, 2025) com extensão PostGIS(POSTGIS, 2025) para lidar com dados geoespaciais; Docker (DOCKER, 2025) para containerização dos ambientes de desenvolvimento e produção; Swagger (SWAGGER, 2025) para documentação automática da API; Passport.js para autenticação via JWT; além de ferramentas para padronização de estilo de código como ESLint e Prettier; e GitLab (GITLAB, 2025) para versionamento, Wiki e integração contínua (CI/CD). No Front-End, destacam-se Next.js, para roteamento e renderização, React (REACTJS, 2025) como biblioteca de interface de usuário, Axios (AXIOS, 2025) para consumo das APIs, Tailwind CSS (TAILWIND CSS, 2025) para estilo utilitário customizado, além de uso de Docker (DOCKER, 2025), ESLint (ESLINT, 2025), e o uso do GitLab para versionamento, documentação e pipelines de CI/CD.

A adoção dessas tecnologias se apoia em princípios de engenharia de software como modularidade, separação de responsabilidades, testabilidade e manutenção de código limpo, conforme discutido em literatura recente sobre desenvolvimento web moderno.

## 5.3 Atividades Desempenhadas Pelo Aluno no Projeto

### 5.3.1 Sprint 0

Na Sprint 0, minhas atividades como AGES IV estiveram voltadas principalmente para dar suporte à equipe e garantir que o projeto tivesse uma base organizacional sólida. Entre as ações previstas estavam o apoio aos AGES I, II e III no início das atividades, a organização das ferramentas de gestão do time, a estruturação inicial do backlog e o alinhamento de requisitos com os stakeholders. Além disso, suportei a documentação da Wiki com informações do projeto, organizei horários para encontros da equipe e auxiliei no suporte técnico e de gestão para a configuração dos repositórios.

Na prática, consegui organizar os alunos no AirTable (AIRTABLE, 2025) e criar um template Kanban no Linear (LINEAR, 2025), o que trouxe mais clareza para a gestão do fluxo de tarefas. Também preparei e divulguei uma tabela de horários para facilitar a marcação de encontros extraclases, além de mapear perguntas para os stakeholders e conduzir o primeiro alinhamento com eles. Outro ponto importante foi o suporte dado aos AGES III na criação e configuração dos repositórios no GitLab, bem como a organização inicial das páginas da Wiki. Participei ainda da criação dos fluxos de telas no Figma e organizei a divisão dos membros em quatro grupos para modelagem dos mockups. Por fim, ofereci apoio direto aos AGES I no alinhamento de expectativas e na orientação para a reunião com os clientes.

Alguns problemas marcaram essa etapa inicial. A comunicação entre os diferentes níveis de AGES foi difícil de alinhar no começo, e a falta de experiência de alguns membros com ferramentas como GitLab e Figma exigiu um suporte maior da minha parte. Além disso, houve desafios em conciliar os horários de todos os participantes para os encontros extraclases e uma incerteza inicial quanto às expectativas dos stakeholders antes da primeira reunião.

Apesar dessas dificuldades, as lições aprendidas foram significativas. Percebi a importância de estruturar e organizar as ferramentas de gestão já na Sprint 0, de modo a garantir fluidez para as próximas etapas. Também ficou claro o valor do alinhamento entre os diferentes níveis de AGES, tanto no aspecto técnico quanto no organizacional, e a necessidade de preparar previamente perguntas e tópicos para reuniões com stakeholders, garantindo objetividade e clareza. Compreendi ainda que o suporte técnico e motivacional aos membros é essencial para manter o ritmo e o engajamento do time.

Como próximos passos, ficou estabelecido que seria necessário estruturar o backlog inicial com base nas informações obtidas após a apresentação dos mockups aos stakeholders, finalizar a organização da Wiki com as primeiras definições do projeto, apoiar os AGES III na evolução da arquitetura do código e da infraestrutura, além de elaborar os primeiros artefatos de gestão, como a Estrutura Analítica do Projeto (EAP)/Release Plan, o Plano de Comunicação e o Plano de Respostas a Riscos.

### 5.3.2 Sprint 1

Na Sprint 1, o foco foi apoiar os AGES II e III no desenvolvimento das atividades técnicas, além de dar continuidade à organização das ferramentas de gestão do time, como o Linear (LINEAR, 2025) e a Wiki do GitLab. Outra frente de trabalho foi a criação de User Stories para o desenvolvimento das funcionalidades previstas, bem como o suporte na criação do diagrama de deploy na AWS.

Durante a execução, atuei na criação e atualização das User Stories voltadas para as rotas de gestão de áreas de plantio no Back-End, ao mesmo tempo em que ofereci suporte às squads de Front-End para a integração com o Back-End. Colaborei também no desenvolvimento de testes unitários e participei ativamente das reuniões de alinhamento com os demais AGES IV. Além disso, estive envolvido na correção de bugs que surgiram ao longo da sprint.

Entretanto, alguns problemas dificultaram a evolução do trabalho. Houve pequenas dificuldades na comunicação entre as squads, somadas à falta de retorno dos membros do Back-End sobre o andamento das tarefas. Também identifiquei baixa adesão da equipe a boas práticas de desenvolvimento, como a realização de commits frequentes, a clareza nas mensagens de commit e a execução de revisões de código. Soma-se a isso a dificuldade recorrente em conciliar os horários da equipe para encontros extraclasse.

As lições aprendidas nesta sprint foram importantes para a melhoria contínua do projeto. Notei a relevância de estruturar as User Stories logo no início da sprint, garantindo clareza no desenvolvimento e alinhamento entre os membros da equipe. Compreendi também a necessidade de reduzir o número de canais de comunicação para evitar ruídos, além de reforçar o valor do alinhamento constante entre os membros da squad. Outro ponto essencial foi a importância de manter atividades em backlog para os membros que terminam suas tarefas antes do previsto, evitando ociosidade e mantendo a produtividade.

Para as próximas etapas, o trabalho seguirá com suporte ao desenvolvimento do Back-End e apoio ao Front-End para integração com o Back-End. Além disso, estão previstos novos alinhamentos com os AGES IV, a correção de bugs, o planejamento da Sprint 2, o suporte no desenvolvimento do deploy na AWS (AWS, 2025) e a migração do código para o GitHub (GITHUB, 2025).

### 5.3.3 Sprint 2

Durante a Sprint 2, minhas atividades como AGES IV concentraram-se no suporte técnico e organizacional às squads, com foco na conclusão de débitos técnicos herdados da Sprint anterior e no avanço de funcionalidades essenciais para o sistema. As principais frentes de trabalho envolveram a criação e revisão de *User Stories* no Linear (LINEAR, 2025), o acompanhamento do desenvolvimento de rotas e correções no Back-End, e o apoio na configuração da infraestrutura em nuvem utilizando a AWS (AWS, 2025).

Entre as atividades executadas, destaco a revisão e aprovação de *Merge Requests* no Back-End, principalmente relacionadas ao Create, Read, Update and Delete (CRUD) de produtos, variedades e plantios, além da verificação de um bug que afetava o gerenciamento de containers e o processo de *seed* do banco de dados. Também realizei ajustes na rota de atualização de plantações e dei suporte à integração final da aplicação para a apresentação da sprint, incluindo a correção de erros de *build* no Front-End. Em paralelo, apoiei a criação das contas e configuração inicial de instâncias EC2 na AWS, parte essencial da preparação para o ambiente de deploy.

Apesar das entregas bem-sucedidas, alguns desafios foram notáveis. A comunicação entre os membros ainda apresentou atrasos, resultando em lentidão na resolução de tarefas e falta de feedback sobre dificuldades técnicas. Essas falhas impactaram parcialmente a agilidade da equipe e reforçaram a necessidade de práticas mais estruturadas de acompanhamento e *follow-up* entre os níveis de AGES.

Como lição, percebi que, embora o grupo já demonstrasse maturidade no fluxo de desenvolvimento, ainda havia lacunas no entendimento do ciclo de informações dentro da aplicação e no uso adequado das ferramentas de versionamento. Reforcei, portanto, a importância de comunicação assertiva, definição clara de prioridades e acompanhamento mais frequente das tarefas no Linear.

Para as próximas etapas, defini como foco a criação de novas *User Stories* para o backlog, o suporte contínuo na configuração dos serviços da AWS, a atualização da Wiki do projeto e a implementação de melhorias voltadas à qualidade e padronização do código, incluindo revisões de estilo e estrutura nos módulos do Back-End e Front-End.

### 5.3.4 Sprint 3

A Sprint 3 foi marcada por um avanço técnico importante, especialmente na consolidação da arquitetura de deploy e na integração dos serviços da AWS (AWS, 2025). Meu papel como AGES IV envolveu liderar o acompanhamento dessas atividades, promover alinhamentos com os demais níveis de AGES e oferecer suporte direto aos AGES III e II em momentos de bloqueio técnico.

Entre as ações realizadas, destaco a criação e atualização de *User Stories* relacionadas à configuração dos serviços de nuvem, a manutenção do diagrama de arquitetura de serviços da AWS e o suporte na integração do *SonarCloud* com os repositórios do projeto no GitHub (GITHUB, 2025). Também participei da revisão de *Merge Requests* nos módulos Back-End e Front-End, garantindo que as implementações seguissem os padrões definidos pelo time e mantivessem a consistência com a arquitetura modular proposta.

Durante essa sprint, percebi uma evolução significativa na organização interna das *squads*. A separação de papéis e responsabilidades tornou a comunicação mais fluida e eficaz, permitindo que as tarefas fossem conduzidas com menor acoplamento e mais autonomia. No entanto, identifiquei oportunidades de melhoria na distribuição de tarefas: alguns membros ficaram momentaneamente sem atividades, o que atrasou parte do cronograma até o realinhamento das demandas.

Também foi notável o crescente uso de modelos de Inteligência Artificial (IA) para a geração de trechos de código. Embora essa prática tenha proporcionado agilidade em algumas implementações, observou-se que o uso excessivo dessas ferramentas resultou em trechos de código mais poluídos e, por vezes, inconsistentes com os padrões estabelecidos pelo projeto. Acredito que o uso de modelos de IA deveria ter sido conduzido com cautela e análise crítica, avaliando não apenas o funcionamento imediato do código gerado, mas também sua qualidade estrutural, legibilidade e aderência às boas práticas de engenharia de software.

Além disso, percebi a necessidade de aprimorar a organização no processo de abertura de *Merge Requests*. Em diversos casos, os títulos não representavam adequadamente o conteúdo das modificações realizadas, e algumas solicitações continham alterações além do escopo definido para determinada *User Story*. Essa falta de padronização impactou o processo de revisão e dificultou a rastreabilidade das alte-

rações, evidenciando a importância de boas práticas de versionamento, padronização de commits e controle de qualidade colaborativo.

As lições aprendidas nesta sprint reforçaram a importância de um planejamento mais granular e de uma gestão de backlog dinâmica, que assegure o envolvimento contínuo de todos os membros. Além disso, a experiência de acompanhar a configuração da infraestrutura em nuvem ampliou minha visão sobre a relação entre arquitetura, deploy e escalabilidade de sistemas.

Como próximos passos, concentrei esforços no suporte às melhorias para a entrega final do sistema, priorizando ajustes de qualidade no código, revisões de arquitetura e o refinamento da documentação técnica na Wiki, com o objetivo de entregar um produto mais estável, padronizado e tecnicamente coeso.

### 5.3.5 Sprint 4

A Sprint 4 representou a etapa final do projeto *Ai Produtor!*, concentrando-se nas entregas conclusivas de funcionalidades e nos ajustes finais da infraestrutura da aplicação. No início da sprint, a maior parte das funcionalidades já havia sido concluída e validada pelos stakeholders, restando apenas a implementação da tela e dos serviços do módulo de *Relatórios*, considerada uma das partes mais estratégicas do sistema por consolidar informações essenciais sobre safras, áreas, produtores e produtividade.

Para dar início a essa etapa, realizamos um alinhamento direto com os stakeholders a fim de detalhar quais análises, métricas e tipos de filtragem deveriam compor a página de relatórios. Com os requisitos devidamente especificados, decidimos dividir novamente a equipe em duas frentes: uma focada no desenvolvimento do módulo Front-End dos relatórios e outra dedicada ao Back-End, responsável pelas consultas, agregações e processamento dos dados provenientes do banco PostgreSQL (POSTGRESQL, 2025). Essa divisão permitiu paralelizar esforços e manter um fluxo de desenvolvimento consistente entre os módulos.

Como AGES IV, atuei tanto no apoio técnico quanto na coordenação dessas frentes, supervisionando o desenvolvimento das novas rotas de consulta, revisando os serviços implementados no NestJS (NESTJS, 2025) e garantindo que o código permanesse aderente às boas práticas de *Clean Code* e *Separation of Concerns*. Também conduzi revisões de nomenclatura, remoção de redundâncias e melhorias de valida-

ção, reforçando a qualidade das entregas antes da integração final ao sistema.

Outra responsabilidade importante nesta sprint foi a finalização da configuração dos serviços na AWS (AWS, 2025). Realizei revisões no ambiente de deploy, ajustes finos na estruturação dos containers Docker (DOCKER, 2025) e acompanhamento da automatização de parte dos processos de *build* e *deploy*. Esse refinamento foi essencial para garantir que a aplicação estivesse acessível, estável e em conformidade com a arquitetura planejada.

Apesar de ser uma sprint relativamente curta e com escopo reduzido, o rendimento da equipe acabou ficando abaixo do esperado. Algumas dificuldades técnicas e de organização de tarefas impactaram o ritmo do desenvolvimento, exigindo realinhamentos no backlog e redistribuição de atividades ao longo da sprint. Ainda assim, conseguimos manter a consistência das entregas e finalizar todas as funcionalidades planejadas dentro do prazo.

Um ponto que voltou a se destacar, assim como na Sprint 3, foi o uso excessivo de ferramentas de Inteligência Artificial para geração de código. Embora úteis em algumas situações, observou-se novamente a produção de trechos mais poluídos e inconsistentes, reforçando a necessidade de uso mais criterioso e acompanhado por análise crítica — especialmente em sprints finais, onde a estabilidade do sistema é prioridade. Além disso, práticas como titulação inadequada de *Merge Requests* e inclusão de modificações além do escopo previsto persistiram em alguns momentos, indicando oportunidades importantes de melhoria no fluxo de versionamento.

Como lição geral desta sprint, percebi a importância de equilibrar a entrega de novas funcionalidades com a manutenção da qualidade técnica, especialmente no fechamento do ciclo de desenvolvimento. A necessidade de alinhamento constante entre squads, documentação clara e revisão cuidadosa do código mostrou-se fundamental para garantir um encerramento coeso e profissional do projeto.

Com a conclusão desta sprint, o sistema *Ai Produtor* alcança um bom nível de maturidade e estabilidade. As entregas consolidadas refletem o esforço coletivo da equipe e reforçam os aprendizados técnicos, organizacionais e colaborativos construídos ao longo de todo o semestre.

## 5.4 Conclusão

A atuação como AGES IV no projeto *Ai Produtor* representou uma das experiências mais completas e desafiadoras da minha formação, exigindo de mim um conjunto equilibrado de competências técnicas, organizacionais e de liderança. Desde a Sprint 0, ficou evidente que o papel de um AGES IV vai muito além do suporte técnico: envolve estruturar processos, alinhar expectativas, orientar decisões e garantir que o trabalho de diferentes níveis de AGES avance de forma coerente, organizada e colaborativa.

Ao longo das sprints, pude aplicar de maneira prática conhecimentos de gestão de backlog, modelagem de arquitetura, documentação técnica, comunicação com stakeholders, práticas de versionamento e automação de processos. Ferramentas como GitLab (GITLAB, 2025), Linear (LINEAR, 2025), Figma (FIGMA, 2025) e AWS (AWS, 2025) tornaram-se parte do trabalho cotidiano, permitindo que eu contribuísse tanto na orientação estratégica quanto na execução técnica — especialmente em momentos críticos, como na configuração da infraestrutura em nuvem e nas revisões de código em Back-End e Front-End.

As Sprints 2, 3 e 4 foram marcadas por desafios significativos, que reforçaram lições importantes sobre qualidade, processos e maturidade da equipe. Um dos pontos mais notáveis foi a crescente utilização de ferramentas de IA para geração de código. Embora esse recurso tenha trazido agilidade, observei impactos negativos na qualidade do código, na legibilidade e na padronização. Esse cenário reforçou a importância do uso criterioso de IA, acompanhado de revisão crítica e de aderência às boas práticas de engenharia de software — especialmente em projetos colaborativos, onde consistência e clareza são fundamentais para a evolução do sistema.

Outro aprendizado importante esteve relacionado ao fluxo de versionamento do projeto. Em diversas situações, identifiquei *Merge Requests* com títulos inadequados, escopos desalinhados com as *User Stories* e modificações excessivas dentro de um único *Merge Request*. Essas falhas impactaram diretamente o processo de code review, aumentando o tempo de análise e reduzindo a rastreabilidade das alterações. Essa recorrência evidenciou a necessidade de reforçar processos formais de versãoamento e revisão, bem como a importância de educar continuamente o time para boas práticas de colaboração e controle de qualidade.

No campo das soft skills, a minha atuação exigiu mediação entre níveis de AGES, alinhamento de expectativas, condução de reuniões e apoio direto aos membros que enfrentavam dificuldades técnicas ou bloqueios. A comunicação entre squads, por vezes falha, gerou atrasos e exigiu intervenções para reorganizar tarefas e evitar gargalos. Essas experiências reforçaram a importância de uma comunicação clara, objetiva e frequente — além da necessidade de manter o backlog atualizado e distribuído de forma equilibrada para evitar períodos de ociosidade ou sobrecarga.

Também ficou evidente o quanto a organização inicial da Sprint 0 impactou positivamente as etapas posteriores. Estruturar ferramentas, definir templates, padronizar processos e criar um ambiente colaborativo desde o início foi fundamental para sustentar o desenvolvimento técnico das sprints seguintes. Da mesma forma, a vivência em disciplinas como Gerenciamento de Projetos, Arquitetura de Software e Banco de Dados ofereceu a base teórica necessária para decisões que influenciaram diretamente na qualidade do produto entregue.

Por fim, o projeto *Ai Produtor* consolidou-se como uma experiência transformadora. Não apenas pelo desenvolvimento técnico do sistema — que alcançou maturidade, estabilidade e confiabilidade ao término da Sprint 4 — mas, principalmente, pela evolução profissional que pude vivenciar. Aprendi a liderar com empatia, a apoiar com responsabilidade, a tomar decisões sob pressão e a equilibrar visão técnica com necessidades humanas e organizacionais.

Concluir o *Ai Produtor* como AGES IV me permitiu enxergar, com clareza, a dimensão de responsabilidade envolvida em um papel de liderança e integração entre múltiplos níveis. Diferente das experiências anteriores, aqui o desafio não estava apenas em orientar tecnicamente, mas em garantir coerência organizacional, qualidade do código e alinhamento contínuo entre stakeholders e desenvolvimento. As dificuldades enfrentadas com versionamento, uso excessivo de IA, padronização de entregas e gestão de tarefas mostraram que a maturidade de um time depende tanto de orientação quanto de disciplina processual. Essa experiência ampliou minha visão sobre liderança Técnica e Estratégica, reforçando que coordenar pessoas e práticas é tão essencial quanto dominar ferramentas e tecnologias. O AGES IV marcou, assim, uma transição clara na minha formação: de desenvolvedor atuante para profissional capaz de conduzir equipes e decisões de forma crítica e estruturada.

## 6 CONSIDERAÇÕES FINAIS

Ao longo da minha trajetória no curso de Engenharia de Software na PUCRS, pude vivenciar uma formação que integrou teoria, prática e vivências reais de trabalho em equipe. A disciplina AGES, distribuída ao longo dos quatro níveis, foi o eixo estruturante dessa experiência. Em cada etapa, assumi responsabilidades distintas — de desenvolvedor iniciante a gerente de projeto — o que me permitiu compreender de forma profunda o ciclo de vida de um projeto, não apenas do ponto de vista técnico, mas também humano, organizacional e estratégico.

Do ponto de vista técnico, a AGES foi decisiva para consolidar conhecimentos em desenvolvimento web, bancos de dados, versionamento, arquitetura de software e integração contínua. Minha evolução se deu não apenas pelo uso de tecnologias como Angular, Spring Boot, React, NestJS, Prisma, Docker e AWS, mas principalmente pela oportunidade de aplicá-las em projetos reais, com clientes reais e expectativas reais. Essa vivência trouxe camadas de entendimento que nenhum exercício em sala de aula conseguiria substituir: lidar com escopo mutável, gerenciar dívidas técnicas, tomar decisões arquiteturais, revisar código de colegas, conduzir deploys e manter uma aplicação minimamente sustentável.

A formação de soft skills, porém, foi igualmente transformadora. Em cada projeto, enfrentei desafios distintos: falta de engajamento no AGES I, dificuldades de comunicação no AGES II, decisões arquiteturais complexas no AGES III e responsabilidades de liderança e coordenação no AGES IV. Essas vivências me ensinaram a lidar com equipes heterogêneas, identificar gargalos, comunicar com clareza, ajudar colegas em bloqueios e, principalmente, desenvolver empatia e adaptabilidade. Percebi que liderar não significa apenas orientar tecnicamente, mas criar condições para que o time trabalhe bem — algo que exige escuta, presença e sensibilidade.

Ao refletir sobre essa trajetória, comprehendo que houve acertos importantes: organização de squads, criação de fluxos de trabalho mais claros, integração consistente entre Front-End e Back-End, apoio a membros menos experientes e participação ativa na melhoria dos processos. Mas também enfrentei dificuldades reais: conciliação de horários, variações de engajamento, pouca adesão inicial a boas práticas, desafios com versionamento e, mais recentemente, o uso excessivo e pouco criterioso de ferra-

mentas de IA para geração de código — um ponto que considero crítico e que precisa ser amadurecido nas próximas edições da disciplina.

Minha visão sobre Engenharia de Software também mudou ao longo do curso. No início, eu enxergava a área majoritariamente como desenvolvimento de código. Hoje, entendo que a prática profissional envolve colaboração, processos bem definidos, clareza organizacional, alinhamento com usuários e responsabilidade técnica sobre cada decisão. A AGES foi crucial para essa mudança de perspectiva, pois me expôs às dinâmicas de equipes multidisciplinares e à responsabilidade de entregar soluções tecnológicas que resolvem problemas reais.

Também reconheço o valor do modelo pedagógico da AGES, que aproxima o aluno do ambiente profissional e permite experimentar diferentes papéis em um ciclo de desenvolvimento completo. Ainda assim, vejo oportunidades de melhoria. A disciplina poderia fortalecer o suporte ao uso adequado de ferramentas de gestão e comunicação, promover capacitações introdutórias para novos alunos e fornecer orientações mais claras sobre boas práticas de versionamento, documentação e uso consciente de ferramentas de IA. Pequenos investimentos nessa direção podem elevar significativamente a qualidade dos projetos e o amadurecimento das equipes.

Concluindo essa jornada, reconheço o impacto profundo que o curso de Engenharia de Software e a AGES tiveram na construção da minha identidade profissional. Hoje, atuando como engenheiro de software na HP Inc. e próximo de concluir minha graduação, percebo que chego ao fim deste ciclo mais preparado, mais consciente e mais responsável tecnicamente. Aprendi sobre sistemas, mas aprendi ainda mais sobre pessoas, processos e propósito. Levo comigo a certeza de que esta trajetória moldou não apenas meu conhecimento, mas minha forma de trabalhar, aprender e liderar — e que os valores desenvolvidos ao longo desses anos continuarão guiando minha prática profissional e meu compromisso com a Engenharia de Software como uma área estratégica, humana e transformadora.

## REFERÊNCIAS

AIRTABLE. **Airtable**. [S.l.: s.n.]. Disponível em: <<https://www.airtable.com/>>.

Acesso em: 12 nov. 2025.

ANGULAR. **Angular**. [S.l.: s.n.]. Disponível em: <<https://angular.io/>>. Acesso em: 12 nov. 2025.

AWS. **Amazon Web Services**. [S.l.: s.n.]. Disponível em: <<https://aws.amazon.com/>>. Acesso em: 12 nov. 2025.

AXIOS. **Axios**. [S.l.: s.n.]. Disponível em: <<https://axios-http.com/>>. Acesso em: 12 nov. 2025.

BUSCHMANN, Frank et al. **Pattern-Oriented Software Architecture: A System of Patterns**. [S.I.]: John Wiley & Sons, 1996.

DOCKER. **Docker**. [S.l.: s.n.]. Disponível em: <<https://www.docker.com/>>. Acesso em: 12 nov. 2025.

ESLINT. **ESLint**. [S.l.: s.n.]. Disponível em: <<https://eslint.org/>>. Acesso em: 12 nov. 2025.

FIGMA. **Figma**. [S.l.: s.n.]. Disponível em: <<https://www.figma.com/>>. Acesso em: 12 nov. 2025.

GITHUB. **GitHub**. [S.l.: s.n.]. Disponível em: <<https://github.com/>>. Acesso em: 12 nov. 2025.

GITLAB. **GitLab**. [S.l.: s.n.]. Disponível em: <<https://about.gitlab.com/>>. Acesso em: 12 nov. 2025.

GRINBERG, MIGUEL. **Flask: Web Development Framework for Python**. [S.l.: s.n.], 2025. Disponível em: <<https://flask.palletsprojects.com/>>. Acesso em: 12 nov. 2025.

JAVA. **Java**. [S.l.: s.n.]. Disponível em: <<https://www.java.com/pt-BR/>>. Acesso em: 12 nov. 2025.

KÖSTER, MICHAEL AND MATTHES, FLORIAN AND SEITZ, FLORIAN. **MASSim — Multi-Agent Simulation Platform**. [S.l.: s.n.], 2017. Disponível em: <<https://multiagentcontest.org/>>. Acesso em: 12 nov. 2025.

LINEAR. **Linear**. [S.I.: s.n.]. Disponível em: <<https://linear.app/>>. Acesso em: 12 nov. 2025.

MARTIN, Robert C. **Clean Architecture: A Craftsman's Guide to Software Structure and Design**. 1. ed. [S.I.]: Prentice Hall Press, 2017.

MAVEN. **Maven**. [S.I.: s.n.]. Disponível em: <<https://maven.apache.org/>>. Acesso em: 12 nov. 2025.

NESTJS. **NestJS**. [S.I.: s.n.]. Disponível em: <<https://nestjs.com/>>. Acesso em: 12 nov. 2025.

NODEJS. **Node.js**. [S.I.: s.n.]. Disponível em: <<https://nodejs.org/>>. Acesso em: 12 nov. 2025.

NPM. **Npm**. [S.I.: s.n.]. Disponível em: <<https://www.npmjs.com/>>. Acesso em: 12 nov. 2025.

OPENSTREETMAP FOUNDATION. **OpenStreetMap: Collaborative Mapping Project**. [S.I.: s.n.], 2025. Disponível em: <<https://www.openstreetmap.org>>.

POSTGIS. **PostGIS**. [S.I.: s.n.]. Disponível em: <<https://postgis.net/>>. Acesso em: 12 nov. 2025.

POSTGRESQL. **PostgreSQL**. [S.I.: s.n.]. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 12 nov. 2025.

PRISMA ORM. **Prisma ORM**. [S.I.: s.n.]. Disponível em: <<https://www.prisma.io/>>. Acesso em: 12 nov. 2025.

PYTHON SOFTWARE FOUNDATION. **Python: A High-Level Programming Language**. [S.I.: s.n.], 2025. Disponível em: <<https://www.python.org/>>. Acesso em: 12 nov. 2025.

REACTJS. **React – A JavaScript library for building user interfaces**. [S.I.: s.n.]. Disponível em: <<https://reactjs.org/>>. Acesso em: 12 nov. 2025.

SPRING BOOT. **Spring Boot**. [S.I.: s.n.]. Disponível em: <<https://spring.io/>>. Acesso em: 12 nov. 2025.

SWAGGER. **Swagger**. [S.I.: s.n.]. Disponível em: <<https://swagger.io/>>. Acesso em: 12 nov. 2025.

TAILWIND CSS. **Tailwind CSS**. [S.l.: s.n.]. Disponível em:

<<https://tailwindcss.com/>>. Acesso em: 12 nov. 2025.

TYPESCRIPT. **TypeScript**. [S.l.: s.n.]. Disponível em:

<<https://www.typescriptlang.org/>>. Acesso em: 12 nov. 2025.

VERCEL. **Next.js — The React Framework**. [S.l.: s.n.], 2025. Disponível em:

<<https://nextjs.org/docs>>. Acesso em: 12 nov. 2025.