

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

PEDRO CASTIGLIA FILIPETTO

**MEMORIAL DE ATUAÇÃO NA AGÊNCIA EXPERIMENTAL DE ENGENHARIA DE
SOFTWARE - SEMESTRE 2025/2
AGES I**

Porto Alegre, Rio Grande do Sul
2025

RESUMO

Este documento relata a experiência do autor no projeto Operações GAECO, desenvolvido na Agência Experimental de Engenharia de Software (AGES) da PUCRS para o Ministério Público do Rio Grande do Sul (MPRS). Durante este período, o autor integra a equipe do projeto Operações GAECO, que foi desenvolvida para o Ministério Público do Rio Grande do Sul (MPRS) que busca criar uma plataforma para gerenciar e documentar as operações feitas pela GAECO. O autor desempenhou um papel ativo na concepção e implementação da plataforma, atuando em tarefas de prototipagem, desenvolvimento de backend e de frontend. O relato evidencia a aplicação prática de metodologias ágeis e explora como os desafios técnicos e a dinâmica de equipe foram fundamentais para o aprimoramento de habilidades de comunicação, proatividade e resolução de problemas por parte do autor.

PALAVRAS CHAVES: AGES, Engenharia de Software, Operações GAECO, PUCRS, operações, documentar, gerenciar.

LISTA DE ILUSTRAÇÕES

Figura 1 – Time Vincula	6
Figura 2 – Modelo do banco relacional	8
Figura 3 – Diagrama de Deploy	9
Figura 4 – Tela de Autenticação	10
Figura 5 – Tela de Listagem de Casos	11
Figura 6 – Tela do Caso	11
Figura 7 – Tela de Visualização de Vínculos	12

LISTA DE SIGLAS

AGES	Agência Experimental de Engenharia de Software	7
MPRS	Ministério Público do Rio Grande do Sul	6

SUMÁRIO

1	APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO	5
2	AGES I — “VINCULA”	6
2.1	Introdução	6
2.2	Desenvolvimento do Projeto	7
2.3	Atividades Desempenhadas Pelo Aluno no Projeto	13
2.4	Conclusão	14
	Referências	16

1 APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

Desde que me conheço por gente,

2 AGES I — “VINCULA”

2.1 Introdução

Desenvolvido para o Ministério Público do Rio Grande do Sul (MPRS), o projeto Vincula visa solucionar uma necessidade crítica do **nimp! (nimp!)**: a ausência de uma ferramenta integrada para cruzamento de dados recebidos em investigações, como os de quebras de sigilo bancário e telefônico.

O processo atual é manual e fragmentado, o que dificulta a identificação de relações complexas. Para resolver esse problema, nosso objetivo é desenvolver uma plataforma web que permita aos analistas e promotores visualizar os vínculos entre as informações de forma gráfica e intuitiva, além de gerar relatórios.

O projeto, que está sendo desenvolvido durante o segundo semestre de 2025, conta com a orientação do Prof. Dilnei Venturini e a colaboração direta dos stakeholders Césio Luiz Velleda Lázaro da Silva e Luciano Ratai Menna Barreto. A Figura 1 mostra o time reunido com o professor e os stakeholders.



Figura 1 – Time Vincula
Fonte: Adaptado de Projeto Vincula (2025b)

2.2 Desenvolvimento do Projeto

2.2.1 Metodologia de Desenvolvimento

O desenvolvimento do projeto é organizado em Sprints de duas semanas, seguindo os princípios da metodologia Scrum (Schwaber; Sutherland, 2020), conforme proposto pela AGES. Nossa equipe está inserida em uma estrutura de squads, onde cada um é liderado por um colega de AGES III e composto por alunos de AGES I e II, o que permite um trabalho mais focado e colaborativo.

Cada Sprint se inicia com uma cerimônia de Planejamento (Sprint Planning), momento em que os colegas AGES IV apresentam as User Stories e as tarefas são distribuídas entre as equipes.

Para garantir o alinhamento contínuo durante a execução da Sprint, adotamos uma rotina de sincronização diária. Durante as aulas presenciais de terça e quinta-feira, realizamos uma Daily Scrum, na qual o andamento das tarefas registradas no Azure DevOps (Microsoft Corporation, 2025) é revisado. Nos demais dias, essa sincronização ocorre de forma assíncrona por meio de um check-in diário em nosso servidor do Discord (Discord Inc., 2025), que também funciona como nosso principal canal para comunicação rápida e colaboração em tempo real.

Ao final do ciclo, a Sprint se conclui com duas cerimônias principais: a Revisão (Sprint Review), na qual apresentamos o incremento de trabalho desenvolvido ao stakeholder para validação, e a Retrospectiva (Sprint Retrospective), uma reunião interna focada na melhoria contínua de nossos processos.

2.2.2 Repositório do Código Fonte do Projeto

O projeto conta com dois repositórios separados, ambos foram mantidos no GitLab da Agência Experimental de Engenharia de Software (AGES). Um engloba o código do frontend e o outro do backend.

- Vincula frontend: <https://tools.ages.pucrs.br/vincula/frontend>
- Vincula backend: <https://tools.ages.pucrs.br/vincula/backend>

2.2.3 Banco de Dados Utilizado

A arquitetura de persistência de dados do projeto Vincula adota uma abordagem híbrida, utilizando duas soluções de banco de dados distintas para otimizar a performance. Para os dados estruturados da aplicação, como o gerenciamento de usuários e casos de investigação, foi escolhido um sistema de banco de dados relacional, o PostgreSQL (PostgreSQL Global Development Group, 2025). Em paralelo, para a funcionalidade central de modelagem e consulta dos vínculos, foi empregado um banco de dados orientado a grafos, o Neo4j (Neo4j, Inc., 2025).

Essa separação estratégica permite utilizar a robustez do PostgreSQL para as operações transacionais e, ao mesmo tempo, aproveitar a alta performance do Neo4j para as complexas consultas de conectividade e análise de redes de relacionamento.

O banco de grafos está em planejamento e será implantado a partir da Sprint 2.

A Figura 2 mostra o modelo do banco relacional.

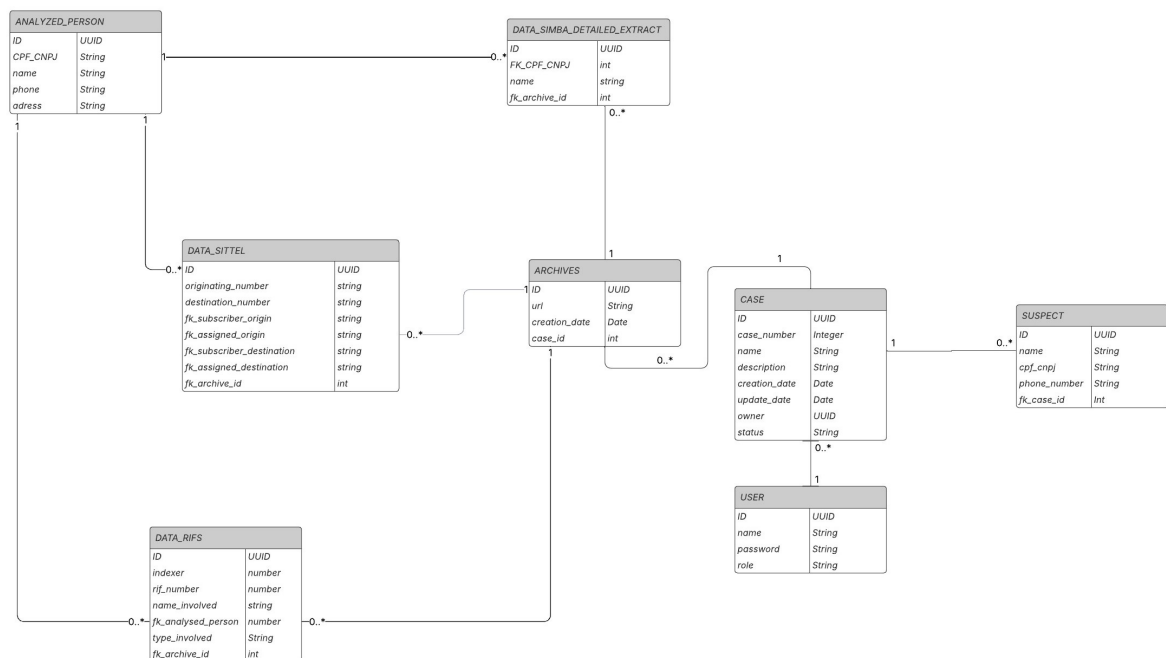


Figura 2 – Modelo do banco relacional
Fonte: Adaptado de Projeto Vincula (2025b)

2.2.4 Arquitetura Utilizada

A arquitetura do projeto Vincula foi projetada para ser executada na nuvem da **aws!** (**aws!**) (Amazon Web Services, Inc., 2025), utilizando uma combinação de serviços gerenciados e um ambiente containerizado para garantir eficiência e automação no ciclo de desenvolvimento. A Figura 3 mostra o diagrama de Deploy na **aws!**.

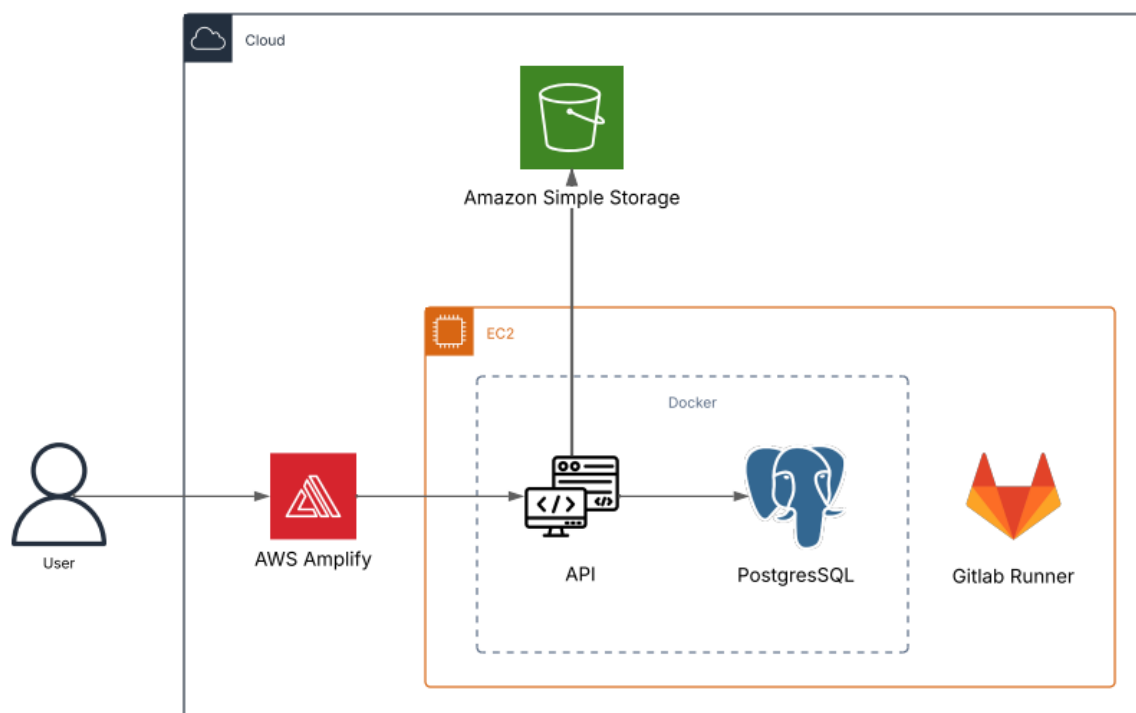


Figura 3 – Diagrama de Deploy
Fonte: Adaptado de Projeto Vincula (2025b)

Conforme ilustrado pelo diagrama da Figura 3, a arquitetura da solução segue um fluxo claro e desacoplado. A interface com o usuário, hospedada no AWS Amplify (Amazon Web Services, 2025c), consome uma API backend em Python (Python Software Foundation, 2025) com FastAPI. Esta API, juntamente com o banco de dados PostgreSQL, opera de forma containerizada com Docker (Docker, Inc., 2025) em uma instância Amazon EC2 (Amazon Web Services, 2025a). O sistema é complementado pelo Amazon S3 (Amazon Web Services, 2025b), responsável pelo armazenamento de arquivos, e por um GitLab Runner (GitLab Inc., 2025) na mesma instância EC2, que automatiza o processo de integração e entrega contínua (CI/CD).

2.2.5 Protótipos das Telas Desenvolvidas

Os protótipos para as telas centrais da aplicação foram desenvolvidos utilizando a ferramenta Figma (Figma, Inc., 2025). O manual de identidade visual do MPRS foi usado como referência para cores, estilo e logomarcas.

Algumas das telas são: Tela de Autenticação (Figura 4), Tela de Listagem de Casos (Figura 5), Tela do Caso (Figura 6) e a Tela de Visualização de Vínculos (Figura 7).

As demais telas estão disponíveis no Figma do projeto:

<https://www.figma.com/design/mvL3UaJUd8Rnnctve1iSpV/Vincula-AGES>



Figura 4 – Tela de Autenticação
Fonte: Adaptado de Projeto Vincula (2025a)

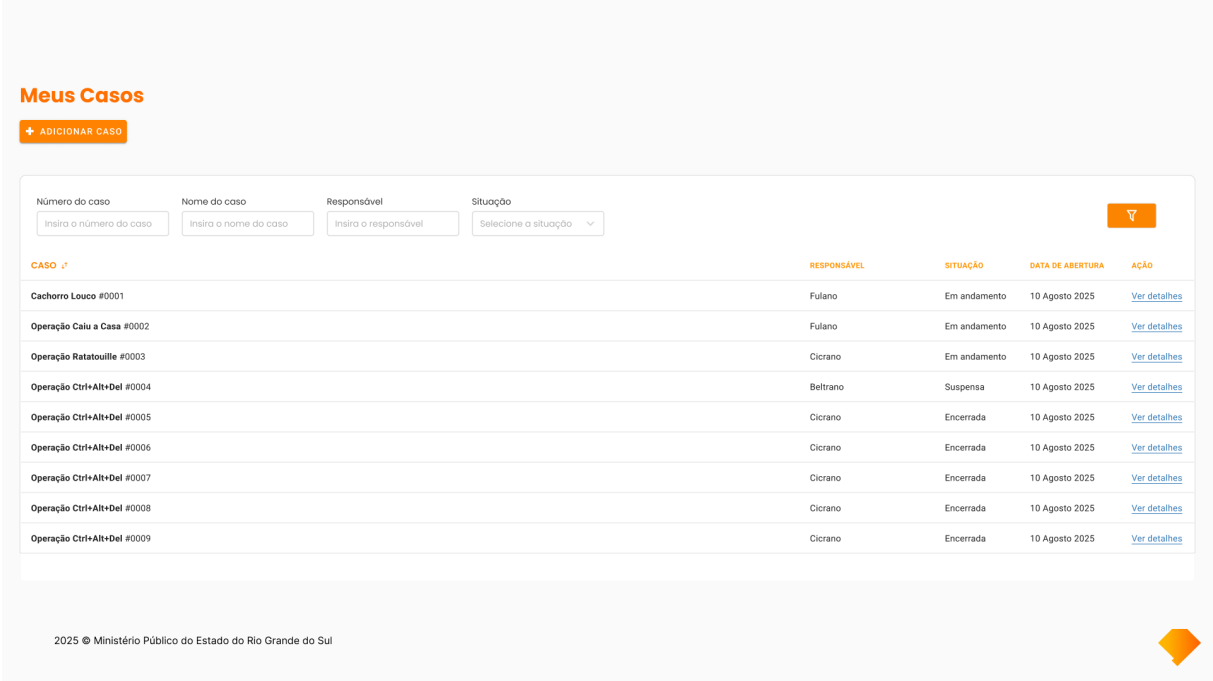


Figura 5 – Tela de Listagem de Casos
Fonte: Adaptado de Projeto Vincula (2025a)

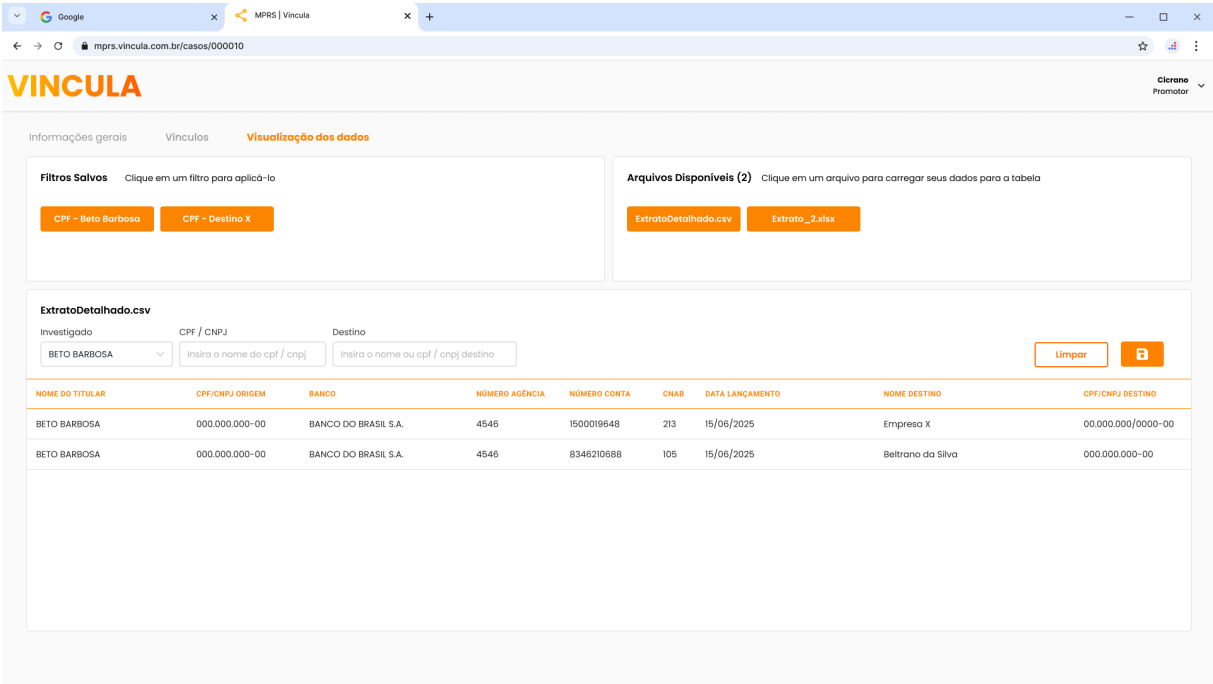


Figura 6 – Tela do Caso
Fonte: Adaptado de Projeto Vincula (2025a)

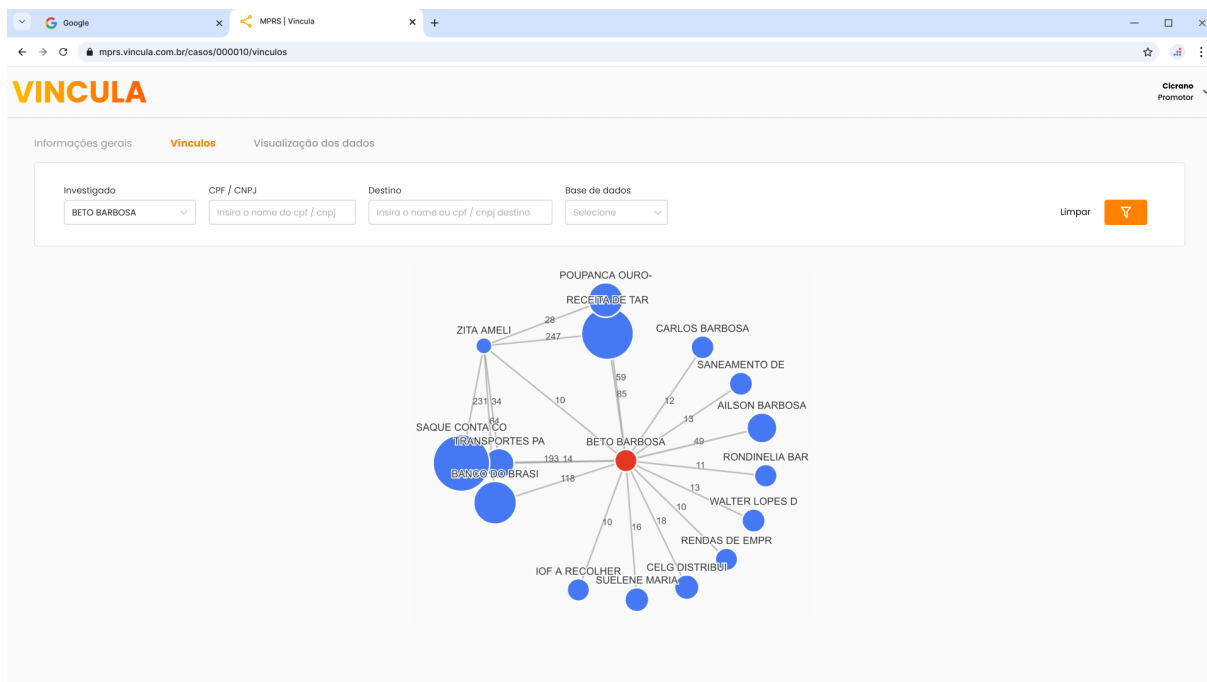


Figura 7 – Tela de Visualização de Vínculos
 Fonte: Adaptado de Projeto Vincula (2025a)

2.2.6 Tecnologias Utilizadas

O projeto é construído utilizando quatro tecnologias principais, sendo elas o React (Meta Platforms, Inc., 2025) com NextJs (Vercel Inc., 2025) e a linguagem TypeScript (Microsoft, 2025) para o frontend, o FastAPI (Sebastián Ramírez, 2025), com a linguagem Python para o backend, PostgreSQL para o banco relacional e Neo4j (Neo4j, Inc., 2025) para o banco de grafos.

O NextJs foi escolhido para o frontend por ser uma framework moderna e que abstrai diversos aspectos trabalhosos do desenvolvimento com a biblioteca React pura.

No backend, logo de início, decidimos que uma framework Python seria essencial para o projeto, visto a necessidade de manipular grandes volumes de dados de forma eficiente. Por isso, optamos pela FastAPI, que torna o processo de programar endpoints simplificado. O processamento de dados será feito com a biblioteca Pandas (The pandas development team, 2025).

O gerenciamento de tarefas, o planejamento das sprints e o controle do backlog são realizados na plataforma Azure DevOps. O controle de versão é feito com Git, hospedado no servidor da AGES. Todo o ambiente de desenvolvimento é containerizado com Docker.

2.3 Atividades Desempenhadas Pelo Aluno no Projeto

2.3.1 Sprint 0

Durante a Sprint 0, o time focou no planejamento e na prototipagem das interfaces da plataforma Vincula. Enquanto os colegas AGES II desenhavam o modelo inicial do banco de dados e os AGES III e IV definiam a arquitetura, minha contribuição inicial foi na colaboração com os mockups das telas essenciais no Figma, como as de Home, Login e a de Visualização de Casos.

Por ser minha primeira experiência na AGES, após a fase inicial de prototipagem, encontrei um desafio em identificar os próximos passos e reconheço que faltou proatividade da minha parte para buscar novas tarefas. Decidi, então, focar em estudar as tecnologias para as futuras sprints de desenvolvimento do backend.

Para isso, desenvolvi um projeto de estudo prático com FastAPI, implementando funcionalidades de cadastro de usuários, login e autenticação com **jwt!** (**jwt!**), o que me deu uma base sólida para as tarefas que viriam a seguir. Adicionalmente, para entender a complexidade dos dados que iremos manipular, criei scripts com a biblioteca Pandas para fazer uma análise exploratória nos arquivos de exemplo disponibilizados pelos stakeholders provenientes do **simba!** (**simba!**) e do **sittel!** (**sittel!**), o que me permitiu entender na prática como os vínculos se formam.

A sprint terminou com a apresentação dos mockups e das User Stories planejadas para a Sprint 1 aos stakeholders. Posteriormente, a retrospectiva da Sprint 0 foi uma experiência de grande importância para meu aprendizado, pois nela pude esclarecer minhas dúvidas e alinhar com os colegas AGES IV questões sobre a comunicação e responsabilidades. A conversa me acalmou quanto às incertezas sobre a organização de tarefas, pois entendi que a natureza da Sprint 0 é realmente mais fluida e exploratória, sem uma definição de atividades individuais tão rígida como nas sprints de desenvolvimento.

A lição mais importante desta sprint foi a importância da comunicação e da proatividade. Aprendi que é minha responsabilidade buscar ativamente o alinhamento com a equipe e pedir direcionamento. Ao final da Sprint 0, senti-me tecnicamente mais preparado para as tarefas de backend e, principalmente, mais ciente da postura colaborativa que o projeto exige para as próximas etapas.

2.3.2 Sprint 1

Com o planejamento concluído, a Sprint 1 marcou o início efetivo do desenvolvimento da plataforma. Fui responsável por duas tarefas principais que abrangeram tanto o backend quanto o frontend, permitindo-me aplicar os conhecimentos adquiridos na fase de estudos. No backend, implementei o endpoint para listagem de casos ('GET /cases'), que incluía a lógica de paginação, filtragem dinâmica e a proteção de rota via token de autenticação. Este trabalho exigiu colaboração com o colega Bryan Leandro, para integrar a funcionalidade com o sistema de login, e foi refinado com base no feedback dos colegas Felipe Cardona e Gabriel Belmonte.

No frontend, desenvolvi um componente de 'Input' genérico e reutilizável com React e Material-UI (MUI Team, 2025), seguindo as especificações do Figma, os princípios de componentes do NextJs e alinhando as decisões de implementação com as colegas Lara Kunrath e Carolina Ferreira. Além das minhas tarefas designadas, procurei manter uma postura colaborativa, auxiliando os colegas Rodrigo Schmitt e Gabriel Pinho com suas tarefas.

O principal desafio desta sprint não foi técnico, mas sim de processo e comunicação dentro da equipe. Observei que a falta de uma sincronia técnica inicial entre desenvolvedores trabalhando em partes interconectadas do sistema levou a inconsistências no padrão de código, o que gerou retrabalho no final da sprint. Essa experiência reforçou a lição da sprint anterior sobre a importância da proatividade.

Apreendi que, para garantir a consistência e a qualidade da base de código, a comunicação técnica no início do desenvolvimento de uma tarefa é tão crucial quanto o code review no final. Ao final da Sprint 1, ambas as minhas tarefas foram concluídas e integradas com sucesso. Sinto que solidifiquei meu conhecimento prático em FastAPI e NextJs e, mais importante, obtive uma visão mais clara de como minha comunicação proativa pode contribuir diretamente para a qualidade técnica e a eficiência de toda a equipe.

2.4 Conclusão

Até o momento, minha experiência atuando na AGES tem sido extremamente positiva, estava ansioso para esse momento desde o início do curso, e estou dando meu

máximo para aprender muito e garantir que consigamos entregar um projeto completo e que supere expectativas.

Os desafios, na sua maioria ligados ao trabalho em equipe, estão me ensinando tanto quanto meus sucessos no desenvolvimento de código. O ambiente da AGES está me possibilitando aprender a colaborar de forma efetiva e entender a importância da comunicação em um projeto de software.

Sigo entusiasmado com a AGES e com o curso de Engenharia de Software, estou cada vez mais convencido que fiz a escolha certa ao voltar a estudar, mesmo depois de mais velho.

REFERÊNCIAS

AMAZON WEB SERVICES. **Amazon Elastic Compute Cloud (EC2)**. Seattle, Washington: Amazon Web Services, Inc., 2025. Disponível em <https://aws.amazon.com/ec2>. Acesso em 6 de setembro de 2025.

AMAZON WEB SERVICES. **Amazon Simple Storage Service (S3)**. Seattle, Washington: Amazon Web Services, Inc., 2025. Disponível em <https://aws.amazon.com/s3>. Acesso em 6 de setembro de 2025.

AMAZON WEB SERVICES. **AWS Amplify**. Seattle, Washington: Amazon Web Services, Inc., 2025. Disponível em <https://aws.amazon.com/amplify>. Acesso em 6 de setembro de 2025.

AMAZON WEB SERVICES, INC. **Amazon Web Services**. Seattle, Washington: Amazon Web Services, Inc., 2025. Disponível em <https://aws.amazon.com>. Acesso em 2 de setembro de 2025.

DISCORD INC. **Discord**. San Francisco, California: Discord Inc., 2025. Disponível em <https://discord.com>. Acesso em 6 de setembro de 2025.

DOCKER, INC. **Docker**. San Francisco, California: Docker, Inc., 2025. Disponível em <https://www.docker.com>. Acesso em 6 de setembro de 2025.

FIGMA, INC. **Figma**. San Francisco, California: Figma, Inc., 2025. Disponível em <https://www.figma.com/pt-br/>. Acesso em 2 de setembro de 2025.

GITLAB INC. **GitLab Runner**. San Francisco, California: GitLab Inc., 2025. Disponível em <https://docs.gitlab.com/runner>. Acesso em 6 de setembro de 2025.

META PLATFORMS, INC. **React**. Menlo Park, California: Meta Platforms, Inc., 2025. Disponível em <https://react.dev>. Acesso em 2 de setembro de 2025.

MICROSOFT. **TypeScript**. Redmond, Washington: Microsoft Corporation, 2025. Disponível em <https://www.typescriptlang.org>. Acesso em 2 de setembro de 2025.

MICROSOFT CORPORATION. **Azure DevOps**. Redmond, Washington: Microsoft Corporation, 2025. Disponível em <https://azure.microsoft.com/products/devops>. Acesso em 6 de setembro de 2025.

MUI TEAM. **MUI**. [S.l.]: MUI, 2025. Disponível em <https://mui.com>. Acesso em 6 de setembro de 2025.

NEO4J, INC. **Neo4j Graph Database Platform**. San Mateo, California: Neo4j, Inc., 2025. Disponível em <https://neo4j.com>. Acesso em 6 de setembro de 2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL**. Canada: PostgreSQL Global Development Group, 2025. Disponível em <https://www.postgresql.org>. Acesso em 2 de setembro de 2025.

PROJETO VINCULA. **Figma do Projeto Vincula**. [S. l.: s. n.], 2025. Disponível em: <https://www.figma.com/design/mvL3UaJUd8Rnnctve1iSpV/Vincula-AGES>. Acesso em 6 de setembro de 2025.

PROJETO VINCULA. **Wiki do Projeto Vincula**. [S. l.: s. n.], 2025. Disponível em: <https://tools.ages.pucrs.br/vincula/wiki/-/wikis/home>. Acesso em 6 de setembro de 2025.

PYTHON SOFTWARE FOUNDATION. **Python**. Beaverton, Oregon: Python Software Foundation, 2025. Disponível em <https://www.python.org/>. Acesso em 2 de setembro de 2025.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game**. Burlington, Massachusetts, 2020. Edição de Novembro de 2020.

SEBASTIÁN RAMÍREZ. **FastAPI**. Berlin, Germany: Sebastián Ramírez, 2025. Disponível em <https://fastapi.tiangolo.com/>. Acesso em 2 de setembro de 2025.

THE PANDAS DEVELOPMENT TEAM. **pandas**. Austin, Texas: NumFOCUS, 2025. Disponível em <https://pandas.pydata.org/>. Acesso em 2 de setembro de 2025.

VERCEL INC. **NextJs**. San Francisco, California: Vercel Inc., 2025. Disponível em <https://nextjs.org/>. Acesso em 2 de setembro de 2025.