

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 Operações GAECO	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 Operações	11
Figura 6 – Tela de Listagem de Alvos	12

LISTA DE SIGLAS

AGES	Agência Experimental de Engenharia de Software	7
MPRS	Ministério Público do Rio Grande do Sul	6
MBA	Mandado de Busca e Apreensão	6

SUMÁRIO

1	APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO	5
2	AGES I — “OPERAÇÕES GAECO”	6
2.1	Introdução	6
2.2	Desenvolvimento do Projeto	7
2.3	Atividades Desempenhadas Pelo Aluno no Projeto	13
2.4	Conclusão	15
	Referências	16

1 APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

Desde que me conheço por gente,

2 AGES I — “OPERAÇÕES GAECO”

2.1 Introdução

Desenvolvido para o Ministério Público do Rio Grande do Sul (MPRS), o projeto Operações GAECO visa solucionar uma necessidade crítica de integração das aplicações de Mandado de Busca e Apreensão (MBA), RECUPERA E CUMPRA-SE de forma com que todos os participantes das operações consigam visualizar os locais, objetos, alvos, entre outras informações e documentá-las.

O processo atual é manual, o que acaba dificultando o acesso das informações a todos os integrantes da operação, além não permitir a edição dos documentos, sem precisar refazê-los por completo. Para resolver esse problema, nosso objetivo é desenvolver um aplicativo que permita aos agentes e aos administradores da operação visualizar e criar os alvos e as operações, além de gerar relatórios com um resumo das apreensões.

O projeto, que está sendo desenvolvido durante o segundo semestre de 2025, conta com a orientação do Prof. Rafael Chanin e a colaboração direta dos stakeholders Rovena Zanchet e Frantiele Rodrigues dos Santos. A Figura 1 mostra o time reunido com o professor e os stakeholders.



Figura 1 – Time Operações GAECO
Fonte: Adaptado de **wiki-Operacoes GAECO**

2.2 Desenvolvimento do Projeto

2.2.1 Repositório do Código Fonte do Projeto

O projeto conta com três repositórios separados, e todos foram mantidos no GitLab da Agência Experimental de Engenharia de Software (AGES). Um engloba o código do frontend, outro do backend e o último contém a Infraestrutura.

- Operações GAECO frontend: <https://tools.ages.pucrs.br/operacoes-gaeco/operacoes-gaeco-mobile>
- Operações GAECO backend: <https://tools.ages.pucrs.br/operacoes-gaeco/operacoes-gaeco-backend>
- Operações GAECO: [https://tools.ages.pucrs.br/operacoes-gaeco-i](https://tools.ages.pucrs.br/operacoes-gaeco/operacoes-gaeco-i)

2.2.2 Banco de Dados Utilizado

A arquitetura de persistência de dados do projeto Operações GAECO adota uma abordagem simples, utilizando uma solução de banco de dados para otimizar a performance. Para os dados estruturados da aplicação, como o gerenciamento de usuários e operações, foi escolhido um sistema de banco de dados relacional, o PostgreSQL (PostgreSQL Global Development Group, 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.

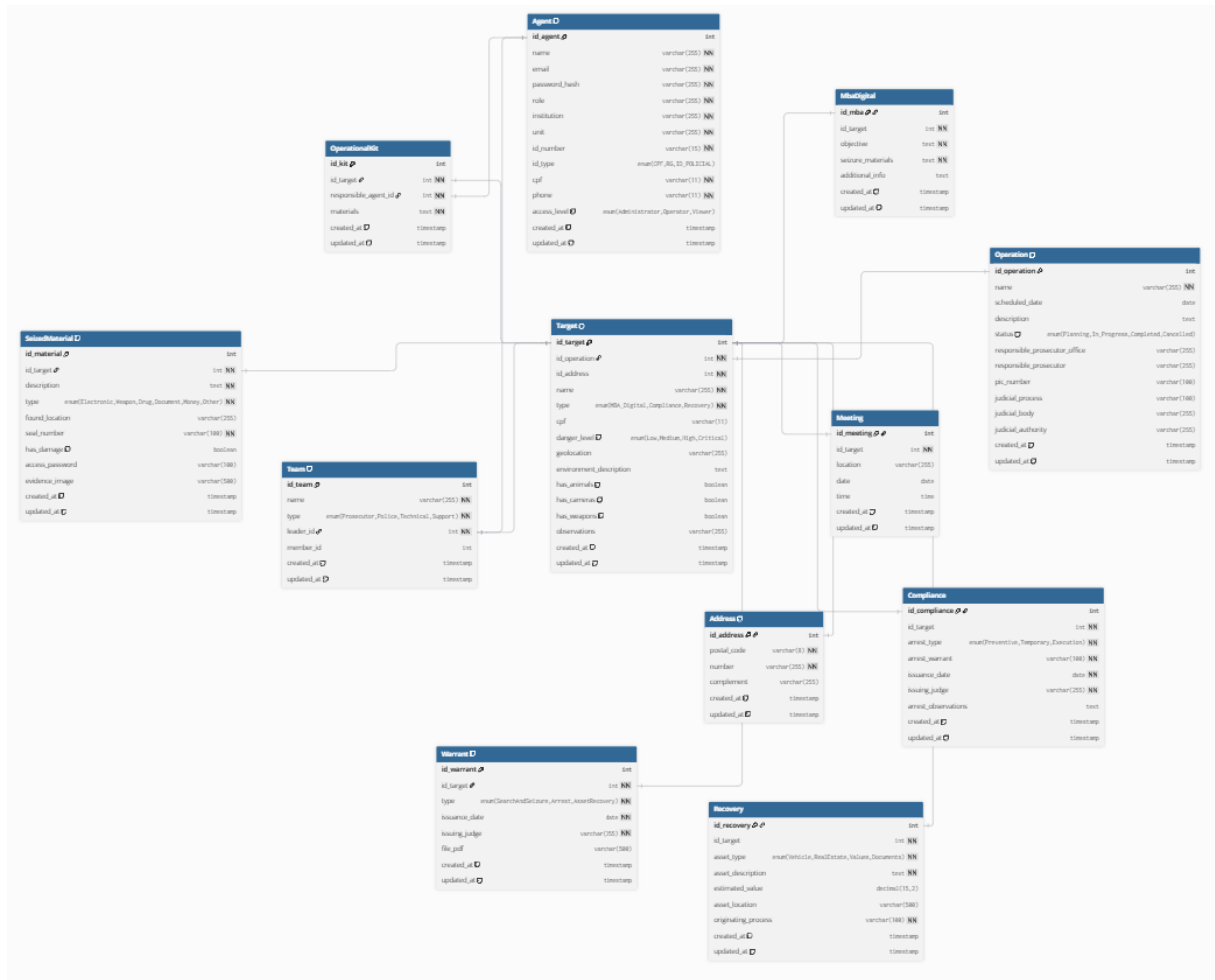


Figura 2 – Modelo do banco relacional
 Fonte: Adaptado de **wiki-vincula**

2.2.3 Arquitetura Utilizada

A arquitetura do projeto Operações GAECO 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!**.

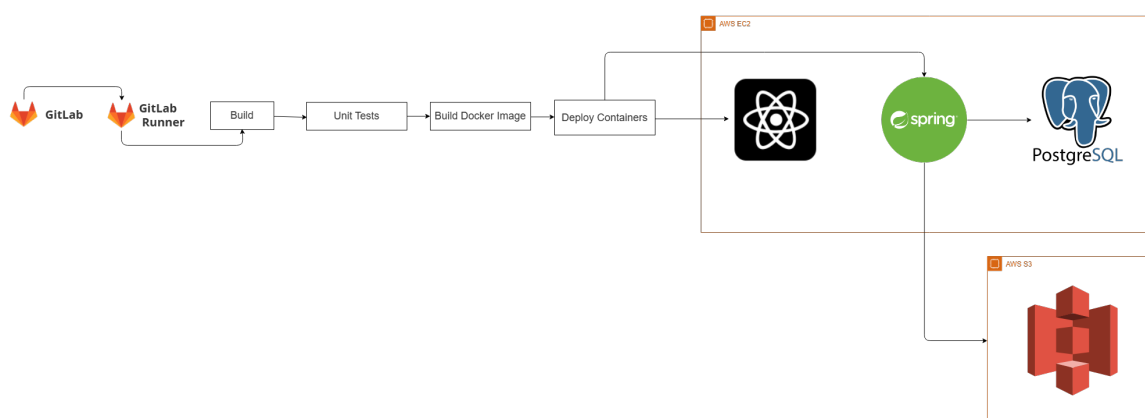


Figura 3 – Diagrama de Deploy
Fonte: Adaptado de **wiki-Operacoes GAECO**

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 Java (Oracle Corporation, 2025) com Spring Boot. 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.

2.2.4 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 Operações (Figura 5) e Tela do Caso (Figura 6).

As demais telas estão disponíveis no Figma do projeto:

<https://www.figma.com/design/eDNVyvDaNWyvONdiL053ff/Opera%C3%A7oes-GAECO>



Figura 4 – Tela de Autenticação
Fonte: Adaptado de **figma-Operacoes GAECO**



Figura 5 – Tela de Listagem de Operações
Fonte: Adaptado de **figma-Operacoes GAECO**

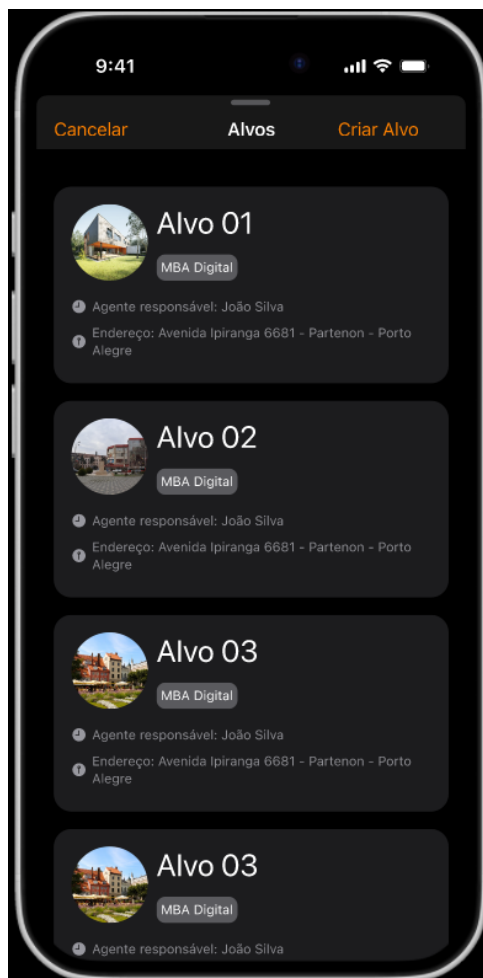


Figura 6 – Tela de Listagem de Alvos
Fonte: Adaptado de **figma-Operacoes GAECO**

2.2.5 Tecnologias Utilizadas

O projeto é construído utilizando quatro tecnologias principais, sendo elas o React Native (**react**) com Expo (**nextjs**) e a linguagem TypeScript (Microsoft, 2025) para o frontend, o Spring Boot (**fastapi**), com a linguagem Java para o backend, PostgreSQL para o banco relacional.

O Spring Boot foi escolhido por sua facilidade de configuração, suporte a microserviços, integração com diversos bancos de dados e ecossistema rico para desenvolvimento de APIs REST.

No backend, logo de início, decidimos que este projeto utilizaria Java 21, uma linguagem robusta, orientada a objetos e amplamente utilizada para aplicações corporativas.

O gerenciamento de tarefas, o planejamento das sprints e o controle do backlog

são realizados na plataforma ClickUp. 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 Operações GAECO. 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 ideias para o desenvolvimento das telas.

Por ser minha primeira experiência na AGES, após a fase inicial de prototipagem, encontrei dificuldade 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 e frontend.

Para isso, me desafiei a praticar desenvolvendo telas do frontend, baseado nos mockups do figma, e comecei alguns rascunhos procurando me aproximar ao máximo das telas desenhadas no figma. Isso me fez entender um pouco da estrutura para que eu pudesse utilizar no desenvolvimento que viria em seguida.

A etapa da Sprint 0 foi concluída com a apresentação dos mockups e das User Stories da Sprint 1 aos stakeholders. Para mim, o momento mais significativo foi a retrospectiva, onde pude alinhar expectativas sobre comunicação e responsabilidades com os colegas mais experientes da equipe (AGES IV). Essa troca de ideias me tranquilizou e esclareceu, pois me fez perceber que a estrutura ser menos rígida e menor volume de tarefas são características próprias de uma sprint exploratória como a Sprint 0, diferentemente do que ocorre nas fases seguintes de desenvolvimento.

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

2.3.2 Sprint 1

Com o planejamento concluído, iniciamos com tarefas para o desenvolvimento das telas em duplas. Até o meio da Sprint 1, desenvolvemos um esboço para a tela de listagem de alvos que foi atribuída a mim e ao colega Felipe Cruz, porém, em seguida os colegas AGES IV julgaram melhor que apenas deveríamos desenvolver os componentes das telas para que após esse processo pudessemos apenas uni-los.

Após replanejarmos a Sprint 1, iniciamos efetivamente o desenvolvimento da aplicação. Fui responsável por dois componentes que abrangeram o frontend, permitindo-me aplicar os conhecimentos adquiridos na fase de estudos. No frontend, implementei a search bar para que tornasse possível a busca por algo nas telas, o segundo componente foi o team indicator member que tornou possível visualizar os agentes adicionados nas operações. Este trabalho exigiu colaboração com o colega Felipe Cruz, e foi feita a validação dos componentes e integração no projeto pelos colegas Erick Muniz e Marcus Raach.

Além das minhas tarefas designadas, procurei sempre manter uma postura colaborativa, auxiliando os colegas com suas tarefas e continuei estudando para as tarefas seguintes de backend.

O principal desafio desta sprint foi o replanejamento em meio a Sprint, o que na minha opinião desacelerou a Sprint, mas acredito ser uma decisão crucial para a eficiência dos processos. Observei uma falta de convicção entre gerentes de projeto AGES IV para a tomada de decisão inicial. Essa experiência reforçou a lição para melhorarmos a comunicação do time.

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 frontend e React Native e, mais importante, obtive uma visão mais clara de como a comunicação e proatividade 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.

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.

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

ORACLE CORPORATION. **Java Platform, Standard Edition**. Austin, Texas: Oracle Corporation, 2025. Disponível em <https://www.oracle.com/java/>. Acesso em 9 de setembro de 2025.

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