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 FREITAS SILVA

memorial de atuação na agência experimental de engenharia de software – período 2023 a 2025

AGES I,II, III ou IV

Porto Alegre, RS

2023

**Dedicatória (opcional)**

Dedicatória: Texto no qual o autor do trabalho oferece homenagem ou dedica o seu trabalho a alguém.

**Agradecimentos (opcional)**

AGRADECIMENTOS

Os agradecimentos devem ser dirigidos àqueles que contribuíram de maneira relevante à elaboração do trabalho, restringindo-se ao mínimo necessário, como instituições (CNPq, CAPES, PUCRS, empresas ou organizações que fizeram parte da pesquisa), ou pessoas (profissionais, pesquisadores, orientadores, etc.).

Os agradecimentos devem ser colocados de forma hierárquica de importância e para trabalhos financiados com recursos de instituições (CAPES, CNPq, FINEP, FAPERGS, etc.) os agradecimentos são obrigatórios a essas instituições.

**Epígrafe (opcional e sem título)**

Epígrafe: É um item onde o autor apresenta a citação de um texto que seja relacionado com o tema do trabalho, seguido da indicação de autoria do mesmo.

(texto iniciando do meio da página alinhado a direita)

Nome do autor da epígrafe

**RESUMO**

Este documento trata da minha trajetória pessoal na Agência Experimental de Engenharia de Software (AGES) ao longo do curso de Engenharia de Software da PUCRS, minhas contribuições e aprendizados ao longo da minha jornada em quatro projetos; o primeiro que tratava do desenvolvimento de um sistema capaz de analisar PDF’s, o segundo cujo foco era ..., o terceiro que buscava ..., e um último que visava ....

**PALAVRAS CHAVES:** AGES, Engenharia de Software, Programação, Desenvolvimento Web, Arquitetura de Sistemas, Trabalho em equipe.

LIsta de ilustrações

[**Foto 1**: Equipe AGES I 10](#_Toc131961349)

**Ilustração 1**: Diagrama de Deploy

Lista de tabelas

[Tabela 1: Preços de alimentos em dólares de 1900-1952 a 1995-1997 2](#_Toc206829730)

Lista de Siglas

ABNT – Associação Brasileira de Normas Técnicas

AGES – Agência Experimental de Engenharia de Software

PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul

SUMÁRIO

[1 - APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO 9](#_Toc130502895)

[2 - PROJETOS AGES I - “Veículos via Montadora” 10](#_Toc130502896)

[2.1 Introdução 10](#_Toc130502897)

[2.2 Desenvolvimento do Projeto 11](#_Toc130502898)

[2.3 Atividades desempenhadas pelo aluno no projeto 12](#_Toc130502899)

[2.4. Conclusão 15](#_Toc130502900)

[3 - PROJETOS AGES II - “NOME DO PROJETO XXXX” 17](#_Toc130502901)

[4 - PROJETOS AGES III - “NOME DO PROJETO XXXX” 18](#_Toc130502902)

[5 - PROJETOS AGES IV - “NOME DO PROJETO XXXX” 19](#_Toc130502903)

[6 - CONSIDERAÇÕES FINAIS (Somente AGES IV) 20](#_Toc130502904)

[REFERÊNCIAS 21](#_Toc130502905)

[APÊNDICES 22](#_Toc130502906)

[APÊNDICE A 22](#_Toc130502907)

1 - APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

Neste tópico o aluno deve fazer um paralelo da evolução dele ao longo do curso e sua trajetória pessoal prática (estágios, bolsas de pesquisa, voluntariado, hackatonas, emprego) **.**

Ages I,II e III podem apresentar seu momento atual.

Desde o começo do curso, evolui muito em diversas áreas da vida e, claro, do trabalho. Quando entrei no curso, em fevereiro de 2022, eu já havia programado alguns sites simples e publicado até mesmo um aplicativo na Play Store, o que me deixa muito feliz de olhar para trás e saber que valeu a pena ter sido curioso e testado muitas coisas.

Mesmo que não tenha me aprofundado e tenha “pulado de galho em galho”, o fato de eu ter estudado PHP, MySQL, Javascript, React Native, Arduino, Python e, claro, Git, desde antes do curso, me deu uma base muito grande para este. Cada parcela, por menor que seja, contribuiu em muito para o curso para o qual eu entrei já pensando que entendia de várias coisas, quando realmente sabia muito pouco de menos ainda.

No 1º semestre da faculdade, eu estava trabalhando como professor de inglês na escola online Skyeng, dando aulas individuais para alunos ao redor do mundo, o que foi muito importante para eu perder muito

2 - PROJETOS AGES I - “Veículos via Montadora”

### Introdução

O projeto Veículos via montadora é um projeto tecnicamente pequeno, mas que prove melhoria significativa para as partes envolvidas, principalmente, os grupos Sinoscar e Tramonto Jeep.

Nos foi apresentado que, para cadastrar um veículo, vários PDF’s eram enviados de diversas fábricas e era o dever de uma pessoa analisar os dados deste documento, tratá-los, e enviar para um sistema próprio da Sinosserra, devendo copiar as informações sem erros. O processo todo leva em volta de 6 horas, e nem sempre é possível finalizar em um dia só devido à ausência de informações ou materiais nos veículos. Foi iniciado em 27/02/2023 e entregue no dia [28/06/2023].

Abaixo, segue foto da equipe, orientada pelo professor Daniel Antônio Callegari.

[](https://tools.ages.pucrs.br/veiculos-via-montadora/wiki/-/raw/main/team/Equipe.jpeg)

Foto 1

Antes de começar o desenvolvimento, é comum que se dedique um período para o estudo das tecnologias mais adequadas e para bem entender as dificuldades do usuário. Este tempo permite um melhor andamento do projeto e distribuição da equipe, e é o que será tratado nesta sessão, juntamente da minha contribuição para ele.

O projeto teve início após a primeira reunião com o cliente, porém, não foi sem complicações. Neste primeiro encontro, descobrimos que o cliente original não poderia mais acompanhar o projeto e, por este motivo, outra pessoa foi enviada para explicar o problema que estaríamos solucionando, mas como este não tinha o mesmo domínio de causa que o original, tivemos de marcar um 2º encontro na primeira semana com a pessoa encarregada de substituir o *stakeholder* inicial o que, por fim, deu início ao projeto.

### 2.2 Desenvolvimento do Projeto

* + 1. Repositório do código Fonte do Projeto

O projeto foi dividido em 3 repositórios de código, cujos links podem ser encontrados abaixo:

* Front-end: <https://tools.ages.pucrs.br/veiculos-via-montadora/frontend>
* Back-end: <https://tools.ages.pucrs.br/veiculos-via-montadora/backend>
* Infraestrutura:\_<https://tools.ages.pucrs.br/veiculos-via-montadora/infrastructure>

E um último para a *Wiki* do projeto, com toda a documentação e outras informações relevantes

* Wiki: <https://tools.ages.pucrs.br/veiculos-via-montadora/wiki/-/wikis/home>
  + 1. Banco de Dados utilizado

Dada a natureza muito dinâmica dos documentos armazenados e da linguagem utilizada no backend, optou-se por utilizar o SGBD não relacional MongoDB.

Um modelo de documento (que está sujeito a alterações) pode ser encontrado na figura disponível na wiki: <https://tools.ages.pucrs.br/veiculos-via-montadora/wiki/-/wikis/banco_dados>.

A alternativa relacional foi analisada, mas foi concluído que, devido a natureza muito volátil das informações armazenada, não faria sentido utilizar esse paradigma.

* + 1. Arquitetura utilizada

A parte da arquitetura está bem detalhada na *Wiki* do projeto e pode ser encontrada no link a seguir: <https://tools.ages.pucrs.br/veiculos-via-montadora/wiki/-/wikis/arquitetura>. Basicamente, foram utilizados recursos da AWS por requisição do cliente; caso ele queira integrar nossa solução com o sistema já existente dele. Abaixo, há uma foto do diagrama de *deploy* com a relação entre os serviços, *containers* e *runners*.

Diagrama de Deploy na AWS
Mostra a integração entre containers, runners e pipelines.

Figura

* + 1. Protótipos das telas desenvolvidas

Link para a Wiki com todas as telas e componentes produzidos no Figma: <https://tools.ages.pucrs.br/veiculos-via-montadora/wiki/-/wikis/mockups>

* + 1. Tecnologias Utilizadas

Abaixo, estão citadas as tecnologias utilizadas, acompanhadas por uma breve descrição e o link da sua documentação.

Geral:

Quase todos os projetos atualmente utilizam de ferramentas que permitem uma maior integração entre os ambientes de desenvolvimento variados da equipe. Com isso em mente, utilizamos o *Git* para versionamento e alterações paralelas no código e o *Docker* em todas as frentes para que todos, independente de configurações externas, possam desenvolver sob as mesmas condições.

- Git: <https://git-scm.com/doc>

- Docker: https://docs.docker.com/get-started/

Frontend:

Foi desenvolvido em *TypeScript* com auxílio das *bibliotecas React* (para estrutura e lógica de componentes), *StyledComponets* e *MaterialUI* (para a parte gráfica). Para a padronização do código e dos *commits* foi utilizado o *Prettier* unido ao *ESLint* para a questão de espaçamentos e re-fatoração de código, além do *Husky* - unido as últimas duas ferramentas citadas - para que toda mudança versionada no *Git* esteja de acordo com as especificações definidas. Ainda, unido ao React foi utilizada a ferramenta *Vite*, que melhora a construção e distribuição do Software e o *Yarn* como gerenciador de dependências. Por fim, para realizar os testes unitários utilizamos o *Jest*.

- Typescript: <https://www.typescriptlang.org/docs/>

- React: <https://react.dev/learn>

- Material UI: <https://mui.com/material-ui/getting-started/overview/>

- Styled Components: <https://styled-components.com/docs>

- Prettier: <https://prettier.io/docs/en/index.html>

- ESLint: <https://eslint.org/docs/latest/>

- Husky: <https://typicode.github.io/husky/>

- Vite: <https://vitejs.dev/guide/>

- Yarn: <https://yarnpkg.com/getting-started>

- Jest: <https://jestjs.io/pt-BR/docs/getting-started>

Backend:

Desenvolvido em *Python* com os *frameworks* *FastAPI* e *Uvicorn* para o servidor, utiliza o *Poetry* para gerenciamento de dependências – a principal delas sendo o *TabulaPY* (para leitura do PDF) - e *Pytest* para testes unitários. Além disso, o código deste repositório é executado em cima de uma função *AWS Lambda* rodando um container *Docker*.

- Python: <https://www.python.org/doc/>

- FastAPI: <https://fastapi.tiangolo.com/>

- Uvicorn: <https://www.uvicorn.org/>

- Poetry: <https://python-poetry.org/docs/>

- Tabula PY: <https://tabula-py.readthedocs.io/en/latest/>

- Pytest: <https://docs.pytest.org/en/7.2.x/>

- AWS Lambda: <https://docs.aws.amazon.com/pt_br/lambda/latest/dg/welcome.html>

### 2.3 Atividades desempenhadas pelo aluno no projeto

2.3.1 Sprint 0

2.3.1.1 Atividades Previstas

Apesar de na Sprint 0 não ter sido prevista nenhuma atividade em especifico para cada um, em dois momentos eu decidi que gostaria de ser proativo. Na primeira semana, junto da primeira reunião com o cliente, eu decidi pesquisar sobre como fazer a leitura de PDF (principalmente em Java ou Javascritpt). Mais para frente, depois da reunião e do time ter encontrado uma biblioteca melhor em Python, decidi estudar e me focar bastante em uma área pra mim pouca explorada; a do design, e resolvi fazer uns mockups e protótipos no Figma.

2.3.1.2 Atividades Concluídas

Consegui extrair o texto bruto de 2 PDF's em Java, e contribui bastante para o desenvolvimento do Figma, principalmente na questão dos protótipos e animações.

2.3.1.3 Problemas Encontrados

Apesar de ter conseguido ler um PDF estritamente falando em Java, apenas consegui extrair o texto, e não o conteúdo da forma originalmente estruturada. Ainda, não consegui ler o arquivo em Javascript, apenas mostrá-lo na tela. Da parte do Figma, não encontrei nenhum problema grande, além da minha falta de conhecimento e prática, que foi rapidamente superada depois de pedir ajuda ao Luiz (AGES IV) e assistir alguns tutoriais no Youtube.

2.3.1.4 Lições Aprendidas

As lições aprendidas nessa Sprint foram várias, mas principalmente técnicas. Aprendi a mexer bem no Figma, em questão de svg, UX, animações, componentes, e outros.

2.3.1.5 Próximos Passos

Ao longo dessa primeira Sprint, foi decidida toda a questão de arquitetura, linguagens e paradigmas que seriam utilizados, sendo estes React + Typescript para o front, Python (com FastAPI) + MongoDB para o back/banco, e diversos serviços da AWS por definição do cliente, que foram gerenciados majoritariamente pelo Arthur (AGES III). Com isso em mente, visto que eu hoje já trabalho com as tecnologias do Front + MongoDB (na Nutrório), meus próximos passos previstos são estudar Python, aplicado a parte de leitura de PDF com a biblioteca definida (Tabula) e gostaria de poder também auxiliar os outros AGES I que estão começando com a parte do Front-End.

2.3.2 Sprint 1

2.3.1.1 Atividades Previstas

Para a Sprint 1, agora com a arquitetura definida, o time foi dividido inicialmente em duas frentes, e decidi que seria mais produtivo ajudar no frontend, visto que já tinha mais experiência com isso e é a área em que poderia ajudar mais pessoas. Depois de divididos entre back e front, dividimos o frontend nos componentes delimitados pelo Figma e cada squad recebeu uma ou mais tarefas. A tarefa da minha equipe era o componente principal de visualização de arquivo, que poderia receber os estados de carregando, carregado e erro, uma opção de apagar, e que deveria se adequar ao tamanho estipulado pelo componente "pai".

2.3.1.2 Atividades Concluídas

Conseguimos finalizar o componente com todos seus requisitos. Em uma visão mais ampla, as outras squads também finalizaram seus componentes e consegui me inteirar de tudo que foi feito nessa frente. Não acompanhei tanto o backend, mas sei que o boilerplate foi concluído e que consegui ajudar na questão dos testes de maneira geral.

2.3.1.3 Problemas Encontrados

O maior problema encontrado foi a questão do tempo, em que a Sprint 1 é muito curta em relação as outras, e ainda não há uma união tão grande na equipe; é pouco presente o espírito de time ainda. Outro problema, que foi descoberto na final da sprint foi o da integração; apesar de cada "squad" ter finalizado seu componente, a junção deles e o fluxo geral da aplicação não foi finalizado, por falta de comunicação e revisão.

2.3.1.4 Lições Aprendidas

Com isso em mente, é necessário prestar mais atenção futuramente na integração de componentes e das equipes.

2.3.1.5 Próximos Passos

Finalizadas as tarefas individualmente, o passo mais urgente e lógico é integrar os componentes do frontend entre si, e integrar o front com o back. Ainda, a realização de testes unitários e de integração segue na escala de importância, acompanhados pela criação dos componentes restantes e última página do mockup do Figma.

2.3.3 Sprint 2

No mínimo uma página contendo tudo que o aluno fez na Sprint 2.

2.3.4. Sprint 3

No mínimo uma página contendo tudo que o aluno fez na Sprint 3.

2.3.5. Sprint 4

No mínimo uma página contendo tudo que o aluno fez na Sprint 4.

2.4. Conclusão

Neste item o aluno deverá refletir sobre:

* Crítica e autocrítica em relação a sua atuação no projeto na parte técnica como na parte de soft skills.
* Comente o relacionamento entre as disciplinas cursadas e a AGES.
* Este projeto foi o melhor projeto trabalhado? Justifique.
* Relate as lições aprendidas (retrospectiva pessoal):
  + - O que foi positivo no projeto.
    - O que podia ter sido melhor (em ternos de banco, desenvolvimento, arquitetura, soft skills....)

(No mínimo uma página de relato)

3 - PROJETOS AGES II - “NOME DO PROJETO XXXX”

Cada projeto deverá iniciar em uma página distinta.

Parágrafos deve ter 1,25 cm.

Figuras centralizadas, com o nome da Figura acima e o nome da Fonte embaixo da figura.

4 - PROJETOS AGES III - “NOME DO PROJETO XXXX”

5 - PROJETOS AGES IV - “NOME DO PROJETO XXXX”

6 - CONSIDERAÇÕES FINAIS (Somente AGES IV)

As considerações finais referem-se a trajetória do aluno no curso, onde se expõe o fechamento da narrativa e são apresentados os resultados alcançados.

Este item é somente para os AGES IV.

Em particular, espera-se neste capítulo:

* contribuições que o curso trouxe para a sua evolução profissional
  + competências (o que) e habilidades desenvolvidas (como), (hardskills e softskills);
  + lições aprendidas (o que deu certo, o que deu errado);
* uma reflexão sobre a visão do aluno sobre a prática da Engenharia de Software, como era no início de sua trajetória, e que visão ele tem hoje;
* eventuais comentários que deseje adicionar;
* sugestão de melhorias, críticas e elogios em relação a AGES.

( No mínimo uma página de relato)

REFERÊNCIAS

ENGEL, J.; BLACKWELL, R. D.; MINIARD, P. W. **Comportamento do consumidor.** 8. ed. Rio de Janeiro: LTC, 2000.

AMADO, Jorge. **Capitães de areia**. Rio de Janeiro: Record, 1991. 233 p.

\_\_\_\_\_\_. **Gabriela cravo e canela.** São Paulo: Martins, 1958. 453 p.

PRADO, Geraldo. **A arena científica. Ciência da Informação**, Brasília, v. 24, n. 2, 1995. Recensões. Disponível em:<http://www.ibict.br/cionline/inicio.htm>. Acesso em: 16 dez. 2003.

JOHANSSON, Henry J. et al. **Processos de negócios: como criar sinergia entre a estratégia de mercado e a excelência operacional**. São Paulo: Pioneira, 1995. (Biblioteca Pioneira de Administração e Negócios). Tradução de: Business process reengineering.

(mais exemplos no site da Biblioteca Central da PUCRS http://www3.pucrs.br/portal/page/portal/biblioteca/Capa/BCEPesquisa/BCEPesquisaModelos).

APÊNDICES

APÊNDICE A – Exemplo1: Análise dos relatórios mensais de uso do serviço de renovação de empréstimos.

**Apêndice(s) (**Este item é elaborado pelo próprio autor do trabalho e serve para complementar a sua argumentação. É um elemento **opcional).**