

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 II**

Porto Alegre, RS
2023

Um bom programador é alguém
que sempre olha para os dois lados antes
de atravessar uma rua de mão única.

Autor Desconhecido

RESUMO

Este documento trata da minha trajetória pessoal ao longo do curso de Engenharia de Software da PUCRS na Agência Experimental de Engenharia de Software (AGES), cujo objetivo é permitir aos alunos que tenham vivências similares as do mercado de trabalho. Mais especificamente, apresentarei aqui minhas contribuições e aprendizados ao longo da minha jornada em quatro projetos, realizando atividades que vão desde o planejamento e design de telas/mockups, banco de dados e histórias de usuário até a programação de funcionalidades complexas para o usuário final e gerenciamento de equipe e de projeto.

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

LISTA DE ILUSTRAÇÕES

Figura 1 – Foto da Equipe	8
Figura 2 – Diagrama de Deploy	11
Figura 3 – Foto da Equipe	30
Figura 4 – Diagrama de Banco de Dados.....	32

LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

AGES – Agência Experimental de Engenharia de Software

CSS – Cascading Style Sheets

CSV – Comma-separated values

E2E – End to End

HP – Hewlett-Packard

LIS – Laboratório de Inovação em Software

MVC – Model View Controller

NPM – Node Package Manager

PDF – Portable Document Format

PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul

SGBD – Sistema Gerenciador de Banco de Dados

SGS – Sistema de Gestão Sinosserra

SQL – Structured Query Language

US – User Story

SUMÁRIO

1. APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO	6
2. PROJETOS AGES I - Veículos via Montadora.....	8
2.1 Introdução	8
2.2 Desenvolvimento do Projeto.....	10
2.2.1 Repositório do código Fonte do Projeto	10
2.2.2 Banco de Dados utilizado.....	10
2.2.3 Arquitetura utilizada.....	10
2.2.4 Protótipos das telas desenvolvidas.....	12
2.2.5 Tecnologias Utilizadas.....	12
2.3 Atividades desempenhadas pelo aluno no projeto	15
2.3.1 Sprint 0	16
2.3.2 Sprint 1	18
2.3.3 Sprint 2	20
2.3.4 Sprint 3	22
2.3.5 Sprint 4	24
2.4 Conclusão	26
3. PROJETOS AGES II – Globo Aplausos 2023	29
3.1 Introdução	29
3.2 Desenvolvimento do Projeto.....	31
3.2.1 Repositório do código Fonte do Projeto	31
3.2.2 Banco de Dados utilizado.....	31
3.2.3 Arquitetura utilizada.....	32
3.2.4 Protótipos das telas desenvolvidas.....	33
3.2.5 Tecnologias Utilizadas.....	33
3.3 Atividades desempenhadas pelo aluno no projeto	35
3.3.1 Sprint 0	36
3.3.2 Sprint 1	38

1. APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

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 tecnologias, algumas que posso citar sendo PHP, MySQL, Javascript, React Native, Arduino, Python e Git. Mesmo que não tenha me aprofundado e tenha “pulado de galho em galho”, o fato de eu ter visto um pouquinho disso tudo desde antes da faculdade me deu uma base muito grande. Cada parcela, por menor que seja, contribuiu em muito para o curso para o qual 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. Embora não seja relacionado a área em que trabalho e que provavelmente vou seguir trabalhando ao longo da minha vida, eu não me arrependo de ter passado esses 6 meses fazendo algo que me desafiou muito e que moldou minha personalidade, sendo muito importante para eu perder muito da minha timidez que tinha até o ensino médio. Ainda sobre o primeiro semestre, foi nele em que construí a maior parte dos amigos que tenho até hoje e que me apoiaram e acompanharam durante o curso, nos momentos bons e nos difíceis. Uma experiência muito marcante que tive logo no começo foi a *Global Legal Hackathon* (GLH) – uma *Hackatona* focada para melhorar a integração das pessoas com o Direito no Brasil. Participaram da GLH aproximadamente 40 advogados e 6 desenvolvedores, todos da PUCRS, sendo 4 meus colegas, também de 1º semestre, e um que foi meu AGES IV logo no primeiro projeto da AGES e com quem desenvolvi uma boa relação. Essa atividade foi um excelente pontapé para perceber que não sabia de quase nada prático, mas que não estava sozinho, e que havia várias pessoas com quem poderia contar quando precisasse, cada uma com um *background* diferente e valioso.

Terminados os primeiros 6 meses, já tinha passado com relativa tranquilidade do 1º semestre, e decidi que gostaria de tentar algum estágio na área de desenvolvimento. Terminei minha parceria com a escola e passei 3 meses nessa busca, participei de alguns processos seletivos, mas nenhum resultou em nada, até

que um colega, que era o principal desenvolvedor de uma Startup no Tecnopuc, acreditou em mim e me ofereceu uma vaga para que eu o ajudasse no desenvolvimento. Eu aceitei e, em agosto de 2022, comecei a trabalhar no *front end*, criando telas responsivas e dando algumas sugestões sobre como implementar funcionalidades mais complexas. Em dezembro do mesmo ano, esse colega saiu também para trabalhar em outro lugar, e eu fiquei responsável por continuar o trabalho dele, que consistia na lógica do *back end* e manutenção do banco de dados – o que era novo para mim, visto que utilizamos outro paradigma de banco de dados (não relacional) com que não estava familiarizado, mas novamente decidi aceitar o desafio, pois sabia que ele ainda poderia me ajudar enquanto me adaptava a isso.

Desde o começo de 2023, sou o principal desenvolvedor da Startup e tenho já um colega responsável pelo *front end* (a quem auxiliei quando entrou para que pudesse seguir com mais facilidade o que eu agora passaria adiante) e outro que passei a treinar para me substituir eventualmente no *back end*. No 3º semestre de faculdade comecei minha trajetória pela AGES e percebo o quão importantes são todas essas oportunidades que tive, embora ainda fique claro o quanto ainda tenho pela frente no curso quando observo meus colegas mais experientes, e espero eventualmente conseguir ajudar os que vierem depois de mim para que todo conhecimento seja disperso, e nunca perdido. Por volta de julho do mesmo ano, mudei um pouco meu foco dentro da *startup* para garantir que o projeto siga em boas mãos, pois sei que é muito promissor, então fiz diversas melhorias em servidor, documentação e outras ferramentas para que cada novo desenvolvedor que tenha a oportunidade de passar pelo projeto tenha mais facilidades que o último. Por fim, em agosto de 2023, depois de precisamente um ano trabalhando quase todos os dias nessa startup, me despedi do projeto, dos colegas e mentores que me acompanharam ao longo dessa jornada de imensos aprendizados, visando alcançar novos patamares e ter outras experiências.

Assim, pouco depois do início do 4º semestre do curso de Engenharia de Software, fui aprovado para uma vaga de Bolsista/Estagiário no Laboratório de Inovação em Software (LIS), que foi criado através de uma parceria entre a HP e a PUCRS em 2011. O objetivo principal deste é permitir uma vivência em projetos de desenvolvimento e de pesquisa com empresas e clientes reais, integrando teoria e prática com a base de conhecimento dos diversos cursos de informática da faculdade.

2. PROJETOS AGES I - VEÍCULOS VIA MONTADORA

2.1 Introdução

O projeto Veículos via Montadora é um projeto relativamente pequeno, mas que prevê melhoria significativa para as partes envolvidas, principalmente, os grupos Sinoscar e Tramonto Jeep, representados aqui pelos *Stakeholders* Leonardo Cunha, Luana Lima Vieira e Fabiano Longaray. Nos foi apresentado que, para cadastrar um veículo, vários PDF's eram enviados de diversas fábricas e era o dever de uma pessoa (a saber, da *Luana*) analisar os dados deste documento, tratá-los, e enviar para um sistema próprio da Sinosserra (SGS), 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. Com tudo isso em mente, nossa missão era desenvolver um programa capaz de extrair e analisar os dados de diversos e variados PDF's para aumentar a produtividade e melhorar a qualidade de vida de todos envolvidos hoje e futuramente no processo.

Foi iniciado em 27/02/2023 e entregue no dia 14/06/2023, tendo sido acompanhado e orientado pelo professor Daniel Antônio Callegari (na imagem, à direita). Abaixo, segue foto do time responsável pelo desenvolvimento.

Figura 1 – Foto da Equipe



Fonte 1: <https://tools.ages.pucrs.br/veiculos-via-montadora/wiki/-/raw/main/team/equipe.jpeg>

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

2.2.1 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>

2.2.2 Banco de Dados utilizado

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

Um modelo de documento 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 à natureza muito volátil das informações armazenadas, não faria sentido utilizar esse paradigma. Essa dinamicidade do modelo não relacional se provou bem útil, visto que na Sprint 3 percebemos que teríamos de remodelar o banco para acomodar uma nova dinâmica de PDF's e veículos, o que teria sido muito mais complexo em um modelo de tabelas.

2.2.3 Arquitetura utilizada

A arquitetura foi bem detalhada e explicada majoritariamente pelo Arthur (Ages III), e pode ser encontrada na *Wiki* do projeto, 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 queiram integrar nossa solução com o sistema já existente dele. Abaixo, há uma imagem do diagrama de *deploy* original com a relação entre os serviços, *containers* e *runners*.

2.2.4 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>.

Um dos maiores requisitos da tela a ser criada era que ficasse similar ao sistema já existente, para diminuir a fricção na aderência do novo sistema. Com isso em mente, o Luiz (Ages IV) fez o design de uma estrutura com componentes dispostos de maneira idêntica ao SGS já existente, e o design em si foi pouco alterado desde então, visto que só ele criou ideias de *mockups* para o sistema.

Minha contribuição inicial ao projeto foi em grande parte justamente aqui - no Figma - visto que, apesar de não conseguir fazer o design bem desde o começo, consegui ajudar bastante na parte de prototipação, “componentização” e navegação do modelo. Com isso, a apresentação inicial aos stakeholders, que já tinha forma, ganhou vida, possuindo componentes interativos que simulavam quase integralmente a experiência final que o time tinha em mente para usabilidade da plataforma. O modelo seguiu evoluindo junto com o projeto e a necessidade de incluir novos itens, principalmente respostas gráficas ao usuário como variações de erro e sucesso para alguns componentes.

2.2.5 Tecnologias Utilizadas

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

- Git: <https://git-scm.com/doc>
- Docker: <https://docs.docker.com/get-started/>

Front end:

Foi desenvolvido em *TypeScript* com auxílio das bibliotecas *React* (para estrutura e lógica de componentes), *StyledComponents* 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>

Back end:

Desenvolvido em *Python* com os frameworks *FastAPI* e *Uvicorn* para o servidor, utiliza o *Poetry* para gerenciamento de dependências – as principais delas sendo *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

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.

Apesar de nesta Sprint 0 não ter sido prevista nenhuma atividade em específico para cada um, em dois momentos eu decidi que gostaria de ser proativo e tentar contribuir mais para o projeto. Na primeira semana, junto da primeira reunião com o cliente, eu decidi pesquisar sobre como fazer a leitura de PDF, preferencialmente em Java ou Javascript que imaginei serem as linguagens mais prováveis de utilizarmos. 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 para mim pouca explorada; a do design, e resolvi fazer uns mockups e protótipos no Figma.

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. Apesar de ter conseguido ler um PDF estritamente falando, 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.

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 e acompanhei as discussões sobre tecnologias e práticas a serem utilizadas no projeto. Nunca tinha escutado falar de algumas técnicas como, por exemplo, para limitar a gramática dos *commits* e tinha um certo preconceito com a plataforma *Trello*, que passei a superar vendo sua utilidade para projetos de médio porte como esse, permitindo distribuir e apresentar as tarefas de maneira dinâmica e interativa.

Ao longo dessa primeira Sprint, foi decidida toda a questão de arquitetura, linguagens e paradigmas que seriam utilizados. Com isso em mente, visto que eu hoje já trabalho com as tecnologias do Front + MongoDB, 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

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 *front end*, visto que já tinha mais experiência com isso e é a área em que poderia ajudar mais pessoas. Depois de divididos entre *back end* e *front end*, dividimos o *front end* 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”.

Conseguimos finalizar o componente com todos seus requisitos e acredito que consegui repassar bastante do que já sabia para meus colegas de equipe, que nunca tinham tido experiência com a programação funcional que utilizamos com o *React* ou com a estilização de componentes por meio de CSS. 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 *back end*, mas sei que o *boilerplate* foi concluído e que consegui ajudar na questão dos testes de maneira geral.

O maior problema encontrado foi a questão do tempo, isto é, a Sprint 1 é muito curta em relação as outras, e ainda não havia 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.

Aprendemos com isso que é necessário prestar mais atenção futuramente na integração de componentes e das equipes externamente. Portanto, finalizadas as tarefas individualmente, o passo mais urgente e lógico é integrar os componentes do front-end entre si e com o *back end*, agora mais completo. 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, que ainda não foi iniciada.

Em suma, apesar de a Sprint 1 ser uma das mais curtas e de a entrega não ter sido tão satisfatória, certamente aprendemos nossa lição e aprimoramentos já estão sendo pensados e já começaram desde já. As equipes foram repensadas, os Ages I receberam um foco maior e as *tasks* foram mais granularmente divididas, então

acredito que a Sprint 2 será melhor – haverá mais entregas e os Ages I aprenderão e contribuirão mais.

2.3.3 Sprint 2

Nesta Sprint 2, sabendo que o maior problema da Sprint anterior fora a integração dos componentes, houve uma melhor estruturação das *User Stories* e divisão destas entre a equipe. Assim sendo, todas as “squads” tinham tarefas mais precisas e independentes, mais especificamente, cada “squad” foi dividida ainda mais entre *front end* e *back end*, com os Ages I sendo responsáveis pela tarefa, Ages II responsáveis por auxiliar os Ages I, e Ages III e IV como supervisores das tarefas. Minha dupla tinha por objetivo a criação do componente do status de extração dos PDF's, que deveria receber um status calculado a partir da análise dos status de cada arquivo individualmente, sendo este status positivo, negativo ou indefinido (carregando).

Conseguimos concluir nosso componente, assim como o resto da squad e, conforme apontado na última sprint, realizamos diversas integrações com outros componentes dos outros times, com o *back end* e com a própria AWS, tendo iniciado a “Dockerização” das duas “frentes” do projeto com suas respectivas *imagens* publicadas e hospedadas na AWS.

Pessoalmente, não tive muitos problemas na Sprint; estava na minha zona de conforto, trabalhando com frontend e auxiliando minha dupla que, apesar de já ter passado por Ages I, não tinha tanto entendimento das ferramentas utilizadas no projeto. Por outro lado, as pessoas da minha squad que ficaram encarregadas do *back end* sofreram um pouco mais e não conseguiram entregar a tarefa proposta, que era de realizar uma leitura e análise do PDF da Chevrolet, visto que não havia nenhum padrão neles; parecia que cada novo veículo era feito em uma ferramenta diferente por uma pessoa diferente, isto é, as tabelas e informações dos veículos cada vez eram dispostos em uma posição e com uma notação diferente da próxima, o que impossibilitou o desenvolvimento de uma solução generalizada para este grupo de veículos.

Aprendemos com esta (falta de) entrega que há de se cuidar muito bem do que é prometido para o cliente e o quão importante é, principalmente para as empresas envolvidas, a própria Agência (que, como diz no nome, é experimental), já que as diferenças que para um ser humano podem simplificar a leitura, quando tenta-se terceirizar e automatizar por meio de máquinas e programação, por vezes dificultam esta conversão, o que foi documentado e repassado para o cliente, que entendeu o

problema e se encarregou de conversar com a fábrica responsável pelo desenvolvimento dos materiais analisados.

Após analisar e refletir sobre as falhas do projeto até agora, e depois de conversar com o *stakeholder* para alterar levemente o escopo de análise, optamos por Iniciar a leitura dos PDFs da Jeep, que aparentava ter mais padrão e uma estrutura que, embora mais “feia” e “crua” para nós, humanos, era mais bem estruturada para uma análise sistemática. Voltando para o *front end* e para as tarefas em que eu viria participar, também foi decidido que agora começaríamos a construir as telas e componentes relacionados a visualização dos dados dos PDFs - o principal ponto de acesso do sistema para a cliente e usuária final do produto.

2.3.4 Sprint 3

Conforme prometido na sprint anterior, estas próximas semanas seriam dedicadas à leitura dos PDFs da Jeep, à atualização do modelo de banco de dados e à construção das telas e componentes relacionados a visualização dos dados destes PDF's. A sprint era muito curta e, por isso, cada squad tinha uma responsabilidade simples e pontual, sendo a minha toda a sessão de listagem de PDFs, desde a integração com o arquivo em si até o componente de cada PDF com seu nome, e decidimos internamente que eu faria o esqueleto da página e que o Marcello (minha dupla e Ages II) criaria o componente menor e a parte mais “bonita”, de estilização da página.

Ao contrário da última Sprint, desta vez conseguimos entregar tudo e nos redimir, tendo feito uma leitura de sucesso dos PDF's previstos e conseguido integrar a parte visual com a parte lógica do projeto, nosso componente da squad foi completo e a listagem de PDF's e apresentação dos dados dos veículos estava clara.

Apesar de termos feito uma entrega de sucesso, esta não foi sem problemas, um dos maiores tendo sido uma suspeita de o *GitLab* estar fora do ar, então tivemos que mover o projeto inteiro para o GitHub e trabalhar por ali durante uma semana – o que não é tão simples quando fazer um *push* para outro repositório apenas, visto que nossos pipelines e validações de código e de cargos estava toda configurada na outra plataforma, mas era algo necessário, ou ao menos acreditou-se ser, já que foi percebido que a plataforma não chegou a ficar indisponível, o que não é muito relevante, mas o time ficou levemente desapontado pelo esforço direcionado em vão e alterar a origem do código duas vezes. De qualquer forma, o maior problema foi que a sprint durou aproximadamente 10 dias apenas, e contemplou apenas 1 final de semana (período em que a maior parte do time está ativa e contribuindo), e isso nos levou a fazer algo “drástico” na terça-feira anterior à apresentação da Sprint 3 – “madrugar”.

O time se reuniu, todos meio apavorados e em dúvida, para juntos conectarmos os últimos pontos que faltavam, e não eram poucos, mas foi uma experiência muito interessante e que possibilitou uma maior integração do time como um todo, independente de squads o que, olhando por este ponto, foi positivo – como discutiremos mais à frente. Felizmente, conseguimos “remendar” as tarefas faltantes entre si e com o servidor, após muitas trocas de ideias, pessoas indo dormir e quase 7h de trabalho ininterrupto. Aproximadamente metade da equipe (em torno de 10

peessoas) estavam unidas e conectadas no *Discord* da equipe entre 23:00 e 5:00, tendo eu, o Gabriel Ferreira (Ages I), o Luiz Mello (Ages IV) e o Mateus Caçabuena (Ages I) ficado até o final (6:30 aproximadamente) para realizar o envio do código para o GitLab e atualizar as tarefas concluídas nesta maluquice. No dia seguinte, após a apresentação, conversamos em equipe sobre o ocorrido e discutimos sobre a “cultura do herói”, em que uma pessoa ficaria responsável por fazer tudo na véspera e como isso era negativo tanto para o herói quanto para o time, o que é algo que entendo bem e concordo, embora não ache que tenha sido o caso deste episódio, visto que apesar de sim, ter sido apenas 1 pessoa (eu) digitando o código, foi um trabalho em equipe, não estava apenas programando por terminar de qualquer jeito ou sem ajuda, todos participavam com ideia e comandos, por exemplo, então acredito que tenha se tratado de, no máximo, uma “equipe de heróis”, que também não é o adequado – idealmente tudo estaria melhor organizado e não chegaríamos neste ponto – mas, dadas as circunstâncias adversas e pouco tempo de sprint, acredito que foi um momento proveitoso e que gerou uma discussão ainda melhor e mais profunda sobre o estado do time.

Com isso, aprendemos que se deve abstrair mais nas *tasks* e priorizar funcionalidade antes de estilo, pois é melhor “funcionando, mas visualmente não atrativo” do que “muito elaborado, mas sem funcionalidades”, o que é uma frase recorrente no projeto. Com motivos de sobra, sabíamos que era vital planejar muito bem a última sprint, e dividir melhor as *tasks* e seus requisitos, já que apesar de termos feito muito, não havíamos ainda conseguido achar uma divisão adequada entre a equipe, o que nos leva ao próximo ponto levantado pelo Pedro Carlucci (Ages II) e por mim durante a retrospectiva da sprint – a divisão do time em 3 squads. Basicamente, depois de muitos argumentos, mas poucas objeções, concordamos que o time estar dividido era um dos motivos de haver tantos “desentendimentos” e falta de comunicação mais global, visto que cada grupo focava apenas na sua bolha cujo escopo era delimitado por 1 Ages IV responsável por cada equipe. Isso não é uma prática ágil e, claramente, tampouco era produtiva e, por isso, decidimos que seria melhor abolir de uma vez a ideia de squads para termos um único time – diversos indivíduos – cada um com autonomia e responsabilidade para escolher qual tarefa realizar e a quem recorrer, visto que isso proporcionaria maior liberdade e dinamismo para todos.

2.3.5 Sprint 4

Nesta sprint, por ser a última, naturalmente boa parte dos nossos esforços tinham que ser direcionados para correção de bugs e dívidas técnicas das sprints passadas, e assim foi. Eu decidi que gostaria de seguir ajudando meus colegas no *front*, por ter mais facilidade com a área conforme já citado, mas que também gostaria de aproveitar estas últimas semanas para contribuir e aprender também no *back end*, então além de ter acompanhado e auxiliado diversas tarefas do *front end* – o que foi possível graças a dissolução das squads decidida na semana anterior - peguei para mim uma tarefa no *back end* de converter o PDF e seus veículos para um arquivo CSV, que era um dos requisitos do sistema.

Do *front end*, auxiliei o Pedro Carlucci (Ages II) a fazer o botão de deletar PDF, o Gabriel Ferreira (Ages I) e o Kevin Ribas (Ages III) a fazerem o *grid* (template) dos dados de cada veículo de um PDF - para deixar mais similar ao Figma proposto inicialmente aos clientes - a Carolina Ferreira (Ages I) a implementar o botão de salvar os campos copiados neste grid, o Davi a implementar setas para navegar entre os veículos de um PDF, o Guilherme de Moraes (Ages I) a implementar os testes automatizados de ponta a ponta utilizando o Cypress e depois o Mateus Caçabuena (Ages I) a fazer o botão de exportar para JSON e CSV os PDFs, que integrava com minha própria tarefa do *back end*. Consegui concluir minha tarefa de exportar os dados, com ajuda do Pedro Carlucci (Ages II), que estava mais familiarizado com o ambiente e, principalmente, com a linguagem utilizada (Python), mas não sem dificuldades.

Apesar de não ter me designado nenhuma tarefa em específico do *front end*, desenvolvi algumas soluções individualmente para auxiliar meus colegas com alguns problemas encontrados, um dos quais era a complexidade de compartilhar dados entre diversos arquivos e módulos envolvidos na página de visualização de PDF. Ainda, no *back end* tive algumas dificuldades com a sintaxe e a lógica da linguagem, mas esta parte foi relativamente tranquila de resolver, o que me trouxe “mais dor de cabeça” foi a questão de transformar o veículo - um objeto com várias propriedades “aninhadas” (ver modelo na sessão 2.2.2) – e suas propriedades para um arquivo CSV, que não possui esta complexidade disponível, para qual a solução foi converter os campos complexos em *strings* separadas por uma barra horizontal (-), o que podia ser feio, mas era uma entrega viável, relativamente simples e que cumpria seu propósito.

Muito pode ser aprendido com esta última sprint, em que muito ficou por entregar, visto que muito havia por fazer - não é que ficamos devendo alguma coisa imprescindível para uma entrega de qualidade, mas certamente não conseguimos fazer tudo que o time tinha visionado para o projeto. Ainda, mais pessoalmente, aprendi muito durante todo o projeto, mas certamente nesta última sprint foi a que eu aprendi mais - me arrisquei no *back end* em outra linguagem, participei de diversas tarefas não relacionadas simultaneamente, o que é um pouco confuso, mas certamente gratificante por ver que consegui contribuir e aprender muito com muitas pessoas. Principalmente, fico feliz de não ter feito mais "por conta", o que é algo que eu normalmente teria feito, mas optei por ter apenas 1 tarefa individual (mesmo querendo fazer mais, hoje vejo que não teria conseguido entregar e poderia ter prejudicado a equipe) para conseguir justamente ajudar o time nesta etapa final, em que o importante era entregar o essencial, já que uma entrega "feia, mas funcional" é melhor do que uma entrega "bonita e com bugs", citando nosso ilustre professor orientador Daniel Callegari.

Agora os próximos passos são revisar os códigos, artefatos e documentos de todo o projeto, apresentar o resultado para os stakeholders, agradecer aos colegas, professores e envolvidos pelo bom processo, e esperar pelo que virá pela frente. Possivelmente, haverá um feedback do cliente no final do ano sobre a produtividade adquirida pelo uso da solução ou então se esta não foi capaz de cumprir sua função e acabou por não ser inutilizada, ambos sendo respostas muito úteis para o futuro da agência e nosso como integrantes de uma comunidade cujo objetivo maior é a solução de problemas práticos.

2.4 Conclusão

Sou muito contente com minha participação no projeto, e espero continuar assim pela frente. Pelo lado social, conversei com a maioria dos meus colegas pelo menos uma vez e trabalhei também com diversos deles, independente da squad; acompanhei o Arthur Ibarra (Ages III) em uma parte da infraestrutura e de testes no *back end*, acompanhei parcialmente o Lucas Susin (Ages III) no esqueleto do *front-end*, dei umas sugestões de organização para o Gabriel Fanto (Ages IV) sobre o Discord e o Google Drive da equipe, acompanhei uma modelagem dos dados a serem salvos no banco com o Pedro Carlucci (Ages II), auxiliei na implementação dos testes “e2e” com o Guilherme da Silva (Ages I), dentre outros. Em geral, acredito que consegui me relacionar bem com todos - e fui até elogiado por diversas retrospectivas de Sprint, tendo recebido elogios junto dos Ages III (que para mim foram “as estrelas” do projeto), o que é algo que me deixa realmente contente e orgulhoso de ter passado um impacto, acredito, tão positivo. Na parte técnica também aprendi muito e pude contribuir em áreas variadas, embora tenha dedicado a maior parte do meu tempo a programar no *front end*, não me limitei a isso e procurei entender bem de todo o processo tanto de servidor quanto nuvem e banco de dados. No início do projeto eu estava mais focado na lógica de *back end* e acabei estudando um mínimo de Python para possivelmente montar a API, mas essa responsabilidade acabou sendo postergada e não voltei a trabalhar efetivamente nisso até a última semana do projeto, durante a última sprint. Já no *front end* eu tinha grande familiaridade com *React* e *TypeScript*, mas fiquei bem contente de poder trabalhar com *Material UI* e *Styled Components*, duas bibliotecas que de estilização com que nunca havia trabalhado; achei elas muito interessantes e certamente estudarei mais sobre elas, embora ainda prefira outras soluções, pelo menos com o uso que tivemos destas durante o projeto.

Embora seja provável que, caso perguntados, meus colegas digam que minha contribuição maior foi no *front end*, visto que foi realmente a parte em que mais trabalhei - em função de tempo (e por estar em constante alteração) - não foi a área em que mais evolui individualmente, como citado no item 2.2.4, a área em que mais contribui e aprendi foi, na minha visão, o Figma. Eu sempre tive um interesse em conseguir criar designs e protótipos, mas nunca tinha tido essa possibilidade em um projeto real, então decidi me focar bastante nisso, tendo criado componentes que serviriam de base para a construção do *front end*, com diversas variações que foram usadas para apresentar o protótipo para os *stakeholders*. Fiquei muito orgulhoso do

resultado e até consegui ajudar o Luiz (Ages IV), que fez o resto do design com meu conhecimento recém obtido.

Ao contrário das outras disciplinas, a AGES é uma cadeira quase completamente prática e que tem um escopo diferenciado dentro do curso, ainda assim, dentre as 13 cadeiras cursadas no primeiro ano de curso devo dizer que a cadeira mais importante foi a de Engenharia de Requisitos, que nos dá um primeiro contato com *stakeholders* reais e com termos e técnicas que utilizamos “em segundo plano”, quase que subconscientemente, dentro da agência. O próprio conceito de *stakeholder*, algumas metodologias de desenvolvimento e a questão de protótipos de alta fidelidade que serviram de base para o Figma podem ser citados como exemplos da cadeira que, apesar de não ser oficialmente pré-requisito para a AGES, é certamente uma base importante. Outras disciplinas como Gerenciamento de Configuração de Software e Programação Orientada a Objetos também são importantes, mas servem mais como bases genéricas e mais teóricas do que fundações para o projeto real.

O projeto Veículos via Montadora foi meu primeiro projeto dentro da agência, então ainda não posso julgar se foi o melhor em que já trabalhei, mas tem um espaço especial reservado e foi certamente uma ótima base para os outros projetos que virão. Os colegas, as tecnologias - em suma, toda a experiência - foi muito recompensadora.

A integração entre os elementos citados acima pode ser citada como um ponto positivo, e acredito que conseguimos nos organizar de tal maneira que não deixasse ninguém descontente com as tecnologias trabalhadas. Certamente muito poderia ser melhorado, mas considerando o escopo de aprendizado e como uma primeira experiência, não posso apontar nada que não tenha sido resolvido ao decorrer do projeto, com exceção de uma coisa que gostaria de contribuir mais no futuro que é a questão dos Ages I no começo do projeto – acredito que possamos ser mais incluídos em questões organizacionais, até para remover um peso dos Ages III e IV, mas principalmente, já que este é para ser nosso primeiro contato com o mercado, deveríamos ser mais consultados e empurrados desde o princípio, visto que são várias coisas pequenas que vão acumulando que acabam pesando para nós em questão de quantidade de matérias novas para aprender em tão pouco tempo.

O projeto foi finalizado e entregue oficialmente quarta-feira, 14 de julho de 2023, e o cliente ficou feliz com a entrega e com a solução apresentada, tendo prometido levar o projeto para testarem internamente e nos dar um feedback. Não conseguimos

entregar tudo que tínhamos em mente, conforme citado no ponto 2.3.5, mas entregamos o que era primordial e certamente o que apresentamos é muito útil e será de grande valor para o usuário final.

No geral, o projeto foi muito bom, o time era participativo e dialogava bastante – embora, claro, existam pessoas e pessoas cada qual com uma personalidade e preferências – e consegui me integrar bem. Não tenho grandes arrependimentos dentro desta primeira experiência, sempre é mais fácil olhar para trás e apontar o que teríamos feito melhor ou mesmo não feito, mas em geral atingi meus objetivos pessoais; conheci pessoas novas, trabalhei em uma equipe muito maior do que estou habituado, não fiquei parado em nenhum momento, sempre estava contribuindo e dando ideias, participei de todas as discussões que pude, fui curioso e corri atrás do que não entendia, seja com ajuda de alguém da equipe ou por meios externos. Foi uma experiência emocionante e gratificante, e que sem dúvidas valeu cada segundo das quase 200h dedicadas no semestre para isso.

3. PROJETOS AGES II – GLOBO APLAUSOS 2023

3.1 Introdução

O projeto Globo Aplausos, ao contrário do meu último projeto na agência, é um projeto bem grande e muito mais complexo, tanto em questão de quantidade de usuários quanto de estrutura - principalmente de armazenamento de dados - que é onde há um enfoque maior por minha parte, por ser o objetivo da segunda passagem pela agência. Novamente tive a oportunidade de fazer parte de um projeto para uma empresa muito reconhecida no Brasil (e no mundo), desta vez o grupo globo, {{ inicialmente }} representado pela *stakeholder* Luciane Fortes, *Project Manager* do grupo, especificamente da área de tecnologia e dos serviços online da plataforma.

A empresa é muito grande e possui diversos colaboradores, porém – ainda mais por causa dos eventos recentes (pandemia) que mudaram a forma como o trabalho é levado em todo o mundo – nem todos se conhecem tão bem ou conseguem trocar tantos *feedbacks*. Com isso em mente, julgou-se necessário criar uma plataforma que permitisse fortalecer as relações interpessoais e que incentivasse os colaboradores a fomentarem um ambiente de trabalho mais positivo e de mútuo reconhecimento.

Por conseguinte, nosso objetivo foi estruturado com o grupo Globo e concordou-se em desenvolver uma plataforma *Web* (digital) na qual todos os participantes recebessem mensalmente um determinado saldo - que se optou por chamar ‘aplausos’ com o qual o usuário poderia ‘aplaudir’, isto é, reconhecer e valorizar, seus colegas. Com estes aplausos recebidos, seria desenvolvida uma ‘loja de recompensas’ através da qual poder-se-ia trocá-los por recompensas ‘reais’ e tangíveis; tais quais vale-presentes, ingressos para cinema, assistentes inteligentes como a Alexa da Amazon, entre outras inúmeras alternativas.

Em termos mais técnicos, outras funcionalidades – não tão centrais, embora igualmente importantes – que se pode citar são: autenticação de usuário, ou seja, fazer login na plataforma apenas caso faça parte do grupo Globo, gerenciamento de perfil e histórico dos *feedbacks* e transações, isto é, visualizar para poder ter um controle dos elogios feitos e recebidos e de pelo que seu ‘reconhecimento’ foi trocado. A última entrega, que não se sabia ao começo do projeto se seria possível entregar com a mesma qualidade do resto do projeto seria um painel administrativo, para que

alguns seletos usuários pudessem alterar certas funcionalidades da plataforma, tais quais; recompensas (e seus preços), o saldo que cada participante receberia ao começo de cada mês, se esta distribuição seria mensal ou então em alguma outra frequência e, por fim, visualizar quais usuários estavam mais ou menos ativos.

Foi iniciado em 07/08/2023 e entregue no dia {{ 20/11/2023 }}, tendo sido acompanhado e orientado pela professora Alessandra Costa Smolenaars Dutra (na imagem, à esquerda, na segunda fileira). Abaixo, segue foto do time responsável pelo desenvolvimento.

Figura 3 – Foto da Equipe



Fonte 3: <https://tools.ages.pucrs.br/globo-aplausos/wiki/-/wikis/uploads/8d58cb4227d49b78816d28e2876e0029/time.jpeg>

3.2 Desenvolvimento do Projeto

3.2.1 Repositório do código Fonte do Projeto

O projeto foi dividido em 2 repositórios de código, cujos links podem ser encontrados abaixo:

- *Front end*: <https://tools.ages.pucrs.br/globo-aplausos/globo-aplausos-frontend>
- *Back end*: <https://tools.ages.pucrs.br/globo-aplausos/globo-aplausos-backend>

Além de um repositório para a *Wiki* do projeto, com toda a documentação e outras informações relevantes:

- Wiki: <https://tools.ages.pucrs.br/globo-aplausos/wiki/-/wikis/home>

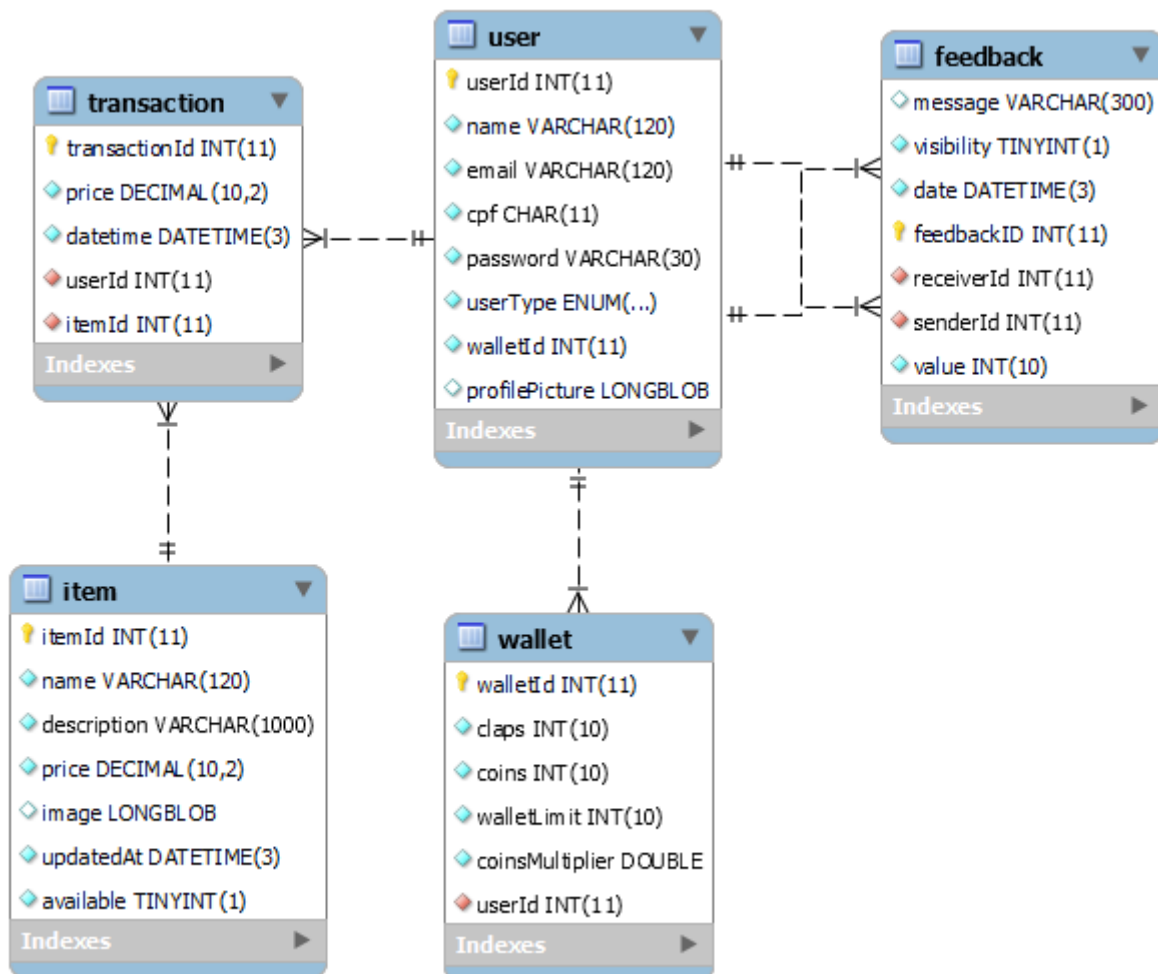
3.2.2 Banco de Dados utilizado

Como ages II, nosso papel principal é justamente o de modelar e estruturar o melhor banco de dados possível para a solução implementada, desde a idealização, tabelas e tipos de dados até a criação delas em algum provedor de banco de dados (SGBD) e futura manutenção, caso necessário. Ainda, é necessário optar por algum dos ‘paradigmas’ de banco de dados, os mais famosos sendo; o ‘relacional’ - que possui a estrutura mais sólida, ou seja, maior estruturação dos dados, maior segurança com relação a integridade dos dados e que possui uma linguagem de *scripting* mais amplamente conhecida, o *SQL*; o ‘não relacional’ - que é mais flexível e ideal para dados que variam muito ou então são muito grandes (como foi o caso do meu primeiro projeto na agência – Veículos via Montadora (ver seção 2.1); e o modelo de grafos, que é ideal para relacionamentos complexos (como redes sociais, por exemplo) e que é muito eficiente para casos mais específicos, embora seja menos utilizado por causa desta especificidade.

O modelo de banco de dados escolhido para o projeto foi o relacional, devido a sua simplicidade e facilidade de implementação. Ainda, o modelo relacional é o modelo mais utilizado atualmente, e é o modelo com que a equipe possui mais experiência e conhecimento, o que tornou a decisão muito simples e direta, que foi um dos pontos – a meu ver – que nos levou a construir um bom banco de dados.

Após definido o modelo, foi necessário escolher o provedor de banco de dados que seria utilizado e, novamente concordamos logo de início que o *MySQL* deveria ser o adotado. Ele é um dos *SGBD's* mais utilizados atualmente, e é também um dos mais utilizados pela equipe, o que facilita na manutenção e no desenvolvimento do projeto. Abaixo, segue um diagrama que representa as tabelas e relacionamentos criados para o projeto.

Figura 4 – Diagrama de Banco de Dados



Fonte 4: <https://tools.ages.pucrs.br/globo-aplausos/globo-aplausos-backend/-/raw/development/prisma/schema.png>

3.2.3 Arquitetura utilizada

A arquitetura do projeto, isto é, sua estrutura e mapeamento de relação entre as partes (*front end*, *back end* e banco de dados) é relativamente simples, e seguiu um padrão muito convencional e ‘padrão’ no mercado, denominado *Model View*

Controller (MVC), que basicamente permite a separação entre as responsabilidades de uma aplicação. Mais informações sobre esta parte da arquitetura foram melhor explicadas pelo Henrique (ages III) e podem ser encontradas na *Wiki* do projeto, no link a seguir: <https://tools.ages.pucrs.br/globo-aplausos/wiki/-/wikis/arquitetura>. Abaixo, segue uma imagem do diagrama de deploy e de relação entre os componentes.

Além desta parte de servidor, que é o foco maior da terceira passagem pela agência, é necessário planejar a integração entre nosso servidor (o ‘*back end*’) e o banco de dados – por meio do *SGBD* escolhido (ver seção 3.2.2). Para fazer esta conexão, decidimos utilizar um *ORM*, cujo trabalho principal é, primariamente, transformar a linguagem específica do banco de dados na linguagem de programação utilizada no código do servidor e vice-versa. Optamos pelo uso da biblioteca *Prisma*, que é muito comumente utilizada principalmente no meio de *Node* e *TypeScript* (ver seção 3.2.5), e que disponibiliza uma interface própria para a criação de tabelas e instruções a serem enviadas para o banco de dados, além de outros.

3.2.4 Protótipos das telas desenvolvidas

A *stakeholder* do projeto foi muito aberta em relação ao design a ser implementado, e não tinha nenhuma ideia pré-pronta e que devesse ser seguida, dando espaço para a criatividade do time. Para ver o resultado final desenvolvido, segue um link para a Wiki com todas as telas e componentes produzidos no Figma, documentado pelo Arthur (ages I): https://tools.ages.pucrs.br/globo-aplausos/wiki/-/wikis/design_mockups.

3.2.5 Tecnologias Utilizadas

Atualmente, é muito incomum encontrar algum projeto que não utilize ferramentas de desenvolvimento colaborativo, como o *Git*, no caso, através do *GitLab* – plataforma própria da agência para versionamento de código entre as equipes e o *Docker*, que não é necessário, mas ajuda na questão de certificar que todos estão desenvolvendo sob as mesmas condições de ambiente. Abaixo, estão citadas as outras tecnologias utilizadas, acompanhadas por uma breve descrição e o link da sua documentação.

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

- *Docker*: <https://docs.docker.com/get-started/>

Front end:

Foi desenvolvido em *TypeScript* com auxílio das bibliotecas *NextJS* (para estrutura e lógica de componentes) e *MaterialUI*, unido ao *CSS* 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, vale ressaltar que o *NextJS* utiliza *React* internamente, e funciona para melhorar a construção e distribuição do Software. Por fim, utilizamos o *NPM* como gerenciador de dependências {{ e, para realizar os testes automatizados utilizamos o *Cypress* }}.

- *TypeScript*: <https://www.typescriptlang.org/docs/>
- *NextJS*: <https://nextjs.org/docs>
- *Material UI*: <https://mui.com/material-ui/getting-started/overview/>
- *CSS*: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>
- *Prettier*: <https://prettier.io/docs/en/index.html>
- *ESLint*: <https://eslint.org/docs/latest/>
- *Husky*: <https://typicode.github.io/husky/>
- *React*: <https://react.dev/learn>
- *NPM*: <https://docs.npmjs.com/>
- *Cypress*: <https://docs.cypress.io/>

Back end:

Desenvolvido também em *TypeScript*, com o *framework* *NestJS* para o servidor, utilizando o *NPM* para gerenciamento de dependências – as principais delas sendo *Prisma* (ver seção 3.2.3) e *Jest* (para testes unitários).

- *NestJS*: <https://docs.nestjs.com/>
- *Prisma*: <https://www.prisma.io/docs>
- *Jest*: <https://jestjs.io/pt-BR/docs/getting-started>

3.3 Atividades desempenhadas pelo aluno no projeto

3.3.1 Sprint 0

Tal qual no projeto passado (ver seção 2.3.1), a primeira *sprint* do projeto - denominada aqui 'Sprint 0' - é uma espécie de 'aquecimento' para a equipe, de alinhamento de expectativas com a cliente e para que todos possam estar de acordo com relação ao que será estudado e, posteriormente, desenvolvido. Nesta seção tratarei da minha participação na Sprint 0, farei alguns paralelos com minha primeira passagem pela agência e buscarei resumir o que foi atingido pela equipe nas primeiras semanas.

As tecnologias utilizadas de *front end*, *back end* e banco de dados foram melhor explicadas anteriormente (ver seções 3.2.5 e 3.2.2), e não dediquei particularmente nenhum tempo muito grande de estudo para elas pois já tinha tido experiência com a maioria. Consequentemente, vou buscar abordar aqui minha participação e visão das outras partes do projeto - como a distribuição de tarefas e desempenho da equipe nas áreas para as quais cada 'nível' (ages I – IV) é esperado contribuir.

Inicialmente, imaginei que aconteceria algo similar ao que aconteceu no meu último projeto (ver seção 2.3.1), no qual não houve muita proatividade e o Figma, principalmente, acabou sendo desenvolvido por 2 pessoas enquanto os outros ficavam estudando. Certamente houve e há muito estudo dentro do projeto, mas não em detrimento dos *mock ups* – todos do time, sem exceção, contribuíram para montar as telas e a navegação esperada do sistema. Em especial, deixo aqui um reconhecimento ao Bernardo Müller (ages II), que montou o menu e o cabeçalho, e ao Arthur Both (ages I), que fez toda a prototipação e navegação esperada, apesar da falta de experiência. Foi muito bom ver que a tarefa de telas foi bem distribuída, dividida e realizada por toda a equipe em conjunto, tendo sido muito elogiada na apresentação (que será tratada mais a frente).

Normalmente, a tarefa dos ages II (principalmente, banco de dados) é desenvolvida depois da Sprint 0, ou então depois de todos dedicarem muito dos seus esforços para contribuir com o Figma, o que não foi necessário devida a adoção geral do desenvolvimento das telas, o que permitiu com que nós (ages II) nos focássemos em desenvolver desde o princípio as estruturas e tipos de dados que viriam a ser utilizados sem grandes mudanças mais a frente no projeto.

Adicionalmente, a 'gerência' e estrutura do projeto em geral, que são lideradas pelos ages III e IV, me gerou sentimentos mistos. Os ages III, apesar de posteriormente terem melhorado, deram início ao projeto e suas tecnologias muito

individualmente; não senti que eles estavam aptos a mudar as ferramentas utilizadas e tampouco consultaram a equipe sobre estas – apenas notificaram no começo que já haviam criado os repositórios com suas devidas bibliotecas e para que as estudássemos para poder contribuir. As ages IV, por outro lado, dividiram a equipe em duplas e dividiram para cada qual um componente por desenvolver, tendo sido muito abertas a sugestões e trocas, o que gerou muita cooperação no time e foi o que permitiu bons resultados desde o começo.

Com isso, aprendemos que a comunicação, de fato, é sempre o processo mais importante em qualquer projeto. Embora não tenha aprendido muito tecnicamente – por falta de necessidade –, gostei muito do modo como foi levado o projeto pelas ages IV e certamente vou tentar fazer algo próximo nas minhas próximas experiências dentro e fora da agência.

Para a próxima sprint, depois de decidida toda a tecnologia e terminadas as telas para apresentação, montou-se uma apresentação com sugestão de entregas para a cliente, que aprovou prontamente e elogiou muito tanto o que havia sido feito quanto a divisão de tarefas e de como seriam desenvolvidos os próximos passos do projeto – o que é um feedback muito importante, visto que a *stakeholder* também é gerente de projetos e entende bem do processo. Estes próximos passos previstos eram, na maior parte, pegar os componentes montados nos protótipos apresentados e transformá-los em componentes que pudessem ser utilizados dentro do código, além de desenvolver as histórias de usuário (*US's*) 1 - fazer login, 2 – fazer logout, 3 – acessar a página inicial e 4 – visualizar saldo de moedas, que serão tratadas na seção seguinte.

3.3.2 Sprint 1

Esta sprint teve início em 23/08, após o aceite da sprint anterior pela cliente, e foi encerrada em 13/09, ou seja, teve uma duração de aproximadamente 3 semanas. Conforme citado na seção anterior, nosso foco era desenvolver as quatro primeiras histórias de usuário e, concorrentemente, todos os componentes da aplicação, o que foi uma estratégia que eu nunca tinha visto antes e que foi muito elogiada pelo time inteiro incluindo a professora orientadora. A estratégia apesar de não gerar uma entrega inicial tão ‘completa’, ou seja, não envolver telas completas desenvolvidas ou grandes funcionalidades, permite uma aceleração do desenvolvimento futuro, ou seja, funciona como uma ‘mola’ para o projeto – é necessário “ir para trás” um pouco para que se possa ir mais rápido para a frente depois.

Novamente, o time foi dividido em duplas, todas compostas por um *ages I* e outro *ages II*, mas por causa do número ímpar de *ages I* minha tarefa acabou sendo composta por mim (*ages II*), pelo Lucas Vieira (*ages I*) e pelo Murilo Vargas (*ages I*), e seu objetivo era construir o componente de ‘select’ – para permitir o usuário visualizar uma lista de usuários e selecionar um (para poder ‘aplaudir’). Nos dividimos internamente entre lógica e estilização, tendo o Murilo ficado responsável pela lógica do componente, isto é, receber uma lista de usuários, criar o componente no código e permitir a seleção de um item da lista, visto que ele já tinha experiência em desenvolvimento web, e eu acabei ficando responsável por auxiliar o Lucas na parte de design do componente, o que conseguimos fazer depois de ler melhor a documentação da biblioteca de componentes (ver seção 3.2.5).

Finalizado o componente, fiquei disponível para auxiliar os outros colegas com suas devidas tarefas, visto que eu já tinha mais experiência com os recursos utilizados e vontade de ajudar. Todos conseguiram terminar seus componentes eventualmente e a equipe foi elogiada pelo trabalho rápido e eficiente. Além destas tarefas mais específicas, decidi por fazer a documentação do banco na *Wiki* do projeto para que todos pudessem entender o funcionamento, estrutura e, principalmente, como instalar o ambiente, e alguns comandos úteis da aplicação para permitir um uso tranquilo e com poucos problemas por todos os membros da equipe.

Em suma, apesar de não termos entregue muita coisa “tangível”, conforme mencionado anteriormente, optamos por desenvolver uma base sólida antes de elaborar muito – o que se provou uma estratégia de muito sucesso. Assim sendo,

esperamos nas próximas *sprints* conseguir desenvolver bem mais da aplicação e fazer uma entrega que cada vez gere mais valor para o cliente.