

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

GABRIEL RABELO ALMEIDA

**MEMORIAL DE ATUAÇÃO NA AGÊNCIA EXPERIMENTAL DE ENGENHARIA
DE SOFTWARE - PERÍODO 2019 A 2022
AGES I, II, III E IV**

Porto Alegre
2022

GABRIEL RABELO ALMEIDA

**MEMORIAL DE ATUAÇÃO NA AGÊNCIA EXPERIMENTAL DE ENGENHARIA
DE SOFTWARE - PERÍODO 2019 A 2022
AGES I, II, III E IV**

Trabalho apresentado à Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Alessandra Costa Smolenaars Dutra

Porto Alegre
2022

Dedico este trabalho à minha mãe e todas as pessoas
que acreditaram sempre nos meus estudos
e me encorajaram a mudar de vida.

*“Não existe vento favorável
a quem não sabe onde deseja ir.”*
(Sêneca)

RESUMO

Este memorial descritivo traz, na visão do aluno, seu portfólio acadêmico que contém os acontecimentos, sucessos e oportunidades de melhoria em toda a trajetória do estudante ao longo do curso Engenharia de Software, nos projetos distribuídos por este, relacionados à Agência Experimental de Engenharia de Software (AGES). As etapas presentes nesse relatório se dão em níveis de um a quatro, e em cada um dos níveis é dado ênfase a uma área específica da engenharia de software, sendo: desenvolvimento de código e testes (AGES I), prototipação da interface com o usuário, definição e manutenção de estrutura de bancos de dados (AGES II), arquitetura de *DevOps* (AGES III), e por último, gerência de projetos (AGES IV).

PALAVRAS-CHAVE: AGES. Engenharia de Software. PUCRS. Experiência do Usuário. Engenharia de Requisitos. Programação. Banco de Dados. Arquitetura de Software. Gerenciamento de Projetos. Desenvolvedor.

LISTA DE ILUSTRAÇÕES

Figura 1 – Equipe no início do projeto Mapas Culturais	13
Figura 2 – Diagrama de <i>Deploy</i>	15
Figura 3 – Diagrama de pacotes back-end	15
Figura 4 – Diagrama de pacotes front-end	16
Figura 5 – Diagrama de classes	16
Figura 6 – Diagrama lógico de banco de dados	17
Figura 7 – Diagrama de Arquitetura	23
Figura 8 – Diagrama Conceitual	24
Figura 9 – Diagrama Lógico	24
Figura 10 – Protótipo da tela inicial	25
Figura 11 – Protótipo da tela de uma pergunta	26
Figura 12 – Protótipo da tela de edição do perfil de usuário	26
Figura 13 – Equipe de desenvolvimento, clientes e professor orientador.	32
Figura 14 – Padrão arquitetural MVC	35
Figura 15 – Diagrama de arquitetura geral	35
Figura 16 – Diagrama genérico de deploy	36
Figura 17 – Diagrama de deploy AWS	37
Figura 18 – Parte 1 da modelagem de classes	38
Figura 19 – Parte 2 da modelagem de classes.	38
Figura 20 – Tela de cadastro do cliente	39
Figura 21 – Tela inicial da área logada de um cliente	40
Figura 22 – Tela de pedidos na visão de parceiro	40
Figura 23 – Tela de entrega do usuário piloto	41
Figura 24 – Repositórios do projeto	42
Figura 25 – Gráfico de commits demonstrando o uso do GitFlow no backend.	42
Figura 26 – Equipe de desenvolvimento, clientes e professor orientador.	48
Figura 27 – Diagrama de componentes	51
Figura 28 – Diagrama de Infraestrutura	52
Figura 29 – Diagrama relacional do banco de dados	52
Figura 30 – Diagrama Lógico do banco de dados	53
Figura 31 – Protótipos iniciais de baixo nível	54
Figura 32 – Tela de cadastro de doador e detalhe de uma necessidade, respectivamente	54
Figura 33 – Tela de listagem de necessidades	55
Figura 34 – Corte da EAP referente ao planejamento para a Sprint 1	57
Figura 35 – Corte da EAP referente a Sprint 2	59
Figura 36 – Corte da EAP referente a Sprint 3	60
Figura 37 – Corte da EAP referente a Sprint 4	62

LISTA DE ABREVIATURAS E SIGLAS

AGES	Agência Experimental de Engenharia de Software	4
PROUNI	Programa Universidade Para Todos	10
CI	Centro de Inovação	10
API	<i>Application Programming Interface</i>	15
DOM	<i>Document Object Model</i>	14
MVC	<i>Model-View-Controller</i>	14
MVP	Mínimo Produto Viável	23
ORM	Object Relational Mapping	14
POC	Prova de Conceito	12
PUCRS	Pontifícia Universidade Católica do Rio Grande do Sul	10
UI	<i>User Interface</i>	33
SEO	<i>Search Engine Optimization</i>	21
TDD	<i>Test Driven Development</i>	64

SUMÁRIO

1	APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO	10
2	PROJETO AGES I — “MAPAS CULTURAIS — IBERÊ CAMARGO”	12
2.1	INTRODUÇÃO	12
2.2	DESENVOLVIMENTO DO PROJETO	13
2.2.1	TECNOLOGIAS	13
2.2.1.1	NODE.JS	13
2.2.1.2	REACT.JS	14
2.2.1.3	MYSQL	14
2.2.1.4	SEQUELIZE	14
2.2.2	ARQUITETURA	14
2.2.3	BANCO DE DADOS	16
2.3	ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO	18
2.3.1	SPRINT 0	18
2.3.2	SPRINTS 1 e 2	18
2.3.3	SPRINT 3	18
2.3.4	SPRINTS 4 e 5	19
2.4	CONCLUSÃO	19
3	PROJETO AGES II — “COMPARTILHANDO EXPERIÊNCIAS — TOWNSQ”	21
3.1	INTRODUÇÃO	21
3.2	DESENVOLVIMENTO DO PROJETO	21
3.2.1	TECNOLOGIAS	21
3.2.1.1	JAVA	22
3.2.1.2	SPRING BOOT	22
3.2.1.3	THYMELEAF	22
3.2.1.4	POSTGRESQL	22
3.2.2	ARQUITETURA	23
3.2.3	BANCO DE DADOS	23
3.2.4	PROTÓTIPOS	25
3.3	ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO	27
3.3.1	SPRINT 0	27

3.3.2	SPRINT 1	27
3.3.3	SPRINT 2	28
3.3.4	SPRINT 3	29
3.3.5	SPRINT 4	29
3.4	CONCLUSÃO	30
4	PROJETO AGES III — “APUS DRONES”	31
4.1	INTRODUÇÃO	31
4.2	DESENVOLVIMENTO DO PROJETO	32
4.2.1	TECNOLOGIAS	32
4.2.1.1	JAVA	32
4.2.1.2	SPRING BOOT	33
4.2.1.3	Flutter	33
4.2.1.4	AWS	33
4.2.1.5	EC2	33
4.2.1.6	ECR	34
4.2.1.7	S3	34
4.2.1.8	MYSQL	34
4.2.2	ARQUITETURA	35
4.2.3	BANCO DE DADOS	37
4.2.4	PROTÓTIPOS	39
4.2.5	REPOSITÓRIO	41
4.3	ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO	43
4.3.1	Sprint 0	43
4.3.2	Sprint 1	44
4.3.3	Sprint 2	45
4.3.4	Sprint 3	45
4.3.5	Sprint 4	46
4.3.6	Sprint 5	46
4.4	CONCLUSÃO	47
5	PROJETO AGES IV - “ADOTE UM AMANHÃ”	48
5.1	INTRODUÇÃO	48
5.2	DESENVOLVIMENTO DO PROJETO	49
5.2.1	TECNOLOGIAS	49

5.2.1.1	JAVA	49
5.2.1.2	SPRING BOOT	49
5.2.1.3	VUE.JS	50
5.2.1.4	POSTGRESQL	50
5.2.2	ARQUITETURA	50
5.2.3	BANCO DE DADOS	51
5.2.4	PROTÓTIPOS	53
5.3	ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO	55
5.3.1	SPRINT 0	55
5.3.2	SPRINT 1	57
5.3.3	SPRINT 2	58
5.3.4	SPRINT 3	60
5.3.5	SPRINT 4	61
5.4	CONCLUSÃO	62
6	CONSIDERAÇÕES FINAIS	64
	REFERÊNCIAS	66

1 APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

O começo da minha vida acadêmica no curso de Engenharia de software se deu de forma bastante conturbada, requerendo muito esforço e alto desempenho, pois vinha de um emprego em que conseguia bancar pouco mais que a minha sobrevivência. Com bastante esforço e apoio de minha mãe, obtive uma bolsa de estudos através do Programa Universidade Para Todos (PROUNI), o que facilitou o meu ingresso no que era o meu sonho profissional: ser um desenvolvedor de software.

No primeiro semestre, já percebi facilidades que eu tinha em criar conexões com colegas do curso, colaborando com o meu processo de integração, melhorando a possibilidade de ter apoio nas disciplinas que estariam por vir e até mesmo possibilitando indicações de trabalhos no âmbito profissional. Com essas conexões, fui informado de uma vaga de estágio que começara logo no segundo semestre da faculdade: a Aceleradora Ágil, programa que na época era uma parceria do Centro de Inovação (CI) da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) com a empresa *Thoughtworks*. Me apliquei para a vaga e fui chamado para ser integrante da 16^a turma, o que possibilitou a troca do meu serviço anterior, não relacionado ao curso, por uma oportunidade de crescimento.

Então chegando no segundo semestre, em 2019, tive duas imensas oportunidades de aprendizado: Meu primeiro estágio e o início do portfólio de projetos que o curso proporciona – a AGES I – que será abordado no próximo capítulo. O estágio foi totalmente sinergético com a ideia que esse curso traz: desenvolvimento ágil e moderno com equipes multidisciplinares e com uma certa autonomia, com grande ajuda de engenheiros de software sêniores.

Outra oportunidade nesse semestre que foi muito relevante foi a *Hackatona* da Engenharia de Software, que proporcionou uma experiência nesse tipo de evento que é muito praticado no mercado de Software. Essa primeira etapa se caracterizou pela busca de conhecimento técnico (*hard skills*), que aparentava ser um mar interminável de experiência a ser adquirida.

Na segunda etapa da AGES, seguido da maioria das disciplinas recomendadas pela grade curricular, obtive muito conhecimento na área de banco de dados e engenharia de requisitos, além de aprimorar ainda mais o lado desenvolvedor. Essa foi uma etapa muito importante, pois se deu de forma remota devido à pandemia do COVID-19. Junto disso, já me encontrava na minha segunda oportunidade de estágio, dessa vez com foco no desenvolvimento *back-end*, na empresa Sicredi. Utilizei essas chances para pôr em prática tudo que estava aprendendo durante o curso. Essa etapa AGES se caracterizou pelo início do interesse em aplicar os conceitos de testes unitários e de integração nos projetos de *back-end*.

Três anos após o início do curso, já no sexto semestre, estava com a missão de ser arquiteto do projeto da minha terceira AGES, buscando as melhores práticas que atendiam as nossas necessidades. Em paralelo, ocorria a minha efetivação na empresa em que estagiava, o que me deu o sentimento de finalmente estar engrenando o que um dia era apenas um sonho.

Como gerente de projeto na AGES IV e em vias de concluir o curso de engenharia de software, já obtive conhecimento em diversas áreas que tangem esse curso, o que me deu uma

visão muito transparente do que quero seguir fazendo, que é ser um Engenheiro de Software, com o sonho de evoluir para Arquiteto de Software. Como desafio dessa etapa final, busquei ampliar ainda mais minhas habilidades de comunicação (*soft skills*), sempre procurando um tom de humor para elevar o engajamento e animação de toda a equipe.

2 PROJETO AGES I — “MAPAS CULTURAIS — IBERÊ CAMARGO”

2.1 INTRODUÇÃO

O projeto que trabalhei como AGES I, tem como problemática a pouca visibilidade da cultura atualmente disponível de forma gratuita em Porto Alegre, com isso o produto pretende fomentar a participação não só de moradores da cidade como também turistas nas atrações culturais, tendo como público alvo jovens (entre 18 e 30 anos), estudantes, comunidade diversa e utilizam transportes públicos e de aplicativos como, por exemplo, ônibus, bicicletas e patinetes elétricos.

Para solucionar o problema, a ideia da aplicação era proporcionar uma mobilidade entre instituições culturais parceiras, disponibilizando informações sobre a instituição mais próxima, com informações de horários de funcionamento, formas de contato, imagens, opção de utilizar um roteiro pré-definido e customizável que sugira a próxima instituição a ser visitada, assim sendo possível visualizar as rotas através de um mapa que facilite e estimule este deslocamento.

Após o levantamento de requisitos no início do projeto foi possível identificar três principais *stakeholders*: Lucas Pierozan, assessor de comunicação do Iberê Camargo, que acompanhou as entregas do projeto, bem como disponibilizou informações referentes à *layout*, fontes específicas e imagens que alimentaram o banco de dados da aplicação. O futuro administrador do sistema, que segundo Lucas seria apenas uma pessoa responsável por isso no Fundação Iberê Camargo e teria como função cadastrar as informações referente às instituições parceiras que queiram fazer parte do contexto da aplicação. E, por fim, o usuário final, principal *stakeholder* do projeto. Desde o início pensamos no valor que entregaremos a este *stakeholder* e como podemos melhorar a experiência que este usuário vai ter utilizando nosso aplicativo.

Conforme o cronograma elaborado pelo professor orientador junto de dois alunos da AGES IV, o período de execução do projeto se deu entre agosto e novembro. Entre o início do desenvolvimento do que seria parte da primeira entrega ao *stakeholder* até a entrega final da Prova de Conceito (POC), houve entregas regulares dando visibilidade ao avanço gradual do projeto.

O projeto teve como professor orientador o Prof. Dr. Ricardo Bastos, que junto dos alunos AGES IV ajudou a conduzir as atividades planejadas para o semestre e acompanhou o desenvolvimento do projeto e dos demais alunos. Uma das funções que mais me chamou a atenção foi sua atuação no que é chamado *one-on-one*, uma reunião com cada aluno de forma individual, trazendo mais visibilidade para os objetivos pessoais de cada um durante o projeto. Além disso, no início do semestre o professor disponibilizou um questionário chamado *performance plan*, que serviu para determinar o que eu esperava de evolução técnica e de relações com os colegas nesse semestre e ao final dele, fazer um comparativo entre as expectativas iniciais e o resultado obtido.

Um registro do primeiro encontro pode ser observado na figura 1. Mais informações do projeto podem ser encontradas na página da wiki (WIKI, 2019b) do projeto.

Figura 1 – Equipe no início do projeto Mapas Culturais



Fonte: Wiki do projeto

2.2 DESENVOLVIMENTO DO PROJETO

2.2.1 TECNOLOGIAS

As tecnologias que descreverei a seguir foram escolhidas com base no conhecimento da equipe (WIKI, 2019c). Na época, o conhecimento em tecnologias que tinham o JavaScript como linguagem era predominantemente, portanto preferimos programar todo o sistema com base nisso.

2.2.1.1 NODE.JS

É um ambiente de execução JavaScript assíncrono orientado a eventos, diferente dos servidores antigos que eram baseados em *threads*, ou seja, o servidor fica online e segura uma conexão até ter a resposta da requisição, tornando a transação mais lenta do que o desejado. O node.js que foi utilizado para servir como *back-end* do projeto, o servidor em que é possível buscar as informações que ficam do lado do servidor e devem ser apresentadas para o usuário. Dan York pauta o seguinte processo que faz o Node.js ser eficiente: “Você usa o seu navegador para fazer uma requisição para a página “/about.html” em um servidor Node.js web. O servidor Node aceita a requisição e chama a função que vai retornar o arquivo do disco. Mas enquanto o

servidor Node espera a resposta a ser retornada, já atende outra requisição. Quando o primeiro arquivo é devolvido é feita uma função de *callback* que insere na fila servidor Node. Então o servidor executa a função que neste caso, renderiza a página “/about.html” e manda de volta para seu navegador.”

2.2.1.2 REACT.JS

Esta biblioteca de JavaScript é utilizada para criar a interface de usuário e faz parte da nossa *View* (conceito de *Model-View-Controller* (MVC) mostrado nas seções anteriores), utilizada e mantida pelo Facebook e outros desenvolvedores independentes, assim como outras companhias. React.js torna o processo de criação de interfaces muito menos complicado através de componentes encapsulados que gerenciam seu próprio estado. Como a lógica do componente é escrita em JavaScript e não em templates, é possível facilmente passar diversas categorias de dados ao longo da sua aplicação e ainda manter o estado fora do *Document Object Model* (DOM). A documentação utilizada para realização do projeto pode ser encontrada no site oficial do React (REACT, 2022).

2.2.1.3 MYSQL

Segundo o site Hostinger, MySQL é um gerenciador de banco de dados que utiliza a linguagem SQL como interface. SQL é uma sigla para “Structured Query Language” que utiliza um modelo relacional de dados. A partir do MySQL um ou mais dispositivos (clientes) conectam em um servidor através de uma rede específica. Qualquer cliente consegue fazer uma requisição pela interface (GUI) nas suas telas e o servidor vai entregar a saída desejada. Basicamente é seguido o seguinte roteiro: MySQL cria um banco de dados para guardar e manipular os dados e definir a relação de cada tabela. Clientes fazem requisições utilizando comandos específicos da linguagem SQL no MySQL. O servidor irá responder com a resposta requisitada e irá apresentar para o lado do cliente.

2.2.1.4 SEQUELIZE

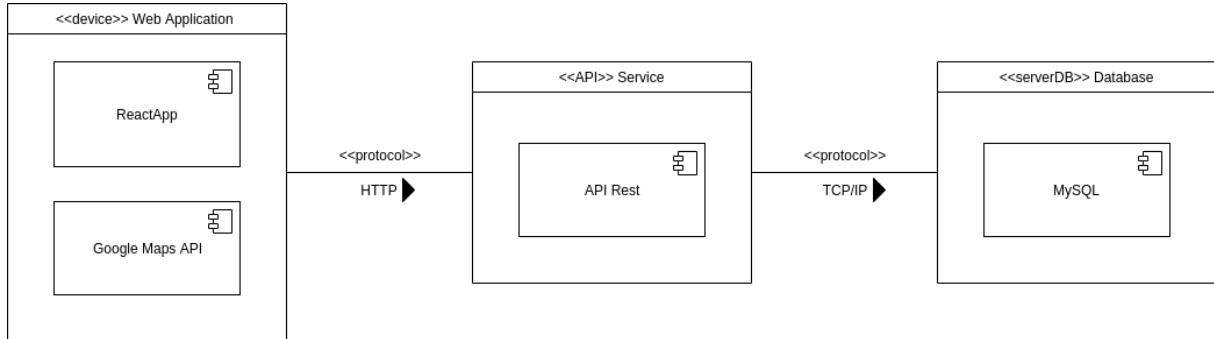
Para otimizar o uso do MySQL, foi utilizado um Object Relational Mapping (ORM): o Sequelize, sendo uma biblioteca que ajuda a mapear informações entre objetos e relações do banco de dados como MySQL. Ele dá uma melhor configuração para os desenvolvedores economizando tempo e esforço para focar em outros problemas. Ajuda a gerenciar tabelas e popular elas quando necessário para o desenvolvimento.

2.2.2 ARQUITETURA

A arquitetura definida pela equipe foi um modelo MVC, como não tinha domínio sobre arquitetura, não tive participação nesse processo de escolha. Conforme mostra a figura 2,

a separação do sistema se dá em *front-end*, *Application Programming Interface* (API) Rest (*back-end*) e banco de dados.

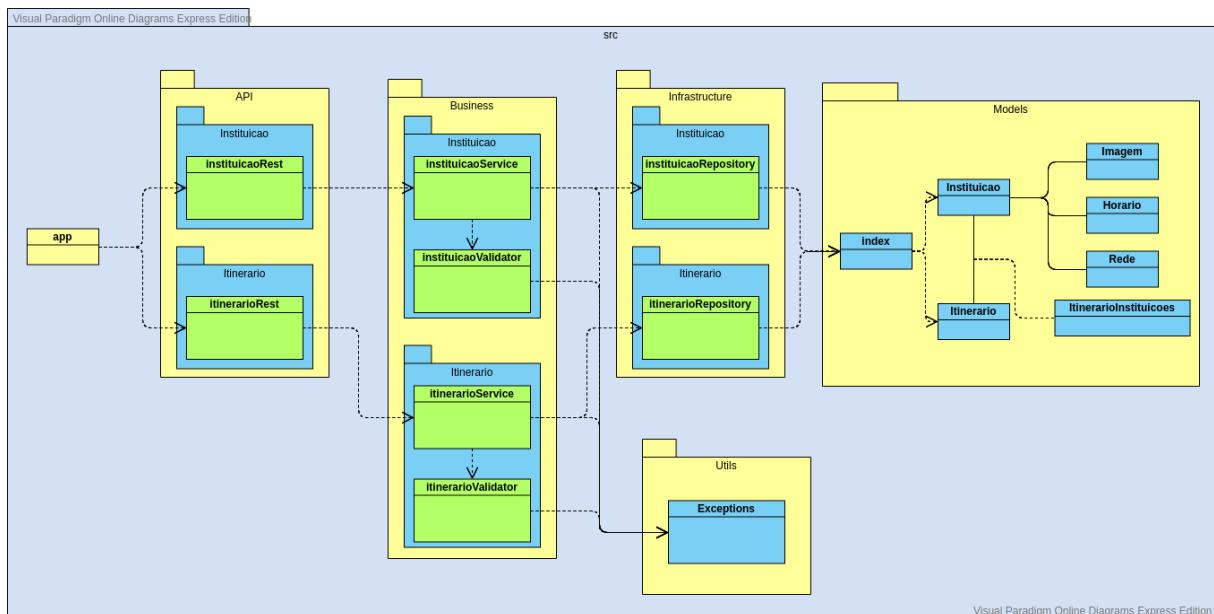
Figura 2 – Diagrama de Deploy



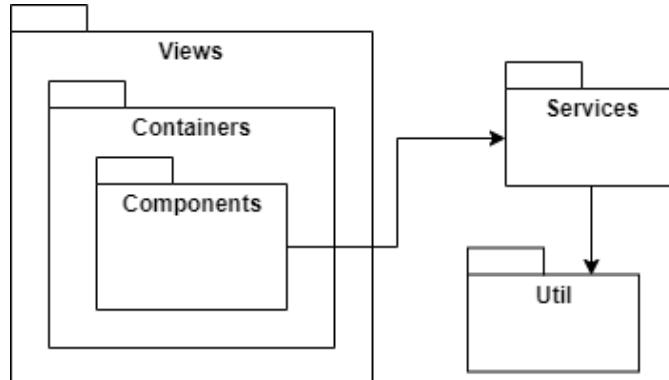
Fonte: Wiki do Projeto

As figuras 3 e 4 demonstram a separação de pacotes utilizada nos projetos do *back* e *front-end*, respectivamente.

Figura 3 – Diagrama de pacotes back-end



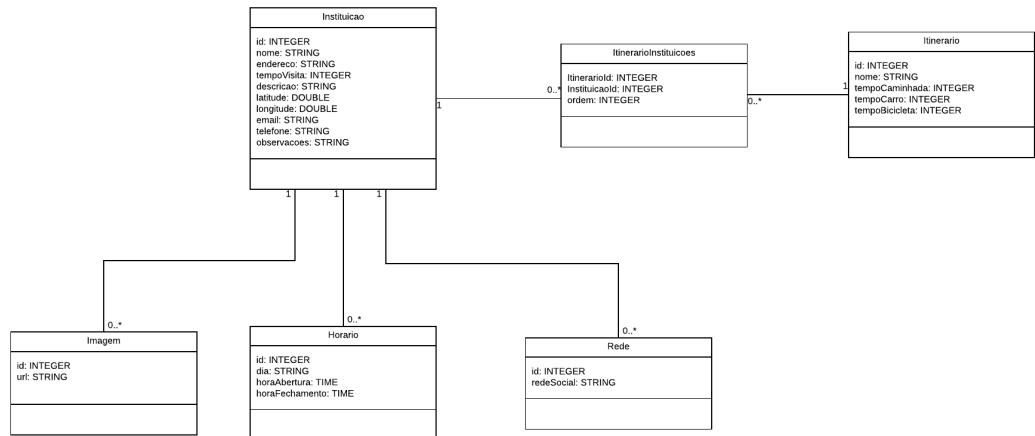
Fonte: Wiki do projeto

Figura 4 – Diagrama de pacotes front-end

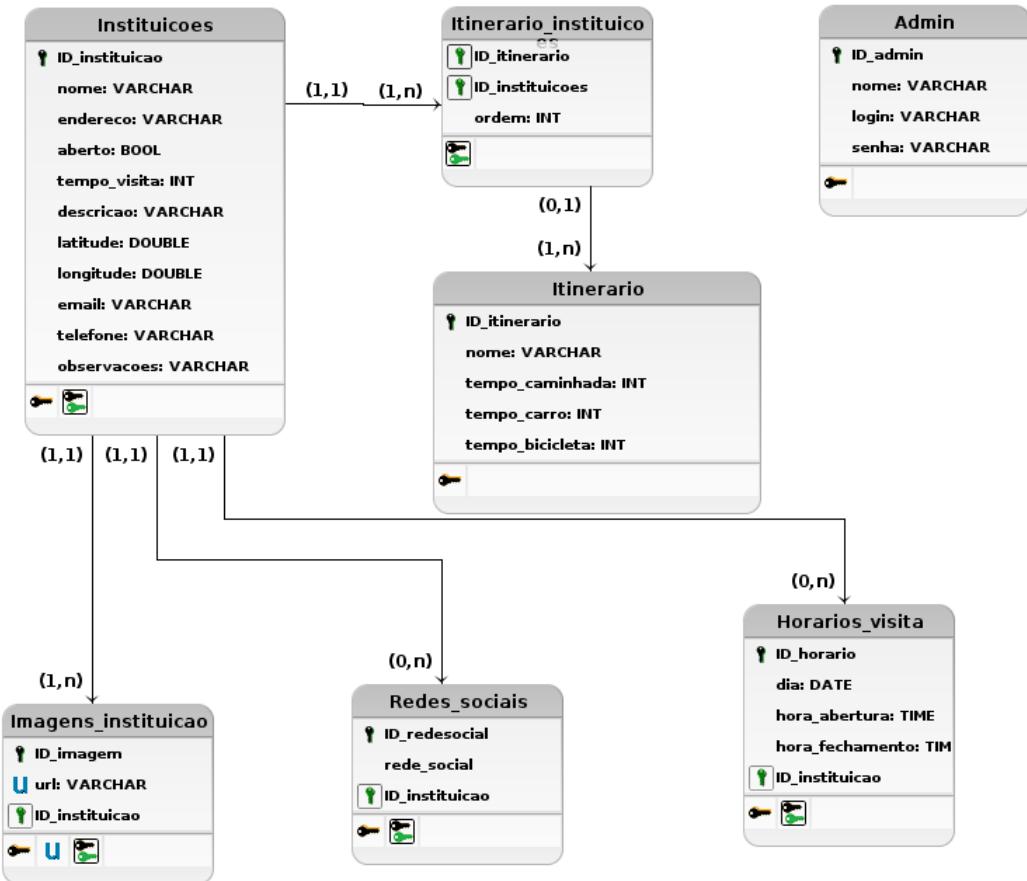
Fonte: Wiki do projeto

2.2.3 BANCO DE DADOS

No banco de dados, foi utilizado uma modelagem lógica da qual não tive participação, por ser AGES I, ela pode ser encontrada na wiki do projeto (WIKI, 2019a). As figuras 5 e 6 mostram como foi pensada, pelos AGES II, as relações entre as entidades do banco.

Figura 5 – Diagrama de classes

Fonte: Wiki do projeto

Figura 6 – Diagrama lógico de banco de dados

Fonte: Wiki do projeto

2.3 ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO

2.3.1 SPRINT 0

No início do projeto houve um debate sobre quais ferramentas seriam utilizadas para o desenvolvimento da aplicação, dando preferência às tecnologias, linguagens e bibliotecas que fossem mais conhecidas pelos alunos participantes, independentemente do nível de AGES em que se encontravam. A seguir será brevemente relatado a minha visão sobre cada etapa desenvolvida, dividida entre tarefas que tive participação.

A modelagem de dados escolhida para o projeto foi criada utilizando uma estrutura relacional. Já a arquitetura utilizada no projeto foi definida pelos alunos da AGES III e IV, conforme citada nas seções anteriores. Após o levantamento dos requisitos e suas respectivas quebras em histórias, o material foi validado com o stakeholder sendo dado início à prototipação. A prototipação foi realizada sem a participação completa da equipe, portanto a minha atuação foi apenas avaliar na apresentação junto à equipe.

2.3.2 SPRINTS 1 e 2

As atividades nas quais tive mais participação no projeto foram as de front-end, por achar mais divertido no início. Antes dessa escolha não havia conhecimento abrangente no assunto, portanto foi necessário seguir um estudo dirigido sugerido pelas pessoas mais experientes no projeto. Após esses estudos, o qual ocorreu do dia 9 ao dia 29 de agosto, foi dado início à tarefa que eu pedi para participar: a criação de um componente no estilo *modal* de informações das instituições. Essa tarefa foi bastante complexa, pois realizei em pareamento com outro colega que também era AGES I, o Vinicius Bazanella. Por sorte ele se identificava mais com o desenvolvedor front-end e liderava o pareamento, permitindo que eu aprendesse mais sobre o assunto. Devido à dificuldade da tarefa, demoramos mais de uma *sprint* para entregar algo, que precisou ser retrabalhado posteriormente.

2.3.3 SPRINT 3

Após a história de criar um *modal* ter sido concluída, passei a realizar mais tarefas de back-end, portanto fui alocado na tarefa de deletar roteiros do banco de dados. Foi complexo entender a linguagem Node.js de início, mas após algum estudo dirigido e pareamento com o Vinicius Bazanella, foi possível dar continuidade a esta atividade. A conclusão desta tarefa se deu no dia em que a equipe decidiu se reunir em um sábado na agência, para termos mais tempo trabalhando em equipe presencialmente. Desta vez, depois da inclusão dos métodos responsáveis por deletar o roteiro, foi dado início aos testes unitários que garantiam que a funcionalidade era consistente. Neste momento tive meu primeiro contato com testes unitários em uma aplicação.

2.3.4 SPRINTS 4 e 5

Finalmente nas últimas tarefas, de volta ao front-end, realizei o conserto da apresentação dos roteiros na aplicação, tendo em vista que estava sendo baseado em dados estáticos, e não tendo dados reais vindos de um banco. Como principal impedimento percebido durante uma das entregas foi a falta de conhecimento sobre a configuração de versionamento do código, não foi observado o fluxo definido no início do projeto assim resultando na quebra do código antes da entrega.

Ao concluir as horas complementares, decidi continuar com as tarefas e recebi muitos feedbacks positivos que fizeram me sentir melhor quanto a minha atuação durante essa primeira AGES, o que me motivou muito em seguir na carreira de desenvolvedor.

2.4 CONCLUSÃO

O projeto trabalhado nesta primeira etapa da AGES proporcionou um aprofundamento no conhecimento que já vinha sendo obtido de forma teórica no curso de Engenharia de Software, como um iniciante na área, e facilitou a perspectiva de uma carreira. Também mostrou, de forma prática, o que funciona e o que não funciona, dependendo das ocasiões do projeto e propiciou uma forma de resiliência ao me mostrar que problemas e imprevistos acontecem, assim estimulando o aprimoramento de minhas habilidades técnicas e comportamentais. De certa forma consigo dizer que mesmo não estando em uma etapa avançada do curso, abriu a possibilidade de ter um conhecimento breve do que estaria por vir, participando de forma ativa com a parte da criação do código e acompanhando também as etapas vinculadas a este processo. Este não foi o melhor projeto trabalhado, pois houveram muitas dificuldades em aprender as tecnologias utilizadas, mas tive orgulho de ter ajudado a desenvolvê-lo e de ter sido reconhecido pelos meus esforços.

A metodologia de desenvolvimento utilizada no projeto, no meu entendimento, é a parte que mais chama atenção ante todos os aspectos analisados. Mostra de uma forma disruptiva, num primeiro momento, como o relacionamento com o stakeholder e a imersão no projeto é extremamente importante para uma harmonia satisfatória com a equipe e o bom desenvolvimento tanto pessoal quanto do resultado esperado. Apesar de ter sido trabalhado utilizando metodologias ágeis, no contato inicial fiquei bem confuso e não absorvi logo de início como as coisas funcionavam em pequenas entregas incrementais por sprints, portanto eu focava apenas na tarefa que estava fazendo, devido às dificuldades técnicas de evoluir nelas.

Quanto a minha atuação no projeto, cabe destacar alguns pontos que foram bons e outros que acabaram sendo um impeditivo para obter melhores resultados. Apesar de ter sido uma disciplina de segundo semestre, com bastante empenho foi possível aprender como funciona de fato a profissão de desenvolvedor e as habilidades necessárias para contornar problemas, que chegam de forma concomitante com o aprendizado. Um dos impedimentos em um primeiro momento foi o entendimento do conteúdo a ser estudado devido sua complexidade, portanto

foi demandado bastante horas para esta atividade. É difícil ter como base fundamentos de programação quando é necessário ter um conhecimento maior para uma boa resolução das tarefas. Após o entendimento parcial dos conteúdos e dado início ao desenvolvimento do código, o primeiro problema encontrado foi a definição da dupla da tarefa. Apesar da aderência ao horário na tabela não houveram muitas oportunidades em que os dois integrantes da tarefa estivessem disponíveis para utilização do método de programação em par, dificultando o desenvolvimento de algumas tarefas e dando a sensação de estar travado nela.

3 PROJETO AGES II — “COMPARTILHANDO EXPERIÊNCIAS — TOWNSQ”

3.1 INTRODUÇÃO

A ideia do projeto Compartilhando Experiências surgiu de uma falta de suporte identificada no mercado de condomínios, pela empresa TownSq que atua neste meio prestando serviços de gerenciamento de condomínios e síndicos. Essa falta de suporte está relacionada à ausência de fóruns de perguntas e respostas gerido por comunidades de especialistas ou não, no gerenciamento destes negócios. Uma ideia muito parecida com a que já é utilizada no mundo da programação como o Stack Overflow, por exemplo. Além dessa solução, a aplicação *web* também servirá como *engineering as marketing*, isto é, uma ferramenta auxiliar cujo objetivo é atrair possíveis clientes para o serviço principal da TownSq.

Como *product owner* e principal *stakeholder*, temos o Marcos Eich, co-fundador da empresa TownSq, ele acompanha as apresentações das entregas, e valida se a entrega foi aceita ou tem pendências. Como possui uma boa experiência na área do desenvolvimento, Marcos ajuda com sugestões e realiza a entrega de padrões internos já adotados na empresa, para que assim possamos já entregar um produto de qualidade e com a devida identidade visual esperada.

O administrador é outro ponto-chave do projeto, é ele quem vai garantir a ordem no sistema, controlando os moderadores e outros aspectos gerenciáveis da aplicação, conforme definido o administrador faz parte da equipe da empresa TownSq.

O usuário final, que pode ser dividido em duas categorias de usuários: o que tem uma dúvida e quer que ela seja solucionada através de uma comunidade de forma prática, e o usuário que faz parte desta comunidade e quer solucionar a dúvida de alguém.

O projeto teve como professor orientador o Prof. Dr. Daniel Callegari, que vem acompanhando o desenvolvimento de forma bastante analítica e fazendo intervenções quando algo precisa ser melhorado e também apontando os acertos da equipe, o que incentiva bastante a ideia de equipes auto gerenciáveis, sendo uma realidade na indústria.

Este projeto não possui foto oficial dos membros reunidos, pois foi um dos primeiros projetos durante a pandemia, e esquecemos de registrar qualquer encontro remoto.

3.2 DESENVOLVIMENTO DO PROJETO

3.2.1 TECNOLOGIAS

As tecnologias deste projeto foram escolhidas pensando na performance de carregamento em qualquer dispositivo, seja computador ou mobile antigo com baixa capacidade de processamento. Buscando uma otimização do *Search Engine Optimization* (SEO) (Otimização para motores de busca), com objetivo de ser estar no topo das pesquisas de buscadores como o Google.

3.2.1.1 JAVA

Java é uma linguagem de programação e plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. Há muitas aplicações e sites que não irão funcionar a não ser que você tenha Java instalado, e muito mais deles são criados todos os dias. Java é utilizado desde notebooks até data centers, video-games até supercomputadores científicos, desde telefones até a internet. Java está em tudo (JAVA, 2022).

A plataforma de software Java consiste na JVM (Java virtual Machine), a API Java e um completo ambiente de desenvolvimento. A JVM faz o parse do código e então interpreta o Java bytecode gerado. A API Java consiste em um extenso conjunto de bibliotecas, incluindo objetos básicos, funções de redes e segurança (IBM, 2022a).

É uma ferramenta utilizada nas disciplinas que precedem a AGES no curso de engenharia de software, com ela vínhamos criando desde aplicações simples até em conjunto com o Spring Boot, resultando em uma aplicação que pode ser utilizada para servidores back-end, sendo este próprio projeto um exemplo dessa combinação.

3.2.1.2 SPRING BOOT

O Spring Boot é uma ferramenta que faz o desenvolvimento de aplicações web e micro-serviços ficar mais fácil e rápido trazendo três principais benefícios: Configuração automática, configuração “opinativa” (opinionated), isto é, ele define diversas configurações de forma padrão, cabendo ao desenvolvedor substituí-las se quiser. Por último, também traz a habilidade de criar aplicações standalone, capaz de rodar de forma local no ambiente de desenvolvimento (IBM, 2022b).

3.2.1.3 THYMELEAF

No front-end, o Thymeleaf foi escolhido pela integração com o Java e o Spring Boot. Além disso, devido aos requisitos do projeto, se faz necessário uma *template engine* que renderize do lado do servidor sem pesar ao usuário, portanto o Thymeleaf foi uma escolha estratégica neste sentido. Segunda a documentação oficial, o objetivo do Thymeleaf é trazer templates naturais para o fluxo de desenvolvimento — HTML que pode ser corretamente reproduzido em *browsers* e também funcionar como protótipos estáticos, permitindo uma colaboração mais forte na equipe de desenvolvimento.

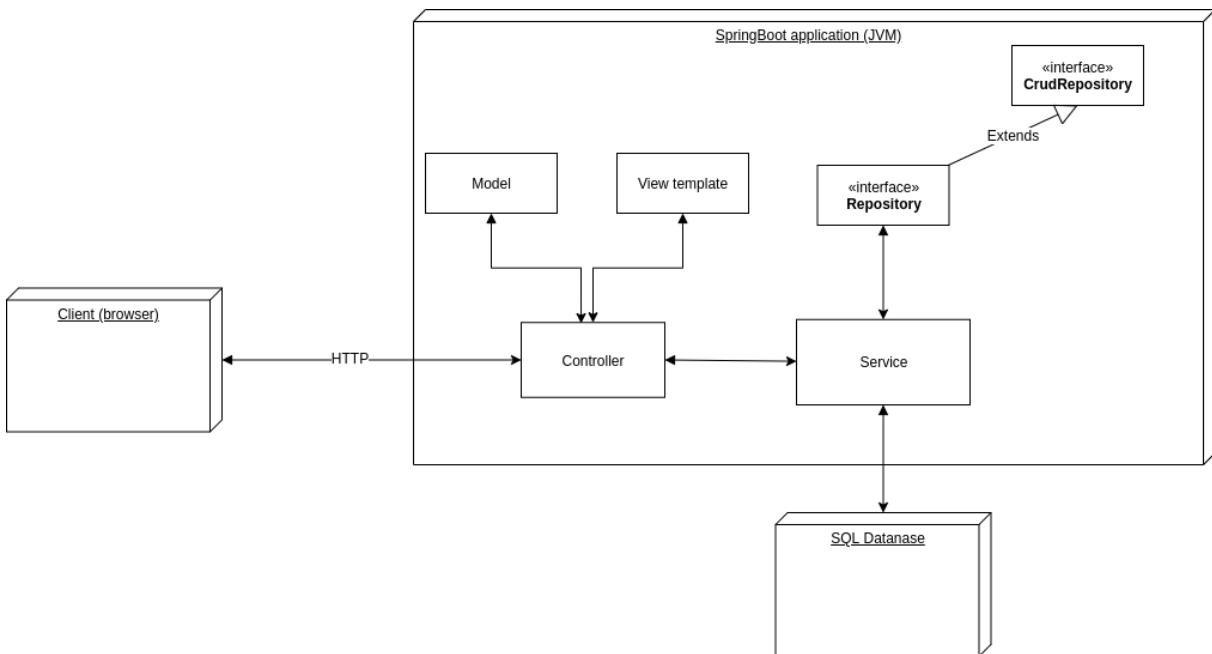
3.2.1.4 POSTGRESQL

PostgreSQL é um banco de dados relacional. Armazena dados em linhas com colunas que representam seus atributos. Uma tabela armazena multiplas linhas. O banco de dados relacional é o mais comum a ser utilizado nas aplicações. Funciona com muitas outras tecnologias e é alinhado com vários padrões de bancos de dados, o que garante sua extensibilidade (SHARMA, 2021).

3.2.2 ARQUITETURA

Para a arquitetura do projeto, foi pensado pelos Ages III o padrão MVC (Figura 7). Sendo o *client (browser)* a parte da visualização, utilizando o Thymeleaf. A maioria do diagrama representa a aplicação *back-end* Spring Boot, que representa o *controller* da aplicação. O *model* é representado pelo banco de dados (SQL). Não tive participação na escolha da arquitetura.

Figura 7 – Diagrama de Arquitetura



Fonte: Wiki do projeto

3.2.3 BANCO DE DADOS

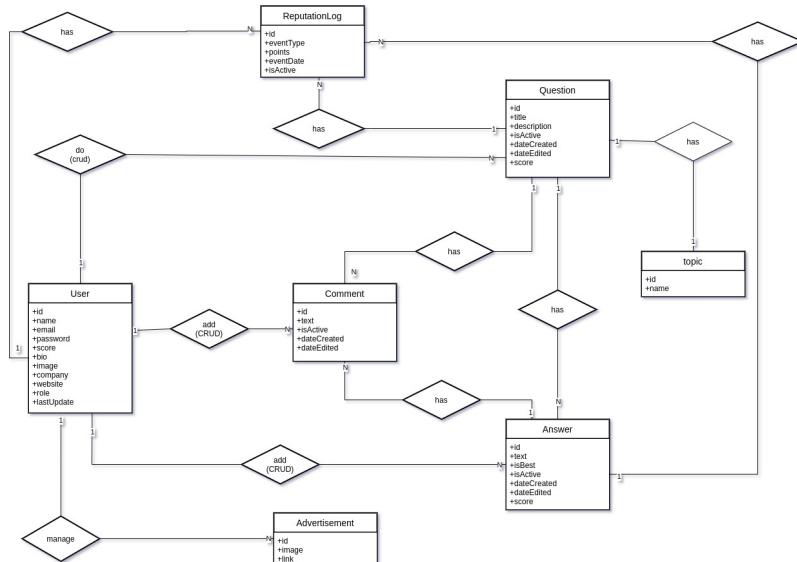
Durante as escolhas das tecnologias do projeto, a equipe optou por uma abordagem relacional na parte da persistência dos dados, portanto foi escolhido o banco de dados PostgreSQL, por ser open source e condizente com o andamento do curso.

Após a etapa de levantamento de requisitos, realizamos a modelagem conceitual de dados (Figura 8), utilizamos a ferramenta Draw.io para ter uma noção inicial do escopo do projeto, referente aos dados utilizados.

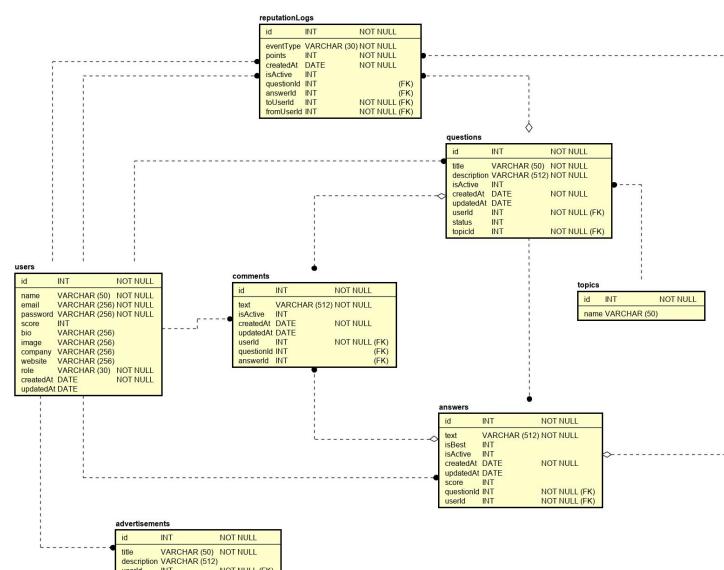
Essa primeira modelagem já contempla grande parte do Mínimo Produto Viável (MVP) acordado após a análise dos requisitos, definindo as entidades principais como *User*, *Comment*, *Answer*, *ReputationLog*, *Question*, *Advertisement*, *Topic*. Como podemos perceber, nesse ponto foi o momento inicial de decisão sobre a linguagem do projeto, ao menos na documentação, que seria em inglês.

Após a validação da primeira modelagem, partimos para a modelagem lógica (Figura 9), utilizamos a ferramenta Astah Professional.

Mais informações podem ser encontradas na wiki do projeto (WIKI, 2020c).

Figura 8 – Diagrama Conceitual

Fonte: Wiki do projeto

Figura 9 – Diagrama Lógico

Fonte: Wiki do projeto

3.2.4 PROTÓTIPOS

A prototipação do projeto foi realizada na ferramenta Figma, nos trazendo uma pequena ideia do design. A seguir, teremos um pequeno exemplo dos protótipos utilizados para a implementação do projeto. Mais telas podem ser encontradas na documentação do projeto (WIKI, 2020d). A tela inicial (figura 10), foi desenhada para a história de usuário número 11 (US11) que tem o escopo de poder visualizar a página inicial da aplicação, que contempla a apresentação das 10 últimas perguntas registradas no sistema.

Figura 10 – Protótipo da tela inicial

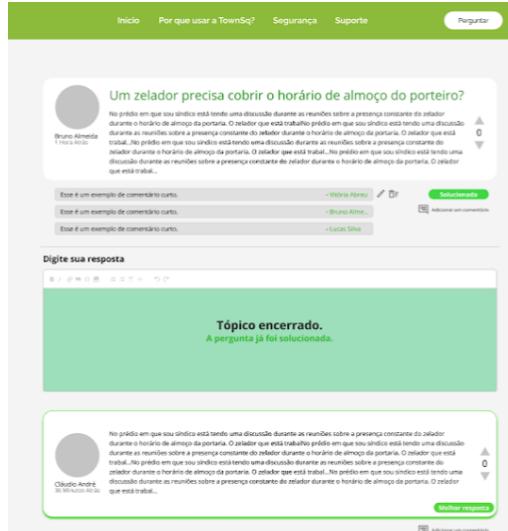


Confira as perguntas mais recentes



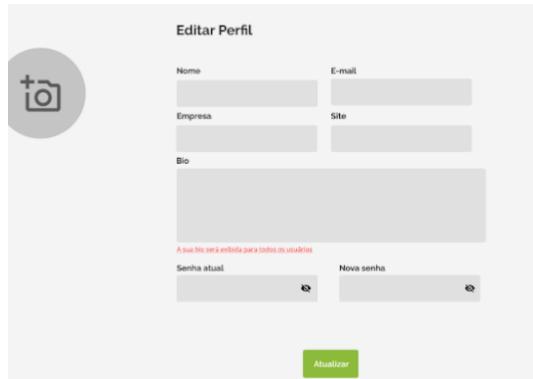
Fonte: Wiki do projeto

Já a tela de uma pergunta (figura 11), faz parte das histórias de usuários referentes às funcionalidades de poder visualizar uma pergunta e ter controles como criação, edição, deleção tanto de uma pergunta, resposta e comentário.

Figura 11 – Protótipo da tela de uma pergunta

Fonte: Wiki do projeto

A tela de edição do perfil de usuário (Figura 12) pertence à história de usuário US04 que representa o controle do usuário no sistema sobre seu próprio perfil, podendo ele editar suas informações pessoais e alterar a senha.

Figura 12 – Protótipo da tela de edição do perfil de usuário

Fonte: Wiki do projeto

3.3 ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO

3.3.1 SPRINT 0

No início do projeto, as primeiras atividades exercidas foram o conhecimento da plataforma que já existe atualmente e a elicitação do projeto junto ao stakeholder Marcos. Após a reunião inicial, foram levantados os requisitos iniciais para serem validados posteriormente. Com os requisitos deu-se início à prototipação das telas e aos diagramas conceituais e lógicos de banco de dados.

Na etapa de prototipação das telas participei logo no início, dando as ideias iniciais e montando com outros AGES II. Nesse momento tivemos alguns problemas de falta de comunicação e desentendimentos quanto aos requisitos a serem desenhados, mas logo foi resolvido e levado para a retrospectiva.

Já na modelagem do banco de dados tive participação mais ativa, começando pela modelagem conceitual, na ferramenta Draw.io. Utilizei os conhecimentos adquiridos na disciplina de Modelagem e Projeto de Banco de Dados, portanto não tive grandes dificuldades. Após a criação do modelo conceitual e sua validação junto a equipe, demos início à modelagem lógica, utilizando a ferramenta Astah. Durante esta etapa, fiz pareamento junto a outro colega, que possuía o ambiente apropriado para a ferramenta.

Por fim, dediquei o tempo que restou para me atualizar quanto às tecnologias utilizadas na parte técnica do projeto. Realizei parte de um projeto como estudo dirigido, o Pet Clinic do Spring Framework Guru, de um curso adquirido na plataforma Udemy. Através dele foi possível ampliar os conhecimentos de Spring Boot e Thymeleaf.

3.3.2 SPRINT 1

Após o *showcase* do trabalho realizado na sprint 0 e o aceite dos requisitos, modelagem e prototipação por parte do cliente Marcos, a equipe deu seguimento ao desenvolvimento do projeto. Para a sprint 1, foi prevista a continuação da modelagem do banco, observando as mudanças solicitadas pelo stakeholder. A equipe adotou a organização em esquadrões, dividindo a equipe em 3 sub-equipes que deram foco em diferentes histórias. A esquadra que participei ficou responsável pelo desenvolvimento da funcionalidade de visualizar e editar o perfil do usuário, atividades essas contemplam tanto o front-end quanto o back-end e o banco de dados. Resolvemos num primeiro momento realizar o desenvolvimento em dojo, isto é, resolver os problemas em conjunto de mais de 3 integrantes da esquadra, se encontrando principalmente em horários fora de aula. Particularmente atuei mais na parte da edição do perfil de usuário, focando na parte do back-end, parte em que posso mais domínio. Procurei na medida do possível realizar testes unitários quando necessários. Além do desenvolvimento, também fiquei servindo de apoio para dúvidas dos AGES I em questão de desenvolvimento e apoio no relatório de andamento.

Na retrospectiva desta primeira sprint, além dos pontos positivos que motivam ainda

mas a entregar um produto completo conforme o combinado, encontramos alguns pequenos problemas inerentes a qualquer equipe. Alguns dos problemas encontrados foram o pouco uso da ferramenta Trello e a má padronização dos *commits* no *git*. Sobraram também alguns débitos técnicos para a próxima sprint, mas nada fora do padrão.

Com o desenvolvimento desta sprint, percebi que ainda preciso melhorar o meu contato com o uso do Thymeleaf e aprender mais sobre o Spring Security, tendo em vista que o segundo é ponto vital do projeto. Conforme as descobertas seguiram, percebemos que a modelagem de dados também continuava em modificações sempre que necessário.

3.3.3 SPRINT 2

A Sprint 2 foi marcada pelas principais funcionalidades do sistema, as que tinham como escopo dar a possibilidade para o usuário criar, editar e deletar perguntas, respostas e comentários (US 5, 7 e 8), e introduziu também as funcionalidades de administrador e moderador, como, por exemplo, criar e alterar banners da propaganda em partes estratégicas do site (US 12). A maioria das tarefas disponíveis nessa sprint também foram realizadas em dojo, priorizando o aprendizado dos AGES I diante das tarefas, e prestando o apoio necessário.

Num primeiro momento, em apenas uma sessão numa sexta-feira, foi possível desenvolver boa parte da história, que contemplava a apresentação da pergunta de forma dinâmica na tela, isto é, trazer informações reais do banco de dados, tendo em vista que estávamos apenas simulando dados para iniciar o layout da página conforme os protótipos. A partir deste momento, foi importante ter em mente, que todo o sistema deveria ser protegido para que usuários anônimos ou não relacionados à alguma pergunta não tenham a possibilidade de realizar ações sobre as informações referente às perguntas, ou qualquer elemento da tela, apenas garantindo a visualização. Após a realização dessa história, focamos nos testes unitários para garantir a qualidade do código e prevenir erros futuros quando o mesmo for refatorado. Dessa vez, sozinho, decidi pegar uma dúvida técnica referente à US 04 no Trello para resolver, não era uma tarefa muito difícil, apenas precisava da criação de um menu no cabeçalho do site que levasse para o perfil de usuário e sair do sistema.

Um problema bastante recorrente nessa altura do projeto foi a integração com o Liquibase, pois a cada momento que alteramos alguma entidade do banco de dados, ele não reconhecia mais as migrações atuais, assim tivemos que ir excluindo migrações locais, para funcionar corretamente. Esse problema se deu pela falta de conhecimento dessa ferramenta, fato levado para a retrospectiva, para ver como a equipe iria atacar essa dificuldade. Além desse problema, foram levantados outros pontos na retrospectiva, como a falta de atenção na hora de registrar os débitos técnicos, alguns não ficando documentados e também o rumo que a aplicação estava tomando, deixando alguns pontos do sistema diferentes do protótipo desenhado inicialmente.

Nessa sprint eu senti que aprendi muito mais sobre o pareamento remoto, mesmo que tenhamos tido mais dojos, é necessário dar a vez para outros realizarem o código para que eles se inteirar de maneira satisfatória no projeto.

3.3.4 SPRINT 3

Com o aceite da maioria das funcionalidades desenvolvidas, mesmo que com alguns débitos técnicos, continuamos o projeto com a certeza de que estávamos no caminho certo, e com a confiança da equipe que todos tinham uns nos outros.

A Sprint 3 tinha como maiores desafios, dar a possibilidade do usuário decidir qual resposta satisfaz mais a sua dúvida, dando um tratamento especial para ela, possibilitando que fique mais destacada em comparação às respostas que não obtiveram tal mérito. Além disso, também era necessário possibilitar que um moderador do sistema pudesse editar e remover perguntas, respostas e comentários, de modo a dar mais autonomia assim mantendo o sistema regulado.

Comecei eu, e um AGES I da minha *squad*, William, fazendo a tarefa da história de favoritar uma pergunta, fui guiando-o para poder aprender mais a fazer com as próprias mãos, e conforme o relato dele, funcionou muito bem. Começamos pelas partes de *front-end* e *layout*, para poder ficar o mais próximo possível dos protótipos, e apenas depois disso tratar a parte lógica do desenvolvimento. Após esse primeiro momento, nos juntamos com o resto da squad para dar um seguimento mais em formato de dojo, assim terminamos mais rapidamente a história, contemplando também testes unitários, o que nos deu mais segurança da qualidade do código. Foi muito importante nesta altura do projeto, que eu continuasse com a vontade de fazer sozinho e guiasse os outros colegas de squad, para que os mesmos ficassem também em testes, pois considero uma parte vital do desenvolvimento. Com o feedback da retrospectiva, fiquei feliz com o resultado dos testes unitários e prática de dojo/*pair programming* que vinham sendo utilizadas.

A história de favoritar uma pergunta foi grande suficiente a ponto de dar trabalho para uma sprint inteira, mas valeu a pena todo o esforço, pois a equipe reconheceu que ficou além do esperado essa funcionalidade tão importante para a usabilidade do sistema.

3.3.5 SPRINT 4

Chegamos ao final do projeto com muita disposição e mais do que o esperado feito, nessa etapa só restavam poucas tarefas e débitos técnicos a fazer, ponto que foi muito elogiado pelo professor orientador, pelo Marcos e reconhecido também pelos alunos.

Depois do planejamento, a minha squad ficou encarregada da funcionalidade de administrar tópicos, isto é, como se fosse uma *tag* da pergunta, funcionalidade que desde o início, tinha como objetivo organizar as perguntas em assuntos diferentes. O desenvolvimento abrangia tanto front-end quanto back-end, para persistir os tópicos criados.

Após o desenvolvimento completo da história incluindo os testes funcionais, passamos para uma das atividades mais importantes do projeto, garantir que o SEO estivesse com uma qualidade no mínimo excelente. A SEO é uma otimização de sites em motores de buscas, ela permite que o site seja facilmente encontrado em ferramentas como o Google, quando configurada corretamente.

Conseguimos por fim, desenvolver todas as tarefas que estavam no escopo do projeto, trazendo orgulho tanto para nós alunos, quanto para a PUCRS através do professor e equipe da AGES e também para o dono do produto.

3.4 CONCLUSÃO

Com o fechamento desse projeto, já posso refletir em alguns aspectos tanto da minha carreira e da evolução nela quanto nos aspectos acadêmicos. Acredito que apoiei a equipe com meus conhecimentos técnicos, fazendo o melhor possível com meus conhecimentos até aqui. Saio desse projeto muito mais preparado para a AGES III que esperava no início dele e com mais vontade ainda de crescer nessa carreira, pois com projetos bons como esse é que se mantém o sonho de ser um profissional excelente. Acredito que foi um consenso entre os colegas de projeto, que este foi um dos melhores projetos que todos os membros já participaram.

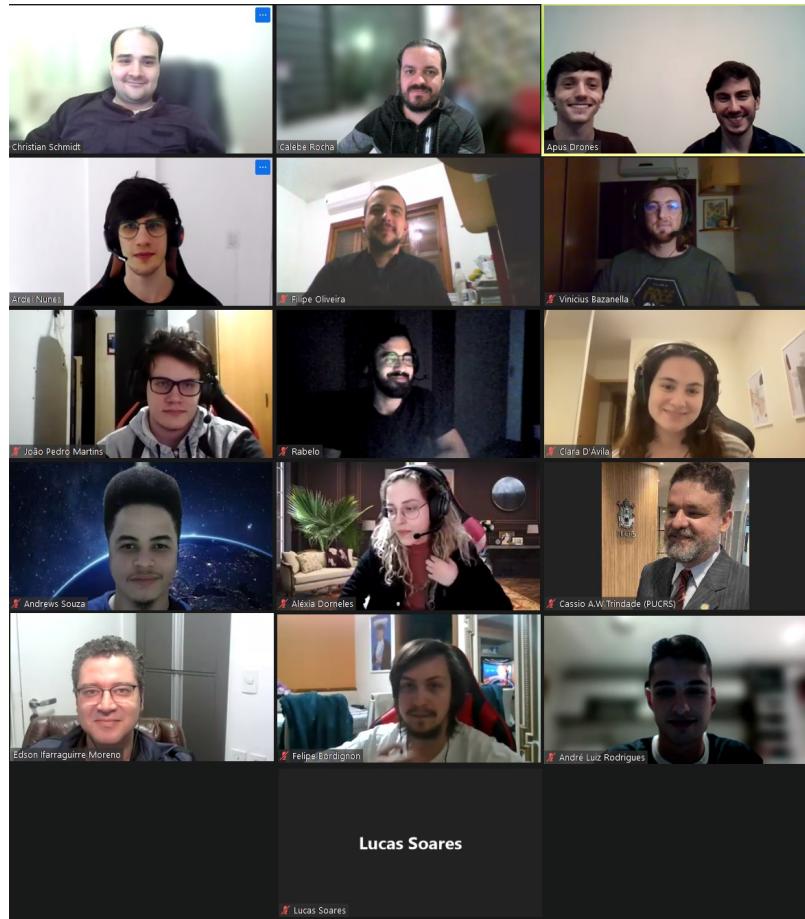
Este não foi meu projeto favorito, apesar de ter sido um dos melhores que já participei. Um dos pontos altos desse projeto foi o belo design que conseguimos entregar, e a participação satisfatória do stakeholder Marcos no projeto. Nesta AGES consegui botar em prática meu conhecimento obtido nas disciplinas de banco de dados, e pude também acompanhar o processo de design, apesar de não ter muito conhecimento sobre o tema.

4 PROJETO AGES III — “APUS DRONES”

4.1 INTRODUÇÃO

O projeto Apus Drones foi idealizado pensando com o intuito viabilizar um serviço delivery realizado por drones, pilotado por operadores, de maneira remota, fora da visão desses, em ambientes urbanos, propiciando entregas mais rápidas e eficientes, com isso reduzindo o nível de emissão de gases agravantes do efeito estufa. Por isso, o objetivo deste projeto é o desenvolvimento de um aplicativo que possibilita a compra de produtos por consumidores, disponibilize esses produtos por uma loja parceira, cadastre um plano de voo dos drones, para que o mesmo possa ser controlado por um operador. Desde o início foi deixado claro que não está no escopo do projeto a realização de transações financeiras. Dos stakeholders do projeto estão definidos os idealizadores e donos do produto: Henrique Dvogeski e Kauan Boerner, vencedores do torneio empreendedor da PUCRS. Lojas parceiras entraram como stakeholders por se beneficiarem desta modalidade de entrega, bem como a equipe de desenvolvimento.

O período de execução está compreendido entre o início do semestre letivo e o final, sendo este aproximadamente de 13/08/2021 a 26/11/2021, conforme o cronograma disponibilizado pela disciplina. O Prof. Edson Moreno trabalhou como orientador desse projeto, atuando de forma efetiva na análise do desempenho dos alunos e intervindo quando necessário no desenvolvimento do projeto. A figura 13 mostra a equipe, que trabalhou de forma remota neste projeto.

Figura 13 – Equipe de desenvolvimento, clientes e professor orientador.

Fonte: Wiki do projeto

4.2 DESENVOLVIMENTO DO PROJETO

4.2.1 TECNOLOGIAS

Nessa sessão, irei descrever que tecnologias foram escolhidas pela equipe neste projeto.

4.2.1.1 JAVA

Java é uma linguagem de programação e plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. Há muitas aplicações e sites que não irão funcionar a não ser que você tenha Java instalado, e muito mais deles são criados todos os dias. Java é utilizado desde notebooks até data centers, video-games até supercomputadores científicos, desde telefones até a internet. Java está em tudo (JAVA, 2022).

A plataforma de software Java consiste na JVM (Java virtual Machine), a API Java e um completo ambiente de desenvolvimento. A JVM faz o parse do código e então interpreta o Java bytecode gerado. A API Java consiste em um extenso conjunto de bibliotecas, incluindo objetos básicos, funções de redes e segurança (IBM, 2022a).

É uma ferramenta utilizada nas disciplinas que precedem a AGES no curso de engenharia

de software, com ela vínhamos criando desde aplicações simples até em conjunto com o Spring Boot, resultando em uma aplicação que pode ser utilizada para servidores back-end, sendo este próprio projeto um exemplo dessa combinação.

4.2.1.2 SPRING BOOT

O Spring Boot é uma ferramenta que faz o desenvolvimento de aplicações web e micro-serviços ficar mais fácil e rápido trazendo três principais benefícios: Configuração automática, configuração “opinativa” (opinionated), isto é, ele define diversas configurações de forma padrão, cabendo ao desenvolvedor substituí-las se quiser. Por último, também traz a habilidade de criar aplicações standalone, capaz de rodar de forma local no ambiente de desenvolvimento (IBM, 2022b).

4.2.1.3 Flutter

Para o *front-end*, foi escolhida o Flutter, uma *framework* de *User Interface* (UI) para criação de interfaces nativas para iOS e Android. Flutter foi publicado em maio de 2017 (THOMAS, 2020). Ele possibilita que seja utilizado a mesma base de código para criar aplicações de forma nativa nas duas plataformas citadas anteriormente. No Flutter tudo é com base em *widget*, algo parecido com componente de uma arquitetura, são pequenos pedaços de telas que juntos formam uma UI completa.

4.2.1.4 AWS

A Amazon Web Services, em 2006, começou a oferecer serviços de infraestrutura de TI para empresas através de serviços web, hoje conhecidos como computação em nuvem. O principal objetivo da computação em nuvem é substituir diretamente gastos com infraestrutura por preços mais baixos, que se ajustam conforme a empresa. Com a nuvem, as empresas não precisam mais planejar ou adquirir grandes máquinas que seriam os servidores, esperando até semanas ou meses de antecedência até possuir tal equipamento (AWS, 2022b). Em vez disso, podem rodar instantaneamente centenas de milhares de servidores em minutos e oferecer resultados rapidamente. Atualmente a Amazon Web Services oferece centenas de produtos utilizados por grande parte do mercado mundial. Neste projeto utilizamos 3 deles:

4.2.1.5 EC2

Amazon Elastic Compute Cloud (Amazon EC2) provê capacidade de computação escalável na AWS Cloud. Usando EC2 elimina a necessidade de investimento em hardware, sendo possível desenvolver e fazer deploy de aplicações de forma mais rápida. Pode ser usado para lançar quantos servidores virtuais forem necessários, configurando a segurança e rede, bem como gerenciando o espaço em disco (AWS, 2022c). No contexto da nossa aplicação, ele é utilizado

para manter o sistema back-end rodando, para deixar o sistema disponível 100% do tempo para demonstração para o stakeholder, além de rodar também um Runner de CI/CD do gitlab.

4.2.1.6 ECR

Amazon Elastic Container Registry (Amazon ECR) é um gerenciador de registros de imagens de containers que é seguro, escalável e resiliente. Suporta repositórios de imagens de containers privados com permissões usando o AWS IAM. Ele é feito dessa forma para que usuários configurados ou instâncias do EC2 possam acessar o repositório de containers e imagens. Pode ser utilizado CLI para enviar, baixar e gerenciar imagens Docker (AWS, 2022d). No atual projeto, o ECR é utilizado para armazenar e versionar imagens do sistema back-end, então disponibilizando estas imagens para que o EC2 possa utilizá-las e mantê-las disponíveis.

4.2.1.7 S3

Amazon Simple Storage Service (Amazon S3) é um serviço de armazenamento de arquivos que oferece alta escalabilidade, disponibilidade, segurança e performance. Pode ser usado para guardar, recuperar qualquer quantidade de dados em qualquer hora em qualquer lugar. Um conceito muito importante é que o S3 guarda dados como objetos em *buckets* (baldes). Um objeto consiste em um arquivo e opcionalmente metadados que descrevem esse arquivo. Para armazenar um objeto no S3, deve-se fazer o upload do arquivo que se deseja salvar em um *bucket*. Quando o arquivo é salvo, pode-se definir permissões no objeto para diferentes usuários. *Buckets* são contêineres para objetos, e pode-se ter um ou mais *buckets*. Para cada um, pode ser controlado o nível de acesso (quem cria, deleta e lista os objetos no *bucket*). Também é possível verificar os *logs* de acesso e seus objetos, além de poder escolher a região geográfica onde o S3 irá armazenar o *bucket* e seu conteúdo. No nosso sistema, utilizamos o S3 para armazenar fotos de usuários e produtos, para que os mesmos possam ser recuperados diretamente pelo usuário na aplicação móvel. (AWS, 2022a)

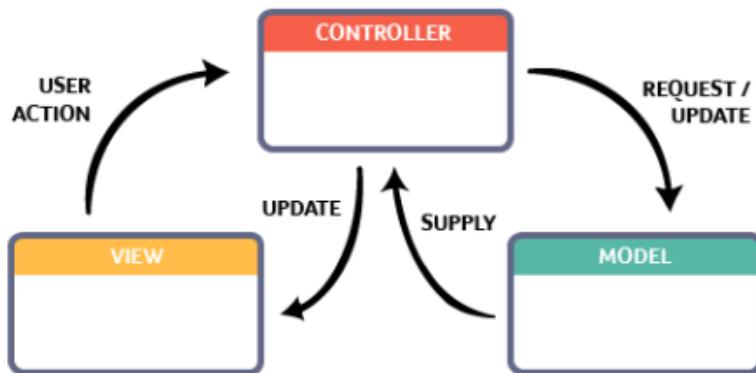
4.2.1.8 MYSQL

Segundo o site Hostinger, MySQL é um gerenciador de banco de dados que utiliza a linguagem SQL como interface. SQL é uma sigla para “Structured Query Language” que utiliza um modelo relacional de dados. A partir do MySQL um ou mais dispositivos (clientes) conectam em um servidor através de uma rede específica. Qualquer cliente consegue fazer uma requisição pela interface (GUI) nas suas telas e o servidor vai entregar a saída desejada. Basicamente é seguido o seguinte roteiro: MySQL cria um banco de dados para guardar e manipular os dados e definir a relação de cada tabela. Clientes fazem requisições utilizando comandos específicos da linguagem SQL no MySQL. O servidor irá responder com a resposta requisitada e irá apresentar para o lado do cliente.

4.2.2 ARQUITETURA

Quando pensamos na arquitetura da aplicação, no início do projeto, imaginamos o padrão MVC (Figura 14), o model sendo persistido no banco de dados e requisitado pelo controller, o backend da aplicação, que por sua vez é chamado pelo view, o front-end da aplicação.

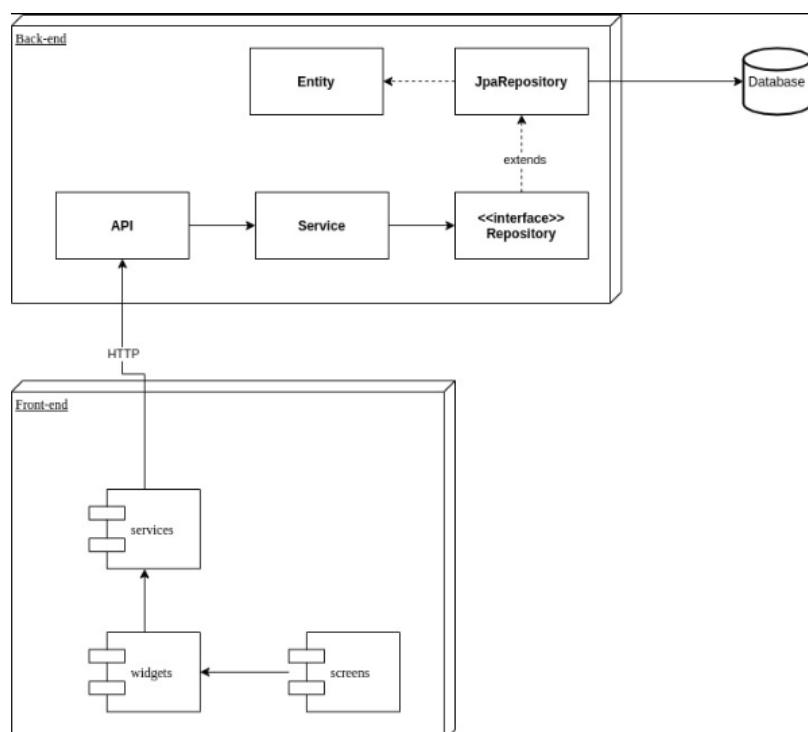
Figura 14 – Padrão arquitetural MVC



Fonte: Wiki do projeto

Essa ideia pode ser melhor visualizada na figura 15, o diagrama da arquitetura do projeto. Ela contempla desde a separação das camadas que compõe a aplicação, até a separação dos principais componentes dentro de cada projeto.

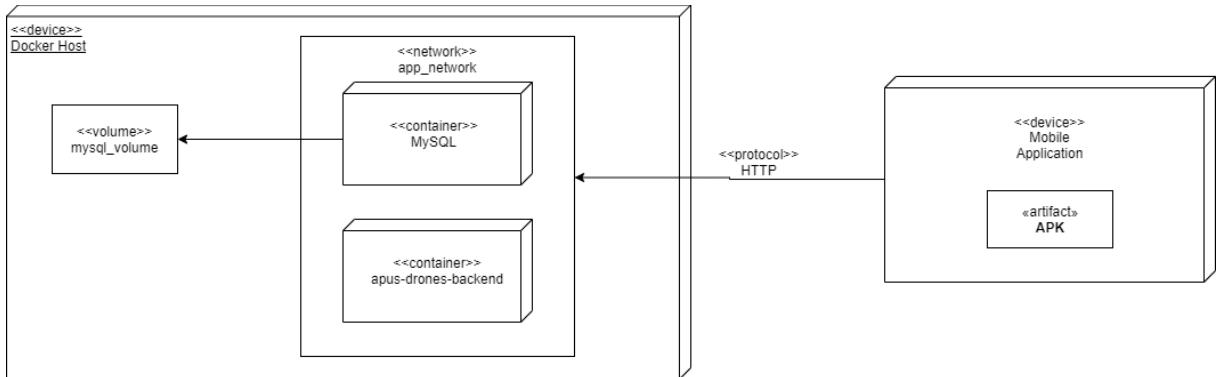
Figura 15 – Diagrama de arquitetura geral



Fonte: Wiki do projeto

Na parte da arquitetura de deploy, o diagrama da imagem 16 representa nossa ideia de separação da aplicação em um container Docker, que poderá ser acessado via http pelo cliente em um dispositivo mobile. Lembrando que este diagrama representa a forma genérica de deploy, em seguida será mostrado o diagrama específico.

Figura 16 – Diagrama genérico de deploy

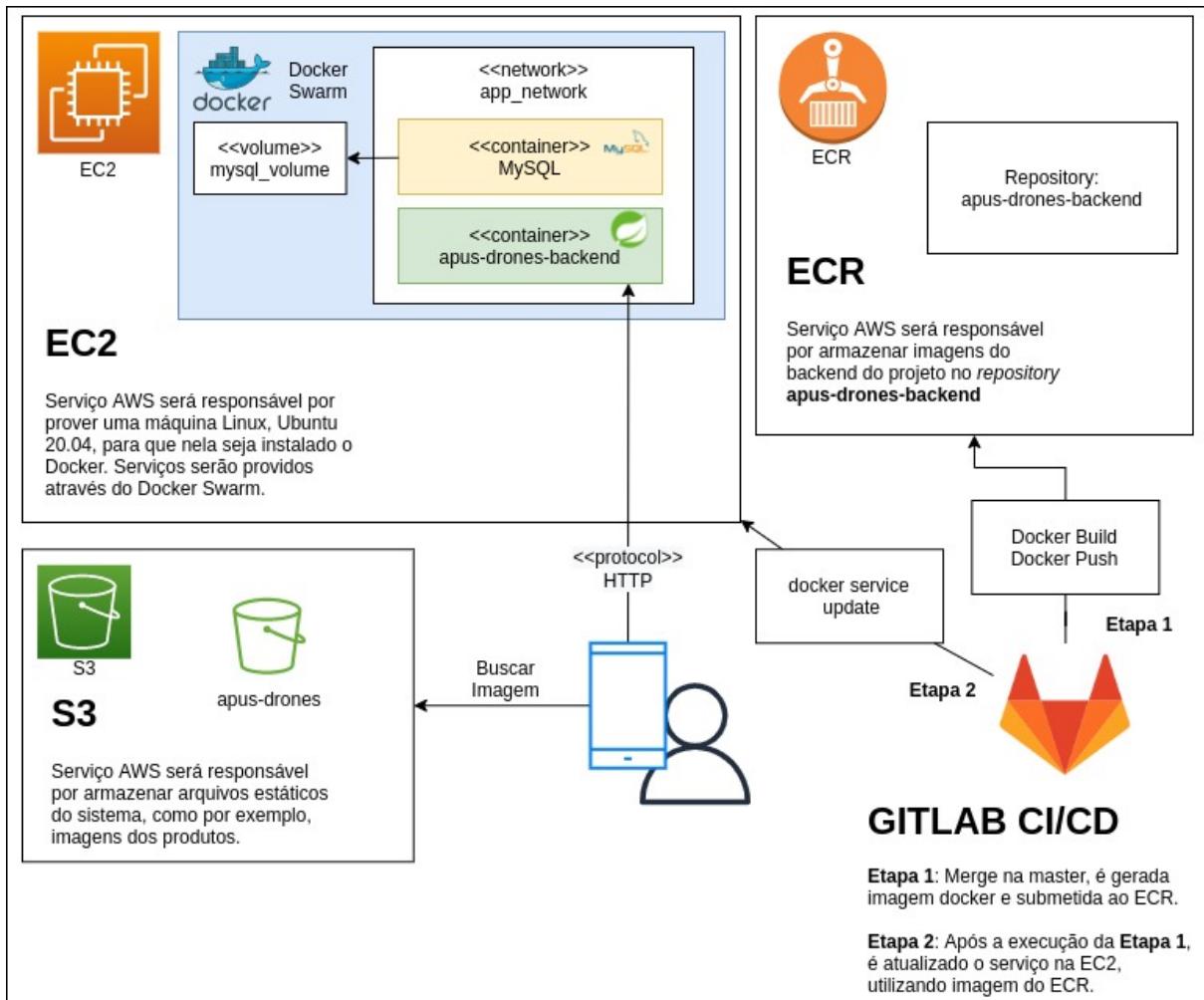


Fonte: Wiki do projeto

A network principal contém dois containers, um que executa o banco de dados MySQL e outro que roda a aplicação backend Spring Boot, identificado no diagrama como apus-drones-backend. Para o container MySQL, há um volume, no diagrama descrito como mysql_volume, que possui um mecanismo melhorado de persistência de dados, ele é melhor descrito na seção de tecnologias.

Já no diagrama específico da AWS (Figura 17), detalhamos os serviços que serão utilizados para o fluxo de deploy, bem como alguns passos de execução. Primeiro a aplicação é desenvolvida e aprovada pela equipe revisora, então quando o código é mesclado com a branch “master”, é gerada uma imagem docker e submetida ao serviço ECR, que por sua vez disponibiliza esta imagem para o EC2, que sobe o serviço apus-drone-backend a partir dessa imagem. O usuário por sua vez, a partir de chamadas HTTP na aplicação front-end consegue acessar a aplicação citada anteriormente. A parte de upload e download das imagens fica a cargo do serviço S3. Todos esses serviços serão descritos melhor na seção tecnologias.

Todos esses diagramas podem ser encontrados na wiki do projeto: < <https://tools.ages.pucrs.br/apus-drones/apus-drones-wiki/wikis/arquitetura> >

Figura 17 – Diagrama de deploy AWS

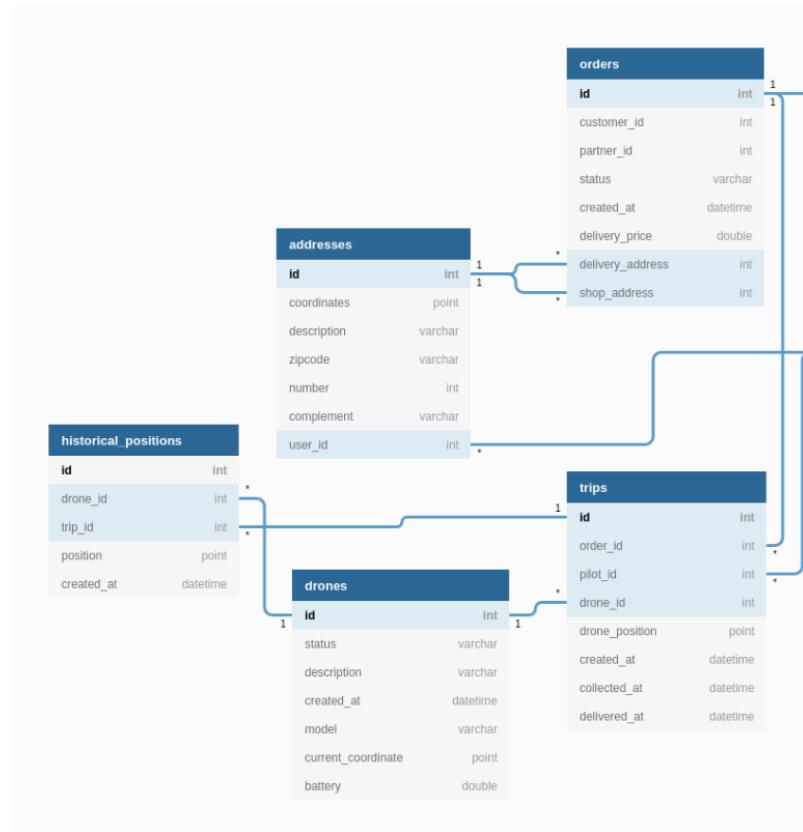
Fonte: Wiki do projeto

4.2.3 BANCO DE DADOS

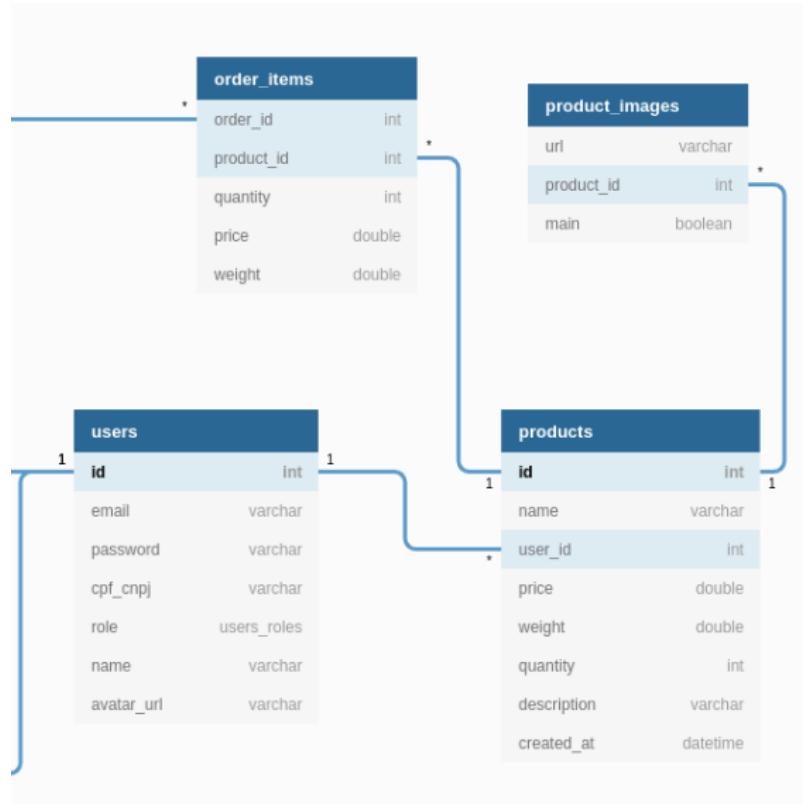
Logo na Sprint 0, foi iniciada a modelagem dos dados e definidas as entidades com base nos requisitos levantados pela equipe responsável por isto. A primeira parte, apresentada na figura 18, diz respeito às entidades de pedidos (orders), endereços (addresses), entregas (trips), drones e histórico de posições de um drone (historical_positions).

A parte 2 desse diagrama, que pode ser visto na figura 19 traz o conceito de itens de um pedido (order_items), produtos e suas imagens (products e product_images, respectivamente) e os usuários do sistema (users), podendo estes ser clientes, parceiros ou pilotos, definido pelo campo “role”.

A documentação completa da modelagem de classes da aplicação pode ser encontrada na wiki (WIKI, 2020a). A escolha do banco de dados não foi unânime desde o início, estávamos indecisos sobre a utilização de um banco relacional ou não relacional, tanto é que no início as dependências do back-end utilizavam MongoDB. Foi concluído que a melhor escolha, dada as

Figura 18 – Parte 1 da modelagem de classes

Fonte: Wiki do projeto

Figura 19 – Parte 2 da modelagem de classes.

Fonte: Wiki do projeto

peculiaridades do projeto (Utilização de coordenadas como localização de usuários e definições de rotas), seria um banco relacional MySQL, que possui funcionalidades de geolocalização e facilitação desses cálculos.

No back-end foi possível realizar configuração para utilização de banco em memória (h2) em ambiente local de desenvolvimento, enquanto em ambiente de homologação (disponibilizado em nuvem) era utilizado banco MySQL.

4.2.4 PROTÓTIPOS

Após a análise dos requisitos e refinamento dos mesmos, foi possível dar início aos protótipos em alto nível da aplicação, utilizando a ferramenta Figma. Os desenhos podem ser encontrados na página da wiki (WIKI, 2020b). A seguir, será mostrado alguns exemplos de telas desenhadas, de diferentes categorias de usuários. A tela apresentada na figura 20, representa a visão da história US11, que trata do cadastro de clientes. Ela contém alguns dados já levantados em refinamentos anteriores que seriam necessários para o cadastro de um cliente no sistema.

Figura 20 – Tela de cadastro do cliente

CADASTRO

Que tipo de usuário você é?

CLIENTE **PARCEIRO** **PILOTO**

Nome
João Silva

CPF
123.456.989-10

Email
cliente@gmail.com

Senha

CEP
12345-678

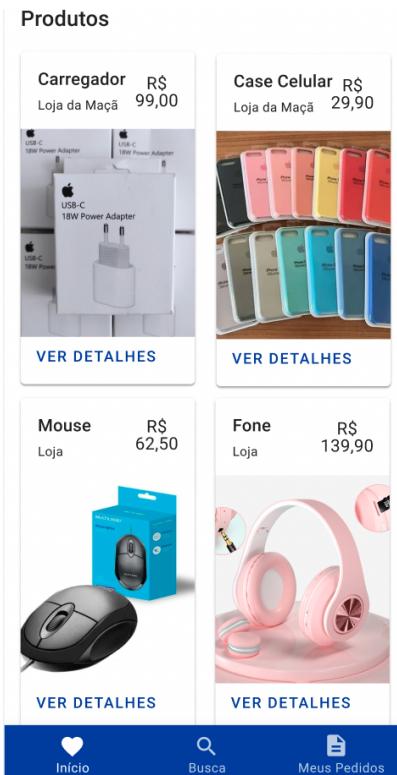
CADASTRAR

Já tem conta? [Fazer Login](#)

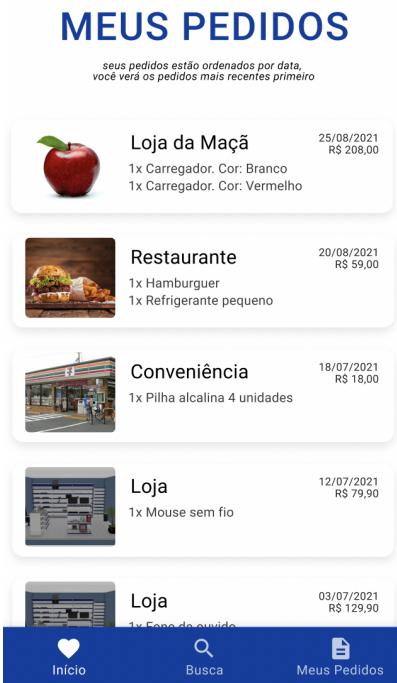
Fonte: Wiki do projeto

Já na figura 21, temos a apresentação da tela inicial para um cliente logado, nela é possível encontrar os parceiros na parte superior, abaixo da barra de pesquisa, e na parte inferior escolher entre os produtos disponibilizados pelo sistema. Esta tela representa a história de código US12.

Trocando de papel, indo agora para as telas de parceiro, a imagem 22 representa a tela de pedidos, que mostra as solicitações e pedidos passados feitas por um cliente. A estória US21 representa os requisitos desta tela.

Figura 21 – Tela inicial da área logada de um cliente

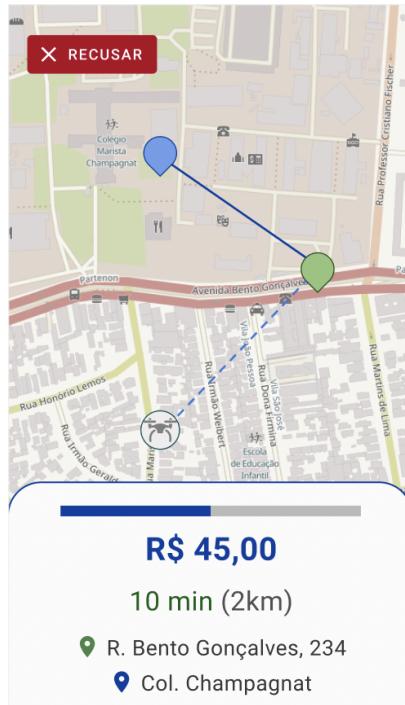
Fonte: Wiki do projeto

Figura 22 – Tela de pedidos na visão de parceiro

Fonte: Wiki do projeto

Agora na parte do piloto, existe a tela de aceitar ou recusar uma “corrida”, que seria uma entrega ou acompanhamento de uma entrega de um produto, de um parceiro para um cliente. A figura 23 representa a história US6.

Figura 23 – Tela de entrega do usuário piloto



Fonte: Wiki do projeto

4.2.5 REPOSITÓRIO

Como gerenciamento de configuração de software, utilizamos o padrão disponibilizado pela agência, o Gitlab próprio da AGES disponível na wiki (WIKI, 2020e), tanto para o armazenamento da wiki quanto para o versionamento do código (figura 24).

A equipe responsável pelo gerenciamento de configuração definiu, com o apoio do resto da equipe, o GitFlow como fluxo de trabalho (figura 25) para o incremento do projeto, fluxo este que pode ser encontrado em <<https://tools.ages.pucrs.br/apus-drones/apus-drones-wiki/wikis/processo>>. Para o merge de branches de features com a branch de desenvolvimento, e a de desenvolvimento com a master, foi combinado que deveria ter pelo menos duas aprovações (joinhas na ferramenta) de colegas diferentes, para assim poder seguir o merge.

Além disso, do Gitlab foi utilizado a integração/deploy contínuo, disponibilizado pela ferramenta de forma nativa, processo esse que será mais detalhado na parte de arquitetura do projeto.

Figura 24 – Repositórios do projeto

Apus Drones

Apus Drones

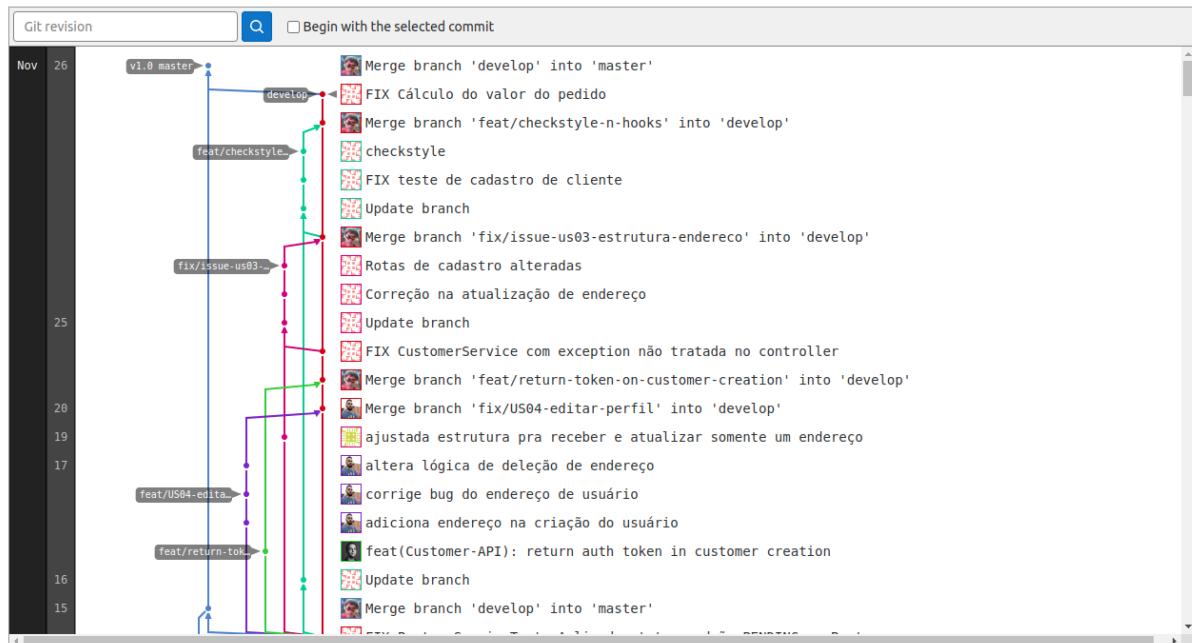
Group ID: 597

Subgroups and projects Shared projects Archived projects

Search by name Updated date

	apus-drones-frontend	★ 0	1 month ago
	Apus Drones Wiki	★ 0	4 months ago
	apus-drones-backend	★ 0	4 months ago

Fonte: Wiki do projeto

Figura 25 – Gráfico de commits demonstrando o uso do GitFlow no backend.

Fonte: Wiki do projeto

4.3 ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO

A seguir, descrevo em detalhes meu desempenho nesse projeto, bem como iniciativas, acabativas e dificuldades encontradas nesse processo.

4.3.1 Sprint 0

Após a apresentação do projeto para os alunos e definição do escopo, a primeira decisão que precisou ser tomada foi referente às tecnologias com que iríamos trabalhar. Duas grandes frentes foram levantadas, a tecnologia do back-end e a que seria utilizada no front-end, como a equipe tinha mais familiaridade com o Java, dado o contexto da faculdade e a experiência dos integrantes acabamos optando por ele junto de Spring Boot.

Logo tendo sido feita a escolha do back-end me prontifiquei para a criação da base do projeto, utilizando a ferramenta Spring Initializr consegui dar início ao projeto, definindo algumas dependências iniciais a ele, como banco de dados mongo inicialmente (que posteriormente foi modificado para utilizar banco relacional MySql). Feito a configuração Inicial aproveitei para adicionar uma documentação viva (Swagger) desta API, que viria a ser útil no futuro para expor de forma facilitada as rotas desenvolvidas, bem como facilitar muito os testes funcionais da aplicação. Houve uma dificuldade na configuração do swagger, pois grande parte das documentações de instalação não estavam atualizadas, causando erros diferentes em cada tentativa de implementação, após horas de tentativas consegui habilitar com sucesso a ferramenta.

Se tratando de qualidade de Software, aproveitei para verificar o interesse da equipe em configurar uma plataforma de verificação contínua de código (SonarQube), ao ser apoiado pelos AGES IV aproveitei esse estado inicial do projeto para adicionar essa configuração. Uma dificuldade encontrada nesse processo foi que diferente de projetos disponibilizado em Github público ou Gitlab oficial, ao utilizar o Gitlab próprio da AGES não tínhamos suporte do Sonar Cloud, ferramenta essa que funciona de forma online, sem a necessidade de utilizar servidor local. A solução encontrada foi utilizar servidor local para o usufruto dessa ferramenta.

Enquanto isso a outra frente levantada, a do front-end, acompanhei a apresentação feita pelo colega Andrews que mostrava um comparativo entre as opções viáveis escolhidas inicialmente: React e Flutter. Dada a preferência do time pelo Flutter iniciei os estudos dirigidos pela tecnologia. Ainda acompanhando os colegas que lideravam esta frente, acompanhei o início do repositório e a estrutura inicial do projeto. Dado o meu contexto na vida profissional, o front-end foi a parte com a qual eu menos me envolvi, por ter menos afinidade.

Dadas as atribuições de um AGES III, nos juntamos todos os alunos que se encaixavam neste papel para discutir temas referente a arquitetura, assim conseguimos desenvolver o diagrama de deploy, requerido pela utilização de serviços da AWS. Começamos pelo desenho do diagrama genérico de deploy, que identifica a disposição da aplicação em que devices elas viriam a ser executadas no futuro. Não sendo suficiente este diagrama genérico, pois não especificava o

uso dos serviços da AWS, foi solicitado que fizéssemos outro diagrama, dessa vez contendo essas especificações, tivemos grande ajuda do colega Filipe Oliveira (AGES II), que possui um conhecimento bastante abrangente em infraestrutura em nuvem. Além disso, participei de forma minoritária no levantamento dos requisitos e critérios de aceitação do projeto, bem como do desenho dos protótipos de alta fidelidade, tarefa esta que eu não pude ser muito útil pela falta de habilidade prática em design.

4.3.2 Sprint 1

Dado o levantamento dos requisitos funcionais feita na Sprint 0, fiquei responsável pelo desenvolvimento do back-end referente a User Story de código US12, que trata da busca de parceiros para disponibilização do usuário na tela inicial da aplicação. Ao ter sido definida a arquitetura da aplicação separada em camadas (api -> service -> repository), comecei a implementação pela parte mais interna dessa disposição – a camada do repositório – e a criação da entidade “User” que seria salva no banco de dados. Junto disso aproveitei para iniciar a prática de testes unitários e de integração, junto ao desenvolvimento, até para confirmar que o código que eu estava fazendo estava de fato funcionando e fazendo sentido.

Finalizada a camada do repositório iniciei o desenvolvimento da camada acima, a de serviço, esta que define as regras de negócio que os dados passarão antes de serem retribuídos para o front-end. Um dos problemas encontrados foi a falta de padrão desenvolvida por outras tarefas que haviam sido feitas antes, como a má separação de camadas e a devolução de entidades salvas em banco como resposta da chamada. A solução encontrada foi adotar o padrão DTO (Data Transfer Object), que consiste em criar objetos que contém apenas o necessário para a finalidade desejada. Este desenvolvimento também foi acompanhado de criação de testes unitários.

Por fim, na camada de api, não houve segredo, apenas deveria ser disponibilizada a rota que daria acesso a utilização do serviço criado no parágrafo anterior. Não houveram grandes dificuldades neste desenvolvimento, tendo em vista que o código necessário para esse acesso é muito pequeno. Ainda sim disponibilizei código de testes para esta camada, garantindo que tivéssemos pelo menos um exemplo de teste unitário/integração por camada.

Além desta sub-tarefa, a US12 também continha a busca de produtos por parceiros, tarefa essa que seguiu o mesmo rito utilizado na anterior, com a diferença na dificuldade encontrada nos testes unitários. Apesar de possuir um certo conhecimento tornou-se mais difícil concluir um teste unitário na camada de serviço, dada a complexidade da tarefa. Por fim desta sprint, realizei a tarefa da US13, referente a barra de busca, que consistia na pesquisa tanto por parceiros quanto por produtos, dada uma String digitada pelo usuário na aplicação, sem grandes dificuldades.

Uma frustração percebida nessa Sprint foi a baixa participação na infraestrutura do projeto, referente às práticas de CI/CD. O colega AGES IV Calebe auxiliou nessa tarefa e realizou a subida da aplicação antes da apresentação.

4.3.3 Sprint 2

Após o aceite das entregas da Sprint anterior pelos stakeholders, busquei realizar tarefas de arquitetura, então fiz estudos referentes à utilização do banco de dados MySql em ambiente de homologação, enquanto em ambiente de desenvolvimento local mantivesse a configuração de banco em memória h2. Ao tentar implementar essa configuração tive problemas na configuração do docker na minha máquina, não possibilitando a subida de um banco MySql local para testes da configuração. Após a desinstalação do docker no meu computador e buscas na internet, consegui seguir com a tarefa e finalizar ela, deixando possível a utilização dos dois bancos dependendo do ambiente. Um ponto alto dessa Sprint foi a implantação do gitlab CI/CD no projeto, iniciado com o colega também AGES III, Vinicius Bazanella. Esta tarefa não foi fácil pela falta de conhecimento sobre o tema, então recorremos a ajuda do colega Calebe, AGES IV, que possui grande expertise no assunto. Tivemos problemas de autenticação para push de imagens da aplicação no ECR, então recorri ao arquiteto da AGES, Cassio Trindade, para viabilizar os acessos necessários. Durante as semanas desta Sprint, recorrentemente fui realizando revisão de merge requests conforme eles iam surgindo.

A última tarefa realizada foi a referente a US25, que tinha como objetivo expirar o pedido do usuário caso não tenha sido aceito por nenhum parceiro após determinado período de tempo. Encontrei bastante dificuldade nesta tarefa, pois foi difícil aplicar a lógica dado os padrões REST para alteração de recursos. Ao conversar com os colegas Ages IV, tendo sugestão de implementação, consegui terminar e entregar a funcionalidade de forma satisfatória, com exceção dos testes unitários, que por falta de tempo acabaram sendo despriorizados.

4.3.4 Sprint 3

Comecei esta sprint ajudando o colega AGES IV Calebe na finalização da implementação do deploy automatizado, tarefa esta que estava um pouco atrasada devido erros de autenticação no repositório do ECR relatado na Sprint anterior. Então atuei no upload de imagem para o serviço S3 da AWS, referente a US24, tarefa esta que já tinha sido começada pelo colega Felipe Bordignon, então entrei como par para dar seguimento a mesma. Comecei verificando que o serviço não estava separado em camadas, possibilitando diversas melhorias para seguir bons padrões de código. Feita esta separação percebi que a funcionalidade deixava brechas como o upload de arquivos de tamanho extremamente grandes, então implementei um limite de tamanho, para 1MB, tamanho suficiente para uma imagem de boa qualidade.

Essa sprint foi a que eu menos contribui no desenvolvimento deste projeto, devido a falta de tempo para dedicar ao projeto, acabei pegando apenas uma tarefa: A US11, cadastro do cliente. Acabei deixando para última semana o desenvolvimento dessa, no mesmo dia do freeze time, portanto não pude aplicar os melhores conceitos de qualidade nesta tarefa, resultando em uma entrega com bastante pendências. Além disso, o cadastro do usuário não contava com o endereço, pois não estava implementado no front-end a forma de registrar o endereço

completo do usuário, resolvi deixar como débito técnico para a próxima sprint. Na retrospectiva houveram feedbacks construtivos devido a esta falta de auxílio pelos AGES III, equipe a qual faço parte, então pude ter noção de que estava deixando a desejar neste desenvolvimento.

4.3.5 Sprint 4

Na sprint 4, comecei atuando em uma issue que permitia que o usuário parceiro pudesse visualizar o carrinho de usuários que ainda não haviam concretizado a compra, ou seja, ainda estavam com o carrinho possibilitando a escolha de novos itens. Verifiquei que a solução não exigia muito esforço, pois bastava adicionar um filtro de status que impedissem que a busca retornasse carrinhos com o status IN_CART, enum esse que caracteriza um carrinho aberto no sistema. Um grande problema encontrado na execução desta tarefa foi que o build do projeto backend estava quebrado, devido a testes falhando na branch principal do projeto, o que já havia sido mapeado para correção, portanto aguardei este desenvolvimento. Nesta sprint também foi finalizado o desenvolvimento da tarefa da US24, relatada na sprint anterior, que tratava do recebimento de upload de imagem pela parte do backend, tarefa essa que realizei a revisão do código, e então corrigi outro problema encontrado, a imagem quando enviada sem extensão (arquivo.png, por exemplo) acabava lançando exceção inesperada.

Aqui também se destaca o trabalho feito em pareamento com o colega Vinicius Bazanella, desta sprint em diante, em conjunto foi possível dar seguimento a tarefa US11, incompleta na sprint anterior, que tinha entre suas pendências o cadastro do endereço do usuário. Ao dar início nesta tarefa verificamos que já estava sendo modificado, em outra tarefa, o objeto de endereço que estávamos trabalhando, portanto, aguardamos esse desenvolvimento para dar seguimento.

Antes da finalização desta sprint, ajudei o colega Calebe na correção de testes unitários que envolviam o Spring Security, era complicado realizar o mock de um usuário logado na aplicação, como já trabalho com Spring há um tempo considerável, pude auxiliar nessa tarefa, procurando a notação que aplicava este mock no método de teste referente a esta funcionalidade. Neste mesmo dia revisei a tarefa que adiciona regra de lint no backend, como mexia em código essencial no projeto, acabou gerando diversos conflitos ao tentar realizar o merge desta tarefa, portanto optei por não seguir em frente com esta funcionalidade, por medo de que as outras funções deixassem de funcionar.

4.3.6 Sprint 5

Comecei este ciclo de desenvolvimento atuando com meu colega Vinicius ainda na tarefa da US11, cadastro de cliente, a qual tinha como pendência ainda o cadastro de endereço para um cliente. Debatemos sobre o uso de uma API pública para a funcionalidade de autocompletar o endereço informado pelo usuário, mas nenhuma API garantia as mesmas funcionalidades que o Google Places, portanto optamos por esta, que não é gratuita de forma integral, após um determinado número de utilizações ela deixa de ser gratuita. Esta tarefa se deu mais pela

parte do Front-end, pois a API é utilizada em um campo de texto, onde o usuário digita parte do endereço desejado, com o número, e aparecem sugestões de busca, com até 5 itens. Com isso foi possível fazer uma boa integração com o backend, pois com apenas um campo era possível dividir as informações deste endereço, buscado pela API do Google, em diversos outros campos, como a coordenada, por exemplo, tornando útil a persistência desses campos.

Ao mover a tarefa relatada anteriormente, atuei em um problema encontrado no mapa na tela de detalhes da entrega do pedido, na visão do cliente. O Mapa não estava movendo para os lados corretamente, se tratando de uma aplicação mobile isto viria a ser um grande problema. Ao procurar a solução na internet, foi encontrado um uso de um componente pronto que provê essa funcionalidade, tornando móvel o mapa a partir de gestos no smartphone.

4.4 CONCLUSÃO

Ao final do projeto, entregamos uma aplicação com o escopo um pouco diferente do proposto inicialmente, removendo funcionalidades de piloto por falta de tempo e conhecimento técnico, o que não fez com que os Stakeholders se decepcionassem, a entrega do App foi um sucesso, deixando apenas uns pontos de melhorias até a apresentação para os alunos da AGES.

Nesse projeto, pude pôr em prática diversas habilidades vistas no curso, como a parte de versionamento, vista na disciplina de Gerenciamento de configuração de software, quanto deploy automatizado, visto na mesma disciplina, bem como padrões de projetos, visto em Projeto e Arquitetura de software. Posso dizer que aprofundei muito mais meus conhecimentos em pontos referente a DevOps, com a utilização de ferramentas de automação de build como Gradle e a própria plataforma de nuvem da AWS. Não considero este o melhor projeto trabalhado, por conta das tecnologias, não gostei da experiência de trabalhar com Flutter, por ser tecnologia mobile, o que dificultou um pouco o desenvolvimento de tarefas de front-end, mas não foi um impedimento. Poderia ter me esforçado mais em tentar entender a tecnologia, mas preferi me manter no campo do backend utilizando Java com Springboot, que é a tecnologia com a qual estou acostumado.

Em relação as soft skills, eu poderia ter tentado me entrosar mais com a equipe a fim de buscar trabalhar mais em conjunto nos finais de semana, pois acabava trabalhando sozinho e ficando sem tempo no meio da semana para parear com algum AGES II ou I, por exemplo. Essas e outras dificuldades, buscarei evitar que ocorram como aluno de AGES IV, tendo o papel de gerente de projeto, função a qual eu já me sinto preparado para exercer.

5 PROJETO AGES IV - “ADOTE UM AMANHÃ”

5.1 INTRODUÇÃO

O projeto Adote um Amanhã pretende proporcionar aos jovens residentes de lares de adoções, uma vida mais digna durante e após o período de moradia, limitado aos 18 anos. Para isso, tivemos como desafio criar funcionalidades que facilitem a obtenção do apoio da sociedade frente a esses problemas como, por exemplo, doação de bens ou serviços para as instituições que abrigam estes jovens.

Tivemos como clientes Nilson Ayala, pai adotivo de duas pessoas, que traz essa ideia tão nobre para nós desenvolvemos, e a Dra. Cinara Dutra, Promotora de Justiça da Infância e da Juventude de Porto Alegre, que com a ajuda do poder público, torna essa ideia uma realidade.

A grande expectativa deste projeto se deu por ele ser um complemento do aplicativo Adoções (PUCRS, 2017) desenvolvido na AGES tendo sido realizado junto com os mesmos clientes atuais. A diferença das duas aplicações se dá na forma de apoio ao jovem. Diferente da sistema atual, o aplicativo anterior - Adoções - tem como intuito facilitar a adoção da criança nessa situação.

No âmbito profissional, este item de portfólio serviu como preparação para gerenciamento de projetos, trazendo como desafios lidar com os clientes e com a equipe, suas habilidades e limitações. Também se caracterizou como a primeira escolha de projeto que haja um impacto social positivo, o que me fez sentir muito mais motivado com as entregas.

O projeto tem como professora orientadora a Profa. Alessandra Dutra e contou com 15 alunos originalmente no projeto. A figura 26 mostra o time reunido no final da *sprint* de levantamento de requisitos, junto dos clientes.

Figura 26 – Equipe de desenvolvimento, clientes e professor orientador.



Fonte: Wiki do projeto

5.2 DESENVOLVIMENTO DO PROJETO

5.2.1 TECNOLOGIAS

As escolhas das tecnologias decorreram de uma pesquisa feita no inicio do projeto, através do Google Forms, a qual se levantou o conhecimento do time acerca das tecnologias mais comuns no mercado e da AGES. O ponto de maior discórdia se deu na escolha da tecnologia para o Front-end, em que tivemos as duas opções como mais viáveis: *Vue.js* ou *React Native*. Algumas desvantagens da primeira em relação a segunda, é que a equipe não tinha conhecimento algum, exceto uma pessoa, o que não foi impeditivo para continuar com essa escolha, pois o *Vue* é conhecida por ter uma boa curva de aprendizagem. Além disso, essa linguagem não permite o desenvolvimento de aplicativos nativos, isto é, que sejam disponibilizados tanto na Google Playstore quanto na Apple Marketplace. Após diversos debates, decidimos fazer uma votação simples no servidor do *Discord* e *Vue.js* foi a tecnologia vencedora.

Todas as escolhas de tecnologias podem ser encontradas na página da wiki do projeto (WIKI, 2022d). A seguir, as tecnologias utilizadas no projeto:

5.2.1.1 JAVA

Java é uma linguagem de programação e plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. Há muitas aplicações e sites que não irão funcionar a não ser que você tenha Java instalado, e muito mais deles são criados todos os dias. Java é utilizado desde notebooks até data centers, video-games até supercomputadores científicos, desde telefones até a internet. Java está em tudo (JAVA, 2022).

A plataforma de software Java consiste na JVM (Java virtual Machine), a API Java e um completo ambiente de desenvolvimento. A JVM faz o parse do código e então interpreta o Java bytecode gerado. A API Java consiste em um extenso conjunto de bibliotecas, incluindo objetos básicos, funções de redes e segurança (IBM, 2022a).

É uma ferramenta utilizada nas disciplinas que precedem a AGES no curso de engenharia de software, com ela vínhamos criando desde aplicações simples até em conjunto com o Spring Boot, resultando em uma aplicação que pode ser utilizada para servidores back-end, sendo este próprio projeto um exemplo dessa combinação.

5.2.1.2 SPRING BOOT

O Spring Boot é uma ferramenta que faz o desenvolvimento de aplicações web e micro-serviços ficar mais fácil e rápido trazendo três principais benefícios: Configuração automática, configuração “opinativa” (*opinionated*), isto é, ele define diversas configurações de forma padrão, cabendo ao desenvolvedor substituí-las se quiser. Por último, também traz a habilidade de criar aplicações standalone, capaz de rodar de forma local no ambiente de desenvolvimento (IBM, 2022b).

5.2.1.3 VUE.JS

O Vue.js (comumente conhecido como Vue, pronunciado em inglês "view") é um framework progressivo do JavaScript de código aberto (open source) para a construção de interfaces de usuário. A integração em projetos que usam outras bibliotecas de JavaScript é facilitada com o Vue porque ele foi projetado para ser adotado de forma incremental. O Vue também pode funcionar como uma estrutura de aplicativos web capaz de alimentar aplicativos avançados de uma única página (FABRICIO, 2017).

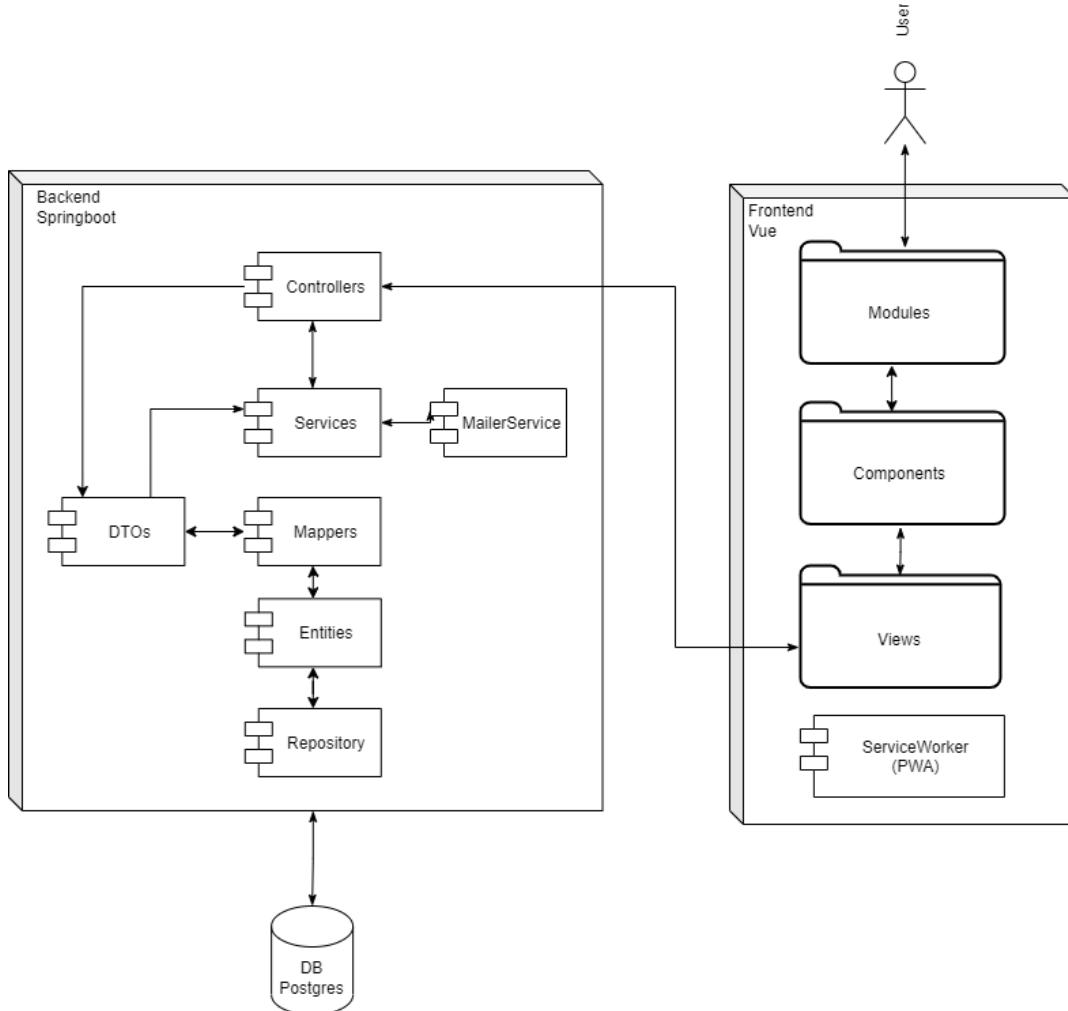
5.2.1.4 POSTGRESQL

PostgreSQL é um banco de dados relacional. Armazena dados em linhas com colunas que representam seus atributos. Uma tabela armazena multiplas linhas. O banco de dados relacional é o mais comum a ser utilizado nas aplicações. Funciona com muitas outras tecnologias e é alinhado com vários padrões de bancos de dados, o que garante sua extensibilidade (SHARMA, 2021).

5.2.2 ARQUITETURA

Após muitos debates sobre possíveis arquiteturas, os alunos do projeto chegaram em uma solução simples que soluciona o problema proposto. Dividimos em três camadas, front-end na tecnologia *Vue*, o *backend* nas tecnologia *Java Spring Boot*, e na camada de persistência o banco de dados *Postgres*, como demonstra a figura 27.

Para a infraestrutura do projeto, indo em encontro com as tecnologias utilizadas no Ministério Público, pensamos em manter hospedado em um servidor da AGES utilizando o docker para orquestrar as imagens dos serviços utilizados, tendo em vista que o cliente não dispõe de serviços em nuvem, como utilizado de costume pela agência. A figura 28 demonstra a orquestração dos serviços utilizando a tecnologia Docker. Toda arquitetura pode ser encontrada na página da wiki do projeto (WIKI, 2022a).

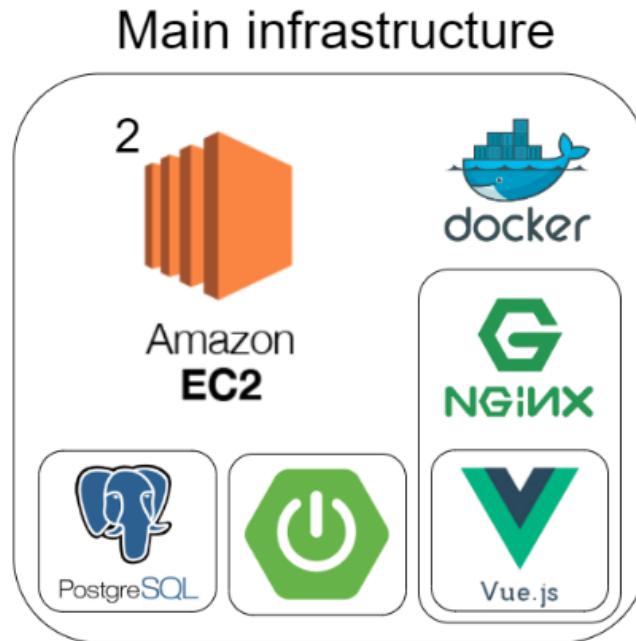
Figura 27 – Diagrama de componentes

Fonte: Wiki do projeto

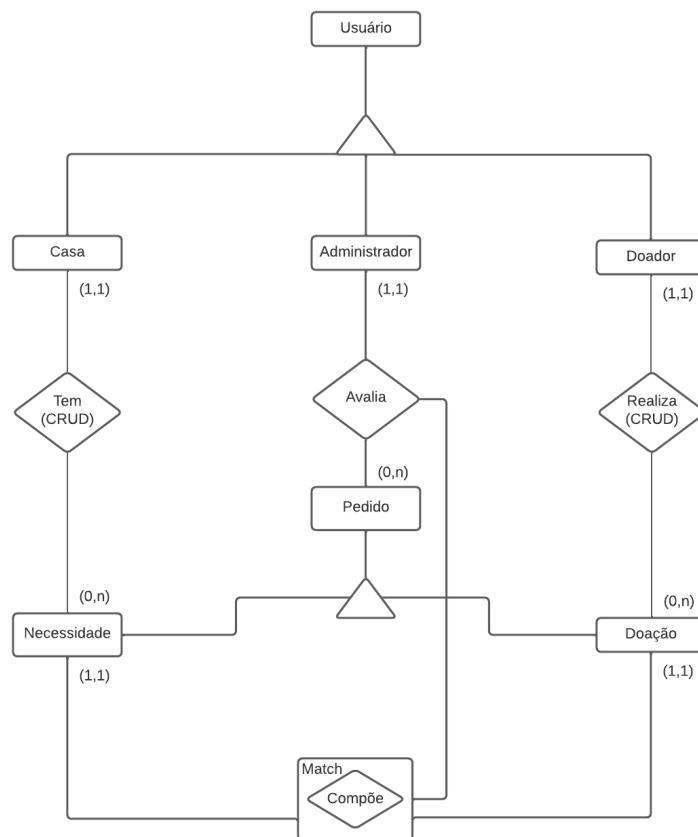
5.2.3 BANCO DE DADOS

Para a modelagem do banco de dados foi utilizada uma estrutura relacional. Como mostrado na figura 29, temos três tipos de usuários que possuem diferentes relações com o mesmo tipo de entidade. O usuário com perfil de Casa, tem acesso à entidade Pedido especializado como Necessidade, pois é um pedido a ser atendido. Já o usuário Doador, tem relação com o pedido em forma de Doação, pois ele está interessado em atender uma necessidade da casa, seja essa uma necessidade existente (doação vinculada) ou não existente (doação desvinculada). O administrador é responsável por avaliar os pedidos, quando uma doação combina com uma necessidade.

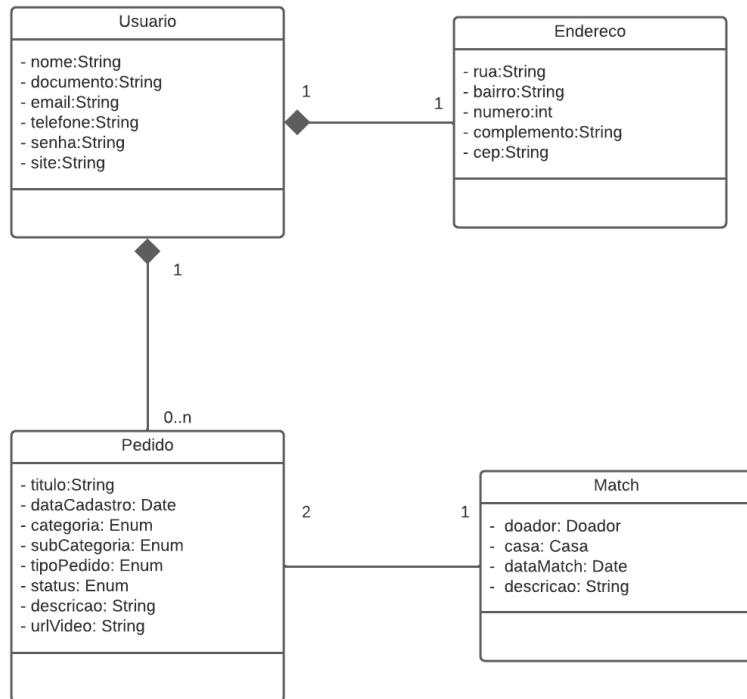
Na modelagem lógica, de forma bem suscinta foi possível representar todos os dados necessários pela aplicação em apenas 4 tabelas, como mostrado na figura 30. Todas informações referente a banco de dados, podem ser encontradas na página da wiki (WIKI, 2022b).

Figura 28 – Diagrama de Infraestrutura

Fonte: Wiki do projeto

Figura 29 – Diagrama relacional do banco de dados

Fonte: Wiki do projeto

Figura 30 – Diagrama Lógico do banco de dados

Fonte: Wiki do projeto

5.2.4 PROTÓTIPOS

A prototipação das telas iniciou com desenhos no quadro, conforme a figura 31, com a ajuda da professora e interação dos colegas, tendo como base os requisitos solicitados pelos clientes.

Após ter uma ideia de como deveria ser pensado o sistema, iniciamos a prototipação de alto nível na ferramenta figma, tendo como líder dessa tarefa a Clara D'ávila (AGES II), que apoiou tanto na criação dos protótipos quanto na busca de uma identidade visual para o sistema.

A figura 32 mostra a ideia inicial de como deveria ser a tela de cadastro de um usuário doador, com informações pessoais, incluindo login (e-mail) e senha. Já na tela de detalhe de uma necessidade, mostramos uma descrição mais abrangente do pedido de ajuda de uma casa, para que o doador tenha uma noção melhor da solicitação.

A listagem de necessidades (figura 33) serve para que o doador possa ver em apenas uma página, tudo que está sendo necessitado pelas instituições de adoção, para assim avaliar o que combina com o que ele tem para doar. Demais protótipos podem ser encontrados na página da wiki (WIKI, 2022c).

Figura 31 – Protótipos iniciais de baixo nível

Fonte: Wiki do projeto

Figura 32 – Tela de cadastro de doador e detalhe de uma necessidade, respectivamente

← Adote um Amanhã
SOLICITAÇÕES



Nome Completo
Exemplo de Souza

CPF/CNPJ
123.456.789.15

CEP
95720-000

Endereço
Av. Ipiranga 6682 - Partenon, 91530-035

Complemento
Casa

E-mail
exemplo@gmail.com

Senha

Arroz e feijão

Data cadastro: 25/12/2022

Categoria: Produto

Subcategoria: Alimentos

Descrição:

Doação de 10 pacotes de arroz da marca são joão validade até 02/2023.


CEPA - Casa de Passagem

← Voltar
Doar

Cadastrar

Fonte: Wiki do projeto

Figura 33 – Tela de listagem de necessidades

Fonte: Wiki do projeto

5.3 ATIVIDADES DESENVOLVIDAS PELO ALUNO NO PROJETO

5.3.1 SPRINT 0

Dado o início do projeto, começamos organizando quem dos gerentes (AGES IV) seria responsável por cada tarefa, fiquei responsável por criar o servidor no Discord, que seria o principal meio de comunicação da equipe. Em conjunto com os outros colegas gerentes, realizei um formulário no Google Forms que serviu como um levantamento de conhecimento da equipe inteira sobre as possíveis tecnologias utilizadas.

Escolhemos a ferramenta Trello como Kanban que seria utilizado durante o desenvolvimento do projeto para guiar e manter a equipe alinhada do que estava sendo feito, portanto, criei um quadro inicial e modelei conforme nossas necessidades de processo de desenvolvimento. Em um primeiro momento, decidimos separar o fluxo na seguinte maneira: *Backlog*, Pronto para desenvolver, em desenvolvimento, revisão de código, integração e testes, e entregue. Porém, isso seria um problema no processo de desenvolvimento que será melhor descrito na sprint 1. Após essa configuração inicial do Trello, adicionei todas as tarefas trabalhadas na sprint 0, para a equipe já se habituar ao modelo de trabalho.

Durante a escolha das tecnologias, os arquitetos (AGES III) sugeriram a utilização de microsserviços, porém me opus desde o início por pensar que o projeto não era escalável o suficiente como POC para justificar o uso de tal arquitetura. Com isso, e conforme conversado com o arquiteto da AGES, Cássio Trindade, entramos em um consenso mantendo o projeto em uma estrutura monolítica.

Quanto as escolhas das tecnologias, ainda não havia sido decidida, através do conhecimento dos alunos, a ferramenta ideal para se trabalhar com o front-end, estávamos indecisos entre duas possíveis opções: Vue.JS ou React Native. Nós gerentes, juntos dos arquitetos, decidimos por Vue.JS, porém a professora orientadora se opôs, pois, não havia conhecimento suficiente na equipe que justificasse tal escolha. Então foi feita uma votação no servidor do Discord, que decidiu que o front-end seria escrito em Vue.js. Após essa escolha me dediquei em um pequeno curso para voltar a aprender esta linguagem, pois a única vez que havia utilizado ela havia sido em meu primeiro estágio, por isso já não recordava como era utilizada.

Dei apoio aos demais colegas no início do design da aplicação na ferramenta Figma, o que foi uma reunião muito produtiva. A colega Clara (AGES II) encontrou uma figura de sol que combina com o nome da aplicação, relacionando sol com amanhã, portanto já tínhamos o início da identidade visual do projeto. Com isso escolhi uma paleta de cores que combinasse com esse padrão. Diante de tanta empolgação, escolhemos uma fonte para o projeto que parecia divertida, mas fomos descobrindo que isso tornava o visual do projeto muito infantilizado e de certa forma, feio. A colega Vitória (AGES I) levou os protótipos para uma pessoa que trabalhava com design que deu dicas de como poderíamos melhorar, o que foi de grande utilidade. Também trabalhei em diversos fluxos do design e transformei o que estava sendo utilizado de forma repetida em diversas telas em componentes, com a ajuda do colega Vinicius (AGES IV).

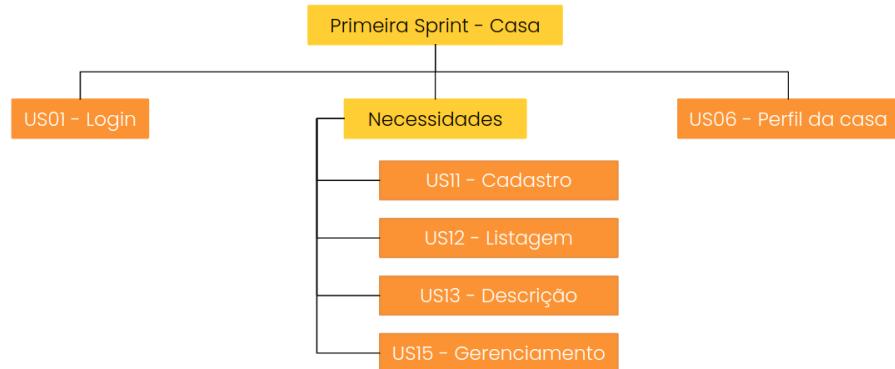
Enquanto isso na parte das histórias de usuário, comecei a desenvolver o modelo que escrita que utilizaríamos, começando pelas histórias referentes ao doador e a casa, que se tornaria nosso foco nas sprints iniciais, neste momento ainda não haviam sido adicionadas observações e critérios de satisfação em cada história de usuário por falta de tempo, portanto deixamos somente o essencial em um primeiro momento. Após a conclusão das histórias de usuários, adicionei todas na ferramenta Air Table, que melhora o gerenciamento delas, facilitando acompanhar elas e editar.

Também apoiei o grupo (AGES II) responsável pelo desenvolvimento dos diagramas de banco de dados, dando dicas de negócio que ajudaram a simplificar os desenhos, pois os fluxos não eram muito complexos, em questão de dados.

Apresentamos para os clientes o nosso trabalho até aqui, que continha os protótipos e histórias de usuários, bem como nossa proposta de desenvolvimento para a próxima sprint, como demonstrado na figura 34. Os clientes ficaram bastante contentes com nossa proposta, o que foi um grande motivador para o nosso trabalho.

Para realizar a retrospectiva da sprint, busquei no site Fun Retrospectives uma ideia divertida e contextualizasse corretamente o nosso momento atual no projeto. Escolhemos uma retrospectiva do Hot-Air Balloon, que separava pontos positivos que ajudavam nossa equipe a decolar, e pontos a melhorar, que impediam que ela subisse. A retrospectiva foi bastante produtiva e deu alguns itens de ações para futuras melhorias, entre eles: monitorar o escopo, trabalho em grupos menores, manter o código limpo e escalável, chamar a equipe para trabalhar

Figura 34 – Corte da EAP referente ao planejamento para a Sprint 1



Fonte: Wiki do projeto

no Discord, não esperar ninguém chamar para começar algo, melhorar a documentação na wiki e finalizar os diagramas de banco de dados.

5.3.2 SPRINT 1

Comecei a primeira sprint adicionando o que havia faltado na sprint anterior nas histórias de usuário no Trello, como protótipos, descrição e critérios de aceite. Também refinei as histórias para poderem ser trabalhadas em pequenas sub-tarefas de front e back-end. Logo no início da sprint, também houve uma chamada para ensinar ao resto da equipe como funcionava a tecnologia Vue.JS. Tive participação desta chamada atuando também de forma prática para aprender.

Como primeira tarefa atribuída a mim, ajudei a colega Mariela (AGES I) a se ambientar com os projetos, para ela poder desenvolver de forma autônoma, sem eu precisar estar sempre junto dela. Após a configuração correta, iniciei a tarefa de backend da história de usuário US06. Aproveitei essa oportunidade para ensinar a Mariela como realizar testes unitários e testes de integração durante o desenvolvimento, para garantir uma melhor qualidade de código, ela se mostrou bastante proativa e entendeu rapidamente como funcionava essa metodologia de desenvolvimento. Utilizamos *pair programming* com a ajuda de um cronômetro para ficarmos a mesma quantidade de tempo no comando do teclado no código. Após finalizar o desenvolvimento da tarefa anterior, adicionei as histórias de usuário da sprint 2 como *backlog* no Trello, para que se algum membro da equipe se disponibilizasse, tivesse tarefa para ele realizar. Também adicionei todas as histórias de usuário na wiki.

Quando as tarefas referente à história de usuário US06 ficaram prontas trabalhei na integração das mesmas, para isso foi necessário o ajuste dos dados no backend, pois estavam sendo criados dados fictícios pouco coerentes com a realidade. Nessa sprint também trabalhei com muita revisão de código, colocando meu ponto de vista e dando dicas de melhorias no código dos colegas.

Finalizada a parte de desenvolvimento, trabalhei na criação da apresentação em PowerPoint

do que havíamos desenvolvido na sprint 1. Enquanto isso, a professora detectou várias falhas ao testar a aplicação que estava sendo executada em ambiente de homologação, para isso foi necessária uma força tarefa para podermos corrigir as falhas antes da apresentação no dia seguinte.

A falha que estava ocorrendo no processo de desenvolvimento nesse ponto do projeto era que não estávamos testando corretamente a aplicação após a integração, ponto que buscamos melhorar para as próximas apresentações. Para isso decidimos separar no Trello a coluna Integração e Testes em duas colunas com atividades específicas: uma para integração e outra para testes.

Na apresentação da sprint 1 conseguimos ter todas histórias desenvolvidas aceitas pelos clientes, mas com devidas ressalvas quanto a falhas percebidos durante a atividade. Aprendemos a não utilizar termos de desenvolvimento com o cliente durante a apresentação, como, por exemplo, débitos técnicos, pois esses poderiam confundi-los. Além disso, foram utilizados protótipos na apresentação, o que não condizia totalmente com as telas desenvolvidas, pois após o desenvolvimento dos protótipos não houve modificações que eram requeridas pelos clientes. Esses pontos deixaram evidentes a falta de preparo que nós, AGES IV, tivemos para a apresentação.

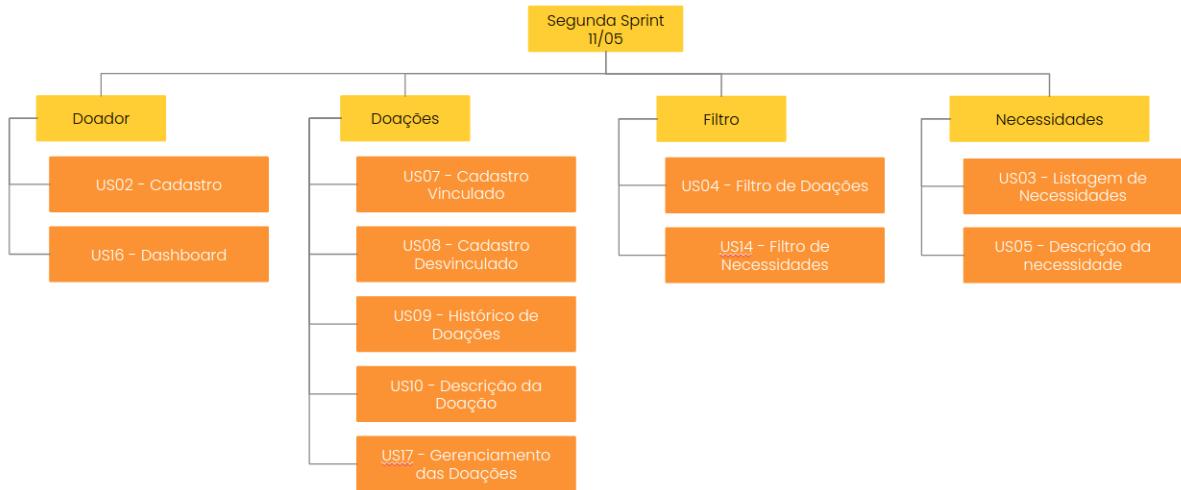
Para a retrospectiva dessa sprint, como a data ficava muito próxima da páscoa, decidi fazer de forma temática, relacionando ao feriado, combinei com os outros colegas de trazer doces e distribuir durante a cerimônia. Como itens de ação desta retrospectiva temos: se preparar melhor para a apresentação, treinar ela com os colegas um dia antes, pesquisar mais antes de pedir ajuda, porque o colega Vinicius estava sendo muito demandado por ser o mais experiente com Vue.JS. Criar pipelines para ter um processo de *deploy* mais rápido e automatizado, testar funcionalmente a aplicação antes de declarar a história como entregue, testar no celular enquanto estiver desenvolvendo.

5.3.3 SPRINT 2

Planejamos esta próxima sprint conforme a figura 35, aparentemente são muitas histórias, porém boa parte das funcionalidades são muito parecidas com o que já havíamos desenvolvido nessa primeira sprint.

Comecei atualizando as tarefas planejadas no Trello, adicionando as descrições e dividindo em sub-tarefas. Com a ajuda dos colegas AGES IV, foi possível refinar as tarefas para podermos componentizar as funcionalidades e reutilizar o melhor possível, para que não houvesse retrabalho ou duas pessoas trabalhando em paralelo em desenvolvimentos muito semelhantes.

Como tarefa de desenvolvimento, escolhi começar com o colega Cleyson a parte do front-end da história US03, referente a tela de listagem de necessidades para um doador, porém meu par já estava mais avançado na tarefa e obtendo ajuda do colega Vinicius, portanto decidi avisar o Cleyson de que iria partir para outra atividade em que era requisitado pelo front-end, que se tratava de retornar as informações do usuário logado.

Figura 35 – Corte da EAP referente a Sprint 2

Fonte: Wiki do projeto

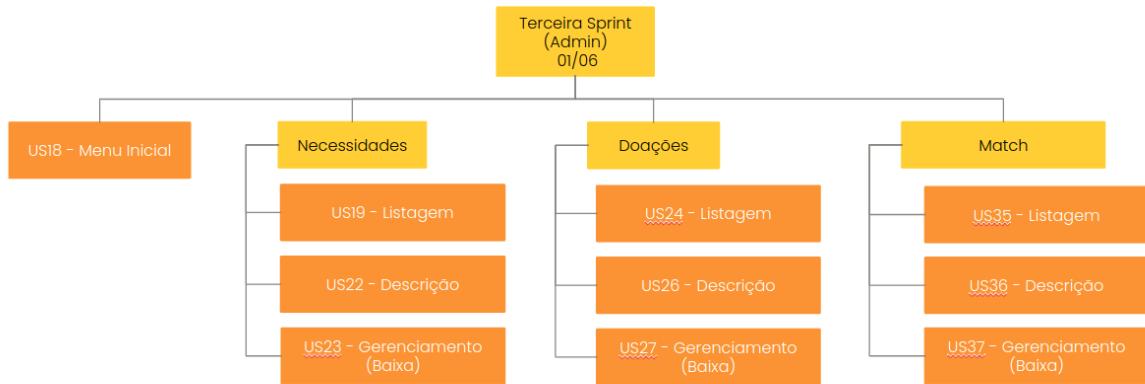
Observando o Trello, percebi que minhas colegas estavam com tarefas de mais no front-end, e como elas preferiam focar nessa parte, decidi pedir uma das tarefas delas de backend para agilizar o processo, então me apropriei da tarefa US10, descrição de uma doação.

Ao ver que o colega Pedro Henrique estava devendo horas complementares, decidi ajudá-lo e chamei para fazer junto comigo a US09, que se tratava de criar uma rota no backend que retornasse o histórico de doações de um usuário doador, com essa interação, consegui passar alguns conhecimentos sobre o desenvolvimento backend e conseguimos fazer alguns testes unitários.

As colegas Vitória e Clara precisavam de ajuda no front-end, área em que meu colega Vinicius Bazanella tem mais domínio, mas como ele não estava disponível para ajudar, me dispus a prestar essa ajuda. Apesar da pouca experiência que tenho com o front-end, acredito que consegui ajudá-las de forma satisfatória.

Após o desenvolvimento das tarefas, fui entender o que faltava ainda para ser entregue, pois, o prazo estava apertado. A entrega seria para a quarta e já era segunda. Então decidi mudar o foco da *daily*. Ao invés de passar um por um integrante da equipe, e cada um dizer o que fez, resolvemos abrir o quadro no Trello na tela principal da sala e nos guiar por cartão de atividade, não mais por quem estava fazendo. Isso nos deu visibilidade do que faltava ser entregue e nos mostrou um grande problema que estávamos passando, a má utilização do quadro, pois não estava sendo atualizado pelos colegas.

Nessa entrega, o Nilson não pode participar, o que causou uma certa perda no valor da apresentação, pois teria sido muito proveitosa o lado crítico dele. Essa apresentação foi feita por mim, e bastante elogiada pelos colegas e pela professora, pois consegui fazê-la com muita calma e conduzindo bem. Conseguimos entregar todas as histórias planejadas anteriormente, salvo algumas pequenas correções percebidas durante e depois do desenvolvimento. Junto da Cinara, combinamos de entregar para a terceira sprint conforme a figura 36.

Figura 36 – Corte da EAP referente a Sprint 3

Fonte: Wiki do projeto

Na retrospectiva, feita pelo colega Marcelo, conseguimos identificar alguns pontos de melhoria no processo e no desenvolvimento. A parte dos empurroezinhos mostrou que alguns colegas trabalharam muito aquém do esperado pela agência, o que junto de um feedback com a professora, resultou na saída de um colega AGES I, Eduardo, por ele não se sentir preparado para essa metodologia de desenvolvimento. Além disso, ficou clara algumas falhas na comunicação que tínhamos no desenvolvimento, assim fizemos alguns itens de ação para corrigi-las. Recebi elogios do colega Guilherme (AGES I), que ficou feliz por eu ter me disponibilizado a ajudá-lo no *back-end*.

Ainda na retrospectiva, identificamos também que havíamos feito poucos testes após o desenvolvimento, por falta de tempo, pois deixamos para desenvolver muito ao fim da sprint. Com isso chegamos a conclusão de deveríamos aplicar um *code freezing*, isto é, botar um limite no tempo de desenvolvimento antes da entrega, reservando os últimos dias antes dela para realizar apenas testes.

5.3.4 SPRINT 3

Após o combinado a ser entregue na sprint 3, mostrado na sessão anterior, iniciamos passando as tarefas acordadas para o Trello. Logo no início do desenvolvimento dessa Sprint, a colega Mariela (AGES I) tomou a iniciativa de puxar as tarefas da história US23, referente a gerência de vínculos de doações e necessidades por parte do administrador. Para isso pediu a minha ajuda para começar a desenvolver. Conseguí parear com ela, mas ela sempre se manteve no comando do código, eu só estava ali para guiar ela durante o desenvolvimento, o que funcionou para um bom desenvolvimento. Com isso terminamos a tarefa de recusar uma necessidade.

Após isso ajudei o colega Guilherme (AGES I) no desenvolvimento da tarefa de listagem de vínculos (US35), conseguimos desenvolver rapidamente esta tarefa, pois já existia implementação

parecida no *back-end* para a busca de pedidos.

De volta na tarefa da US23 com a Mariela, pensamos que a tela nos protótipos havia ficado com uma péssima experiência e difícil de ser implementada, para isso chamei outros dois colegas AGES IV, Vinicius e Marcelo para replanejar a execução dessa história. Com isso chegamos a uma nova tela redesenhada e que apresentou uma experiência melhor para o administrador a ser utilizada. A Colega mariela conseguiu implementar boa parte da tela, mas com o Vinícius consegui finalizar essa tarefa, tornando a tela mais agradável de ser utilizada.

No meio do desenvolvimento, houveram algumas falhas na comunicação e algumas colegas vieram até mim, relatar um problema de que um colega tinha se apropriado da tarefa e desenvolvido sozinho, sem avisar que faria isto. Para isso decidi falar com ambas as partes no privado e tentar resolver esse conflito sem gerar mais estresse.

Antes de considerar as tarefas entregues, fui repassando no Trello o que faltava ser entregue ainda, para buscar possíveis falhas no processo de entrega, desta vez havia fluido melhor o fluxo, após os levantamentos da retrospectiva da sprint passada. Após essa revisão trabalhei com o colega Pedro em algumas correções que eram necessárias ainda antes da entrega.

Participei com a equipe de arquitetura e gerência no *deploy* da aplicação para a apresentação da Sprint 3, foi uma tarefa divertida, pois nos reunimos e acabamos interagindo e dando muitas risadas durante o processo. Conseguimos realizar o *deploy* após algumas dificuldades na automação desse processo.

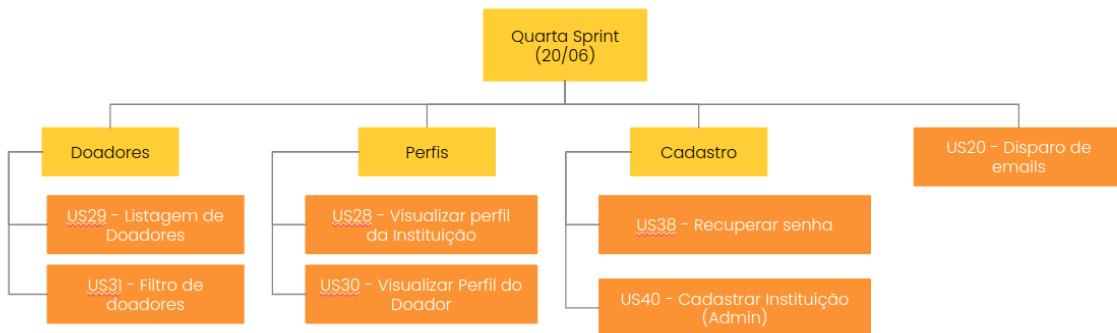
Na apresentação dessa entrega, ficou evidente a falta de documentação que tínhamos sobre o fluxo de doações e necessidades, pois os clientes tinham muitas dúvidas de como funcionaria esses processos. Após alinharmos todos esses passos, saímos com a tarefa de documentar corretamente o fluxo para ser mais fácil de entender quais as atuações das instituições e do administrador no sistema. Fora este pequeno problema, consideramos a entrega bem sucedida, com todos itens entregues.

5.3.5 SPRINT 4

Durante a reunião de planejamento da Sprint 4, a professora solicitou que removêssemos uma funcionalidade referente a bloqueio de usuários no sistema, por pensarmos que traria menos valor ao produto que a recuperação de senha, por exemplo. A figura 37 mostra as histórias combinadas para essa Sprint.

Como gerente do projeto e em final do curso, tive que me organizar para balancear o tempo gasto no memorial descritivo, para que não atrapalhasse a equipe no desenvolvimento. Para isso me disponibilizei caso tivessem dúvidas em alguma tarefa ou estivessem bloqueados.

Fiquei responsável junto do colega Pedro, AGES III, pela tarefa de back-end das histórias US29 e US31, que tinham como objetivo listar e filtrar doadores pelo nome, para poder ser acessado de forma fácil pelo administrador. Enquanto desenvolvia ela, percebi que não fazia sentido o filtro de número de doações, previstas nos protótipos, então conversei com o Vinicius Bazanella, que estava responsável pelo front-end dessas histórias, e decidimos remover esse

Figura 37 – Corte da EAP referente a Sprint 4

Fonte: Wiki do projeto

filtro, deixando apenas possível buscar pelo nome do doador.

Antes de chegar o período final de desenvolvimento, para que não houvesse entregas atrasadas ou maiores imprevistos, resolvi verificar todas as tarefas no Trello e conversar com cada responsável por cada uma delas para atualizar o quadro e saber se a história estava em vias de acabar.

Na apresentação da sprint 4 para os clientes, tivemos um convidado especial da empresa que provavelmente dará seguimento ao aplicativo visto que acabamos o desenvolvimento desse projeto inicial na AGES. Fizemos uma parte especial da apresentação para facilitar o entendimento das funcionalidades que desenvolvemos desde o início do projeto.

Nessa Sprint não teve retrospectiva do nosso projeto, apenas a retrospectiva geral da AGES, para isso juntamos os gerentes de todos os projetos e fizemos a cerimônia com todas as equipes. Foram levantadas pautas referentes a AGES, como a infraestrutura, materiais disponibilizados para o estudo dos alunos, etc.

5.4 CONCLUSÃO

Com o fim deste projeto, encerra-se um grande ciclo de aprendizado, e com isso, acredito que conclui o que se tornou um grande objetivo durante a vida acadêmica: passar um pouquinho de conhecimento de cada área que obtive durante o curso inteiro para as próximas gerações de participantes dos projetos da agência.

Este foi o melhor projeto que participei e o meu preferido, pois foi o que mais enxerguei propósito para fazer as coisas darem certo. Desde a animação dos clientes diante das nossas entregas, até ver que o produto ia ser realmente utilizado após nosso desenvolvimento, e ia trazer mudanças reais na vida de pessoas que se encontram em situações de necessidade, precisando conectá-las com doadores. Além disso, a participação e engajamento da equipe tornou tudo mais agradável de se trabalhar, pois, mesmo com a falta de conhecimento da tecnologia de front-end, não nos deixamos abater e conseguimos entregar um produto mínimo

viável que pode ser mantido por uma empresa terceira.

Fiquei bastante satisfeito com esse projeto, pois tive feedbacks muito positivos dos meus colegas quanto a minha atuação como AGES IV. O conhecimento que obtive na disciplina de gerenciamento de projetos se mostrou muito importante ao ter que lidar com tempos e movimentos para as entregas desse projeto.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento de software mudou a minha vida. Com todo o apoio da minha família e muito estudo, consegui ir atrás dos meus sonhos. A conclusão deste curso é um dos maiores deles, o que me possibilitou conseguir um emprego na área que eu amo e em uma empresa que eu acredito, e que também levou orgulho para a minha mãe.

No início do curso não compreendia nem metade do que era a área de engenharia de software, desde escrever simples programas de soma, até compreender todo o processo de criação de um sistema, levou um pouco de tempo. A cada etapa vivenciada na agência, o quebra cabeça que ia se montando passava a fazer mais sentido na minha vida, o que aumentava o propósito do que eu estava procurando. Hoje após tudo o que foi vivenciado, tenho uma visão muito mais ampla do que consiste esse processo de criação.

Durante o desenvolvimento de qualquer projeto, busquei pôr em prática tudo aquilo que havia aprendido, seja por disciplina teórica do curso ou o que aprendia na prática nos estágios e no trabalho. Isso proporcionou uma troca de experiências com meus colegas, dificilmente vista em outros cursos de forma tão proveitosa em momentos tão iniciais da graduação. A agência cumpre um papel muito útil na área de engenharia de software, trazendo essa possibilidade para nós, alunos do curso e futuros engenheiros.

Nas etapas iniciais da agência, fui buscando primeiro aprender as partes técnicas, tentando dominar as tecnologias com as quais iria trabalhar, sempre focando no back-end, pois era o que tinha mais afinidade. Por ter esse lado mais técnico, no começo não havia um senso de importância para o lado comportamental, portanto comecei sendo muito tímido e pouco expressivo, omitindo opiniões por achar bobas, ou pensar que não teria domínio para debater, mas conforme evoluía na parte técnica e ganhava mais domínio sobre o desenvolvimento. Comecei a dar a devida importância para a parte comportamental e busquei participar mais ativamente nos debates, pois, havia percebido que poucas decisões eram tomadas com base neles. Ao invés disso, o aluno que demonstrava ter mais experiência tomava as decisões e a equipe não tinha base suficiente para tentar outros caminhos.

Na parte técnica, desde que aprendi sobre testes unitários e os benefícios de se utilizar no seu desenvolvimento, sempre busquei incentivar meus colegas a utilizá-los e tomei iniciativas de iniciar testes unitários durante e até antes do desenvolvimento, tentando utilizar quando possível, técnicas de *Test Driven Development* (TDD).

Na parte comportamental, sempre busquei ser sincero sobre minhas atividades durante todos os projetos, tentando dar a maior publicidade sobre o que estava fazendo, e sempre indo atrás de receber e dar feedbacks aos meus colegas e professores, pois acredito que todos são iguais e capazes de aprender uns com os outros.

As partes técnicas e comportamentais se encontram no momento em que eu passo a transmitir todo conhecimento que eu tenho para os colegas, como se fosse um profissional sênior passando conhecimento ao estagiário que está no início de carreira.

Como contribuição para a agência, acredito que deveríamos incentivar e dar mais autonomia

para os alunos a darem feedbacks aos seus pares, colegas com quem trabalham durante a disciplina inteira, ensinando até mesmo a aceitar um feedback, sem levar para o lado pessoal quando é feito de forma construtiva. Essa ideia trago desde a minha primeira experiência, na Aceleradora Ágil, onde eramos ensinados a dar e pedir feedbacks em cada interação com um colega, o que tornava muito mais efetivo o aprendizado.

Além da questão do feedback nas interações, percebi que as disciplinas iniciais no curso contam com monitores para auxiliar na resolução dos exercícios, como, por exemplo, cálculo ou fundamentos de programação. Mas, por que não aplicar isso na principal disciplina do curso de Engenharia de Software? Penso que traria muitos benefícios, tanto para alunos que recém estão começando a disciplina inicial da AGES, quanto para os que estão o ajudando na resolução das tarefas, pois, acredito que grande parte do aprendizado está em ensinar os outros. Esse sistema poderia ser facilmente evoluído no futuro se tornando um sistema de mentoria, onde alunos mais experientes se tornam mentores de um ou mais participante da agência.

Por fim, levarei todo o conhecimento obtido durante todas as etapas da agência para as minhas próximas experiências.

REFERÊNCIAS

- AWS. **Getting started with Amazon S3.** 2022. Acesso em: 26 de setembro de 2021. Disponível em: <<https://aws.amazon.com/s3/getting-started/>>. Citado na página 34.
- _____. **Sobre a AWS.** 2022. Acesso em: 26 de setembro de 2021. Disponível em: <<https://aws.amazon.com/pt/about-aws/>>. Citado na página 33.
- _____. **What is Amazon EC2?** 2022. Acesso em: 26 de setembro de 2021. Disponível em: <<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>>. Citado na página 33.
- _____. **What is Amazon Elastic Container Registry?** 2022. Acesso em: 26 de setembro de 2021. Disponível em: <<https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html>>. Citado na página 34.
- FABRICIO. **O que é Vue?** 2017. Acesso em: 24 de abril de 2022. Disponível em: <<https://www.devmaster.com.br/vue-js-tutorial/38042#Vue>>. Citado na página 50.
- IBM. **What is Java?** 2022. Acesso em: 24 de abril de 2022. Disponível em: <<https://www.ibm.com/cloud/learn/java-explained>>. Citado 3 vezes nas páginas 22, 32 e 49.
- _____. **What is Java Spring Boot?** 2022. Acesso em: 24 de abril de 2022. Disponível em: <<https://www.ibm.com/cloud/learn/java-spring-boot>>. Citado 3 vezes nas páginas 22, 33 e 49.
- JAVA. **What is Java and why do I need it?** 2022. Acesso em: 24 de abril de 2022. Disponível em: <https://www.java.com/en/download/help/whatis_java.html>. Citado 3 vezes nas páginas 22, 32 e 49.
- PUCRS. **Convênio permite criação de aplicativo que facilita adoção de crianças no RS.** 2017. Acesso em: 24 de abril de 2022. Disponível em: <<https://www.pucrs.br/blog/convenio-permite-criacao-de-aplicativo-que-facilita-adocao-de-criancas-no-rs/>>. Citado na página 48.
- REACT. **React Uma biblioteca JavaScript para criar interfaces de usuário.** 2022. Acesso em: 24 de abril de 2022. Disponível em: <<https://pt-br.reactjs.org/>>. Citado na página 14.
- SHARMA, Kris. **What is postgresql.** 2021. Acesso em: 26 de abril de 2022. Disponível em: <<https://ubuntu.com/blog/what-is-postgresql>>. Citado 2 vezes nas páginas 22 e 50.
- THOMAS, Gaël. **What is Flutter and Why You Should Learn it in 2020.** 2020. Acesso em: 26 de setembro de 2021. Disponível em: <<https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>>. Citado na página 33.
- WIKI. **Página de banco de dados do projeto Mapas Culturais.** 2019. Acesso em: 28 de abril de 2022. Disponível em: <https://tools.ages.pucrs.br/Ibere/wiki/-/wikis/banco_dados>. Citado na página 16.
- _____. **Página inicial da wiki Mapas Culturais.** 2019. Acesso em: 24 de abril de 2022. Disponível em: <<https://tools.ages.pucrs.br/Ibere/wiki/-/wikis/home>>. Citado na página 12.

- _____. **Tabela de tecnologias do projeto Mapas Culturais.** 2019. Acesso em: 24 de abril de 2022. Disponível em: <<https://tools.ages.pucrs.br/Ibere/wiki/-/wikis/tabela-tecnologias>>. Citado na página 13.
- _____. **Banco de dados do projeto Apus Drones.** 2020. Acesso em: 26 de abril de 2022. Disponível em: <https://tools.ages.pucrs.br/apus-drones/apus-drones-wiki/wikis/banco_dados>. Citado na página 37.
- _____. **Mockups do projeto Apus Drones.** 2020. Acesso em: 26 de abril de 2022. Disponível em: <https://tools.ages.pucrs.br/apus-drones/apus-drones-wiki/-/wikis/design_mockups>. Citado na página 39.
- _____. **Modelagem de dados do projeto Compartilhamento.** 2020. Acesso em: 12 de junho de 2022. Disponível em: <https://tools.ages.pucrs.br/compartilhamento/compartilhamento-wiki/-/wikis/banco_dados>. Citado na página 23.
- _____. **Protótipos de alto nível do projeto Compartilhando Experiências.** 2020. Acesso em: 24 de abril de 2022. Disponível em: <<https://tools.ages.pucrs.br/compartilhamento/compartilhamento-wiki/-/wikis/mockups>>. Citado na página 25.
- _____. **Wiki do projeto Apus Drones.** 2020. Acesso em: 26 de abril de 2022. Disponível em: <<https://tools.ages.pucrs.br/apus-drones>>. Citado na página 41.
- _____. **Arquitetura do projeto Adote um Amanhã.** 2022. Acesso em: 26 de abril de 2022. Disponível em: <<https://tools.ages.pucrs.br/adote-um-amanha/adote-um-amanha-wiki/-/wikis/arquitetura#arquitetura-geral-da-aplicação>>. Citado na página 50.
- _____. **Banco de dados do projeto Adote um Amanhã.** 2022. Acesso em: 26 de abril de 2022. Disponível em: <https://tools.ages.pucrs.br/adote-um-amanha/adote-um-amanha-wiki/-/wikis/banco_dados>. Citado na página 51.
- _____. **Protótipos do projeto Adote um Amanhã.** 2022. Acesso em: 26 de abril de 2022. Disponível em: <https://tools.ages.pucrs.br/adote-um-amanha/adote-um-amanha-wiki/-/wikis/design_mockups>. Citado na página 53.
- _____. **Tecnologias do projeto Adote um Amanhã.** 2022. Acesso em: 26 de abril de 2022. Disponível em: <<https://tools.ages.pucrs.br/adote-um-amanha/adote-um-amanha-wiki/-/wikis/arquitetura#backend>>. Citado na página 49.