

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

JOÃO PEDRO SALLES DA SILVA

**MEMORIAL DE ATUAÇÃO NA AGÊNCIA EXPERIMENTAL DE ENGENHARIA DE  
SOFTWARE - PERÍODO 2023 A 2025**

**AGES I, II, III**

Porto Alegre

2025

## **Dedicatória**

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

## **AGRADECIMENTOS**

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

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

*“Epígrafe: É um item onde o autor apresenta a citação de um texto que seja relacionado com o tema do trabalho, seguido da indicação de autoria do mesmo. (texto iniciando do meio da página alinhado a direita) “*

**Autor da Epígrafe**

## **RESUMO**

Neste relato, compartilho minha jornada na AGES (Agência Experimental de Engenharia de Software), começando como AGES I, onde minha principal responsabilidade era contribuir para o desenvolvimento e testes do software. Ao progredir para a posição de AGES II, assumi um papel mais aprofundado na construção e no aprimoramento do banco de dados, bem como na implementação de novas funcionalidades. Na AGES III, ajudei a produzir toda a arquitetura de software do projeto, além de desimpedir e ajudar em questões técnicas aqueles que têm mais dificuldades e os AGES IV fazem parte da gerência.

**PALAVRAS CHAVES:** AGES, Engenharia de Software, Aprendizado, Projeto, Desenvolvimento.

## **LISTA DE ILUSTRAÇÕES**

Figura 1 – Time . . . . .	10
Figura 2 – Protótipo no Figma . . . . .	12
Figura 3 – Time . . . . .	16
Figura 4 – Protótipo no Figma . . . . .	18
Figura 5 – Time - Se Doce Fosse . . . . .	25
Figura 6 – Banco de Dados . . . . .	26
Figura 7 – Diagrama de Arquitetura . . . . .	27
Figura 8 – Protótipo no Figma - Home . . . . .	28
Figura 9 – Protótipo no Figma - Página de Produtos . . . . .	29
Figura 10 – Protótipo no Figma - Tabela de pedidos (Admin) . . . . .	30
Figura 11 – Protótipo no Figma - Página Admin . . . . .	30

## **LISTA DE TABELAS**

## **LISTA DE SIGLAS**

**AWS** Amazon Web Services . . . . . 27

## SUMÁRIO

<b>1</b>	<b>APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO . . . . .</b>	<b>9</b>
<b>2</b>	<b>AGES I — IMIGRANTES E REFUGIADOS . . . . .</b>	<b>10</b>
<b>2.1</b>	<b>Introdução . . . . .</b>	<b>10</b>
<b>2.2</b>	<b>Desenvolvimento do Projeto . . . . .</b>	<b>11</b>
<b>2.3</b>	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	<b>12</b>
<b>2.4</b>	<b>Conclusão . . . . .</b>	<b>14</b>
<b>3</b>	<b>AGES II — “LET ME TRIAL” . . . . .</b>	<b>16</b>
<b>3.1</b>	<b>Introdução . . . . .</b>	<b>16</b>
<b>3.2</b>	<b>Desenvolvimento do Projeto . . . . .</b>	<b>17</b>
<b>3.3</b>	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	<b>19</b>
<b>3.4</b>	<b>Conclusão . . . . .</b>	<b>23</b>
<b>4</b>	<b>AGES III — “SE DOCE FOSSE” . . . . .</b>	<b>24</b>
<b>4.1</b>	<b>Introdução . . . . .</b>	<b>24</b>
<b>4.2</b>	<b>Desenvolvimento do Projeto . . . . .</b>	<b>25</b>
<b>4.3</b>	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	<b>31</b>
<b>4.4</b>	<b>Conclusão . . . . .</b>	<b>37</b>
<b>5</b>	<b>AGES IV — “NOME DO PROJETO XXXX” . . . . .</b>	<b>39</b>
<b>5.1</b>	<b>Introdução . . . . .</b>	<b>39</b>
<b>5.2</b>	<b>Desenvolvimento do Projeto . . . . .</b>	<b>39</b>
<b>5.3</b>	<b>Atividades Desempenhadas Pelo Aluno no Projeto . . . . .</b>	<b>39</b>
<b>5.4</b>	<b>Conclusão . . . . .</b>	<b>40</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>41</b>
	<b>Referências . . . . .</b>	<b>42</b>

## 1 APRESENTAÇÃO DA TRAJETÓRIA DO ALUNO

Minha trajetória na área de Software inicia antes da faculdade. Aos meus 11 anos jogava Minecraft e era fascinado em como os plugins deixavam o jogo muito mais atrativo e diferente e comecei a buscar na internet maneiras de editar e criar um plugin próprio. Foi neste momento que me deparei com a linguagem Java, no começo foi estranho e diferente, mas consegui concluir alguns projetos e fiquei muito feliz por isso.

Não continuei na área e minhas pretensões eram outras, mas na hora de escolher meu curso, movido pela paixão e por aquele sentimento do passado, resolvi entrar no curso de Engenharia de Software no segundo semestre de 2022. Desde então procuro evoluir como frontend e sigo um entusiasta da área.

Em maio de 2023, participei da Hackatona de Engenharia de Software na PU-CRS, onde deveríamos resolver uma solução com o tema “Como utilizar a Inteligência Artificial para promover o bem social?”. Nossa equipe apresentou a solução da Comunicação Universal, projeto que consiste em um tradutor de libras em tempo real. Obtivemos o segundo lugar entre 14 equipes.

## 2 AGES I — IMIGRANTES E REFUGIADOS

### 2.1 Introdução

O objetivo principal do projeto é criar um aplicativo de celular cuja finalidade é facilitar o acesso à informação para imigrantes e refugiados. Os clientes Éfren Alvarado, Alex Guilherme e Henrique, vieram com os problemas que são a falta de informações para imigrantes e refugiados sobre seus direitos, assim como a ausência de educação. O aplicativo visa minimizar estes obstáculos reunindo as informações necessárias, de tal maneira que ONGs e instituições de ensino (universidades e escolas) possam ofertar programas com detalhamento. O projeto está sob a orientação do professor Dilnei Venturini e começou em 4 de agosto de 2023, com previsão de conclusão em 8 de dezembro de 2023.

**Figura 1 – Time**



Fonte: wiki do projeto

## 2.2 Desenvolvimento do Projeto

### 2.2.1 Repositório do Código Fonte do Projeto

Os repositórios foram divididos entre backend, frontend, e wiki do projeto.

Repositório para o [backend](#)

Repositório para o [frontend](#)

Repositório para [wiki](#)

### 2.2.2 Banco de Dados Utilizado

No banco de dados, estamos utilizando o SQL Server. Este foi implementando nas Sprints finais e, eu, como AGES I não obtive participação simbólica para a implementação.

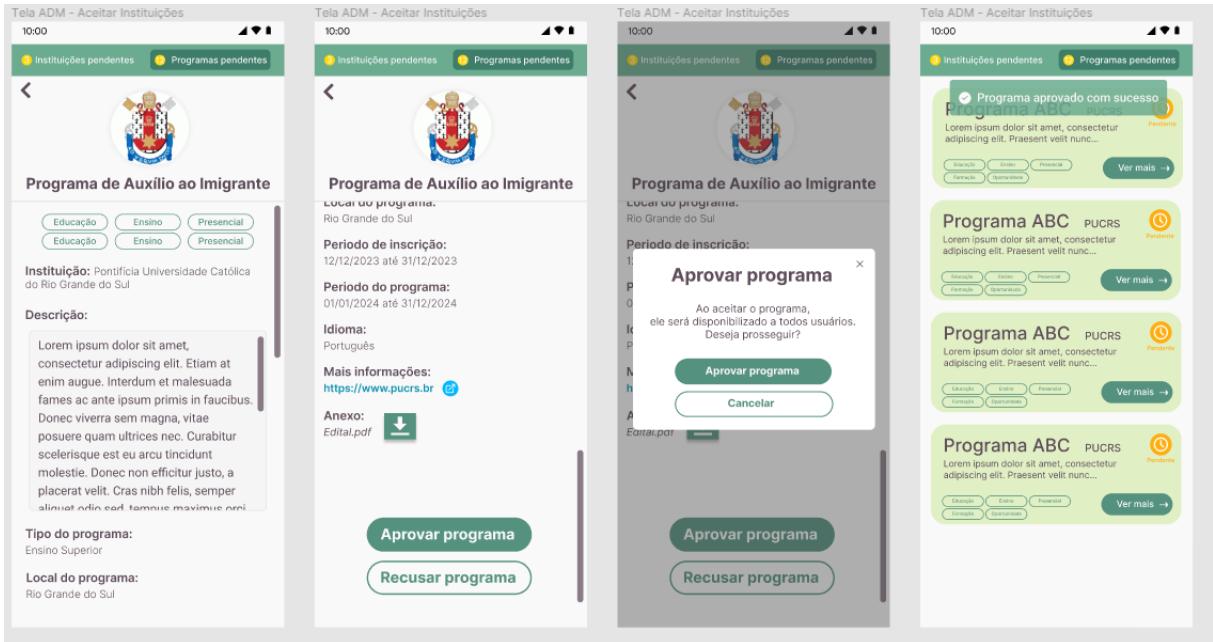
### 2.2.3 Arquitetura Utilizada

O projeto usa Java 11 e Spring Boot e possui três camadas: Controller (apresentação), Service (lógica de negócios) e Repository (acesso a dados). Os módulos incluem configurações, controle de rotas, objetos de transporte de dados, entidades de banco de dados, enums, mensagens, comunicação com o banco de dados, segurança, regras de negócios e utilitários. No frontend, há estruturas para recursos, componentes, telas, rotas, serviços, utilitários, hooks e acesso a APIs. Essas estruturas organizam o projeto de forma eficiente.

### 2.2.4 Protótipos das Telas Desenvolvidas

Utilizamos o Figma para o protótipo de telas, onde mostramos para o cliente a ideia inicial do visual do aplicativo e como implementaríamos as funções que eles gostariam que o programa tivesse.

**Figura 2 – Protótipo no Figma**



Fonte: figma do projeto

### 2.2.5 Tecnologias Utilizadas

Na primeira aula, iniciamos a discussão de quais tecnologias seriam utilizadas. Pelo backend, foi definido que utilizariamos Java, com o framework Spring Boot, visto que o grupo todo tinha familiaridade com a linguagem, já que esta é ensinada nas disciplinas da faculdade. Já no lado do frontend, decidimos utilizar TypeScript com o framework React Native, já que o aplicativo seria desenvolvido tanto para iOS, tanto para Android.

## 2.3 Atividades Desempenhadas Pelo Aluno no Projeto

### 2.3.1 Sprint 0

Na sprint 0 definimos as tecnologias que seriam utilizadas. Dado isso, dediquei boa parte dessa sprint para aprender *React Native* ([Facebook, 2023b](#)) e *TypeScript* ([Microsoft, 2023](#)), visto que seria meu primeiro contato. Os estudos foram produtivos e não obtive dificuldade. Nesta sprint também fizemos os mockups no Figma, onde encontrei dificuldades no começo, já que não conhecia a plataforma. Os problemas foram superados e consegui entregar as telas de administrador. Em virtude de ser minha primeira AGES, foi uma sprint de reconhecimento e pude entender mais como

funciona um ambiente de trabalho

### 2.3.2 Sprint 1

Na sprint 1, definimos as User Stories e nos dividimos em duas squads. Os alunos puderam escolher a área de preferência para atuar, seja no frontend, seja no backend. Nesta sprint escolhi o frontend, pois toda minha experiência com programação foi desenvolvida na faculdade, por consequência nunca havia interagido com esse lado. Depois de separar as squads, fomos orientados pelos AGES III e eles separaram as tarefas para todos. Fiquei responsável por estilizar a tela de acolhimento e criar a chamada API de textos. Na parte de criar a tela obtive sucesso e nenhum problema foi encontrado, mas em contrapartida, por não ter experiência com frontend, encontrei dificuldades para criar a chamada API de textos. Neste momento pude receber ajuda de colegas mais experientes e finalizamos a tarefa.

### 2.3.3 Sprint 2

Após uma boa entrega da sprint 1, mantivemos as squads e eu decidi continuar na parte do frontend. Dessa vez, fiquei responsável por criar um componente de anexo de arquivos. Comecei essa sprint estudando mais sobre React Native e suas funções. Dado isso, iniciei minha tarefa tentando utilizar a biblioteca Document Picker, que não foi implementado devido a colisões com o EXPO. Para encontrar a solução, encontrei o expo document picker, esta biblioteca foi aceita e a tarefa foi finalizada após alguns dias tentando entender o funcionamento dela.

### 2.3.4 Sprint 3

Dando continuidade no projeto, na Sprint 3 permaneci no frontend e fiquei encarregado de criar e estilizar as tabs de navegação, criar a tela de meus programas e o dropdown para tipos de programas. Após quase 2 meses de projeto, já estava confiante e tinha muito mais conhecimento nas linguagens que estávamos utilizando, por isso não obtive muitas dificuldades em executar as tarefas que foram passadas. Nas tabs de navegação, utilizei o Material Top Tabs Navigator, do React Navigation, e as estilizei de maneira que ficassem o mais fidedignas com as desenvolvidas no Figma. Criei a tela de meus programas, apenas com os mocks para que depois, quando

necessário, fizessem a integração. Nesta Sprint também fiquei responsável por um débito técnico na tela de de cadastro de programas, onde o usuário selecionava as informações do programa, faltava um dropdown para escolher se o tipo de programa era “Ensino Básico, Ensino Superior ou uma ONG”. Todas as tarefas foram resolvidas e, até o momento considerava essa a sprint onde me senti mais pronto e realizado. Também entendi muito mais os conceitos do Git e comecei a subir minha branch e commits mais cedo, assim dando margem para que os AGES III pudessem analisar meu código e me passar o feedback caso alguma coisa não esteja como o esperado.

### 2.3.5 Sprint 4

Então chegamos na reta final, na Sprint 4 fiquei com a tarefa de desenvolver a tela de aceitar e recusar programas, que serviria como base para a tela de aceitar e recusar instituições. Por mais que já estivesse habituado com a linguagem e a parte de estilizar uma tela, esta foi a que mais me deu trabalho. Foi minha primeira vez utilizando um ScrollView dentro de outro ScrollView, o que de início foi trabalhoso, visto que o scroll de dentro não funcionava, mas com algumas pesquisas consegui resolver o problema com um simples comando. Também criei um modal para que, na hora de aceitar um programa, o usuário veja se realmente quer completar esta interação. Para isso, utilizei o modal do React Native, o que foi bem simples e muito mais tranquilo do que eu esperava. Esta Sprint foi a qual mais aprendi, visto que mergulhei dentro de conceitos novos tanto de estilização tanto do próprio React Native e integrações. No final, fizemos nossa última retro (que remetia o projeto todo) e finalmente senti que tudo estava acabando e que esse meu tempo como AGES I foi proveitoso.

## 2.4 Conclusão

O projeto agradou muito aos stakeholders, nos parabenizando e ficando encantados com todo o trabalho feito. Eu também gostei, foi muito interessante entender como funcionava todo o desenvolvimento de um projeto e claro, ter uma parte que foi feita por mim é muito gratificante e empolgante para quem está apenas iniciando na área.

Tive um desenvolvimento muito bom como AGES I, cheguei com conhecimento sobre frontend praticamente nulo e consegui ter uma boa curva de aprendizado. Não sabia nada sobre integração de backend com frontend, como utilizar o Postman e com

um pouco de conhecimento sobre versionamento e agora posso dizer que entendo bastante sobre estes conceitos. Também me destaco pela parte não técnica, não pude participar de nenhuma “daily” de terça de maneira síncrona no discord, pois estava cursando a disciplina de Engenharia de Requisitos no mesmo horário, mas sempre atualizei meus progressos e impedimentos pelo canal de texto no dia. Foi muito agregador entender como funciona a parte mais “burocrática” de desenvolver um software, as dailys, retros e plannings me proporcionaram experiências que não teria vivido se não fosse a AGES. Por tanto, agora saio muito mais preparado para embarcar no mercado de trabalho e começar a aplicar meus conhecimentos adquiridos nesta etapa.

O projeto por si só foi muito bom, não tenho experiências passadas para comentar com mais detalhes, mas a organização foi excelente. Acredito que o trabalho do AGES IV foi muito bom, mas aqui destaco a eficiência de todos os AGES III que contribuíram muito e foram essenciais no andamento do projeto e no crescimento dos AGES I e II. Por contrapartida, acredito que o projeto pecou um pouco na hora de designar as tarefas, pois não ficava muito explícito o que deveria ser feito, resultando em dúvidas e mais trabalho para os AGES III. Também creio que o projeto seria melhor se fosse desenvolvido web, entretanto esta é uma questão definida pelo stakeholder

### 3 AGES II — LET ME TRIAL

#### 3.1 Introdução

O projeto Let Me Trial visa facilitar o encontro de pacientes com ensaios clínicos, realizados através de triagens em um site que será utilizado pelos médicos, onde verificarão se os critérios de estudos são compatíveis com as características e doenças dos pacientes. Os clientes Dr. Giovani Gadonski e Matheus Stortti, reclamaram que, esta pré-triagem seria fundamental, visto que muitos pacientes perdem tempo indo até o hospital, sendo que em uma consulta, com este processo mais automatizado, poderia já ser descartado ou pré-encaminhado, aumentando assim as chances de sucesso. O projeto está sob a orientação do professor Jorge Audy e começou dia 1º de março de 2024, com previsão de conclusão em 5 de julho de 2024.

**Figura 3 – Time**



Fonte: wiki do projeto

## 3.2 Desenvolvimento do Projeto

### 3.2.1 Repositório do Código Fonte do Projeto

Os repositórios foram divididos entre web-medico, web-site, web-administrador, api-medico, api-administrador, auth, infra e wiki.

Repositório para o [web-site](#)

Repositório para o [web-medico](#)

Repositório para o [web-administrador](#)

Repositório para o [api-medico](#)

Repositório para [auth](#)

Repositório para [infra](#)

Repositório para wiki: [wiki](#)

### 3.2.2 Banco de Dados Utilizado

No Banco de Dados, optamos por utilizar o PostgreSQL pela sua robustez. Entidades do projeto Paciente: Armazena as informações pertinentes ao projeto do paciente. Medico: Armazena as informações pertinentes ao projeto do médico. Area: Armazena as subáreas da qual pertencem os estudos. Resposta: Armazena a resposta de um paciente sobre um determinado critério. Estudo: Armazena informações de estudos e os critérios pertencentes através da entidade CritérioEstudo. Criterio: Armazena a pergunta sobre um critério. CritérioEstudo: Armazena os dados sobre um critério referente a um determinado estudo, como resposta esperada por ele, se é um critério opcional e relaciona o critério ao estudo.<sup>1</sup>

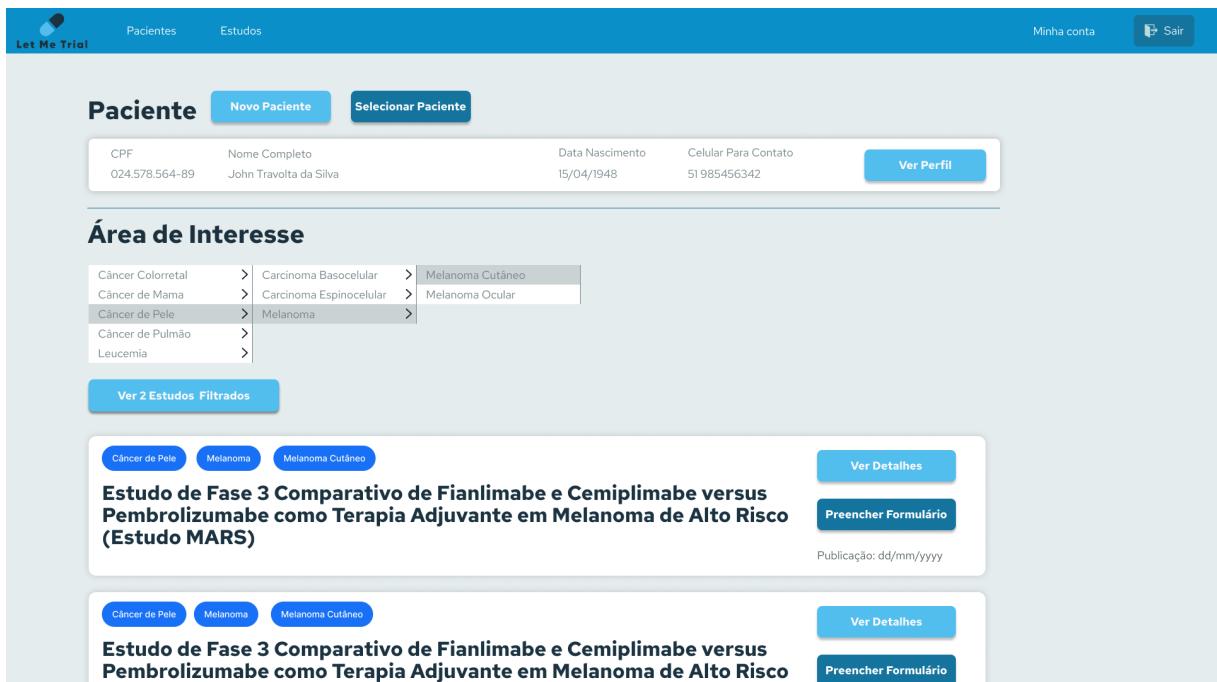
### 3.2.3 Arquitetura Utilizada

No backend, estamos utilizando a Arquitetura Hexagonal, que é um modelo de design de software que divide o sistema em três partes principais: Domínio: o coração do sistema, onde estão as regras de negócio e entidades importantes. Aplicação: coordena as ações do sistema e lida com interações entre o domínio e o mundo externo. Adaptadores: conecta o sistema ao mundo externo, permitindo interações sem que o domínio precise saber dos detalhes técnicos. No frontend, utilizamos o núcleo/core e seus submódulos, como core/domain para definir as entidades principais da aplicação.

ção, como usuários e médicos. O core/useCases é essencial para dividir a lógica de negócios em casos de uso 13 comuns e específicos para diferentes tipos de usuários, enquanto o adapters/presenter nos ajuda a adaptar os dados da lógica de negócios para a interface do usuário. Os frameworks/ui são vitais para organizar os componentes e páginas da interface do usuário, enquanto os frameworks/services facilitam a comunicação com APIs externas. Por fim, os utils são úteis para funções utilitárias, como autenticação e verificação do tipo de usuário

### 3.2.4 Protótipos das Telas Desenvolvidas

**Figura 4 – Protótipo no Figma**



Fonte: figma do projeto

### 3.2.5 Tecnologias Utilizadas

No backend, estamos utilizando Java 21 ([Gosling, 2018](#)) com o framework Spring Boot ([Pivotal Software, 2023](#)), além do JUnit para testes, pois, a maioria do grupo era familiarizado com a linguagem, o que facilitaria e aumentaria a velocidade e a qualidade do projeto. No frontend optamos por utilizar a biblioteca React, com TypeScript e o framework Next. Para o frontend, utilizei o Postman ([Postman, 2023](#)) para testar as APIs antes de alguma integração.

### 3.3 Atividades Desempenhadas Pelo Aluno no Projeto

#### 3.3.1 Sprint 0

Na sprint 0, depois da visita dos stakeholders, definimos metas e como faríamos para manter o projeto. Ficou acordado que teríamos dailys assíncronas no Discord todos os dias, onde deveríamos reportar o que fizemos, o que faremos e se havia ou não algum impedimento. Depois disso, foi definida qual linguagens e frameworks utilizaríamos no projeto, ficando acordado pelo Java 21 com Spring Boot e React com TypeScript e Next.

Após toda a cerimônia, comecei a desenvolver uma tela de login, que serviu como modelo para outras, até chegarmos no produto final. Sempre que pude, estive dando ideias e produzindo ou modificando algum componente, para que ficasse de acordo com o que os clientes esperavam. Nosso primeiro encontro da sprint foi produtivo e conseguimos alinhar o que faltava para entregarmos ao cliente todo o fluxo e a interface do usuário.

Depois de semamans produzindo, nosso cliente chegou e gostou muito do que foi proposto no quesito fluxo e beleza das telas. Entretanto, tínhamos uma dúvida pertinente, que era como seria feito o filtro de estudos e, após uma aula toda, chegamos em um consenso.

Após isso, fizemos a retro, que, por falta de comando, acabou tomando muito tempo com discussões inúteis que não precisariam de tanta minutagem para serem debatidas, o que nos custou a aula toda. Sendo assim não foi possível realizar a planning na sala presencialmente. Uma pena, já que eu considero a planning um dos momentos mais importantes para uma sprint.

#### 3.3.2 Sprint 1

Começamos a sprint 1, sem planning e sem repositórios do frontend, visto que nossos AGES III atrasaram a entrega, assim sobrecregando nossos AGES IV que cumpriram além de suas funções.

Dividimos-nos em squads, onde a minha possuía dois responsáveis pelo backend e dois responsáveis pelo frontend. Fiquei no frontend com a Carolina e demos inícios as tarefas.

Depois de alguns dias, começaram a surgir as tarefas e logo me prontifiquei para realizar o desenvolvimento do componente de botão. De certa forma foi fácil, apesar de ser uma primeira experiência minha com o React na sua versão web, consegui já deixar pronto para os outros colegas poderem utilizar em seus componentes.

Minha squad acabou ficando com uma User Story e demos início ao desenvolvimento do componente de card de estudos. Este foi um pouco mais complicado, visto que lidava com outros componentes e o processo de estilização foi mais complexo. Aproveitei para criar o componente de Tag, que serviria para nosso card e demos início ao desenvolvimento. Em poucos dias estava pronto, faltava apenas a integração.

A integração foi uma fraqueza do grupo todo, visto que o frontend era pouco conhecido pelos participantes, inclusive pelos AGES IV, então tivemos uma aula de discussões e tentando entender como funcionaria todo o processo de integração sem colocar a mão no código. A missão de casa e pra última semana de sprint era entender como funcionava a integração.

Criei um componente de filtro de estudos, para utilizar com nosso card. Mas ainda não sabia como integraria tudo, até que criei um componente wrapper que “embrulhava” os dois, fazendo a chamada API no próprio wrapper, que era utilizado pelo filtro, para depois ser disponibilizada uma lista de estudos disponíveis, para então só depois colocar na página, assim evitando que a página inteira fosse renderizada novamente a cada interação.

### 3.3.3 Sprint 2

Após apresentarmos para o cliente, ficou decidido que o filtro de estudos se transformaria em um *dropdown*, além de uma reformulação completa na parte de integração, já que o *frontend* adotara uma nova abordagem para realizar chamadas API. Inicialmente, imaginei que seria uma tarefa simples, pois já estava habituado com o filtro anterior. No entanto, logo percebi que a implementação seria mais desafiadora do que previsto. Decidimos criar um *dropdown* com dois níveis, acompanhado por um botão para limpar as seleções e outro para filtrar e exibir os estudos.

Optamos por utilizar a biblioteca Material UI ([Material-UI Team, 2023](#)), e eu decidi empregar o *React Select* para implementar o *dropdown*. O início foi complicado, pois estava lidando com uma nova ferramenta. No entanto, a biblioteca se mostrou bas-

tante intuitiva, o que facilitou a replicação do esperado. Conseguí concluir a tarefa mais rápido do que o planejado. Contudo, enfrentamos desafios devido às enchentes que atingiram o estado do Rio Grande do Sul, resultando no cancelamento das aulas por duas semanas, seguidas por um retorno online. Por conta disso, fiquei um bom tempo afastado do projeto até a retomada da sprint e, em seguida, a entrega para o cliente. A entrega não saiu exatamente como esperávamos, mas conseguimos entregar o filtro quase completo, faltando apenas alguns ajustes básicos que não comprometiam o funcionamento da funcionalidade. Conforme previsto devido ao caos gerado pelas enchentes, alguns débitos técnicos foram deixados para serem resolvidos na próxima sprint. Apesar das dificuldades, a experiência acumulada com as bibliotecas utilizadas e a gestão de crises externas foi valiosa para a continuidade do projeto.

### 3.3.4 Sprint 3

Durante a Sprint 3, minhas tarefas incluíam ajustes no funcionamento do filtro de estudos no botão de limpar, visto que ele não funcionava corretamente ao selecionar outro estudo. Também fiquei responsável para corrigir um erro na estilização da página de estudos que não tinha o comportamento esperado quando os cards de estudos apareciam na tela. Além disso, fiquei com uma tarefa extra, que só seria desenvolvida caso sobrasse tempo, que seria o início do desenvolvimento da tela de login.

Consegui implementar com sucesso o botão de limpar do filtro, que agora funciona conforme o esperado, e finalmente deixamos uma das partes mais importantes da aplicação completamente funcional e pronta para entrar na *branch* principal. Para a correção de estilos foi bem simples, visto que era um problema de espaçamento e como eu já tinha experiência com HTML ([World Wide Web Consortium \(W3C\), 2023b](#)) e CSS ([World Wide Web Consortium \(W3C\), 2023a](#)), foi a parte mais tranquila dessa sprint. Como ainda me sobrou uma semana para continuar desenvolvendo, decidi começar a tela de login, e combinei com meus colegas que apenas faria a parte estática, ou seja, qualquer tipo de integração com a api-auth seria completamente deixada de lado e iniciada apenas na próxima sprint. Achei o modelo no figma e comecei os trabalhos, o que não foi novamente muito difícil, visto que já estava habituado com as tecnologias. Aprendi como fazer uma função de mostrar a senha, para que fosse possível caso o usuário desejasse visualizar sua senha antes de enviá-la, além de

entender mais como funcionam os formulários. Esta sprint foi uma das melhores que tive no projeto, já que consegui adiantar todas minhas tarefas e ficar disponível caso alguém precisasse da minha ajuda, além de me sentir cada vez mais familiarizado com as tecnologias utilizadas. No fim, ficou pendente a parte de integração do login, que seria discutida no futuro com os AGES IV e AGES III.

### 3.3.5 Sprint 4

Na apresentação passada, o cliente nos informou que não poderíamos seguir utilizando dados do paciente como nome, CPF e telefone. Com isso, nos reorganizamos e passamos a utilizar apelido, CEP e sexo. Em virtude do que foi mencionado, fiquei responsável pelo *remap* dos dados do paciente no *frontend*, tendo que mudar diversos arquivos. Comecei trocando nome por apelido, já que seria basicamente uma questão mais visual, já que ambos são simples *strings*. Já para mudar CEP para CPF, foi um pouco mais trabalhoso, por contarmos com validadores e formatadores, o que no final foi excelente, fazendo com que eu trabalhasse com coisas novas e serviu de aprendizado para o futuro. Ao adicionar o sexo, imediatamente criei a opção de *dropdown* na hora de cadastrar o pacientes, e estilizei novamente todo o componente, para que ficasse adequado. Partindo para a parte mais difícil, faltava integrar a tela de login com o sistema de autenticação. Foram horas de estudo, junto com o final de semestre, mas consegui fazer a autenticação funcionar através do *Context API* do *React*. A partir disso, criei uma Rota Privada, assim só sendo possível acessar qualquer página da aplicação se o *token* fosse válido e estivesse ativo, caso contrário o usuário seria redirecionado para a tela de login novamente. Como não possuímos um meio de cadastro, toda essa parte foi feita por fora e de forma manual. No fim, esta sprint foi um pouco conturbada, pois ainda passávamos por consequências geradas pelas encheres, já que o calendário acadêmico ficou mais curto e as provas e trabalhos acabaram coincidindo no final do semestre. Porém, mesmo com tudo isso, o trabalho feito foi excelente e toda a equipe conseguiu contribuir da maneira que dava. Fiquei muito feliz por aprender mais sobre *keycloak* e maneiras de realizar uma autenticação, além de trabalhar pela primeira vez com docker ([Docker, Inc., 2023](#)).

### 3.4 Conclusão

O projeto Let Me Trial foi um marco totalmente diferente para mim no quesito de desenvolver software, já que, por falta de alunos com maior conhecimento em *frontend*, tive que fazer diversas coisas de maneira solitária, mas que contribuíram de maneira significativa para meu crescimento dentro da área de desenvolvimento de software. Por consequência, acredito que consegui contribuir de uma forma muito boa devido aos limites tanto dos meus conhecimentos, quanto dos colegas. Tive um bom relacionamento e uma boa comunicação com todos do projeto, assim como fui proativo e sempre estive proposto a ajudar todos com dúvidas e tentar resolver problemas que foram aparecendo, além de manter a pontualidade em todas as reuniões e sempre com a presença em dia.

Definitivamente este foi o projeto, até então, em que eu mais aprendi em relação a tudo, tanto quanto a parte do desenvolvimento de código, quanto a importância dos processos de software. Visto que o grupo todo conseguiu entender como tudo funcionava na metodologia ágil, além de uma boa contribuição de todos sempre que possível, o time foi harmônico e cada um teve seu momento de crescer e disparar no projeto e no final foi bonito e inspirador ver o tanto que evoluíram. Aprendi muito mais sobre *frontend* e suas técnicas, que por mais que o nível inicial de conhecimento de todos era mais baixo, foi o maior ponto positivo do projeto, já que todos estiveram focados em aprender e repassar o conhecimento que foi adquirido em pouco tempo. Conseguir, sempre que possível, diagnosticar minha situação nais dailys e participei ativamente das discussões, principalmente voltado para a área da interface de usuário. O projeto poderia ter tido um início mais calmo e mais organizado, fomos desfavorecidos no começo pelas retros exageradas e muito calorosas, que nos fez perder uma planning que seria muito importante, já que seria a primeira dessas na parte de desenvolvimento de código. Faltou uma participação maior dos AGES III, visto que nosso projeto, por exemplo, ainda está sendo apresentado em *localhost*, além de deixar uma sobrecarga muito grande nas costas dos nossos AGES IV.

## 4 AGES III — “SE DOCE FOSSE”

### 4.1 Introdução

O projeto Se Doce Fosse visa criar um site institucional para divulgação da marca e produtos, além de um sistema de controle de estoque que permita gerenciar de forma eficiente os ingredientes utilizados na produção de doces — especialmente cookies e brownies. A empresa, fruto do sonho de uma empreendedora recém-formada em Engenharia de Alimentos, busca ampliar sua presença digital e otimizar seu processo de produção, evitando perdas por compras excessivas ou falta de insumos para atender à demanda. Os clientes Carolina Padilha e Alessandro Machado Padilha destacaram que a ferramenta será essencial para apoiar o crescimento da marca, aproximar o público-alvo por meio da internet e das redes sociais, além de melhorar o fluxo de trabalho interno. O projeto está sob a orientação do professor Dilnei Ventirini e terá início em 04/08/2025, com previsão de conclusão em 28/11/2025.

**Figura 5 – Time - Se Doce Fosse**



Fonte: wiki do projeto

## 4.2 Desenvolvimento do Projeto

### 4.2.1 Repositório do Código Fonte do Projeto

Os repositórios foram definidos em 3 conteúdos sendo eles:

Repositório para o [frontend](#), onde colocamos toda a parte web do projeto.

Repositório para o [backend](#), onde colocamos a lógica de negócios com as APIs e banco de dados.

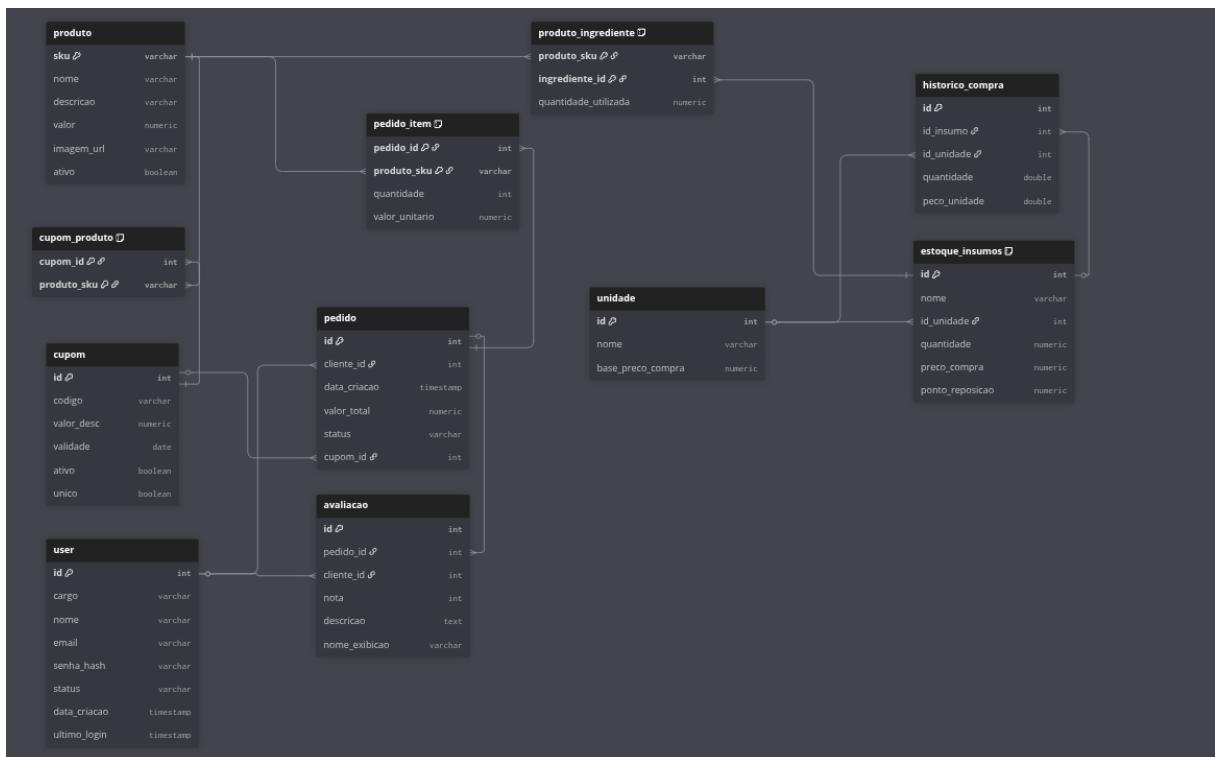
Repositório para a [wiki](#), onde documentamos tudo que envolvia o projeto de uma forma mais didática.

### 4.2.2 Banco de Dados Utilizado

No desenvolvimento do sistema, optou-se por uma abordagem dupla de bancos de dados, utilizando o PostgreSQL e o MongoDB, cada um desempenhando papéis com-

plementares. O PostgreSQL ([PostgreSQL Global Development Group, 2024](#)), banco de dados relacional amplamente reconhecido por sua robustez e confiabilidade, foi empregado para armazenar informações que exigem alta integridade e consistência, como dados de usuários, controle de estoque e histórico de pedidos. Já o MongoDB ([MongoDB, Inc., 2024](#)), banco de dados não relacional orientado a documentos, foi escolhido para gerenciar elementos mais dinâmicos, como o carrinho de compras e listas temporárias, permitindo maior flexibilidade na estruturação e manipulação dessas informações. Essa combinação possibilita um equilíbrio entre desempenho, escalabilidade e integridade, explorando as vantagens específicas de cada tecnologia.

**Figura 6 – Banco de Dados**



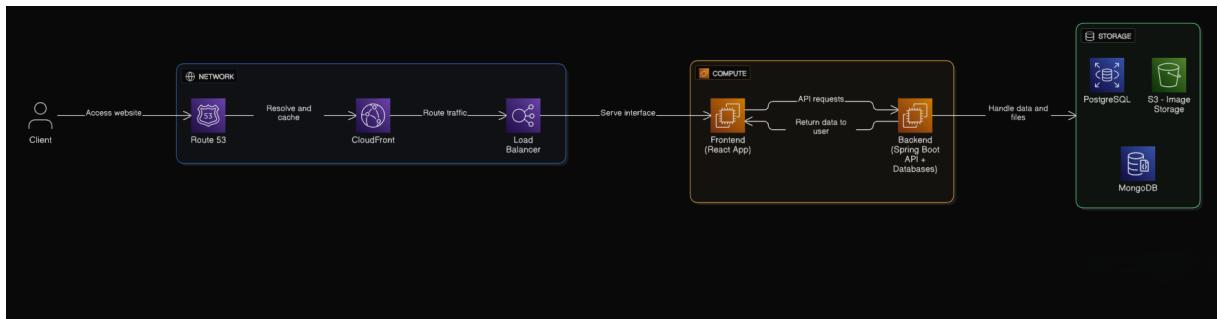
Fonte: Wiki do Projeto

#### 4.2.3 Arquitetura Utilizada

No projeto, adotou-se uma arquitetura em camadas (*Layered Architecture*) para o desenvolvimento do backend, implementado em Java ([Gosling, 2018](#)) com o framework Spring Boot ([Pivotal Software, 2023](#)). Essa organização separa claramente as responsabilidades entre as camadas de apresentação, serviços, repositórios e entidades, promovendo maior manutenibilidade, testabilidade e clareza no fluxo de dados. No frontend, foi utilizada a biblioteca React ([Facebook, 2023a](#)), estruturada de forma

modular, permitindo o reaproveitamento de componentes e a escalabilidade da aplicação. Para o estilo, optou-se pelo uso de SCSS Modules, que garantem isolamento de escopo e facilitam a manutenção visual da interface, contribuindo para uma organização consistente entre lógica, apresentação e estilos em toda a aplicação. Na parte de Infraestrutura, utilizamos a Amazon Web Services (AWS) ([Amazon.com, Inc., 2025](#)) para hospedar a aplicação, criando uma EC2 para o frontend, uma para o backend junto dos bancos de dados. O frontend utiliza NGINX para servir arquivos estáticos, atuar como reverse proxy para o backend, realizar terminação SSL/TLS, fazer cache e compressão, e gerenciar balanceamento de carga quando necessário, garantindo desempenho e segurança. Para hospedar imagens, utilizamos o Amazon S3, que oferece armazenamento escalável e seguro para os arquivos estáticos do projeto, como imagens de produtos. Também utilizamos o GitHub Actions para realizar o CI/CD do projeto, automatizando os processos de build e testes.

**Figura 7 – Diagrama de Arquitetura**



Fonte: Wiki do Projeto

#### 4.2.4 Protótipos das Telas Desenvolvidas

Na Sprint 0, protótipos de telas foram desenvolvidos e entregues aos clientes. Usamos elas como base para construir o frontend e validar com o clientes, além de auxiliar no backend, para entender quais endpoints seriam necessários. O protótipo foi desenvolvido no Figma, uma ferramenta colaborativa de design de interfaces que permite a criação de protótipos interativos e compartilháveis. A seguir, imagens do protótipo desenvolvido:

**Figura 8 – Protótipo no Figma - Home**

**Se Doce Fosse**

Produtos    Encomendas    Sobre Nós    Entrar    Carrinho

Doces artesanais feitos com amor  
Cada doce é um reflexo de cuidado, paixão e criatividade!  
Seg-Sex: 8h-18h | Sáb: 8h-16h

## Doces

**Cookie Chocolate Branco**  
Lorem ipsum dolor sit amet. Eum amet culpa aut commodi accusamus vel culpa suscipit!  
**R\$ 15,00** [+ Adicionar](#)

**Cookie Oreo com Nutella**  
Lorem ipsum dolor sit amet. Eum amet culpa aut commodi accusamus vel culpa suscipit!  
**R\$ 15,00** [+ Adicionar](#)

**Cookie Dark Chocolate**  
Lorem ipsum dolor sit amet. Eum amet culpa aut commodi accusamus vel culpa suscipit!  
**R\$ 15,00** [+ Adicionar](#)

**Cookie Limão Siciliano com Mirtilio**  
Lorem ipsum dolor sit amet. Eum amet culpa aut commodi accusamus vel culpa suscipit!  
**R\$ 15,00** [+ Adicionar](#)

## Comentários

Ana  
Lorem ipsum dolor sit amet.  
Eum amet culpa aut

Ana  
Lorem ipsum dolor sit amet.  
Eum amet culpa aut

Ana  
Lorem ipsum dolor sit amet.  
Eum amet culpa aut

Horário de funcionamento  
Seg-Sex | 8h-18h  
Sáb | 8h-16h  
Dom | Fechado

Contato  
[Icones de Instagram, WhatsApp e Email]

Queremos saber sua opinião! Avalie nossa loja!

**Deixe um comentário**

Comentário\*  
Nome\*  Telefone\*

**Enviar**

Fonte: Figma do projeto

**Figura 9 – Protótipo no Figma - Página de Produtos**

The screenshot shows a web browser displaying a product page for a bakery named "Se Doce Fosse". The header includes the logo, navigation links for "Produtos", "Encomendas", and "Sobre Nós", and user authentication options. The main content area is titled "Produtos" and displays a grid of 12 different cookie varieties, each with a small image, the name, the price (all listed as R\$ 15,00), and a "Adicionar" (Add) button.

- Cookie Oreo com Nutella**: Stacks of cookies with Oreo pieces and Nutella filling.
- Cookie Chocolate Branco**: A single cookie with white chocolate chunks.
- Cookie Dark Chocolate**: Stacks of dark chocolate cookies.
- Cookie Dark Maracujá**: A stack of cookies with maracujá (passion fruit) filling.
- Cookie Duetto**: A stack of two distinct types of cookies.
- Cookie de Framboesa**: A stack of cookies with raspberry filling.
- Cookie Kinder Bueno**: Stacks of cookies with Kinder Bueno candy pieces.
- Cookie Lotus**: Stacks of cookies with Lotus Biscoff spread.
- Cookie Nozes com Doce de Leite**: Stacks of cookies with nut and caramel filling.
- Cookie de Pistache**: Stacks of pistachio-flavored cookies.
- Cookie Red Velvet**: Stacks of red velvet cookies.
- Cookie Romeo e Julieta**: Stacks of cookies with different fillings, possibly chocolate and strawberry.

The footer contains contact information, social media links, and a feedback form titled "Deixe um comentário" (Leave a comment) with fields for message, name, and phone number, along with an "Enviar" (Send) button.

Fonte: Figma do Projeto

**Figura 10 – Protótipo no Figma - Tabela de pedidos (Admin)**

Fonte: Figma do Projeto

**Figura 11 – Protótipo no Figma - Página Admin**

Produto	Categoria	Preço	Estoque	Status	Ações
Cookie Biscoito	Cookies Tradicionais	R\$ 9,90	10	Ativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	
Cookie Dark Mara...	Cookies Recheados	R\$ 17,90	25	Inativo	

Fonte: Figma do projeto

#### 4.2.5 Tecnologias Utilizadas

Escolhemos uma stack que equilibra aprendizado e robustez: Java 21 ([Gosling, 2018](#)) com Spring Boot ([Pivotal Software, 2023](#)) no backend, PostgreSQL ([PostgreSQL](#)

Global Development Group, 2024) para dados relacionais críticos e MongoDB (MongoDB, Inc., 2024) para dados mais flexíveis, e React com TypeScript (Facebook, 2023a; Microsoft, 2023) no frontend.

Java e Spring Boot foram adotados pela familiaridade dos alunos e pela produtividade na criação de APIs testáveis. PostgreSQL garante integridade e consultas complexas, enquanto MongoDB facilita dados com esquema variável (ex.: carrinho). React + TypeScript oferece componentes reutilizáveis e verificação de tipos em desenvolvimento. Complementamos com SCSS Modules para estilos, NGINX para servir o frontend e GitHub Actions para CI/CD. A infraestrutura utiliza AWS (Amazon.com, Inc., 2025) (S3 e EC2) quando necessário. Essa combinação também expõe os alunos a paradigmas diferentes (relacional e documental) sem aumentar muito a complexidade operacional.

Essas escolhas priorizam aprendizado prático e entrega eficiente, evitando complexidade desnecessária ao mesmo tempo em que expõem os alunos a tecnologias e padrões usados em produção. Além disso, priorizamos práticas como testes automatizados e integração contínua para reforçar formação profissional e preparar os alunos para contextos reais de desenvolvimento.

### 4.3 Atividades Desempenhadas Pelo Aluno no Projeto

#### 4.3.1 Sprint 0

Durante a Sprint 0, meu trabalho esteve totalmente voltado para preparar o ambiente necessário ao desenvolvimento do projeto, garantindo que toda a equipe tivesse as condições ideais para iniciar as próximas etapas. A primeira ação foi a inicialização do repositório principal, criando uma base organizada para armazenar o código-fonte do frontend e demais materiais relacionados ao sistema. Além disso, adicionei todos os membros da equipe aos repositórios, concedendo as permissões adequadas para que cada integrante pudesse colaborar de forma integrada desde o início, além de estabelecer regras para proteger as branches.

Também realizei a dockerização do ambiente do frontend, criando uma estrutura que permitisse aos integrantes responsáveis pelo backend testar a interface de forma simples e rápida, sem precisar configurar dependências locais. Configurei os arquivos e parâmetros do Docker com esse objetivo, garantindo que o front pudesse ser exe-

cutado em qualquer máquina de maneira padronizada, apenas subindo o container. Essa etapa foi essencial para reduzir barreiras entre as áreas, facilitar a integração e evitar problemas relacionados a diferenças de versões ou de configuração.

Paralelamente às tarefas mais técnicas, estudei as tecnologias ligadas à infraestrutura que seriam utilizadas ao longo do projeto. Também dediquei tempo para estruturar e documentar os processos que a equipe deveria seguir. Criei uma seção específica na Wiki chamada “Processos”, onde registrei todas as definições acordadas, incluindo fluxos de trabalho, regras para criação de branches, padrões de commits e boas práticas para organização do código. O objetivo foi garantir que ninguém tivesse dúvidas sobre como proceder, permitindo que todos soubessem exatamente quais passos adotar no dia a dia.

Ao longo dessa sprint, mantive um olhar atento para assegurar que cada detalhe estivesse finalizado antes do encerramento do ciclo. Verifiquei o funcionamento do espelhamento entre as plataformas, testei os containers Docker para confirmar que estavam executando corretamente e revisei as permissões dos membros nos repositórios. Além disso, organizei os arquivos iniciais do projeto, deixando tudo estruturado para que a equipe pudesse começar a desenvolver as funcionalidades planejadas sem enfrentar bloqueios relacionados à configuração do ambiente.

Concluindo, a Sprint 0 foi um período dedicado a preparar a base técnica e organizacional do projeto. As atividades que realizei criaram uma base sólida para o trabalho das próximas sprints. Essa preparação garantiu que o time tivesse clareza sobre as práticas adotadas e que todos dispusessem de um ambiente uniforme, seguro e pronto para receber as implementações do sistema.

#### 4.3.2 Sprint 1

Na Sprint 1, foquei em organizar o trabalho do frontend e em dar suporte aos AGES I e II, garantindo que todos pudessem avançar sem bloqueios. Logo no início, separei as tarefas de acordo com o nível de cada integrante, definindo o que seria mais adequado para os AGES I e para os AGES II. Essa divisão ajudou bastante a dar clareza sobre as responsabilidades e permitiu que o time tivesse uma visão mais concreta do que precisava ser entregue.

Além da organização das tasks, trabalhei diretamente na implementação de alguns

componentes essenciais para o projeto, como o *Input* e o *Textarea*. Esses elementos serviram como base para o restante da interface e já deixaram o repositório com exemplos práticos para os outros integrantes, e também já deixava pronto esses componentes que seriam reutilizados em tarefas futuras. Também revisei, comentei e aprovei diversos Pull Requests, ajudando a manter a qualidade e a consistência do código que estava sendo incorporado ao projeto.

Outro ponto marcante dessa sprint foi o tempo dedicado a desimpedir colegas que estavam com dificuldades para entender o funcionamento do frontend. Muitos dos integrantes, especialmente os do AGES I, estavam tendo contato com essas tecnologias pela primeira vez e se sentiram um pouco perdidos. Para ajudar, organizei “aulas” no Discord, ferramenta que utilizamos como base para comunicação, onde expliquei o passo a passo de como os componentes funcionam, apresentei a estrutura do projeto e mostrei boas práticas para a escrita de código. Em vários momentos também fiz *pair programming*, acompanhando o desenvolvimento em tempo real e orientando sobre a melhor forma de resolver problemas específicos, além de ajudar a entender funções específicas do TypeScript.

Percebi que esse apoio constante fez diferença no engajamento do grupo. À medida que as dúvidas eram resolvidas, os colegas conseguiam evoluir mais rápido e participar ativamente das entregas. Ainda assim, notei que apenas explicações ao vivo não eram suficientes para todos, por isso, comecei a pensar na ideia de criar materiais em vídeo que complementassem a documentação já disponível, tornando o aprendizado mais acessível e facilitando o trabalho de quem prefere revisar os conteúdos de outra maneira.

Enquanto essas atividades aconteciam, também iniciei estudos sobre a infraestrutura que pretendemos usar na AWS. Esse planejamento começou de forma paralela, para que as decisões sobre deploy e hospedagem não atrasem o projeto nas próximas sprints.

De forma geral, a Sprint 1 foi muito voltada a facilitar o caminho dos colegas e a consolidar a base do frontend. Conseguí equilibrar a produção de código com o suporte técnico, garantindo que todos tivessem condições de avançar. O trabalho de orientação, revisão e organização foi essencial para criar um ambiente colaborativo e produtivo, preparando o time para as próximas etapas do desenvolvimento.

#### 4.3.3 Sprint 2

Na Sprint 2, foquei em desimpedir os AGES I e os AGES II e em começar a implementar soluções que facilitassem as próximas etapas do projeto. Logo no começo identifiquei os principais pontos de bloqueio no frontend e passei a orientar os colegas de forma assíncrona e direta, respondendo dúvidas no Discord e revisando trechos de código quando necessário. Paralelamente, comecei a levantar opções de deploy e participei das discussões iniciais sobre a arquitetura na AWS para que as decisões de infraestrutura não atrasassem as entregas.

Uma das frentes centrais foi a implementação da autenticação no frontend. Trabalhei na proteção de rotas e criei mecanismos de segurança para a área de ADMIN. Além disso, padronizei as chamadas para a API, estabelecendo convenções que tornaram as integrações mais previsíveis e fáceis de entender para os integrantes menos experientes. Essas padronizações foram documentadas de forma simples para servir como referência e reduziram dúvidas recorrentes durante a sprint.

Durante a sprint ajudei efetivamente os AGES I e AGES II em suas tasks no frontend. Desta vez dediquei mais tempo à assistência assíncrona, mas ainda assim consegui transferir conhecimento e orientar boas práticas. Também contribuí na definição de diretrizes iniciais para o deploy na AWS, alinhando expectativas com o time e apontando os caminhos a seguir.

Enfrentei problemas ao proteger algumas rotas: ao aplicar a proteção, as rotas chegavam a aparecer vazias e o conteúdo não era renderizado corretamente. Para contornar isso, resolvi mover a lógica de proteção para o layout do admin, evitando sobrecarregar as rotas e permitindo que o conteúdo fosse carregado e autorizado no nível do layout. Essa solução se mostrou simples e eficaz para o caso que estávamos enfrentando.

Como lição, ficou claro que nem tudo precisa sair perfeito desde o início. Aprendi a resolver problemas em condições não ideais e que uma comunicação rápida e objetiva muitas vezes resolve bloqueios que pareceriam complexos. De forma geral, a Sprint 2 foi importante para consolidar autenticação, padronizar chamadas de API e desbloquear o time para que as entregas pudessesem evoluir com mais estabilidade.

#### 4.3.4 Sprint 3

Na Sprint 3, novamente concentrei meus esforços em desimpedir os AGES I e II, atuando tanto na parte de API quanto no código do frontend. Não foi necessário gravar o vídeo previsto para as integrações, pois os colegas compreenderam rapidamente os conceitos após as explicações, exemplos práticos e sessões de acompanhamento que realizei. Assumi tarefas de integração entre frontend e backend: padronizei as chamadas, ajustei formatos de dados quando necessário e validei fluxos básicos com ferramentas de teste (ex.: requisições manuais e mocks), o que permitiu que várias funcionalidades passassem a se comunicar corretamente com a API. Embora o deploy final não tenha sido concluído nesta sprint, avançamos bastante no planejamento da entrega e em testes locais colaborativos com os demais dos AGES III.

Durante o processo enfrentei problemas de integração relevantes: o backend ainda apresentava instabilidades e, em alguns momentos, alterações em campos da tabela de produtos não eram persistidas, impedindo a finalização de funcionalidades no frontend. Para identificar a causa, conversei com a equipe de backend, inspecionei logs, testei endpoints isoladamente e adotei verificações intermediárias no frontend que evidenciaram quais valores não chegavam corretamente ao banco. Também houve dependência de decisões sobre reorganização dos bancos de dados, o que exigiu ajustes na arquitetura a serem feitos antes do deploy definitivo.

Além disso, realizei pair programming com AGES I e revisei pull requests para manter a qualidade do código. Essas ações reduziram o impacto dos problemas e mantiveram o ritmo das entregas, além de gerar material e exemplos que facilitaram o aprendizado dos colegas.

As lições aprendidas nesta sprint reforçaram que frontend e backend devem caminhar mais alinhados desde o início, com contratos de API claros e testes de integração contínuos. A prática de escrever e executar testes, mesmo simples e manuais nas fases iniciais, ajudaria a detectar inconsistências e evitar retrabalhos que serão feitos. Em suma, a Sprint 3 foi produtiva para desbloquear o time, consolidar integrações e avançar no planejamento do deploy, as pendências remanescentes servem agora como pontos de ação claros para as próximas sprints.

#### 4.3.5 Sprint 4

Na Sprint 4 atuei diretamente auxiliando os AGES I nas integrações do frontend. Todos finalizaram as integrações com sucesso e passaram a rodar por conta própria.

Também trabalhei nas integrações da tabela da tela de estoque e concluí a funcionalidade de produto, incluindo a gestão de ingredientes por meio do modal de criar e editar produtos. Com isso foi possível aplicar descontos diretamente na tabela de estoque de forma consistente. Corrigi problemas de CSS relacionados a modais e revisei pull requests, aceitando mudanças corretas e solicitando ajustes quando necessário. Durante a sprint executei testes manuais de ponta a ponta e preparei exemplos de dados que facilitaram o trabalho dos colegas menos experientes.

Posteriormente finalizei o deploy na AWS, configurei o ambiente para rodar com NGINX e apontei corretamente para o backend. Isso permitiu testar a aplicação em um ambiente mais próximo ao de produção e identificar pontos finos que não eram visíveis em testes locais. Na integração inicial houve um bloqueio de CORS. As requisições foram impedidas até que o endereço do frontend, o IP elástico, fosse incluído nas liberações de CORS no backend. Identificamos a causa, ajustamos as regras de CORS e atualizamos variáveis de ambiente. Após o redeploy do backend os testes em produção passaram a funcionar normalmente e, em seguida, realizamos a demonstração ao cliente. A apresentação foi bem sucedida e o deploy foi validado pelo cliente.

Como lição, ficou evidente que decisões de infraestrutura na nuvem, como provisãoamento de IP e regras de CORS, devem ser priorizadas mais cedo no cronograma, pois resolver esses pontos tarde gera retrabalho e atrasos para a entrega final do projeto. Sendo assim, um planejamento antecipado dessas etapas em sprints passadas, ajudaria a mitigar riscos e garantir uma entrega mais fluida sem contratemplos técnicos de última hora.

Ao mesmo tempo a prioridade em apoiar o AGES I foi justificável porque desde que entrei no projeto, meu maior objetivo era passar conhecimento para quem não tinha tanta experiência, pois sei que isso faz uma diferença enorme no resultado final da AGES, já que a disciplina é feita para os alunos aprenderem e crescerem tecnicamente. Foi justamente na AGES que tive meu primeiro contato com desenvolvimento web e sem esse apoio inicial, talvez eu não teria conseguido chegar onde cheguei

hoje.

#### 4.4 Conclusão

O projeto Se Doce Fosse representou um marco significativo na minha trajetória como desenvolvedor de software, proporcionando um ambiente desafiador e enriquecedor para aplicar e expandir meus conhecimentos técnicos, já que dessa vez atuei como Arquiteto de Software e Líder técnico do Frontend. Entender melhor sobre infraestrutura e realizar um deploy na AWS foram experiências valiosas que complementaram minha formação e me prepararam para desafios futuros na área de desenvolvimento de software. Durante o projeto, enfrentei diversos desafios técnicos e de comunicação que exigiram soluções criativas e colaborativas. A necessidade de alinhar as expectativas entre as equipes de frontend e backend, bem como a gestão de tarefas entre os diferentes níveis de experiência dos membros do time, foram aspectos cruciais que contribuíram para meu crescimento profissional.

Sobre o projeto em si, acredito que consegui contribuir de maneira significativa para o sucesso das entregas, especialmente ao apoiar os AGES I e AGES II no frontend e na resolução de bloqueios técnicos, minha maior meta era fazer com que o aprendizado deles não fosse deixado de lado, pois como já visto na primeira reunião, eles nunca tinham tido contato com esse tipo de tecnologia antes e creio que a AGES não é só sobre entregar um projeto, mas sim sobre o aprendizado e crescimento de todos os envolvidos. Um dos maiores problemas era ter um número alto de AGES IV, já que isso acabava por nenhum deles ter o controle total do projeto, fazendo com que um dependesse do outro para resolver problemas, o que não é o ideal, já que precisamos tomar decisões mais assertivas o quanto antes. Esse problema seria solucionado se eles mesmo tivessem sido designados cada um para um tipo de tarefa, como por exemplo, um para infraestrutura, outro para backend e outro para frontend, assim cada um teria total controle sobre sua área e poderia tomar decisões mais rápidas e eficazes.

Em resumo, o projeto Se Doce Fosse foi uma experiência transformadora que ampliou meus horizontes técnicos e interpessoais. As lições aprendidas sobre liderança técnica fizeram com que eu me tornasse um profissional mais completo e preparado para enfrentar os desafios do desenvolvimento de software em equipe. Ademais, a

oportunidade de ensinar alguém com pair programming foi extremamente gratificante, pois pude ver o crescimento dos colegas e a evolução do projeto como um todo, já que no final, os AGES I estavam realizando tarefas de maneira autônoma, o que demonstra o sucesso do processo de aprendizado e colaboração implementado ao longo do projeto, que era minha meta inicial.

## 5 AGES IV — “NOME DO PROJETO XXXX”

Para cada projeto o aluno deverá argumentar sobre o que foi desenvolvido, seu aprendizado com críticas e autocritica.

### 5.1 Introdução

Na introdução o aluno deve apresentar os seguintes itens: Descrição do Projeto, Stakeholders do Projeto, Período de Execução, Professor Orientador e uma foto do time.

### 5.2 Desenvolvimento do Projeto

#### 5.2.1 Repositório do Código Fonte do Projeto

Deverão ser apresentados os links da Wiki, com uma breve descrição.

#### 5.2.2 Banco de Dados Utilizado

Deverão ser apresentados os links da Wiki, com uma breve descrição.

#### 5.2.3 Arquitetura Utilizada

Deverão ser apresentados os links da Wiki, com uma breve descrição.

#### 5.2.4 Protótipos das Telas Desenvolvidas

Deverão ser apresentados os links da Wiki, com uma breve descrição.

#### 5.2.5 Tecnologias Utilizadas

Deverão ser apresentados os links da Wiki, com uma breve descrição.

### 5.3 Atividades Desempenhadas Pelo Aluno no Projeto

#### 5.3.1 Sprint 0

No mínimo uma página contendo tudo que o aluno fez na Sprint 0.

### 5.3.2 Sprint 1

No mínimo uma página contendo tudo que o aluno fez na Sprint 1.

### 5.3.3 Sprint 2

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

### 5.3.4 Sprint 3

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

### 5.3.5 Sprint 4

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

## 5.4 Conclusão

Neste item o aluno deverá refletir sobre:

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

(No mínimo uma página de relato)

## 6 CONSIDERAÇÕES FINAIS

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

Este item é somente para os AGES IV.

Em particular, espera-se neste capítulo:

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

(No mínimo uma página de relato)

## REFERÊNCIAS

AMAZON.COM, INC. **Amazon Web Services (AWS): Cloud Computing Services.** [S. I.: s. n.], 2025. Accessado em: 16/11/2025. Disponível em: <https://aws.amazon.com/>.

DOCKER, INC. **Docker: Empowering App Development for Developers.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://www.docker.com/>.

FACEBOOK. **React.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://reactjs.org/>.

FACEBOOK. **React Native.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://reactnative.dev/>.

GOSLING, James. **The Java Language Specification.** 3rd. [S. I.]: Addison-Wesley Professional, 2018. Accessado em: 05/07/2024. ISBN 0133260224. Disponível em: <https://docs.oracle.com/javase/specs/jls/se8/html/>.

MATERIAL-UI TEAM. **Material-UI.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://mui.com/>.

MICROSOFT. **TypeScript: JavaScript with syntax for types.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://www.typescriptlang.org/>.

MONGODB, INC. **MongoDB: The Developer Data Platform.** [S. I.: s. n.], 2024. Acessado em: 14/09/2025. Disponível em: <https://www.mongodb.com/>.

PIVOTAL SOFTWARE. **Spring Boot.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://spring.io/projects/spring-boot>.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL: The World's Most Advanced Open Source Relational Database.** [S. I.: s. n.], 2024. Acessado em: 14/09/2025. Disponível em: <https://www.postgresql.org/>.

POSTMAN. **Postman.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://www.postman.com/>.

WORLD WIDE WEB CONSORTIUM (W3C). **CSS: Cascading Style Sheets.** [S. I.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em: <https://www.w3.org/Style/CSS/Overview.en.html>.

WORLD WIDE WEB CONSORTIUM (W3C). **HTML: HyperText Markup Language.** [S. l.: s. n.], 2023. Accessado em: 05/07/2024. Disponível em:  
<https://html.spec.whatwg.org/multipage/>.

## APÊNDICES

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