



GABRIELLE ALMEIDA CUBA

RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO
***FRONT-END* NA SQUADRA DIGITAL**

LAVRAS – MG

2021

GABRIELLE ALMEIDA CUBA

RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO *FRONT-END* NA SQUADRA DIGITAL

Relatório de Estágio apresentado à Universidade Federal de Lavras, como parte das exigências do curso de Ciência da Computação, para obtenção do título de Bacharel

Prof. Dr. Maurício Ronny de Almeida Souza

Orientador

LAVRAS – MG

2021

GABRIELLE ALMEIDA CUBA

RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO *FRONT-END* NA SQUADRA DIGITAL

Relatório de Estágio apresentado à Universidade Federal de Lavras, como parte das exigências do curso de Ciência da Computação, para obtenção do título de Bacharel

APROVADA em 04 de Novembro de 2021.

Prof. Dr. Maurício Ronny de Almeida Souza UFLA
Profa. Dra. Ana Paula Piovesan Melchiori UFLA
Profa. Dra. Juliana Galvani Greggi UFLA

Prof. Dr. Maurício Ronny de Almeida Souza
Orientador

**LAVRAS – MG
2021**

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me permitido chegar aqui. Agradeço meus pais Marluce e José Expedito, minha irmã Maria Julia e minha tia Marly e todos os outros familiares por todo apoio, força e amor durante toda minha vida, especialmente durante a graduação. Agradeço também todos amigos e amigas que conheci durante o curso e que contribuíram para que eu chegasse até este momento.

Agradeço aos professores Maurício e Ana Paula por todo apoio durante este trabalho. Agradeço também a professora Juliana por fazer parte da banca e além disso por todo apoio e recomendações durante a graduação. Agradeço a todos os outros professores por todo ensinamento durante o curso.

RESUMO

A Squadra Digital é uma empresa que atua com desenvolvimento de *software* com foco em áreas de conhecimento de negócio como: Governo, Telecom, Finanças, Indústrias, Comércios e Serviços. Com isso, este documento apresenta o relato das atividades desenvolvidas pela autora durante o estágio supervisionado na empresa, no período de Jun/2020 a Mar/2021, que englobou: um treinamento que apresentou todos os conceitos relacionados a projetos, versionamento, tecnologias utilizadas na empresa; um projeto principal, que era uma aplicação web desenvolvido utilizando o React como biblioteca principal no *Front-End* e no *Back-end* foi utilizado .Net; um projeto secundário que era o *Front-End* de uma aplicação que também utilizava React como biblioteca.

A estagiária foi contratada para atuar na área de *Front-End*, sendo assim durante o período a autora realizou atividades relacionadas ao desenvolvimento web, como a criação de protótipos, telas e serviços para o cliente no qual atuou e no processos de manutenção dos serviços essenciais para o mesmo. Entre as habilidades comportamentais desenvolvidas no período do projeto, podemos ressaltar o trabalho em equipe, responsabilidade, argumentação e proatividade.

Palavras-chave: *Front-End*. Engenharia de *software*. Desenvolvimento de *software*.

ABSTRACT

Squadra Digital is a company that operates with software development with a focus on areas of business knowledge like: Government, Telecon, Finance, Industries, Trades and Services. As follows, this document presents the report of the activities performed by the author during the supervised internship at the company, during the period of Jun/2020 to Mar/2021. The activities during this period included, training related to projects, versioning, technologies used in the company; Developed a web application, front end was built using React and back-end was built using .Net; A secondary project, developed the front-end for an application using React.

The intern was hired to work in the Front-End area, then during the period the author carried out activities related to web development, such as the creation of designs, screens and services for the customer in which she worked and in the maintenance processes of the essential services for it.

Keywords: Front-End. Software Engineering. Software Development.

LISTA DE FIGURAS

Figura 3.1 – Exemplo de como os estados se comportam sem e com o Redux	11
Figura 3.2 – Scrum Framework	14
Figura 3.3 – Modelo Spotify	15
Figura 4.1 – Protótipo da tela inicial referente aos estudos utilizavam um único número de processo	22
Figura 4.2 – Protótipo da tela da inicial referente aos estudos de Cessão Parcial e Englobamento	23
Figura 4.3 – Protótipo da tela da inicial referente ao estudo de Mudança de Regime	24
Figura 4.4 – Protótipo da tela do mapa mostrando suas funcionalidades	25
Figura 4.5 – Protótipo da tela do Visualizar Estudo	28
Figura 4.6 – Protótipo da tela do Visualizar Estudo com o modal de imagens aberto	29
Figura 4.7 – Protótipo da tela do Processar Estudos	30
Figura 4.8 – Protótipo da tela de Corrigir Poligonal	32

SUMÁRIO

1	INTRODUÇÃO	7
2	SOBRE A ORGANIZAÇÃO	9
3	CONCEITOS E TECNOLOGIAS UTILIZADAS	10
3.1	FRONT-END	10
3.2	METODOLOGIAS ÁGEIS	11
3.2.1	Scrum	11
3.2.2	Modelo Spotify	13
3.3	CONTROLE DE VERSÃO	15
3.3.1	Git	16
3.3.2	Azure Devops	16
4	ATIVIDADES DESENVOLVIDAS	18
4.1	TREINAMENTO	18
4.2	PROJETO PRINCIPAL	20
4.2.1	ATIVIDADES DESENVOLVIDAS	21
4.2.1.1	Tela de Realizar Estudo	21
4.2.1.2	Tela do Mapa	24
4.2.1.3	Relatório	27
4.2.1.4	Cancelar Estudos	27
4.2.1.5	Visualizar Estudos	27
4.2.1.6	Processar Estudos	29
4.2.1.7	Excluir Poligonal	30
4.2.1.8	Corrigir Poligonal	31
4.2.2	DESAFIOS DO PROJETO	32
4.3	PROJETO SECUNDÁRIO	34
4.3.1	ATIVIDADES DESENVOLVIDAS	34
4.3.2	DESAFIOS DO PROJETO	34
5	CONCLUSÃO	35
	REFERÊNCIAS	37

1 INTRODUÇÃO

O desenvolvimento de *software* envolve a criação e a manutenção de *software* feito por programadores qualificados através de um computador. Esses programas são usados para ajudar as pessoas a alcançar diferentes objetivos, metas ou processos que quando feitos manualmente levam muito tempo. São ferramentas que possuem o intuito de melhorar a produtividade e automatizar certos trabalhos (BEAPYTHON, 2020).

Sabendo disso, a autora buscou um estágio voltado ao desenvolvimento de *software*. Portanto, o presente relatório refere-se às atividades realizadas pela acadêmica de Ciência da Computação na empresa Squadra Digital, que é uma companhia que atua com essa área. O estágio teve duração de 1584 horas, sendo realizado no horário das 09:00 ao 12:00 e das 13:00 às 16:00, no período do dia 03 de junho de 2020 a 02 de março de 2021 de forma remota.

O referido estágio foi uma oportunidade da autora entrar em contato com o mercado profissional, proporcionando uma experiência de trabalho em equipe, através de vivência nos projetos, com processo ágil. Além disso, como foi focado na área de *Front-End*, ou seja, é uma área inclui tudo aquilo processado e exibido no navegador, tudo que o usuário consegue enxergar (QUEIROS; PORTELA, 2020). O estágio fez com que a estagiária tivesse contato com tecnologias que estão em alta no mercado, como é o caso do *React* e *JavaScript*. O estágio proporcionou um breve aprendizado na área de UI/UX, através da criação de protótipos para algumas funcionalidades. Apresentou também a oportunidade da estagiária poder interagir com cliente e se envolver em reuniões decisórias, fazendo com que habilidades interpessoais e de liderança de equipe fossem desenvolvidas.

O objetivo deste relatório de estágio não é só descrever as atividades desenvolvidas pela autora, mas também apresentar a organização onde o estágio foi realizado, relatar e explicar as tecnologias utilizadas. Além disso, mostrar todo processo das atividades, indo desde o treinamento, projeto principal até o projeto secundário, abordando conceitos, mostrando como essas atividades ocorreram e quais foram as dificuldades da autora durante o período do estágio.

Além deste capítulo introdutório, este documento está organizado nos seguintes capítulos. O capítulo 2 descreve sobre a empresa na qual a autora realizou o estágio. No capítulo 3 são abordados conceitos e tecnologias que foram apresentadas e usadas durante todo o processo do estágio. No capítulo 4 tem-se todas as atividades desenvolvidas durante o processo, dividindo-as em treinamento, projeto principal e projeto secundário. O capítulo 5 contém a conclusão que mostra quais as

contribuições do estágio para a vida da autora e quais disciplinas da graduação que ajudaram neste processo.

2 SOBRE A ORGANIZAÇÃO

A empresa em que o estágio foi realizado foi a Squadra Digital, que faz parte do Grupo Squadra – um grupo composto por outras empresas e que possui mais de 500 colaboradores. Com sua sede em Belo Horizonte e com unidades espalhadas pelo Brasil, é uma empresa que está há mais de 34 anos no mercado atuando como provedora de tecnologia para organizações de todos os portes e segmentos (DIGITAL, 2021). A Squadra Digital atua com desenvolvimento de software de alta qualidade, com foco em estruturas técnicas e comerciais direcionadas nas áreas de conhecimento de negócio: Governo, Telecom, Finanças, Indústrias, Comércio e Serviço (DIGITAL, 2021).

Cada sede tem um responsável pelo time, que tem como objetivo alocar os colaboradores em seus respectivos clientes. Além disso, a empresa conta com departamentos internos como departamento pessoal, rh, marketing, finanças, entre outros.

O cliente no qual a estagiária atuou era um cliente do Governo e o time passou por diversas mudanças durante todo o processo. Inicialmente o time era formado por, aproximadamente, 30 pessoas e era dividido em *Front-End*, *Back-End* que é responsável por processar as informações, disponibilizá-las para toda a aplicação e salvá-las no banco de dados, QA que é uma série de atividades destinadas a garantir que os produtos ou serviços fornecidos atendam ao nível de qualidade exigido e *Scrum master* que é responsável por estabelecer o *Scrum* conforme definido no Guia do *Scrum*. O *Scrum master* faz isso para que todos entendam a teoria e a prática do *Scrum*, seja em uma equipe *Scrum* e na organização, possuía também um arquiteto de *software* e um gerente de projetos.

Haviam subtimes que foram formados inicialmente para focar na produção de histórias específicas. Porém, com o decorrer do tempo, aproximadamente 2 semanas de trabalho, a equipe notou que a quantidade de pessoas no projeto era muito grande e estava mais atrapalhando do que ajudando. Portanto foi necessário fazer uma redução de time para as tarefas fluírem.

3 CONCEITOS E TECNOLOGIAS UTILIZADAS

O sistema desenvolvido foi um sistema Web, ou seja, um *software* hospedado na internet onde o usuário poderia acessá-lo através de um navegador Web. Foi dividido em *Front-End* e *Back-End*. O foco deste capítulo é abordar tecnologias que foram usadas no gerenciamento de equipe e projeto, no qual a estagiária teve contato. Além disso, também pretende-se abordar conceitos sobre tecnologias focadas na área que a autora foi contratada, que foi o *Front-End*.

3.1 FRONT-END

O *Front-End* é a parte visual da aplicação, ou seja, é a interface que interage diretamente com o usuário. O desenvolvimento na aplicação foi feito com *React.js*¹ que, segundo seus criadores, é uma biblioteca *JavaScript* declarativa, eficiente e flexível para criação de interfaces de usuários (REACT, 2021). Em 2011 ela foi usada em uma parte da interface da rede social Facebook, porém foi só em 2013 que ocorreu o lançamento oficial e se tornou *open source* (HOUSE, 2020). É uma biblioteca flexível. A ideia é ter componente reutilizável e que gerencia seu próprio estado, fazendo com que a criação de *UI's* iterativas seja uma tarefa fácil (REACT, 2021).

Juntamente com essa biblioteca foi utilizado o *Redux*², criado por Dan Abramov, que também é uma biblioteca e tem como objetivo gerenciar os estados da aplicação, tornando-os unidirecionais (um exemplo é mostrado na figura 3.1), simplificando assim o controle de estados. Essa biblioteca nasceu a partir da arquitetura *Flux*³, criada pelo Facebook para contribuir nas aplicações de interface de usuário (GUEDES, 2020).

Além disso, foi utilizado o *Ant Design*⁴ que é uma biblioteca de componentes de *design*, utilizada para padronizar o CSS e alguns elementos, deixando a interface mais agradável. Com isso não é necessário criar componentes do zero. Para editar estilos de algum elemento foi utilizado o *SASS*⁵ que é um pré-processador de CSS, que ajuda na produtividade do código, possibilitando a criação de variáveis e funções, principalmente quando se trata na duplicação de uma ação.

¹ <https://pt-br.reactjs.org/>

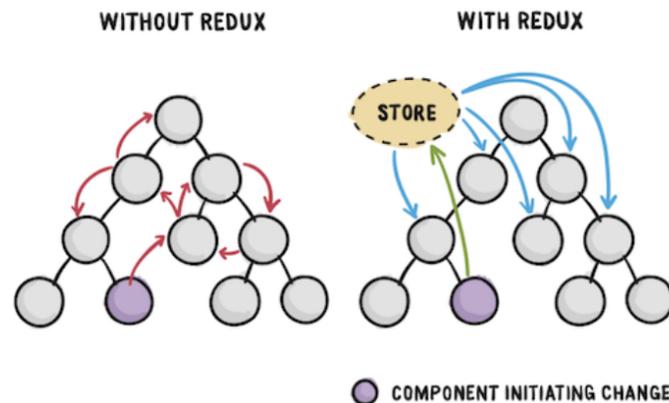
² <https://redux.js.org/>

³ <https://facebook.github.io/flux/>

⁴ <https://ant.design/>

⁵ <https://sass-lang.com/>

Figura 3.1 – Exemplo de como os estados se comportam sem e com o Redux



Fonte: <https://kenzie.com.br/blog/o-que-e-redux/>

3.2 METODOLOGIAS ÁGEIS

A característica principal de processos ágeis é possuir ciclos curtos e iterativos de desenvolvimento. Sendo assim o sistema é implementado de forma gradativa, começando por aquilo que é mais urgente para o cliente (VALENTE, 2020).

Processos ágeis são ciclos curtos que duram aproximadamente um mês. Sendo assim, o sistema vai sendo construído de forma incremental e cada incremento deve ser aprovado pelos clientes (VALENTE, 2020). Processos ágeis são considerados processos leves, quase sem prescrições e documentação (VALENTE, 2020).

As metodologias ágeis chegaram com o intuito de resolver processos organizados em etapas de produção muito longas, problemas com definição de entregas, comunicações pobres entre as equipes e o desalinhamento de informações entre o time interno e os clientes (VINAL, 2018).

Esse relatório aborda uma metodologia ágil muito conhecida que é o Scrum e um estilo que se inspirou em um método ágil que é o modelo *Spotify*.

3.2.1 Scrum

Scrum é uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos (SUTHERLAND; SCHWABER, 2020).

O *Scrum* envolve grupos de pessoas que, coletivamente, possuem todas as habilidades e conhecimentos para fazer o trabalho e compartilhar ou adquirir essas habilidades conforme necessário (SUTHERLAND; SCHWABER, 2020).

O *Scrum* combina quatro eventos formais para inspeção e adaptação dentro de um evento contido, o *Sprint*. Esses eventos funcionam porque implementam os pilares empíricos do *Scrum* de transparência, inspeção e adaptação (SUTHERLAND; SCHWABER, 2020).

O *Scrum* funciona da seguinte forma, requer um *Scrum Master* para promover um ambiente onde, segundo (SUTHERLAND; SCHWABER, 2020):

1. Um *Product Owner* tem a responsabilidade de ordenar o trabalho de um problema complexo em *Product Backlog* que é uma lista ordenada e emergente do que é necessário para melhorar o produto.
2. O time *Scrum* transforma uma seleção do trabalho em um valor durante a *Sprint*.
3. O time *Scrum* e seus *stakeholders* observam os resultados e se ajustam para o próximo *Sprint*.
4. Repetir.

A time *Scrum* consiste em um *Scrum Master*, um *Product Owner* e Desenvolvedores. Times *Scrum* são multifuncionais. Sendo assim os membros possuem todas as habilidades necessárias para criar valor a cada *Sprint*. Os times também são autogerenciados, o que significa que decidem internamente quem faz o quê, quando e como (SUTHERLAND; SCHWABER, 2020).

O Time *Scrum* é pequeno o suficiente para permanecer ágil e grande o suficiente para concluir um trabalho significativo em uma *Sprint*, normalmente 10 ou menos pessoas. Todo o Time *Scrum* é responsável por criar um incremento valioso e útil a cada *Sprint* (SUTHERLAND; SCHWABER, 2020).

O *Scrum* é um contêiner para todos os outros eventos, de acordo com (SUTHERLAND; SCHWABER, 2020). Cada evento no *Scrum* é uma oportunidade formal para inspecionar e adaptar os artefatos do *Scrum*. Um exemplo pode ser visto na figura 3.2. Os eventos são usados no *Scrum* para criar regularidade e minimizar a necessidade de reuniões não definidas no *Scrum*. Os eventos são:

- *Sprint*: são o coração do *Scrum*, onde ideias são transformadas em valor. Eles são eventos de duração fixa de um mês ou menos para criar consistência. Um novo *Sprint* começa imediatamente após a conclusão do *Sprint* anterior.
- O *Sprint Planning* inicia o *Sprint*, definindo o trabalho a ser executado para o *Sprint*. Este plano resultante é criado pelo trabalho colaborativo de todo o Time *Scrum*. O *Product Owner* garante que os participantes estejam preparados para discutir os itens mais importantes do *Product Backlog* e como eles são mapeados para a Meta do Produto.
- O *Daily Scrum* tem como objetivo inspecionar o progresso em direção ao objetivo do *Sprint* e adaptar o *Sprint Backlog* conforme necessário, ajustando o próximo trabalho planejado. O *Daily Scrum* é um evento de 15 minutos para os Desenvolvedores da Equipe *Scrum* sendo realizado todo dia no mesmo horário.
- A Revisão do *Sprint* tem como objetivo inspecionar o resultado do *Sprint* e determinar as adaptações futuras. O Time *Scrum* apresenta os resultados de seu trabalho para os principais interessados e o progresso em direção ao objetivo do produto é discutido.
- Retrospectiva *Sprint* é uma reunião que tem como objetivo é planejar maneiras de aumentar a qualidade e a eficácia.

O *Scrum* é gratuito e é uma framework imutável (SUTHERLAND; SCHWABER, 2020) .

3.2.2 Modelo Spotify

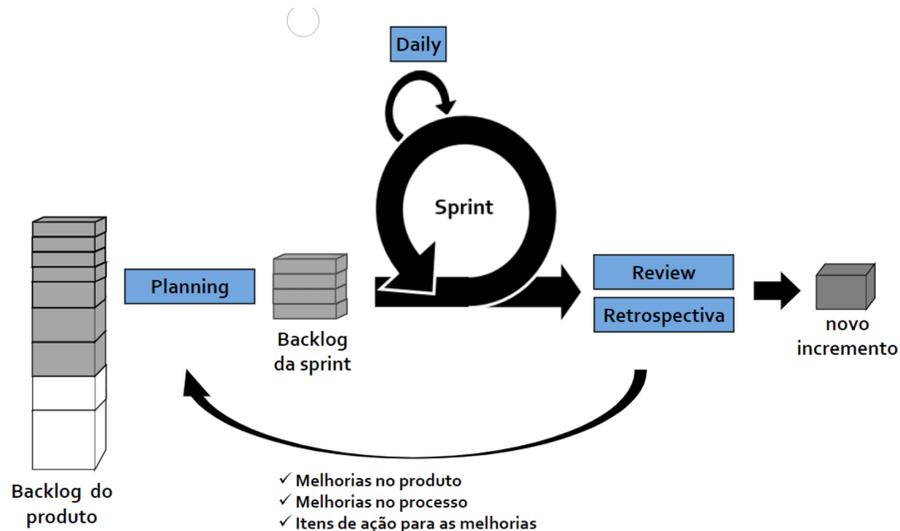
O modelo Spotify⁶ se concentra em como as empresas podem estruturar uma organização para permitir agilidade (CRUTH, 2021).

Foi criado pelo Spotify a partir do momento que eles viram que com o aumento das equipes, a metodologia *Scrum* estava mais atrapalhando do que ajudando. Então resolveram deixar tudo opcional, concluindo-se que os princípios do ágil são mais importantes do que um método específico (KNIBERG, 2014). As mudanças do *Scrum* para esse novo modelo foram:

- *Scrum Master* passou a se chamar *Agile Coach*, pois precisavam de mais líderes servidores do que mestres em processos.

⁶ <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>

Figura 3.2 – Scrum Framework



Fonte: <https://thomazribas.com/agile/scrum>

- Time *Scrum* mudou para *Squad*, que é um time multifuncional, auto organizado, geralmente com menos de 8 pessoas. Possuem responsabilidade de ponta a ponta do que é construído como *design*, *commit*, *deploy*, manutenção, operações. Têm autonomia e decidem o que construir, como construir e como trabalhar junto durante o processo. Existem algumas regras como *Squad Mission*, *Product Strategy* que são aplicadas para qualquer área com a qual eles estejam trabalhando e *Short-term goals* que são renegociados a cada 3 meses.

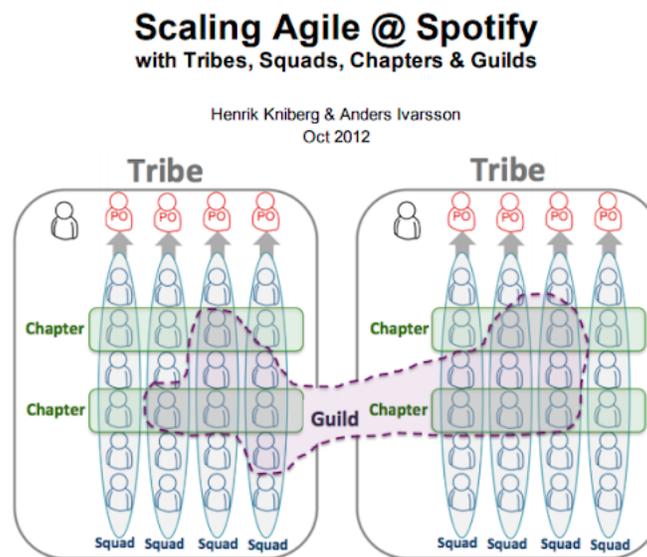
Todo modelo foi pensado para criar um ambiente autônomo que gere motivação, pois acreditam que a equipe motivada cria soluções melhores e mais rápidas.

Há também outras estruturas que compõem este modelo são as *Tribes*, *Guilds* e *Chapters*. A organização delas é mostrado na figura 3.3.

- *Chapters*: São áreas de competência, por exemplo desenvolvimento, qualidade, design ou *agile coaching*. Cada *Chapter* possui um *Chapter Lead*, que é o gerente ou líder-servidor que atua no desenvolvimento de uma determinada competência.
- *Tribes*: É uma estrutura matricial leve, composta por um conjunto de *Squads* e *Chapters*.
- *Squads* possuem um *Product Owner*, que atua de forma semelhante ao que prega o *Scrum*, definindo as prioridades da *Squad* e refinando o *Backlog*.

- *Guilds* são comunidades focadas em um determinado tema, por exemplo *design*, que podem atravessar diferentes *Tribes*, *Squads* e *Chapters*. São de caráter voluntário e qualquer um pode participar.

Figura 3.3 – Modelo Spotify



Fonte: <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>

3.3 CONTROLE DE VERSÃO

Controle de versão é uma atividade da gerência de configuração de software, cujo objetivo é controlar alterações em um arquivo ou um conjunto de arquivos ao longo do tempo para que seja possível recuperar versões específicas (GIT, 2021a). Git, Subversion e CVS são exemplos de ferramentas de controle de versão.

Além disso é um tipo de software que ajuda a equipe de desenvolvedores a comunicar e gerenciar de forma eficiente todas as alterações feitas no código-fonte, junto com as informações de quem fez uma determinada alteração (GEEKS, 2020).

É criado um *branch*, ou seja, uma ramificação no git que aponta para as alterações feitas nos arquivos do projeto, separado para cada contribuidor que fez as alterações e elas não são mescladas ao código fonte, ao menos que sejam analisadas e as mudanças sejam aprovadas. O branch não apenas mantém o código-fonte organizado, mas também melhora a produtividade (GEEKS, 2020).

3.3.1 Git

O Git ⁷ é um sistema de controle de versão de código aberto e gratuito, projetado para lidar com todos os tipos de projetos, desde pequenos até projetos grandes com velocidade e eficiência (GIT, 2021b). Além disso, devido à sua estrutura interna é uma máquina do tempo extremamente rápida e um robô integrado muito competente (AQUILES; FERREIRA, 2014).

O Git possibilita a integração de fluxos de trabalho realizados por vários colaboradores com o passar do tempo, para um determinado repositório. Além disso, tem um fluxo de trabalho de contribuição e uma terminologia exclusivos para dar suporte ao modelo distribuído. Por exemplo, não há um bloqueio de arquivo normalmente associado a operações de *check-in/check-out*, ou seja mudanças iniciais e mudanças finais. Na verdade, o Git se preocupa com alterações em um nível ainda maior, comparando arquivos *byte* por *byte* (MICROSOFT, 2018).

Usa estrutura de camadas para gerenciar e armazenar o conteúdo de um projeto, usando os repositórios que são a maior unidade de armazenamento e podem conter um ou mais *branches*. *Bran-ches* são uma unidade de armazenamento que contém os arquivos e as pastas que compõem o projeto. As *branches* separam o fluxo de trabalho e todas as contribuições são feitas para um *branch* específico e que está no escopo do projeto. Todos os repositórios têm um *branch* padrão chamado de "master" que funciona como a versão atual, sendo assim todos os *branches* criados no repositório nascem a partir do *branch* master (MICROSOFT, 2018).

3.3.2 Azure Devops

O Azure DevOps fornece serviços de desenvolvedor para as equipes de suporte planejarem o trabalho, colaborar no desenvolvimento de código e criar e implantar aplicativos. O Azure DevOps oferece suporte a uma cultura e conjunto de processos que reúnem desenvolvedores, gerentes de projeto e colaboradores para concluir o desenvolvimento de software. (MICROSOFT, 2021)

O Azure Devops abrange a amplitude do ciclo de vida de desenvolvimento para ajudar os desenvolvedores a enviar o produto com mais rapidez e qualidade.

Ele representa a ferramenta mais completa na nuvem pública. Inclui o Azure Boards, Azure Pipelines para compilar, testar e implantar CI/CD, ou seja, integração contínua e entrega contínua, que funciona com qualquer linguagem e possui suporte ao github.

⁷ <https://git-scm.com/>

Possui o Azure Repos que é uma ferramenta que possibilita a criação de repositórios git. Tem também o Azure Boards que ajuda a monitorar o trabalho usando quadros *Kanban* configuráveis. Além disso, permite com que os colaboradores trabalhem juntos em projetos com mais eficiência. (COOL, 2018)

O Azure Devops foi utilizado no projeto principal, para gerenciar o repositório de *Front-End* e de *Back-End*, dando destaque para o Azure Boards que foi utilizado para ter controle das tarefas que eram passadas para os desenvolvedores.

4 ATIVIDADES DESENVOLVIDAS

Nesse capítulo são descritas as principais atividades desenvolvidas pela estagiária na empresa Squadra Tecnologia. As atividades estão organizadas em relação aos principais projetos em que a estagiária esteve envolvida: um treinamento (Seção 4.1) e dois projetos de desenvolvimento (Seções 4.2 e 4.3).

4.1 TREINAMENTO

Ao ingressar na organização, a estagiária foi submetida a um treinamento. O treinamento consistiu em um curso voltado para *Front-End* e uma aula/mentoria voltado para o *Back-End* e conceitos de programação. Ocorreu dos dias 03/06/2020 até 19/06/2020, as aulas ocorreram no horário das 13:00 às 17:00 e o curso poderia ser feito na parte da manhã para completar as 6 horas de estágio. Os estagiários ingressantes não eram separados em relação às atividades de *Front-End* e de *Back-End*, todos tiveram que aprender os mesmos conceitos.

As aulas foram divididas em duas partes. Na primeira parte foram abordados todos os conceitos teóricos, com alguns exemplos de aplicação na prática. Já a segunda parte abordou todos os conceitos teóricos junto com a principal plataforma que a empresa utiliza que é a .Net. No horário da tarde a organização alocava uma mentora que ensinava os principais conceitos. Além disso, havia um mentor dedicado a tirar dúvidas referentes ao curso de React.

As aulas tinham um cronograma a ser seguido: o primeiro tema abordado foi o Git. Foram discutidos os seguintes tópicos: o que é git?; a utilização do sistema de versionamento de código; principais plataformas; o que são ramificações; entendendo o fluxo das ramificações; por onde devo iniciar; comandos básicos e trabalhando com fluxo.

Nesse tema foram abordados a história do Git, o seu significado e o seu criador. Logo em seguida vieram as explicações das principais plataformas. Foram citadas GitLab, GitHub e Bitbucket. No tópico destinado a ramificações e seus fluxos, foram discutidos sobre o que é uma *branch* e o que é a *branch master*, foi apresentada a importância de se trabalhar em *branches* secundárias ao invés de trabalhar diretamente na *master*. Houve uma parte prática onde foram apresentados os principais comandos como *git pull*, *git push*, *commit* e *git add*. Depois foi apresentado o *git flow*, a sua importância e o porquê do seu uso, com uma demonstração teórica e uma demonstração prática.

O segundo tema abordado foi sobre Programação Orientada a Objeto (POO). Neste tema foram explicados os tópicos: o que é um paradigma de programação; o conceito de programação procedural ou imperativa; programação funcional e o conceito de POO. Neste último tópico, abordou-se os principais conceitos e suas respectivas diferenças, abstração, herança, polimorfismo e encapsulamento. Logo em seguida, foi discutido o que é uma classe, objeto, instância, propriedade e método. Ambos os exemplos foram teóricos e a parte prática ficaria para aprender junto com a aplicação do .Net .

O próximo tema abordado foi sobre boas práticas de programação. Foram apresentadas sugestões sobre como nomear variáveis e métodos, como deixar o código limpo, uso de comentários dentro dos códigos, repetição e aproveitamento dos códigos. Foi abordado previamente o conceito de *SOLID-Single Responsibility Principle* (Princípio da responsabilidade única) e indicados 2 livros para serem lidos futuramente : *Clean Code* (MARTIN, 2008) e *Clean Architecture* (MARTIN, 2017).

Outro tema tratado foi o de padrões de projeto. Foram definidos conceitos, os 23 padrões de projetos mais conhecidos, usando como referência o livro "Padrões de Projetos: Soluções Reutilizáveis de Software Orientados a Objetos"(GAMMA, 2009). Para os padrões de projeto, foram apresentados seus componentes que são: nome, objetivo, problema, solução e consequências. Apresentou-se também os requisitos de um bom sistema de padrões e dois critérios. O primeiro é a finalidade, refletindo no que o padrão faz, podendo ter finalidade de criação e estrutural e o segundo critério é o escopo que pode ser dividido em classes e objetos. Além disso, foram apresentados em maior detalhes dois padrões de criação: Factory Method e o Singleton. Foi utilizado o site "*Refactoring Guru*"¹ para apresentar melhor esses conceitos.

Sobre metodologia ágil foi apresentada a sua ideia e logo em seguida discutido sobre o modelo ágil criado pelo Spotify (KNIBERG, 2014). Nessa discussão foram apresentados os conceitos de *Squads*, *Tribes*, *Chapter* e *Guilds*. Além disso, foi mostrado como gerenciar todas essas equipes.

O penúltimo tópico foi sobre testes. Foram abordados os testes de unidade com .Net Core, apresentando o seu conceito, motivação e o princípio de cobertura de códigos. Também foram apresentados alguns exemplos práticos com comparativos de como fazer testes da maneira correta e de maneira errada. Além disso, foi passado um *podcast* que tem como conteúdo testes de unidade, integração, aceitação e cargas.

¹ <https://refactoring.guru/pt-br/design-patterns/behavioral-patterns>

O último tópico mostrado foi o .Net. Este tópico foi apresentado de maneira prática, englobando todos os conceitos abordados previamente no treinamento. A mentora compartilhava a tela e nela ia fazendo todo segmento de código, criava as classes, objetos e também fazia os testes. Devido ao pouco tempo de treinamento a abordagem não foi tão profunda na parte prática, porém abordou o necessário para ter o primeiro contato com a linguagem e entender o que está acontecendo nela.

O curso de React foi disponibilizado na plataforma Cod3r². O curso apresenta conceitos desde o básico ao avançado. Inicialmente foram apresentados conceitos como o porquê utilizar o React, o conceito de SPA (*Single Page Application*) e exemplos práticos fazendo pequenos exercícios como tabelas e calculadora. Inicialmente foi utilizado o modelo de classe e estados no React e após passar esse nível mais básico, foram apresentados os *hooks*, ou seja, funcionalidades que permite que você use estado e outras funcionalidades do React sem ter a necessidade de escrever uma classe, e conceitos de estados aplicados nos *hooks*. Ao final deste módulo, foi feito um projeto de um *ToDo List*. Após finalizar este projeto foi apresentado o conceito Redux, explicando o que é uma *action*, uma *store* e um *reducer*, porque deve ser utilizado e como facilita a vida do desenvolvedor para gerenciar estados globais. No final deste módulo, foi feita uma refatoração do projeto de *ToDo List*, para utilizar ele com Redux. Já o projeto final foi mais complexo, exigindo todo o conteúdo que foi apresentado no curso. O projeto consistia em fazer uma aplicação de controle financeiro. Foi um curso bem completo que conseguiu apresentar os principais conceitos do React.

O treinamento forneceu a base para o projeto no qual a estagiária iria trabalhar.

4.2 PROJETO PRINCIPAL

O intuito do projeto era desenvolver um sistema WEB chamado 'SIG ÁREAS', utilizando o React com ANTD, HTML, SASS e Javascript como tecnologias no *Front-End* e .Net no *Back-End*. O projeto é uma refatoração de um sistema já existente da ANM (Agência Nacional Mineradora), procurando aperfeiçoar e melhorar o seu funcionamento. Tinha como objetivo principal analisar e estudar áreas que foram solicitadas por mineradores e ver se elas eram propícias para mineração ou não.

O período do projeto foi do dia 21/06/2020 à 13/01/2021, logo em seguida houve uma pausa de 2 semanas para questões de atualização dos servidores. O projeto foi retomado dia 01/02/2021

² <https://www.cod3r.com.br/courses/React-redux-fundamentos-e-2-apps-do-absoluto-zero>

e foi até dia 02/03/2021. A equipe do projeto foi composta por 2 desenvolvedores *Front-End*, 3 desenvolvedores *Back-End*, 1 Líder de Projeto, 1 *Scrum Master*, 1 analista de requisitos, 2 *testers* e 1 Gerente de Projeto.

4.2.1 ATIVIDADES DESENVOLVIDAS

A atividade principal do projeto era fazer o *Front-End* da aplicação funcionar. Cada História de Usuário do projeto correspondia a uma nova funcionalidade e possuía uma documentação. O projeto tinha em torno de 40 histórias, portanto em torno de 40 funcionalidades a serem desenvolvidas, incluindo estudos, relatórios e outras telas.

Havia diversas funcionalidades no sistema porém os "estudos das áreas", eram os principais. Os estudos são: Retirada de Interferência, Opção de Área, Redução de Área, Mudança de Regime, Cessão Parcial, Englobamento, Arrendamento Parcial e Desmembramento. As outras funcionalidades que estavam sendo feitas até o momento em que a estagiária ficou no projeto foram: o cancelamento dos estudos, visualizar estudos, processamento manual, recuperar documentos, excluir poligonal e corrigir poligonal.

Para cada uma dessas funcionalidades foram desenvolvidas novas telas, embora houvessem telas parecidas em termos de *design*. No entanto, não foram feitos aproveitamentos de código devido aos parâmetros recebidos em cada funcionalidade serem muito diferentes. Assim, o reuso de código ocasionaria maior esforço, portanto foi decidido manter uma duplicidade de código com parâmetros diferentes.

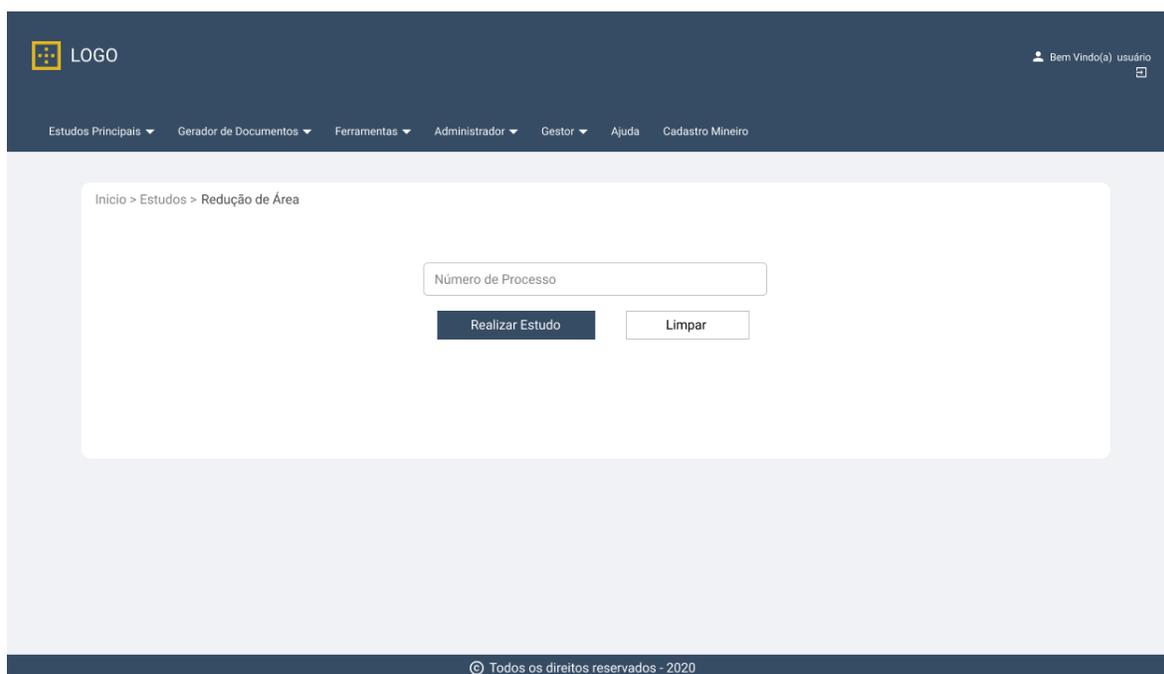
4.2.1.1 Tela de Realizar Estudo

O fluxo principal do sistema funcionava da seguinte maneira: (i) o minerador entrava com um pedido para minerar uma área específica em outro sistema; após isso, (ii) era criado um número de processo referente àquele chamado, que ficava aberto até que (iii) um técnico da ANM pegasse este número de processo e estudasse ele no SIG ÁREAS.

Para poder estudar o número de processo o técnico da ANM precisa identificar a qual estudo ele estava relacionado. Após a seleção do estudo, era aberta uma nova tela (ver Figura 4.1). Essa tela estava inicialmente pronta apenas para a funcionalidade de estudo de Retirada de Interferência, porém ao analisar os estudos seguintes que iriam entrar em desenvolvimento, foi observado que a tela

era exatamente a mesma em questão de *design* para alguns estudos. O que mudava era apenas as chamadas do *endpoint* do *Back-End*, pois as validações para cada tela eram diferentes. A estagiária ficou responsável por passar os parâmetros para o componente da tela de validação e de criar o serviço que iria fazer a requisição para o *Back-End* na funcionalidade "Redução de Área".

Figura 4.1 – Protótipo da tela inicial referente aos estudos utilizavam um único número de processo



Fonte: Elaborada pelo autor no Figma

Relacionado a essa mesma tela, porém para outros estudos, a estagiária ficou responsável por adicionar novas funcionalidades e criar o serviço que chamava as validações do *Back-End* referente a cada estudo desenvolvido, conectando o *Front-End* com o *Back-End*.

Na funcionalidade de estudo de Cessão Parcial, ainda nessa mesma tela, era necessário acrescentar mais um campo de entrada e ao lado deste campo, um botão ("adicionar") para inserir esse número em uma tabela e exibi-la em tela. Uma vez exibida a tabela, seria possível excluir e adicionar novos números. Além disso, associadas ao botão de adicionar, validações deveriam ser feitas em *endpoints*, que é um ponto de comunicação entre o *Front-End* e o *Back-End*, diferentes das validações usadas do botão de realizar estudo. Portanto, foi preciso criar um serviço para buscar o *endpoint* e fazer a conexão do *Back-End* com o *Front-End*. Essas mudanças podem ser vistas na Figura 4.2.

Figura 4.2 – Protótipo da tela da inicial referente aos estudos de Cessão Parcial e Englobamento

Logo

Bem Vindo(a) usuário

Estudos Principais Gerador de Documentos Ferramentas Administrador Gestor Ajuda Cadastro Mineiro

Início > Estudos > Cessão Parcial

Número de Processo

____ / ____ - ____

Adicionar

Realizar Estudo

Limpar

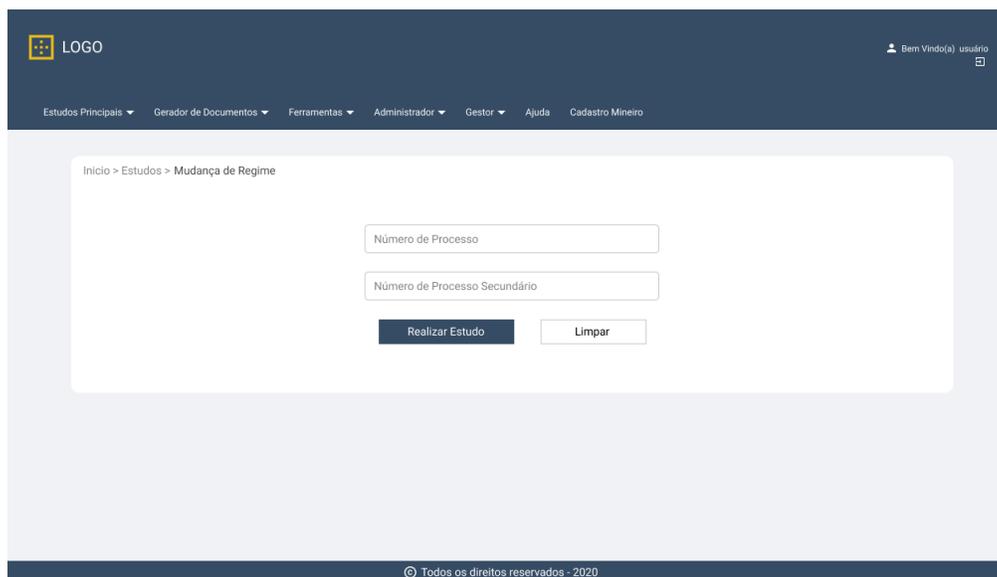
Processo	Ano	Data de Prioridade	Ação
12345.678912/2021-01	2000	00/00/0000	X
00000.678912/2021-01	2007	00/00/0000	X
00000.678912/2021-01	2021	00/00/0000	X

© Todos os direitos reservados - 2020

Fonte: Elaborada pelo autora no Figma

Para a funcionalidade de estudo de Mudança de Regime a tela de realizar estudo (Figura 4.3) deveria ter dois parâmetros de entrada. Já para a tela de estudo de Englobamento, ela era a mesma utilizada na tela de estudo de Cessão Parcial (Figura 4.2). Portanto a estagiária apenas criou o serviço que buscava as validações corretas para cada estudo.

Figura 4.3 – Protótipo da tela da inicial referente ao estudo de Mudança de Regime



O protótipo da tela de Mudança de Regime apresenta uma interface com um cabeçalho escuro contendo o logo e o nome de usuário. Abaixo, há um menu de navegação com opções como 'Estudos Principais', 'Gerador de Documentos', 'Ferramentas', 'Administrador', 'Gestor', 'Ajuda' e 'Cadastro Mineiro'. O conteúdo principal da tela é um formulário branco com o título 'Início > Estudos > Mudança de Regime'. O formulário contém dois campos de entrada de texto: 'Número de Processo' e 'Número de Processo Secundário'. Abaixo dos campos, há dois botões: 'Realizar Estudo' (em azul escuro) e 'Limpar' (em cinza). Na base da tela, há uma barra de rodapé com o texto '© Todos os direitos reservados - 2020'.

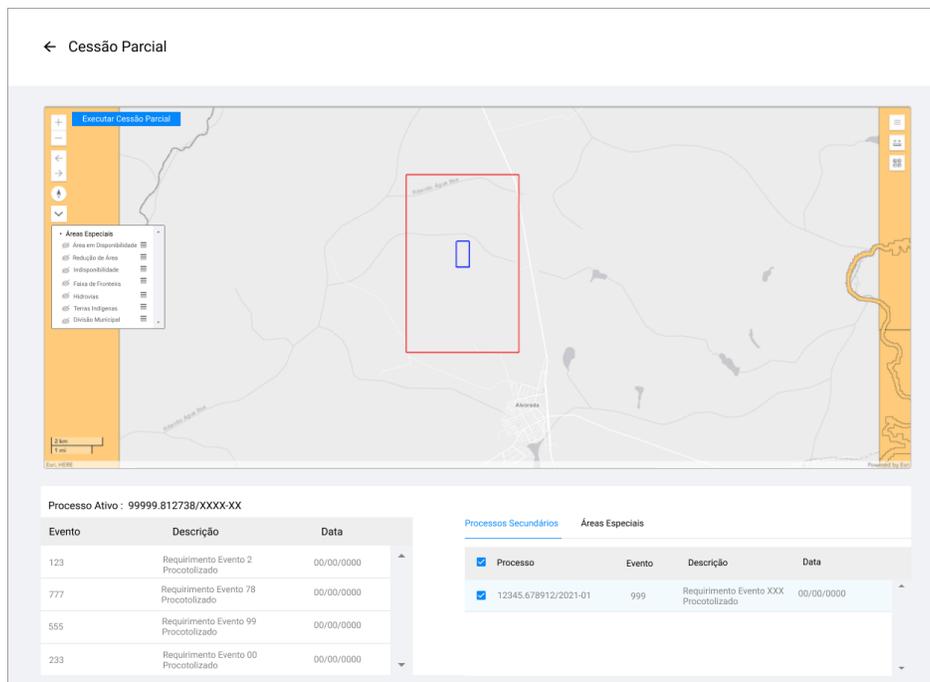
Fonte: Elaborada pelo autora no Figma

Além das validações do *Back-End*, haviam validações realizadas no *Front-End* que eram visuais. Por exemplo, tentar realizar o estudo sem o número estar preenchido, ou o número estar incompleto. Sendo assim, essas validações foram criadas no *Front-End*, com a apresentação de uma mensagem de erro informando qual o erro apresentado.

4.2.1.2 Tela do Mapa

Após passar essa tela inicial de realizar estudo, era aberta uma tela com o mapa (ver Figura 4.4). Esse mapa era consumido de uma API externa, ou seja, era consumido de uma aplicação externa. O componente de mapa já estava pronto para o estudo de Retirada de Interferência, porém era muito semelhante para os outros estudos, só que ao invés de ocorrer o reuso de código, ele foi duplicado por questões já apresentadas neste relatório.

Figura 4.4 – Protótipo da tela do mapa mostrando suas funcionalidades



Fonte: Elaborada pela autora no Figma

A tela do mapa consumia a API do Arcgis³ portanto aparecia um mapa. O *Front-End* era responsável por apresentar no mapa a área do número de processo principal e o processo secundário ou o número de processo que estava relacionado ao principal. Na cor azul aparecia o contorno da área referente ao número de processo em que o técnico solicitou o estudo, ou seja o processo principal, e na cor vermelha apareciam processos secundários ou relacionados ao principal.

No lado esquerdo do mapa havia um menu na vertical, no qual mostrava as áreas especiais, ou seja, áreas que estavam próximas a área estudada ou que até mesmo interferiam a área do processo, eram elas rodovias, hidrelétricas, rios, reservas, processos inativos (número de processos que não estão mais ativos), processos ativos, áreas em disponibilidade, divisão municipal, divisão estadual, faixa de fronteira entre outras. Era possível deixá-las visíveis ou não, bastava selecionar no menu.

Havia um botão “executar estudo + nome do estudo”, por exemplo, “executar retirada de interferência”, que ao clicar nele e passar pelas referentes validações do mapa, o botão mudava para “gerar relatório” no qual ao clicar nele, era gerado um relatório e após o relatório ser gerado o botão

³ <https://www.arcgis.com/index.html>

mudava para “finalizar o estudo” e após clicar no mesmo, o estudo era finalizado e apresentava uma mensagem de estudo finalizado com sucesso.

Como todos os estudos utilizam o mapa, foi criado um componente referente a ele, que teria funções destinadas a área do mapa, por exemplo, plotar uma área, mostrar quais camadas deveriam ser visíveis de acordo com o número do estudo, fazer a troca de botões, de executar para gerar relatório e depois finalizar o estudo. O único estudo que não utilizava esse componente era o estudo de Retirada de Interferência, pois ele já possuía esta funcionalidade pronta quando a estagiária ingressou no projeto e foi pedido para não alterá-la.

Na parte inferior do mapa possuía do lado esquerdo uma área destinada aos eventos do número de processo, por exemplo mostrava-se quais eventos aquele número de processo possuía. Esses eventos eram utilizados nas validações do número de processo para poder entrar na tela do mapa. Do lado direito, ficava uma área que geralmente apresentava duas abas mostrando quais processos interferiam ou estavam relacionados com aquele número estudado, essa funcionalidade variava de estudo para estudo e na outra aba ficava as áreas especiais que aquele processo possuía. Esses detalhes podem ser notados na figura 4.4

Cada estudo tinha um componente que chamava as funções do componente do mapa, como plotar e remover a área. Além disso, tinham funcionalidades específicas, por isso não era possível fazer um componente de estudos e tentar reutilizá-lo, porque tinham muitos parâmetros que mudavam. Além disso, os dados deveriam ser buscados no *Back-End*. Portanto essa ligação era feita através do serviço, chamando cada *endpoint* correto, não era um serviço completo, por exemplo “serviceOpção-DeArea”, era um serviço para cada *endpoint*.

Além de fazer a ligação através do serviço, a estagiária ficou responsável por montar todo componente e toda a parte visual da tela do mapa. Sendo ela a apresentação do mapa, a apresentação dos números de eventos, as abas com os processos relacionados e as áreas especiais, para os seguintes estudos: Opção de Área, Redução de Área, Mudança de Regime, Cessão Parcial e Englobamento. Além disso, a autora ficou responsável por plotar as áreas dos números de processos no mapa, passar todos os parâmetros necessários para gerar a imagem inicial e final que posteriormente seriam utilizadas no visualizar estudo e analisar quais áreas especiais iriam vir visíveis ou não.

Havia validações que eram feitas no *Front-End*, que eram específicas da tela do mapa. Por exemplo, para todo estudo que tivesse interferência com faixa de fronteira deveria aparecer um *pop-up* com uma mensagem de atenção, avisando o técnico desta interferência. Sendo assim a estagiária teve

que criar uma função para analisar quais áreas especiais que aquele número de processo possuía e além disso verificar das áreas retornadas quais delas eram áreas com nome de faixa de fronteira. Ao identificar essas áreas deveria ser emitido um *pop-up* alertando da sua existência.

4.2.1.3 Relatório

Em relação ao relatório, a estagiária não ficou responsável por criá-los em HTML, ficou responsável por criá-los em PDF. Foi utilizada uma biblioteca do React chamada React-pdf⁴ e toda vez que era gerado um relatório em HTML era feito o *download* automaticamente deste relatório em PDF. A autora atuou em correção de erros apresentados nesta funcionalidade.

O *Front-End* também enviava informações para o *Back-End* salvar. Esta ação poderia ser gerada a partir de um clique em algum botão, como por exemplo no botão de "finalizar o estudo" o *Front-End* enviava as informações atualizadas para o *Back-End* salvar no banco de dados.

Referente aos estudos, essas foram as principais atividades, o estudo de englobamento estava pela metade quando chegou o final do estágio. Já os estudos de Arrendamento Parcial e Desmembramento não tinham começado a ser produzidos.

4.2.1.4 Cancelar Estudos

A tela de cancelar estudos era uma tela que já estava pronta quando a estagiária ingressou no projeto, portanto acabou atuando apenas em resolução de erros referentes a ela.

4.2.1.5 Visualizar Estudos

Para a história de Visualizar Estudo, apresentada na Figura 4.5, teve que ser criada uma nova tela, que iria consultar todos os números de processos que estavam em estudos ou que já foram finalizados. Para cada linha da tabela, ou seja, para cada número de processo que aparecia, tinha um botão que quando clicado, era aberto um modal que apresentava duas imagens (Figura 4.6). A primeira imagem era a imagem inicial do estudo e a segunda imagem era uma imagem final, ou seja, após finalizar o estudo. Portanto essas imagens eram salvas quando estudava-se um processo e era finalizado. Caso o processo estivesse em estudo não era possível visualizar essas imagens.

⁴ <https://React-pdf.org/>

Existia um *endpoint* do mapa que era chamado toda vez que o mapa era aberto, para salvar a imagem inicial e esse mesmo *endpoint* era chamado para salvar a imagem final, o que mudava era apenas os parâmetros passado, inicialmente a área era uma e no final era outra.

Quando clicava-se no botão de gerar relatório, conseguia-se gerar a imagem final, devido aos parâmetros passados, da área do processo, que mudava no começo e no final, muitas vezes. Essas imagens eram salvas no servidor, quando o *Front-End* mandava as informações para o *Back-End* após finalizar o estudo.

Com isso era possível o Visualizar Estudos pegar as imagens que estavam salvas no servidor e exibir em tela, sendo possível a comparação. Portanto a estagiária ficou responsável por desenvolver toda essa história no *Front-End*. A única coisa em que não atuou, foi na passagem das Imagens do *Back-End* para o *Front-End*, que quem acabou atuando foi o arquiteto de software, pois a passagem deste parâmetro estava apresentando um erro no servidor, que não estava permitindo trazer as imagens.

Figura 4.5 – Protótipo da tela do Visualizar Estudo

Logo

Bem Vindo(a) usuário

Estudos Principais Gerador de Documentos Ferramentas Administrador Gestor Ajuda Cadastro Mineiro

Início > Estudos > Visualizar Estudos

Número de Processo

Consultar Limpar

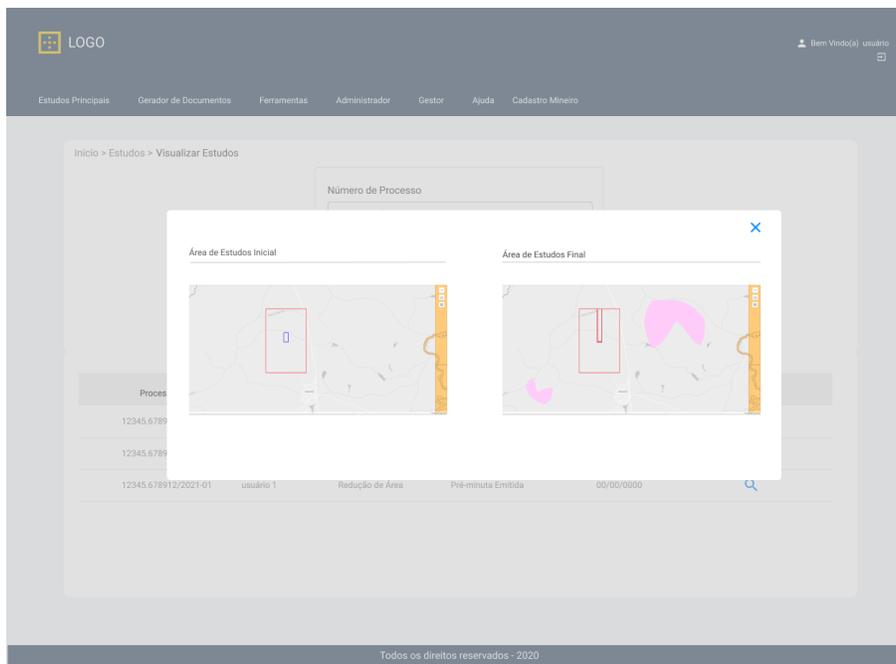
Listar Processos em Estudos

Processo	Usuário	Tipo de Estudo	Situação	Data de Início do Estudo	Ação
12345.678912/2021-01	usuário 1	Redução de Área	Pré-minuta Emitida	00/00/0000	🔍
12345.678912/2021-01	usuário 1	Redução de Área	Pré-minuta Emitida	00/00/0000	🔍
12345.678912/2021-01	usuário 1	Redução de Área	Pré-minuta Emitida	00/00/0000	🔍

Todos os direitos reservados - 2020

Fonte: Elaborada pelo autora no Figma

Figura 4.6 – Protótipo da tela do Visualizar Estudo com o modal de imagens aberto



Fonte: Elaborada pelo autora no Figma

4.2.1.6 Processar Estudos

A funcionalidade de Processar Estudo, que pode ser visto na Figura 4.7, era muito semelhante ao Visualizar Estudo. Teve que ser feita toda a parte de *design* dele em cima de uma software que já existia, tentando seguir a mesma ideia e melhorando funções de *UI/UX* já que o projeto não possuía um protótipo a ser seguido.

Portanto a autora teve que criar um novo componente e um novo *service*, pois as validações para esta história eram diferentes das apresentadas no Visualizar Estudos.

A diferença em questão do *design* é que além dos números aparecerem dentro de uma tabela eles apareciam com a opção de caixa de seleção, sendo possível selecionar mais de um número. Aparecia também um botão escrito processar estudo, que iria processar o estudo seguindo as regras de negócio impostas nas documentações, por isso não foi utilizado o mesmo componente do Visualizar Estudos.

Figura 4.7 – Protótipo da tela do Processar Estudos

Início > Estudos > Processar Estudos

Número de Processo

Consultar Limpar

Listar Processos em Estudos

<input type="checkbox"/>	Processo	Usuário	Tipo de Estudo	Situação	Data de Início do Estudo	Ação
<input checked="" type="checkbox"/>	12345.678912/2021-01	usuário 1	Redução de Área	Pré-minuta Emitida	00/00/0000	🔍
<input checked="" type="checkbox"/>	12345.678912/2021-01	usuário 1	Redução de Área	Pré-minuta Emitida	00/00/0000	🔍
<input type="checkbox"/>	12345.678912/2021-01	usuário 1	Redução de Área	Pré-minuta Emitida	00/00/0000	🔍

Processar Estudo Limpar

Todos os direitos reservados - 2020

Fonte: Elaborada pelo autora no Figma

4.2.1.7 Excluir Poligonal

O estudo de Excluir poligonal foi um dos que a estagiária ficou responsável do início até o final. O estudo tinha uma tela inicial que tinha um campo para inserir um número de processo e selecionar qual camada que ele deveria aparecer e após isso iria para a tela do mapa que era muito semelhante a dos estudos.

Para essa primeira tela de consulta foi criado um componente global, pois essa mesma tela iria ser usada no Corrigir Poligonal, só mudavam alguns parâmetros, como título da tela, a lista de camadas para selecionar, porém ambas propriedades eram recebidas como parâmetro.

A tela do mapa como dito acima é muito semelhante a dos estudos, particularmente com o de Opção de Área, pois a ideia era apresentar quais áreas estavam naquela camada selecionada e poder excluir alguma delas dessa determinada camada.

No componente do lado direito e inferior a tela do mapa, apareciam os ID's, ou seja, números únicos relacionados as áreas, com uma caixa de seleção e conforme selecionada, a cor dessa área mudava no mapa para vermelho, deixando claro que ela foi selecionada.

Além disso, esse número era um link que quando clicado, era aberto um modal, ou seja uma caixa de diálogos, com mais informações sobre o processo. Lembrando que todos os estudos possuíam essa funcionalidade.

4.2.1.8 Corrigir Poligonal

Para o estudo de Corrigir Poligonal, apresentado na Figura 4.8, a estagiária teve que fazer um protótipo para redesenhar toda a tela novamente e mostrar esse protótipo para o cliente, pois como o sistema todo não tinha um protótipo de fato e era feito em cima de um sistema legado, as funcionalidades de *UI/UX* em alguns casos eram muito ruins e no corrigir poligonal era um desses casos.

Portanto, após o protótipo ser apresentado com possíveis melhorias de tela e aprovado pelo cliente, o componente foi montado seguindo o protótipo, aproveitando a tela em comum com o Excluir Poligonal.

Após passar pela tela inicial e por todas validações, era aberta uma nova tela, uma das mais complexas de todo projeto. Este estudo tinha diversas funcionalidades, inicialmente era apresentada uma busca, que dependendo do campo selecionado mudava o layout de tela que deveria apresentar-se na sequência.

Além disso, em um desses layouts a estagiária teve que trabalhar com *upload* de arquivos com extensões específicas, fazendo a leitura desses arquivos e apresentando os dados em tela, na forma de uma tabela que deveria apresentar a possibilidade de editar os campos e na hora de salvar deveria validar se a edição estava correta. Além de todo o *design* também foi criado um serviço para cada validação, fazendo a conexão do *Front-End* com o *Back-End*.

Além da parte de desenvolver novas funcionalidades, a estagiária atuou também na correção de erros reportados através de pessoas que testavam o projeto.

A autora ajudou outros desenvolvedores que entraram depois no projeto a entenderem mais sobre ele, entenderem como cada funcionalidade se comportava e como o código está organizado, mostrando o que estava feito e o que faltava a ser desenvolvido.

Além de conhecimento técnico, a estagiária acabou desenvolvendo um conhecimento de negócio sobre o projeto, o que a proporcionou participar de diversas reuniões para discutir assuntos técnicos e de possíveis melhorias para funcionalidade do projeto.

Figura 4.8 – Protótipo da tela de Corrigir Poligonal

Processo: 12345.XXXXX/2000-00 Camada: Processos Ativos

Formato de descrição do poligonal

Coordenadas Geodésicas Coordenadas Cartesianas

9dsa1-938192j2-dlaaskd3-kkkkkk32

Buscar Limpar

Dados Gerais

Processo: 12345.XXXXX/2000-00

Camada: Processos Ativos

Processo de Origem: 12345.XXXXX/1111-00

Cota Mínima Cota Máxima

Adicionar Coordenadas

Rumos

Rumo Verdadeiro Rumo Diverso

Importar Arquivo Upload

Latitude

Longitude

Latitude	Longitude	Editar	Ação
-00°00'88"888	-00°00'88"888	Edit	X
-00°00'88"888	-00°00'88"888	Edit	X

Validar Poligonal Excluir Todos

Todos os direitos reservados - 2020

Fonte: Elaborada pelo autora no Figma

4.2.2 DESAFIOS DO PROJETO

Inicialmente o maior desafio foi entender como funcionava o projeto, porque ele era composto de tecnologias que a estagiária nunca tinha trabalhado, então isso acabava dificultando o entendimento do fluxo de código, mesmo depois da realização do curso de treinamento voltado para React.

A estagiária apresentou dificuldade para entender onde deveriam ser feitas as modificações para alguma funcionalidade aparecer na tela, além disso teve dificuldade também na parte de trabalhar com integrações com o *Back-End* e até mesmo para testar e simular algum tipo de erro que era relatado. Mas essa dificuldade apresentou-se inicialmente, porque conforme foi passando o tempo, a autora foi entendendo como funcionava o fluxo dos componentes, as integrações e os testes.

Junto com a dificuldade de entender o projeto, apareceu uma dificuldade de entender as documentações, inicialmente foi pelo fato da estagiária nunca ter tido contado com este tipo de documentação. Mas conforme foi passando o tempo, foi notado que a documentação não estava muito clara, ou estava muito básica, pois possuía os requisitos mínimos para fazer uma nova funcionalidade e os requisitos extras, ou muitas vezes, requisitos essenciais tinham que ser buscados através de reuniões com cliente ou até mesmo conversando novamente com analista de requisitos para entender o que deveria ser feito. Isso tudo acaba atrapalhando o andamento das coisas.

Outra dificuldade que a estagiária apresentou, foi em relação a metodologia ágil, por nunca ter tido contato, a autora não entendia muito bem como funcionava. O modelo utilizado era o *Scrum*, então as *Sprints* tinham duração de 10 dias. Porém as vezes, a *Daily* ultrapassa a ideia de ser uma reunião rápida e acabava durando de 1 a 2 horas. Possuía *Planning* e a primeira *Planning* chegou a durar 8 horas. Então, pela estagiária ter tido experiências como essas acabou atrapalhando o entendimento de como funciona uma metodologia ágil na prática.

Junto com a metodologia ágil, umas das maiores dificuldades apresentadas pela estagiária foram as entregas. Justamente porque surgiam muitos *bugs* e o time não conseguia resolver dentro do *sprint*. Isso acabava comprometendo a entrega de uma nova funcionalidade. Muitos dos *bugs* às vezes eram de um serviço secundário, então às vezes, o time tinha que esperar, o serviço ser arrumado, para poder atuar nos *bugs* reportados e tudo isso afetava no tempo da entrega.

Acompanhando essa parte de entrega fora do prazo, algo que dificultava muito, era a queda do servidor do ARCGIS que ficava localizado no cliente e a lentidão do mesmo. Como o servidor do ARCGIS buscava informações referente ao mapa e ao número de processo, com o servidor fora do ar, o trabalho do time e a entrega do produto eram atrasados.

Além das dificuldades já apresentadas, outra dificuldade bem grande que se apresentou durante o projeto, no qual a estagiária atuou, foi uma constante mudança na equipe. Inicialmente a equipe era composta por quase 15 pessoas, porém foi passando o tempo e percebeu-se que as tarefas ficavam muito divididas e uma dependia da outra, fazendo com que alguns desenvolvedores ficassem travados, até que outras partes do código fossem liberadas. Esse problema persistiu do início do projeto até Agosto do mesmo ano, sendo resolvido com a redução da equipe. Porém, ao chegar em Dezembro, houve a saída de 3 pessoas no projeto. A partir de então, desde Dezembro até Fevereiro a equipe ficou bagunçada, entraram novas pessoas, mas elas não entendiam muito bem como funcionava o projeto, não tinha todo o conhecimento, isso acabou atrasando todas as entregas previstas para esses meses.

4.3 PROJETO SECUNDÁRIO

O projeto tinha como objetivo fazer todo o *Front-End* da CEB (Companhia Energética de Brasília), seguindo o protótipo proposto e utilizando o React, Ant Design e SASS.

O período do projeto foi do dia 14/01/2021 até o dia 29/01/2021. Era composto por 2 desenvolvedores *Front-End* e 1 *scrum master*.

4.3.1 ATIVIDADES DESENVOLVIDAS

As atividades desenvolvidas dentre essas duas semanas, foram desenvolver o *Front-End* da aplicação seguindo o protótipo proposto.

A estagiária ficou responsável por fazer parte da página principal, como menu, slide de imagens e algumas outras funcionalidades presentes, que tratavam de colocar imagem estática na página principal, fazer telas e posicionamento de botões. Além disso, a autora fez o design de algumas telas que seriam do sistema interno utilizando HTML, SASS, React e Ant Design.

Por ser um projeto em que só tinha dois desenvolvedores *Front-End* a estagiária conseguiu participar um pouco da arquitetura do projeto e opinar sobre ela, definindo assim hierarquias de pastas e organização delas.

4.3.2 DESAFIOS DO PROJETO

O projeto tinha um protótipo a ser seguido e como optamos em utilizar o Ant Design para estilizar os componentes. Essa escolha acabou dificultando em questão de deixar o *software* fiel ao protótipo, devido às propriedades impostas nos elementos do Ant Design. Sendo assim, muitas vezes, era necessário mudar uma determinada propriedade de um componente e isso refletia em outros elementos. Seria muito mais fácil se tivesse sido utilizado CSS puro no projeto.

Além disso, como o foco do projeto era desenvolver apenas o *Front-End*, o *Back-End* já estava pronto, porém era muito antigo, porque o cliente só queria a atualização do *Front-End*. O cliente queria utilizar o *Back-End* do projeto antigo. Isso estava dificultando a integração de *Back-End* com *Front-End*, pois como era muito antigo tinha como padrão de requisição o *SOAP*, porém React utiliza o *REST*, isso acabou atrasando o desenvolvimento do projeto.

5 CONCLUSÃO

Este relatório descreve as atividades de estágio realizadas pela autora na empresa Squadra Digital, no período de 03 de Junho de 2020 a 02 de Fevereiro de 2021. As atividades desenvolvidas compreenderam treinamentos e a atuação em dois projetos de desenvolvimento. As principais habilidades desenvolvidas estão relacionadas ao desenvolvimento web, em especial o desenvolvimento Front-End, a participação em processos de desenvolvimento que usam metodologias ágeis, e o uso de ferramentas de controle de versão. Espera-se que o conhecimento adquirido durante essa experiência seja proveitoso para outras oportunidades de trabalho voltadas para essa área ou até mesmo em outras áreas.

A partir desta experiência foi possível que a autora colocasse em prática conceitos práticos e teóricos ensinados ao decorrer da graduação. A citar algumas disciplinas que tiveram conceitos vivenciados na prática, durante o estágio:

- Introdução a Algoritmos e Estrutura de Dados aplicado ao conhecimento necessário de algoritmos para poder entender e desenvolver os conceitos básicos de programação;
- Engenharia de Software, que mostra toda a parte de levantamento de requisitos e documentações;
- Processo de Software que apresentou alguns estilos de metodologias ágeis facilitando o entendimento quando a estagiária entrou no projeto;
- Programação Web, que aborda todas as etapas necessárias para desenvolver um software. Porém com uma realva
- Interação Humano Computador, que mostra como olhar a visão do cliente ao desenvolver um protótipo;
- Banco de Dados que apresentou conteúdos compatíveis para o entendimento de armazenamento de dados e possíveis consultas no mesmo.

Uma ressalva de que se a disciplina de Programação Web fosse obrigatória ou que fosse ofertada no início do curso, ajudaria na visão do projeto e com que autora entendesse de forma mais clara o que é o *Front-End* e o *Back-End* e como fazer serviço de requisições. Na visão da autora,

se a matéria de Programação Web fosse ofertada antes de Engenharia de Software, seria mais fácil aproveitar a disciplina de Engenharia de Software, porque as definições de software estariam claras.

Essa mesma ideia de deixar a matéria obrigatória no curso, se aplica para a disciplina de Processos de Software, que aborda temas importantes, como metodologias ágeis e essas metodologias são muito utilizadas nas empresas hoje em dia. Na visão da autora, tornar essa disciplina obrigatória ajudaria no processo de desenvolvimento profissional.

Com isso foi possível que a autora desenvolvesse habilidades de trabalho em equipe através da experiência com o time, responsabilidade e proatividade para poder cumprir com as tarefas que foram passadas e argumentação através da participação em reuniões.

REFERÊNCIAS

- AQUILES, A.; FERREIRA, R. **Controlando versões com git e github**. 1. ed. Reading: Casa Ferreira, 2014.
- BEAPYTHON. **Software Development's Importance In Our Modern World**. 2020. <<https://beapython.dev/2020/06/30/software-developments-importance-in-our-modern-world/>>.
- COOL, J. **Introducing Azure DevOps**. 2018. <<https://azure.microsoft.com/en-us/blog/introducing-azure-devops>>. Acesso em 14 Aug. 2021.
- CRUTH, M. **Discover the Spotify model**. 2021. <<https://www.atlassian.com/agile/agile-at-scale/spotify>>. Acesso em 14 Aug. 2021.
- DIGITAL, S. **Squadra Digital**. 2021. <<https://www.squadra.com.br/>>. Acesso em 26 Out. 2021.
- GAMMA, E. **Padrões de projetos: soluções reutilizáveis**. [S.l.]: Bookman editora, 2009.
- GEEKS, G. F. **Version Control Systems**. 2020. <<https://www.geeksforgeeks.org/version-control-systems/>>. Acesso em 25 Set. 2021.
- GIT. **Getting Started - About Version Control**. 2021. <<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>>. Acesso em 25 Set. 2021.
- GIT. **GIT**. 2021. <<https://git-scm.com/>>. Acesso em 25 Set. 2021.
- GUEDES, M. **O que é Redux?!**. 2020. <<https://www.treinaweb.com.br/blog/o-que-e-redux>>. Acesso em 27 Out. 2021.
- HOUSE, N. **Guia prático sobre React**. 2020. <<https://natahouse.com/pt/guia-pratico-sobre-react>>.
- KNIBERG, H. **Spotify engineering culture(part 1)**. 2014. <<https://engineering.atspotify.com/2014/03/27/spotify-engineering-culture-part-1/>>. Acesso em 14 Aug. 2021.
- MARTIN, R. C. **Clean Code**. [S.l.]: Prentice Hall PTR, 2008.
- MARTIN, R. C. **Clean Architecture**. [S.l.]: Pearson, 2017.
- MICROSOFT. **Conceitos básicos do Git e do GitHub para o Docs**. 2018. <<https://docs.microsoft.com/pt-br/contribute/git-github-fundamentals>>. Acesso em 13 Aug. 2021.
- MICROSOFT. **What is Azure DevOps**. 2021. <<https://docs.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops-2020>>. Acesso em 14 Aug. 2021.
- QUEIROS, R.; PORTELA, F. **Desenvolvimento Avançado para WEB**. 1. ed. Reading: FCA – Editora de Informática, Lda, 2020. Disponível em: <https://issuu.com/lidel/docs/9789727229154_desenvolvimento_avan_ado_para_a_web_>.
- REACT. **REACT**. 2021. <<https://pt-br.reactjs.org/>>. Acesso em 25 Set. 2021.
- SUTHERLAND, M. J.; SCHWABER, M. K. **The 2020 Scrum Guide™**. 2020. <<https://scrumguides.org/scrum-guide.html>>.

VALENTE, M. T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. [S.l.]: Moderna, 2020.

VINAL, V. Metodologias Ágeis: o que são, como implementar as 4 principais. 2018.
<<https://rockcontent.com/br/blog/metodologias-ageis/>>.