



GUILHERME HENRIQUE DE MELO

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO:
DESENVOLVIMENTO WEB, MANUTENÇÃO E
AUTOMAÇÃO DE SERVIÇOS NA EQUALS**

LAVRAS – MG

2021

GUILHERME HENRIQUE DE MELO

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO: DESENVOLVIMENTO WEB,
MANUTENÇÃO E AUTOMAÇÃO DE SERVIÇOS NA EQUALS**

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

GUILHERME HENRIQUE DE MELO

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO: DESENVOLVIMENTO WEB,
MANUTENÇÃO E AUTOMAÇÃO DE SERVIÇOS NA EQUALS**

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 11 de Novembro de 2021.

Prof. Dr. Maurício Ronny de Almeida Souza UFLA
Profa. Dra. Renata Teles Moreira UFLA
Prof. Dr. Bruno de Abreu Silva UFLA

Prof. Dr. Maurício Ronny de Almeida Souza
Orientador

**LAVRAS – MG
2021**

AGRADECIMENTOS

Agradeço, a todos os meus familiares pelo apoio e por tornarem possível minha progressão no mundo acadêmico. Agradeço meus Pais e meu irmão por sempre me apoiarem na minha escolha. Agradeço a minha namorada, Marcela, por ter me apoiado muito a ter iniciado o estágio e também nunca me deixar desistir. Agradeço aos meus amigos, por várias manhãs e noites passadas estudando e rindo muito.

Agradeço, o professor Maurício por todo o apoio durante este trabalho e sempre dando toda atenção para o sucesso, além de me apoiar em escolhas difíceis. Agradeço aos professores Renata e Bruno por estarem presentes em minha banca e também por tornar o DCC um local feliz e de muito aprendizado.

RESUMO

A empresa Equals atua no nicho da gestão financeira inteligente. Nesse sentido esse documento apresenta o relato das atividades desenvolvidas pelo autor durante o estagio supervisionado na empresa no período de fev/2020 a maio/2021. Durante o período o estagiário realizou atividades relacionadas desenvolvimento web com a criação de paginas para principal site da empresa e no processos de automação e manutenção dos serviços essenciais para a empresa. Entre as habilidades desenvolvidas no período podemos ressaltar trabalho em equipe, responsabilidade, argumentação, oratória e escuta ativa.

Palavras-chave: Estágio. Desenvolvimento. Web. Microsserviços. Engenharia de software. Node. React. Spring.

ABSTRACT

The company Equals operates in the niche of intelligent financial management. In this sense, this document presents the report of the activities developed by the author during the supervised internship in the company in the period Feb/2020 to May/2021. During the period the trainee performed activities related to web development with the creation of pages for the main site of the company and in the processes of automation and maintenance of essential services for the company. Among the skills developed in the period we can highlight teamwork, responsibility, argumentation, oratory and active listening.

Keywords: Internship. Development. Web. Microservices. Software Engineering. Node. React. Spring.

LISTA DE FIGURAS

Figura 3.1 – Protótipo do fluxo seguido pelo Jira na equipe onde o estagiário trabalhou.	13
Figura 3.2 – Board de Desenvolvimento.	14
Figura 3.3 – Ilustra um projeto fictício utilizando arquitetura de monólitos e microserviços . . .	14
Figura 4.1 – Fluxo de desenvolvimento no Git seguido pela equipe	19
Figura 4.2 – Tela responsável por demonstrar os registros de monitoramento para a equipe de análise	21
Figura 4.3 – Modal de edição dentro da tela de monitoramentos, onde é possível editar um monitoramento	21
Figura 4.4 – Figura que demonstra o slide que descreve a tecnologia de Nesting do SASS. Essa tecnologia é responsável por tornar o CSS mais semelhante a um código HTML.	24
Figura 4.5 – Figura que demonstra o slide que descreve tecnologias do SASS chamadas de Extend/Inheritance	25
Figura 4.6 – Interface padrão do Storybook encontrada em seu site	25

SUMÁRIO

1	Introdução	7
2	Sobre a empresa	8
3	Conceitos e Tecnologias utilizadas	10
3.1	SpringBoot	10
3.2	Banco de Dados	11
3.3	React	11
3.4	Git	12
3.5	Jira	12
3.6	Arquitetura de Microsserviços	13
3.7	Node.js	15
3.8	Métodos Ágeis	15
4	Atividades realizadas	17
4.1	Treinamentos	17
4.2	Equipe de suporte	19
4.2.1	Desenvolvimento Web	19
4.2.2	Automação	22
4.3	Objetivos semestrais	23
4.4	Desafios e Aprendizados	26
5	Conclusão	27
	REFERÊNCIAS	28

1 INTRODUÇÃO

Durante o período de Fevereiro de 2020 até setembro de 2021, o estudante Guilherme Henrique de Melo atuou como estagiário na Empresa estabelecida em São Paulo e Lavras, chamada Equals. O estagiário passou por um processo seletivo onde os participantes foram entrevistados e testados através de um *case* de desenvolvimento. Entre os trabalhos desenvolvidos, o estagiário foi escolhido por ter o melhor resultado.

O objetivo deste documento é descrever as atividades realizadas pelo estagiário no período citado. O período inicial na empresa foi através de um treinamento buscando uma aperfeiçoamento nas tecnologias, conjunto do aprendizado de novas tecnologias e seguido de demandas executadas para o entendimento do fluxo de desenvolvimento e trabalho da equipe onde foi alocado. A equipe era composta por Líder, PO, *tester* e desenvolvedores. A equipe seguiu o Scrum como metodologia ágil, com cerimônias diárias e quinzenais. O estagiário também lecionou treinamentos e participou de diversos cursos na empresa.

O estagiário aprendeu algumas tecnologias como os serviços da AWS – Amazon Web Service ¹ – e automação de serviços utilizando orquestradores e agendadores. Outro papel importante foi o desenvolvimento do *website*, atuando no *Frontend* e *Backend* da aplicação que é utilizado somente internamente pela maioria das equipes da empresa. Uma das maiores conquistas do estagiário foi a refatoração completa de um dos serviços da empresa, utilizando bastante conhecimento teórico e prático.

Além deste capítulo introdutório, este documento está organizado nos seguintes capítulos. No Capítulo 2, será apresentada empresa que o estagiário residiu. No Capítulo 3 as principais tecnologias que o estagiário utilizou, conjunto com a dificuldade passada durante o período de estágio. No capítulo 4 as atividades desenvolvidas durante o período de estágio. Por fim, no Capítulo 5 será feita uma conclusão descrevendo como o estágio e a importância da UFLA na vida de Guilherme.

¹ <https://aws.amazon.com/pt/>

2 SOBRE A EMPRESA

Este Capítulo apresenta a empresa Equals, onde ocorreram as atividades de estágio. A empresa Equals é uma *fintech*, onde seu foco é tecnologia e também atuação no cenário de finanças, posicionada dentro do "guarda-chuva" de empresas adquiridas pela Stone¹.

A Equals trabalha com a conciliação de recebíveis de pequenas, médias e grandes empresas. A conciliação de recebíveis é responsável por validar se as vendas, taxas e impostos pagos pelo comerciante estão corretos, baseados no acordado com a adquirente. Um possível cenário onde a conciliação de recebíveis disponibilizada pela Equals seria quando um comerciante acorda uma taxa fixa de 1% das suas vendas com a adquirente, entretanto a real taxa seria de 1,5% e a discrepância nas taxas seria detectada.

Atualmente a Equals possui dois principais produtos, sendo eles Raio-X e Equals-Core. O Raio-X é responsável pelos comerciantes de baixo e médio porte e possui um número de transações que chega a abranger centenas de milhares por dia, já o Equals-Core se responsabiliza pelos comerciantes de grande porte, entretanto com um número bem menor de transações por dia.

A Equals atualmente está organizada em dois polos, um estabelecido em São Paulo (SP) e outro polo em Lavras (MG). O polo que está em São Paulo é responsável pela parte de negócios feitos pela empresa, como assuntos ligados à diretoria, comercial, e financeiro. O polo de Lavras, onde o estagiário atuou, é responsável pela parte de tecnologia, se relacionando com desenvolvimento, novas tecnologias e central de funcionamento.

Atualmente em Lavras a Equals está organizada em 9 equipes, com uma média de 8 integrantes, formados por líder técnico, líder de pessoas, desenvolvedores, testadores e PO (*Product Owner*). Cada time estabelecido na central de Lavras é responsável por um produto/responsabilidade distinta na empresa. O estagiário atuou no time OPX, onde exercia um papel muito importante na empresa que prestava suporte para todas as equipes e também a melhoria e manutenção das automações das etapas de conciliação da empresa.

Os ritos feitos pelo time OPX, time onde o estagiário atuou, foram:

- Daily Meeting (Reunião diária)
- Retro (Retrospectiva quinzenal)

¹ <https://www.stone.com.br>

- Planning (Planejamento da sprint)

A *daily* é a reunião executada diariamente para deixar o time alinhado das demandas feitas no dia anterior, o que será feito no dia e no caso de algum impedimento alinhar o time que é necessário ajuda, o momento da *daily* também é utilizado para dar alinhamentos, para equipe, sobre possíveis exames médicos ou saídas de emergências durante o dia. A *retro* é um rito realizado quinzenalmente e seu objetivo é inspecionar o trabalho da equipe da última *sprint* e criar planos de ação para pontos que precisam de melhorias. A *planning* é o rito que acontece todo início/fim de *Sprint*, na *planning* a equipe faz a leitura de todas as atividades que serão executadas na *sprint* e discute cada ponto que deve ser desenvolvido e sobre as regras de negócio para um entendimento integral de cada demanda. Durante a *planning*, para cada demanda é feita uma votação entre os membros que atuam nas atividades para buscar uma concordância na nota que será atribuída a cada demanda. A nota tem um impacto importante pois determina a quantidade de tarefas que serão executadas no período da *sprint* e também no caso de nota máximas concluímos que a atividade necessita de uma divisão em tarefas menores. Os ritos que são seguidos pela equipe são originados da metodologia ágil chamada *Scrum*.

3 CONCEITOS E TECNOLOGIAS UTILIZADAS

Neste Capítulo, são descritas as tecnologias utilizadas pelo estagiário na empresa Equals. Durante o estágio, diversas tecnologias já conhecidas pelo estagiário foram aperfeiçoadas, como Spring-boot (Seção 3.1) e Banco de Dados SQL (Seção3.2). Outras tecnologias, como React (Seção3.3), GitHub (Seção3.4), Jira (Seção3.5) e Node.js (Seção3.7) foram aprendidas durante o tempo de estudo.

3.1 SpringBoot

O Spring Boot ¹ é uma estrutura de software livre projetada para facilitar esse processo em aplicações Java. Portanto, traz mais agilidade ao processo de desenvolvimento, pois os desenvolvedores podem reduzir o tempo gasto na configuração inicial do projeto (SPRING.IO, 2021). Utilizando o Spring Boot é possível facilitar a configuração de várias etapas da configuração de um ambiente como servidores, Dependências, Bibliotecas e *Health Check* da aplicação (SPRING.IO, 2021).

Para completar todo o processo de desenvolvimento, o Spring Boot usa um conceito chamado “convenção sobre configuração” que é uma ferramenta que pode decidir a melhor maneira de fazer algo. Isso é o que chamamos de “ferramenta opinativa”, que toma decisões por nós com base em convenções e aplica configurações padrão e torna o trabalho mais fácil. No entanto, não é inflexível e ainda permite configurações diferentes da configuração padrão se o usuário desejar (SPRING.IO, 2021). Por exemplo, é possível alterar o servidor da *Web* que deseja usar, o Tomcat ² é usado por padrão. Uma das maiores vantagens que o Spring Boot traz para o desenvolvimento é que todas essas configurações não precisam mais ser feitas por *XML*, embora ele ainda suporte esse tipo de configuração. A maioria das configurações pode ser feita programaticamente por meio de *annotations*.

Essa tecnologia já havia sido utilizada pelo Estagiário em um trabalho de Engenharia de Software na graduação. No processo seletivo, que foi baseado no desenvolvimento de uma aplicação *FrontEnd* e *Backend*, o já conhecimento da tecnologia foi crucial para a admissão do estagiário na empresa.

¹ <https://spring.io/projects/spring-boot>

² <http://tomcat.apache.org/>

3.2 Banco de Dados

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é um conjunto de programas que permite aos usuários criar e manter bancos de dados. Portanto, o SGBD é um sistema de software com o propósito de facilitar o processo de definição, construção, operação e compartilhamento de bancos de dados entre vários usuários e aplicativos. Definir um banco de dados envolve a especificação do tipo de dados, estrutura e restrições dos dados a serem armazenados no banco de dados. O modelo de dados relacionais básico representa um banco de dados como um conjunto de tabelas, cada uma das quais pode ser armazenada como um arquivo separado. A maioria dos bancos de dados relacionais usa uma linguagem de consulta de alto nível chamada SQL (ELMASRI et al., 2005).

A programação SQL pode ser usada para analisar ou executar tarefas em tabelas, principalmente através dos seguintes comandos: inserir, pesquisar, atualizar e excluir (ELMASRI et al., 2005). Diversos treinamentos lecionados pela equipe de Banco de Dados da Equals foram importantes para um conhecimento da maneira correta de criar consultas eficientes e o mais leves possíveis, pois devido ao fluxo de dados, sobrecargas no banco podem acontecer e comandos construídos da maneira correta eram necessários no dia a dia do estagiário.

Conhecimento sobre banco de dados aprendido pelo estagiário durante o período de cursos na UFLA. Esse conhecimento ajudou tanto no processo seletivo da empresa como no aperfeiçoamento do estagiário no dia a dia na empresa.

3.3 React

O React ³ é uma biblioteca criada em 2011 pela equipe do Facebook que surgiu com a responsabilidade de facilitar a criação de interfaces gráficas interativas (REACTJS.ORG, 2021). O React é renderizado de forma eficiente à medida que os dados na tela são alterados e é codificado utilizando Java Script.

O React funciona pensando na criação de vários componentes encapsulados que gerenciam seu próprio estado, ou seja, cada componente presente em uma tela é independente e possui o seu próprio funcionamento e ciclo de vida. O React existe também no mundo Mobile, conhecido como React Native ⁴, entretanto o estagiário não trabalhou com aplicações mobile.

³ <https://reactjs.org/>

⁴ <https://reactnative.dev/>

Apesar de muita dificuldade, o React passou a ser a Biblioteca preferida do estagiário quando se trata de aplicações *Frontend* e esteve diariamente presente no dia a dia da empresa.

3.4 Git

Git ⁵ é a ferramenta de controle de versão mais utilizada no mundo dos desenvolvedores. Git é um ferramenta responsável por administrar o controle de versão de um projeto, criado por Linus Torvalds (também criador do Linux ⁶), onde todas as alterações feitas em seu projeto são armazenadas (GIT.SCM, 2021).

O Git é conhecido pela sua eficiência e flexibilidade em seus processos. Operações de *commits* de novas alterações, *branches*, mesclagem e comparação de versões anteriores, são otimizadas para desempenho (GIT.SCM, 2021). O Git possui uma segurança extremamente rígida, onde todo código possui uma hash de criptografia segura chamado SHA1 para proteger o código de visualização e alterações maliciosas (GIT.SCM, 2021).

O conhecimento de Git absorvido no curso de Ciência da Computação na UFLA foi crucial para o profissional do estágio. Palestras lecionadas por estudantes e trabalhos feitos por professores que exigiam um controle de versão ajudaram fortemente no aprendizado de git, que hoje em dia é tão importante e utilizado em qualquer empresa de desenvolvimento.

O Github ⁷ é uma plataforma utilizada para gerenciar seu código e criar um ambiente de colaboração entre desenvolvedores, utilizando o Git como sistema de controle (GITHUB.COM, 2021). Na equipe OPX diariamente era necessário atuar no Github e cautela era muito importante no processo de versionamento para manter o padrão do histórico de desenvolvimento da empresa.

3.5 Jira

O Jira ⁸ Software faz parte de uma série de produtos projetados para ajudar todos os tipos de equipes a gerenciar seu dia a dia de trabalho. O Jira se tornou uma ferramenta de gerenciamento poderosa para todos os tipos de casos de uso, desde casos de teste e gerenciamento de requisitos até o desenvolvimento ágil de software (ATLASSIAN.COM, 2021).

⁵ <https://git-scm.com/>

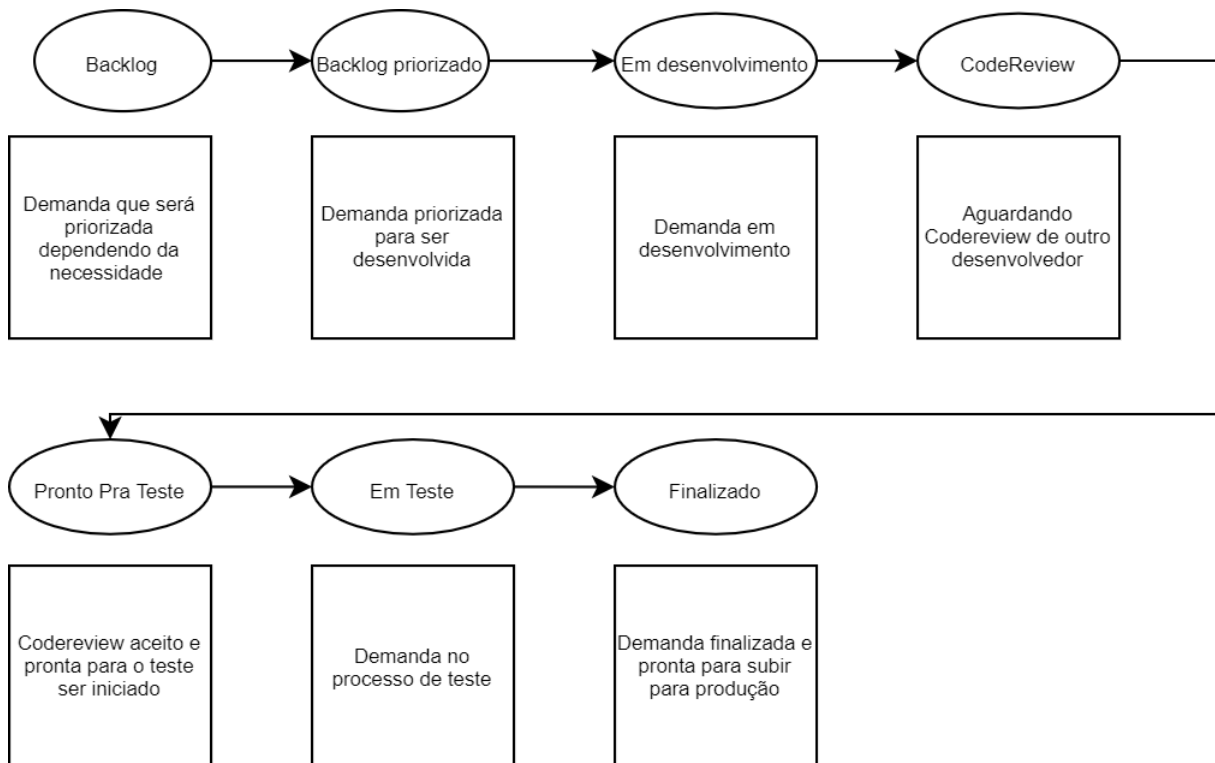
⁶ <https://www.linux.org/>

⁷ github.com

⁸ <https://www.atlassian.com/br/software/jira>

Na Equals, o Jira é o principal agente para organizar o fluxo de desenvolvimento. Toda e qualquer demanda executada pelo estagiário passava o fluxo de desenvolvimento pré-definido pela empresa, apresentado nas Figura 3.1 e Figura 3.2.

Figura 3.1 – Protótipo do fluxo seguido pelo Jira na equipe onde o estagiário trabalhou.



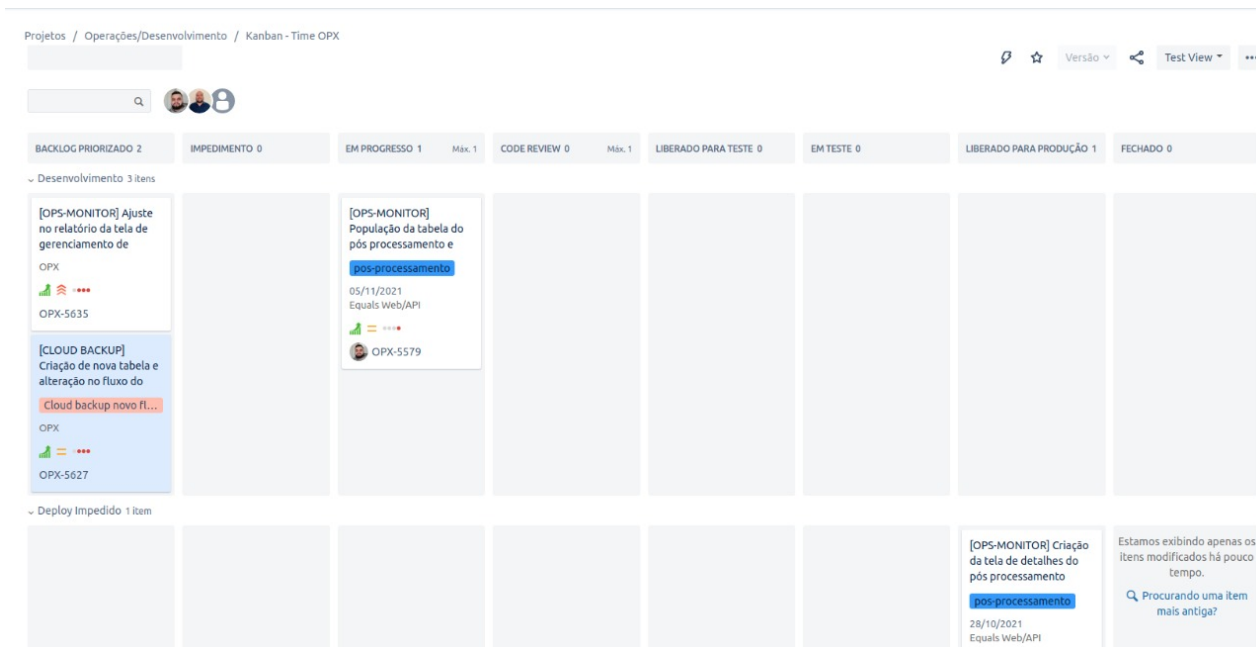
Fonte: do autor

3.6 Arquitetura de Microsserviços

O time de OPX, possui uma arquitetura baseada em microsserviços e não em monolitos. Na Figura 3.3, podemos ver a organização de um projeto utilizando a arquitetura monolito e um projeto com a arquitetura de microservices.

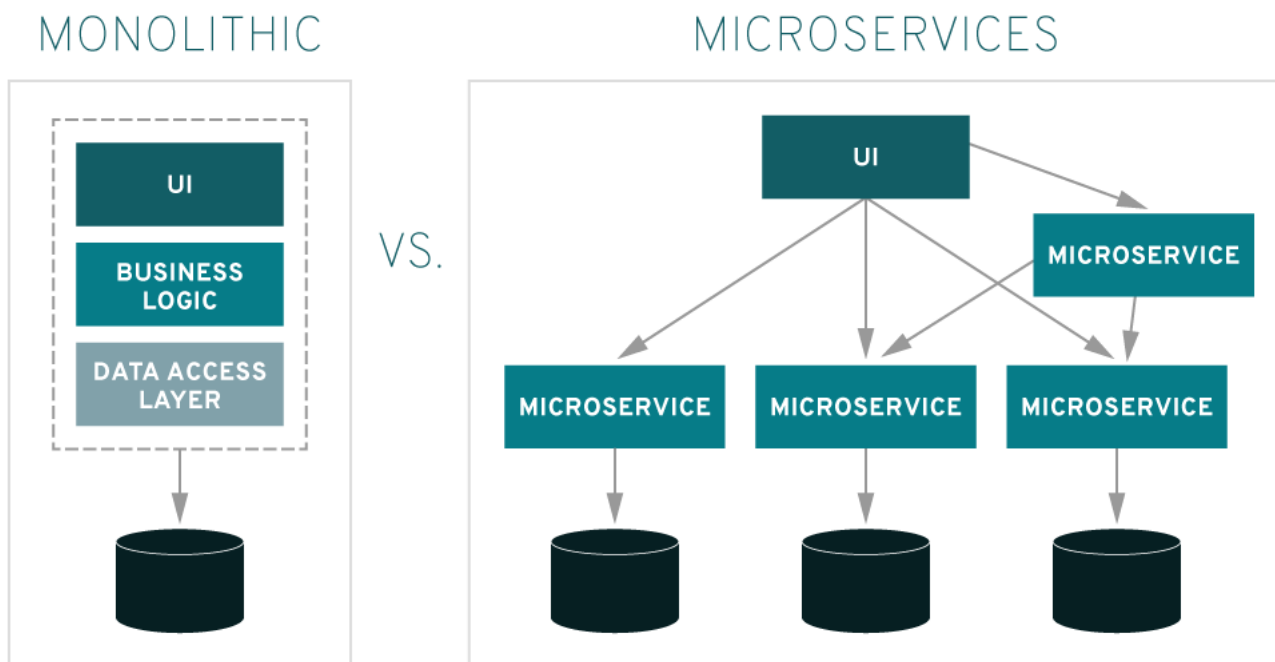
A utilização de micro serviços tem como objetivo desmembrar aplicações, as tornando mais independentes, e comunicando entre si através de APIs. Esse desmembramento é importante pelo fato de trazer agilidade na manutenção, facilidade no descobrimento de *bugs* e eficiência no desenvolvimento de novas funções. Quando falamos sobre monolitos erros podem se propagar por todo sistema e assim impedindo toda aplicação de funcionar.

Figura 3.2 – Board de Desenvolvimento.



Fonte: do autor

Figura 3.3 – Ilustra um projeto fictício utilizando arquitetura de monólitos e microserviços



Fonte: <https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>

3.7 Node.js

O Javascript é uma linguagem de programação popular entre os programadores e aparece predominantemente no *Frontend (Cliente Side)*. O Node.js⁹ mostra que o Javascript também pode ser muito eficiente no ambiente de execução do *Backend (Server Side)* (NODEJS.ORG, 2021). Node.js é muito recente quando comparado com outras linguagens de programação mas já é muito utilizada em grandes empresas como (NODEJS.ORG, 2021), por exemplo Netflix¹⁰ e Uber¹¹(NODEJS.ORG, 2021).

O Node.js possui capacidade de escala em suas aplicações e, também por possuir uma arquitetura baseada em flexibilidade e baixo custo, o torna uma boa escolha para implementações em microsserviços (NODEJS.ORG, 2021). A principal característica do Node.js que o diferencia de outras tecnologias é o fato que sua execução é *single-thread*, ou seja, apenas uma *thread* executa a aplicação. O Node.js está conquistando cada vez mais público devido a sua flexibilidade, leveza, produtividade e também facilita por ser a mesma linguagem utilizada no *Frontend* (NODEJS.ORG, 2021).

3.8 Métodos Ágeis

Scrum é um método de gerenciamento de projetos ágil, iterativo e incremental, utilizado na equipe do estagiário. Foi proposto por Jeffrey Sutherland e Ken Schwaber em um artigo publicado pela primeira vez em 1995. Entre os métodos ágeis, Scrum é o mais famoso e mais comumente usado. Provavelmente parte do sucesso desse método se deva à existência de indústrias relacionadas à sua adoção, diversos cursos, consultoria e certificação (VALENTE et al., 2020).

A principal função é que o proprietário do produto (PO) deve possuir a visão do produto que será construído, sendo responsável também por maximizar o retorno do investimento feito no projeto. O Scrum Master é uma função única e exclusiva do Scrum, ele é o Scrum Expert da equipe, responsável por garantir que as regras do método sejam seguidas, a lista de afazeres do produto é uma lista de histórias classificada por prioridade, a história é escrita e priorizada pelo proprietário do produto e é uma breve descrição das funções que devem ser implementadas no projeto. Já além das

⁹ <https://Node.js.org/en/>

¹⁰ <https://netflix.com>

¹¹ <https://uber.com>

funções dos integrantes, o Product Backlog é um artefato dinâmico, ou seja, deve ser constantemente atualizado para refletir as mudanças nos requisitos e na visão do produto.

Sobre o fluxo, seu desenvolvimento é dividido em sprints e dura até um mês. No final da sprint, produtos que tenham valor tangível para o cliente devem ser entregues. O Sprint Planning é uma reunião em que toda a equipe se reúne para determinar a história que será implementada na próxima sprint. Portanto, o evento marcou o início do sprint. O Sprint Backlog é o artefato gerado no final do plano do Sprint. É uma lista de tarefas de sprint e a duração da tarefa (VALENTE et al., 2020).

4 ATIVIDADES REALIZADAS

Neste Capítulo, são descritas as atividades realizadas pelo estagiário na empresa Equals. As atividades incluem: Treinamentos (Seção 4.1), atividades na equipe de suporte (Seção 4.2), desenvolvimento Web (Seção 4.2.1), automação dos serviços da empresa (Seção 4.2.2), e objetivos semestrais (Seção 4.3). Ao fim do capítulo, também são apresentadas algumas dificuldades enfrentadas ao longo do estágio (Seção 4.4).

4.1 Treinamentos

No período de treinamento, o estagiário teve como foco um melhor aprendizado nas tecnologias utilizadas pela equipe para uma melhor integração. Inicialmente foram feitos os seguintes cursos na plataforma Alura ¹ :

- Curso Spring Boot API Rest: Segurança da API, Cache e Monitoramento: abordando os conteúdos de Páginas com API Rest, Spring Cache e monitoramento de *endpoints*.
- Curso Apache Camel: O framework de integração entre sistemas. Abordando os conteúdos de Apache Camel, rotas e endpoints, práticas de padrão de integração.
- Curso Java OO: Introdução à Orientação a Objetos: Abordando o conteúdo de orientação a objetos completa para Java.

Após os cursos serem finalizados, foi feito um estudo mais avançado sobre Java 11, com o intuito de criar uma apresentação, para equipe, sobre as novidades e também sobre o que foi aprendido nos cursos. Neste momento inicial de treinamento também foi aprendido pelo estagiário os ritos que são seguidos pela equipe. Após o fim da apresentação e dos treinamentos, o estagiário foi submetido a iniciar o desenvolvimento com o auxílio de outro desenvolvedor em uma demanda do dia a dia da equipe . A demanda que foi desenvolvida tinha como objetivo inserir uma nova adquirente no fluxo já existente. A primeira atividade desenvolvida pelo estagiário, foi necessário esforço para leitura do código já existente e melhorar o entendimento da regra de negócio da empresa. Foi uma atividade simples em questão de codificação, porém complexa no sentido de entendimento no código já existente.

¹ <https://www.alura.com.br/>

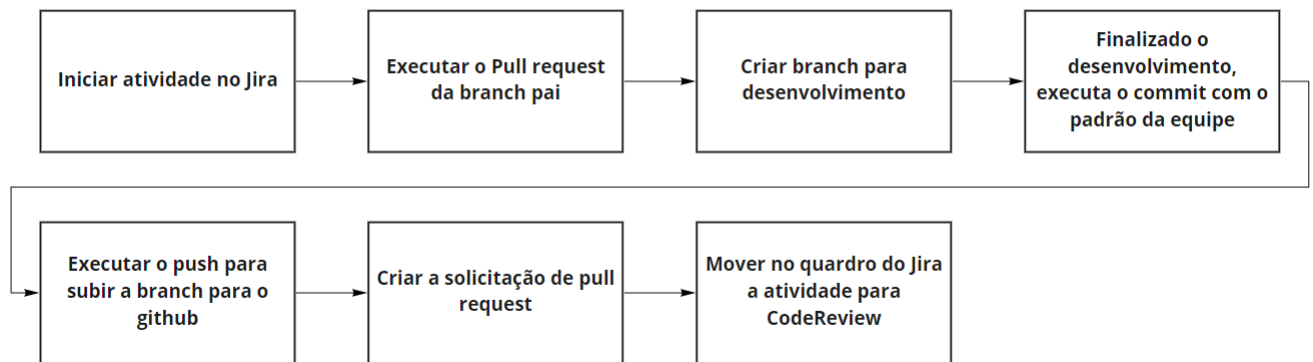
Na equipe onde o estagiário atuou, todo desenvolvedor deve seguir um fluxo de desenvolvimento para que não existam discrepâncias entre o caminho seguido durante o processo de execução das tarefas e também para um melhor rastreamento do que foi desenvolvido. O fluxo é gerenciado por meio das atividades cadastradas na ferramenta Jira, chamadas internamente de "Jiras". O fluxo que deve ser seguido ao iniciar uma atividade de desenvolvimento possui os seguintes passos:

1. Movimentação do "Jira" para Desenvolvimento "Em Progresso"
2. Criação da *Branch* no GitHub para desenvolvimento da demanda
3. Leitura da demanda e anotação dos pontos que deverão ser executados
4. Iniciar desenvolvimento
5. Testar se o que foi solicitado está correto e se não teve impacto em demais pontos
6. Criação de um Teste unitário
7. *Commit* no padrão da equipe
8. Criação do *pull request*
9. Movimentação do "Jira" para *Codereview*

Na Figura 4.1, é possível visualizar o fluxo do Git que deveria ser executado com cautela durante o desenvolvimento das demandas. O primeiro passo é iniciar a demanda no Jira e em seguida a execução do *pull request* para garantia do código fonte estar atualizado. A criação de uma *branch* para o desenvolvimento utilizando o padrão da equipe para nomenclatura, no caso da demanda que está em desenvolvimento na Figura 3.2 sua nomenclatura começará com OPX-5579. O próximo passo é executar o *commit* e o *push* para subir a *branch* para o repositório do *Github* e assim criando o *pull request* e alterando a demanda para CodeReview.

Após esse desenvolvimento supervisionado, foi iniciado um desenvolvimento mais independente onde as atividades eram executadas com menos supervisão, mas sempre atentos para sanar dúvidas. Esse período onde uma maior responsabilidade foi designada ao estagiário foi muito importante para um amadurecimento e confiança nas demandas executadas e sempre que bem sucedido um feedback positivo era recebido da equipe.

Figura 4.1 – Fluxo de desenvolvimento no Git seguido pela equipe

Fluxo de execução do git

Fonte: do autor

4.2 Equipe de suporte

A equipe OPX possui diversas aplicações e é responsável pela manutenção/suporte de suas aplicações. A manutenção visa corrigir o erro o quanto antes, no caso de problemas em produção devia ser feito um *hotfix*, sendo uma correção de extrema urgência, quando não é um problema urgente ele será executado na próxima sprint. No suporte é necessário executar melhorias no que já foi criado e implementações de novas funções, sendo muita das vezes sugeridas por outras equipes ou pelos próprios desenvolvedores do time. A equipe possui um "Bombeiro", que é responsável por cuidar da análise dos suportes que são descritos pelas equipes auxiliadas e a partir das análises, criar demandas para serem entendidas na *planning* e executadas na *sprint*.

4.2.1 Desenvolvimento Web

O time OPX é responsável por todo desenvolvimento e manutenção da página *web* que dá suporte a equipe "Coração" da equals. Essa aplicação demonstra todo o fluxo de recebimento de arquivos do dia a dia da empresa através de gráficos e tabelas robustas. A aplicação *Frontend* foi desenvolvida utilizando a biblioteca React e fazendo o gerenciamento de estados gerais através do Redux, que é uma biblioteca responsável por gerenciar os estados do componentes da aplicações, para isso eventos são disparados e tratados no *reducer* da sua aplicação gerenciando seus componente como desejar.

A aplicação do lado do cliente faz requisições para a *API (Backend)*. Entretanto, há também um *middleware* que está implementado em Node.js, por onde todas as requisições que são enviadas pelo *Frontend* passam para serem validadas. O *middleware* é importante pois a aplicação do *Frontend* possui algumas funcionalidades importantes na empresa, garantindo segurança e rastreamento de requisições para a *API*.

O *Backend* foi desenvolvido na linguagem Java utilizando o *framework* Spring. No *Backend* existem diversas rotas no modelos CRUD que são responsáveis por executar comandos, como insert, delete e update, para buscar as informações para popular o *Frontend*. A comunicação com o banco de dados da Oracle está sendo feita utilizando Hibernate, juntamente com o Mybats, que são responsáveis por mapear objeto-relacional e melhorar a comunicação entre banco de dados e aplicação.

As Figuras 4.2 e 4.6 apresentam as telas responsáveis pelo gerenciamento de monitoramentos, possuindo uma tabela com todos os dados necessários para a equipe de análise de processamentos diários. Na Figura é possível ver uma das funcionalidades, que é a modal de edição de monitoramentos. As telas têm como função fazer o gerenciamento das informações das adquirentes da empresa. Para a construção da tela foi necessário construir diversos componentes. Inicialmente foi feito um arquivo onde está todas as configurações do Redux-Saga. Para a construção da tabela foi criado um componente onde toda configuração da tabela é criada, incluindo filtros, ordenações e visualização completa dos dados da tabela através de uma modal. Também na mesma tela também existem duas funções: editar e excluir. Para isso foram usados *modais* onde é possível fazer alterações diretamente nos dados da tabela. É interessante ressaltar que todas as funções necessárias para a página, sendo elas, buscar dados, editar e inserir, também foram criadas pelo estagiário no *Backend* da aplicação.

A tela Gerenciamento dos Monitoramentos 4.2, informa a existência de atraso de arquivos por parte das adquirentes e a situação dos contratos dos clientes. Essa tela possui todas as funcionalidades de um CRUD (Create, Read, Update e Delete).

A tela de Configuração de Adquirentes, Figura 4.6, é composta por uma tabela que nos mostra as principais configurações das adquirentes que reflete informações do banco de dados, porém, conseguimos interagir através do OPS-Monitor. Essa tela possui as funcionalidades de um CRUD (Read e Update).

¹ Hibernate é um framework responsável por fazer o mapeamento relacional na linguagem java

¹ Mybats é uma estrutura onde pode-se armazenar instruções SQL no formato XML

¹ Redux-saga é uma biblioteca que foca em fazer os efeitos colaterais, como chamadas assíncronas para buscar dados em uma API, em aplicações React/Redux serem mais fáceis e simples de se criar e manter.

Figura 4.2 – Tela responsável por demonstrar os registros de monitoramento para a equipe de análise

ID CLIENTE	NOME FANTASIA	CONTRATOS	ADQUIRENTES	LAYOUT ERP	TIMEOUT	TIPO DE MONITORAMENTO	AÇÕES
[blurred]	[blurred]	[blurred]	Cielo		11:25	Situação do contrato	[edit] [delete]
[blurred]	[blurred]	[blurred]	Cielo		11:29	Situação do contrato	[edit] [delete]
[blurred]	[blurred]	[blurred]	Cielo		10:10	Situação do contrato	[edit] [delete]
[blurred]	[blurred]	[blurred]	Cielo Hau Bol... Beleza n...		08:25	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]	Recarga...		23:44	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]	AMEX Good Ca... Sodexo + 5		12:00	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]	Stone		11:00	Situação do contrato	[edit] [delete]
[blurred]	[blurred]	[blurred]	Stone		13:00	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]	Stone		19:00	Situação do contrato	[edit] [delete]
[blurred]	[blurred]	[blurred]	Stone		18:00	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]	Stone		23:00	Situação do contrato	[edit] [delete]
[blurred]	[blurred]	[blurred]			16:21	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]			11:26	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]	Cielo		11:25	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]			16:20	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]			16:20	Atraso de arquivo - Adquirente	[edit] [delete]
[blurred]	[blurred]	[blurred]			16:20	Atraso de arquivo - Adquirente	[edit] [delete]

Fonte: do autor

Figura 4.3 – Modal de edição dentro da tela de monitoramentos, onde é possível editar um monitoramento

ADQUIRENTE	TAMANHO DA FILA	FREQUÊNCIA DEFINIDA	MÁX. DE DIAS SEM RECEBIMENTO	HORÁRIO DE CORTE	E-MAILS DE COBRANÇA	AÇÕES
[blurred]	8	NÃO		15:15	[blurred]	[edit] [delete]
[blurred]	12	SIM		00:02	[blurred]	[edit] [delete]
[blurred]	10	SIM		11:55	[blurred]	[edit] [delete]
[blurred]	10	SIM		22:23	[blurred]	[edit] [delete]
[blurred]	10	SIM		10:10	[blurred]	[edit] [delete]
[blurred]	15	NÃO		10:10	[blurred]	[edit] [delete]
[blurred]	15	NÃO		10:10	[blurred]	[edit] [delete]
[blurred]	10	SIM		11:33	[blurred]	[edit] [delete]
[blurred]	10	SIM			[blurred]	[edit] [delete]
[blurred]	10	NÃO			[blurred]	[edit] [delete]
[blurred]	10	SIM			[blurred]	[edit] [delete]
[blurred]	10	SIM			[blurred]	[edit] [delete]
[blurred]	10	SIM	5		[blurred]	[edit] [delete]
[blurred]	10	SIM	5		[blurred]	[edit] [delete]
[blurred]	1000	NÃO	5		[blurred]	[edit] [delete]
[blurred]	10	SIM	5	02:42	[blurred]	[edit] [delete]
[blurred]	10	SIM	5		[blurred]	[edit] [delete]
[blurred]	10	NÃO	5		[blurred]	[edit] [delete]
[blurred]	1000	SIM	5	04:30	[blurred]	[edit] [delete]

Fonte: do autor

O processo de previsão de arquivo é responsável por realizar a previsão de chegada de arquivos no dia a dia da empresa. Esses serviços que executam a previsão de arquivos foram desenvolvidos em

Java, também se comunicando com o banco de dados e utilizando conceito de funções *lambdas* e rotas em suas execuções. As funções *lambdas* são o método de uma aplicação em Java que está hospedado no servidor da AWS e quando um evento for acionado, por exemplo a chegada de uma mensagem em alguma determinada fila, essa função será executada.

4.2.2 Automação

O estagiário trabalhou com a automação de serviços da empresa, como relacionamento de arquivos e também o processamento retroativo de arquivos. A automação é importante devido a quantidade de arquivos que são recepcionados e processados na empresa (chegando a centenas de milhares de arquivos por dia). A automação dos serviços da empresa torna possível processar essa gigantesca massa de arquivos, diminuindo ao máximo o serviço manual. Esta automação possibilita o acompanhamento visual, por meio de gráficos, do funcionamento da empresa. Neste contexto, o estagiário realizou as seguintes atividades relacionadas à automação: (i) recepcionamento de arquivos, (ii) agendamentos, (iii) desenvolvimento para API do Jira e (iiii) estudo sobre Docker.

Sobre o recepcionamento de arquivos (i), foi utilizado o Camel, é feito o monitoramento de uma pasta onde a empresa recebe arquivos e inicia-se o processo de cadastro desses arquivos no sistema. O processo de *backup* é iniciado através dos serviços da AWS logo após o cadastro dos arquivos, onde é enviado uma mensagem através do SNS para a fila de outro serviço para ser validado o arquivo e executado seu backup em um repositório nacional e um internacional. Vários serviços são citados no processo, isso é devido à utilização de micro serviços e não de monolitos na arquitetura da equipe.

Sobre os agendamentos (ii), os *Schedules* são responsáveis por executar agendamentos que serão executados em horário específicos durante o dia, são utilizados nos processos da equipe, onde algumas aplicações iniciam suas execuções em determinadas horas ou em intervalos. Esse processo é necessário pois existem vários horários de pico de CPU nas máquinas da empresa e devido a isso é preciso a utilização de agendamentos(Schedules) em horários específicos para evitar a sobrecarga. Sobre o desenvolvimento para API do Jira (iii), as atividades responsáveis pelo desenvolvimento utilizando a integração com o Jira, essas atividades são responsáveis por criar demandas em cenários de

¹ O Camel é responsável por fazer o monitoramento de um diretório e tomar as ações desejadas.

¹ O SNS, também conhecido como Simple Notification Service, é responsável por enviar mensagens para as filas desejadas do serviço da AWS

erro nos arquivos recebidos pela empresa ou para notificação dos clientes através de emails. Devido a massa de arquivos diários é possível que existam arquivos com erro, baseado nessa quantidade torna-se inviável a notificação manual de cada ocorrência de um erro e o serviço com integração do Jira faz a notificação e já cria uma demanda, para o time responsável, no caso de erro.

Sobre o estudo sobre Docker (iiii), o estagiário começou um estudo em Docker. Com Docker, é possível lidar com os containers como se fossem máquinas virtuais modulares leves. Além disso, os containers oferecem maior flexibilidade para você criar, implantar, copiar e migrar um container de um ambiente para outro. O estudo teve como objetivo migrar os micro serviços administrados pelo estagiário para containers a fim de melhorar os serviços e evitar erros de ambiente.

4.3 Objetivos semestrais

Semestralmente, todo funcionário da empresa é submetido a uma avaliação, onde é necessário executar alguns objetivos. O estagiário foi submetido duas vezes a essa avaliação de seus objetivos.

No primeiro ciclo de avaliação o estagiário teve dois objetivos, sendo eles: uma apresentação sobre Typescript e um artigo sobre padrões de projeto. Os padrões de projeto são resoluções para problemas comuns recorrentes no mundo dos softwares. Essas soluções são muito bem vistas nos dias atuais e muito utilizadas em grandes projetos.

Para o primeiro objetivo, foi feita uma apresentação de Typescript para a equipe onde o foco era desenvolver uma aplicação com *Frontend* em React e o *Backend* em Node.js, ambos utilizando Typescript. O artigo criado sobre padrões de projeto, para o segundo objetivo, foi publicado na plataforma Medium ² e no artigo foi contemplado a explicação com exemplos e diagramas sobre os padrões de projeto *Strategy* e *Builder*, ambos implementados em Javascript utilizando Typescript.

No segundo ciclo de avaliação o estagiário teve três objetivos: (i) uma apresentação para todos os desenvolvedor da empresa sobre SASS, (ii) criação de uma documentação para o *Frontend* da equipe, utilizando StoryBook e (iii) a aplicação de um padrão de projeto em um dos microsserviços da equipe.

A apresentação de SASS foi feita para todos os desenvolvedores da empresa e a apresentação teve foco em distribuir conhecimento obtido no estudo de SASS. O SASS ³ (*Syntactically Awesome*

² <https://medium.com/equals-lab/design-patterns-fácil-implementação-de-alguns-padrões-de-projetos-em-javascript-com-typescript-f2137859a49e>

³ <https://sass-lang.com/>

Style Sheets) é um pré-processador de CSS onde possibilita uma escrita mais semelhante a escrita do HTML juntamente com várias funcionalidades para utilizar no css. Nas figuras 4.4 e 4.5, está demonstrados alguns dos slides que foram utilizados na apresentação para os desenvolvedores.

Figura 4.4 – Figura que demonstra o slide que descreve a tecnologia de Nesting do SASS. Essa tecnologia é responsável por tornar o CSS mais semelhante a um código HTML.

Nesting

No seu html é possível ver que existe uma hierarquia das tags, visual. Dessa forma o SASS permitirá seguir essa hierarquia visual do HTML.

```
1 <header class="header">
2   <figure class='header__logo'>
3     
5   <nav class="header__menu menu">
6     <ul class="menu__list">
7       <li href='index.html' class="menu__links"><a
8 href="index.html" class="menu__link">Pagina Inicial</a></li>
9       <li href='index.html' class="menu__links"><a
10 href="about.html" class="menu__link">Sobre nós</a></li>
11      <li href='index.html' class="menu__links"><a
12 href="index.html" class="menu__link">Caes</a></li>
13      <li href='index.html' class="menu__links"><a
14 href="index.html" class="menu__link">Gatos</a></li>
15      <li href='index.html' class="menu__links"><a
16 href="index.html" class="menu__link">Passaros</a></li>
17    </ul>
18  </nav>
19 </header>
```

Fonte: do autor

Em relação ao segundo objetivo, o Storybook é uma biblioteca onde é possível documentar os componentes de uma aplicação *Frontend*. Para documentar cada componente é necessário criar um arquivo onde o componente original será importado e para que o storybook entenda aquele arquivo basta retornar o seu componente com a sua documentação. No processo de documentação é possível documentar cada parâmetro de seu componente, colocar valores padrões que o parâmetro recebe e também determinar o tipo do parâmetro. Outra vantagem do Storybook é que na página web que é criada é possível interagir em tempo real com o componente, criar eventos e logs e assim facilitando ainda mais futuro usos do componente.

Para o terceiro objetivo, o padrão de projeto escolhido foi o *Strategy*. O *Strategy* é um padrão de projeto comportamental que permite que você defina uma família de algoritmos, coloque-os em classes separadas, e faça os objetos deles intercambiáveis. Um cenário muito interessante para

Figura 4.5 – Figura que demonstra o slide que descreve tecnologias do SASS chamadas de Extend/Inheritance

Extend/Inheritance

Segundo SASS, Extend/Inheritance é uma das features mais importantes e usadas atualmente. Extends é muito semelhante ao Mixins, mas possui uma grande diferença no código css gerado. Extends tem como função manter o seu SASS o mais ‘seco’ possível. Extends criados e não usados não serão compilados.

```

1 %contentButton-style {
2   border: 1px solid #ccc;
3   padding: 10px;
4   color: #333;
5   font-size: 14px;
6 }

```

Extend

```

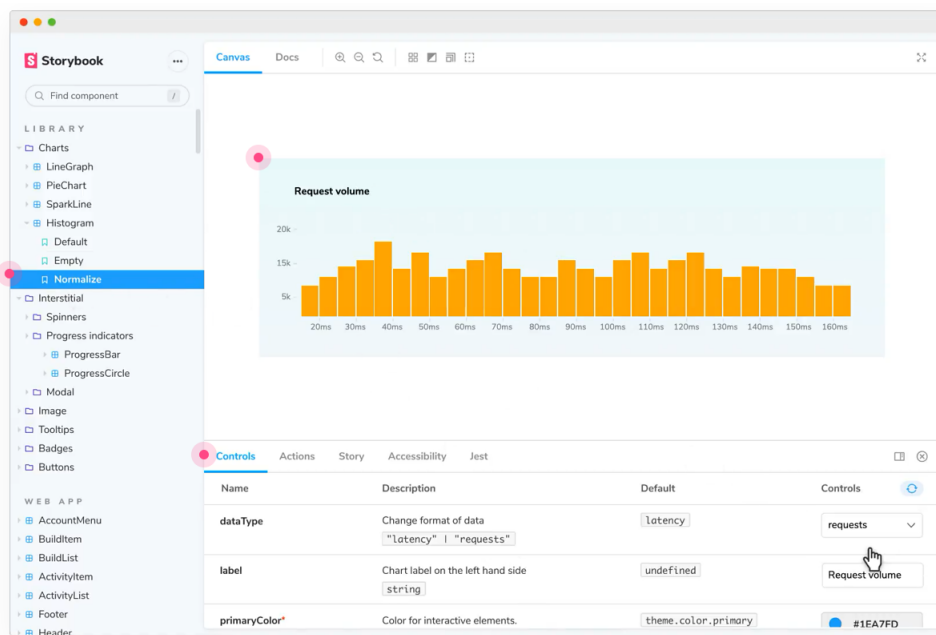
1 @mixin contentButton-style {
2   border: 1px solid #ccc;
3   padding: 10px;
4   color: #333;
5   font-size: 14px;
6 }

```

Mixins

Fonte: do autor

Figura 4.6 – Interface padrão do Storybook encontrada em seu site

Fonte: <https://storybook.js.org/>

simplificar o funcionamento do pattern é o de maneira de cobrança. Considere um software que faz a cobrança de impostos no salário de um indivíduo baseado em qual banco o indivíduo tem conta. Considere que cada banco tem a sua estratégia para calcular o determinado imposto e dessa forma cria-se uma interface que declara a função `cobrarImposto(valorSalario)` e todos os bancos devem implementar determinada interface e assim cada um cria o método `cobrarImposto(valorSalario)` da sua maneira.

Na implementação, inicialmente o serviço de *backup* possui somente um fluxo de *backup* mas com a implementação de um outro fluxo, muito semelhante ao já existente, originou-se uma quantidade de código duplicado, sendo uma prática que não segue o SOLID. Para que seja implementado o pattern, foi criada uma interface com os métodos mais utilizados e semelhantes encontrados e agora cada fluxo deve implementar essa interface e declarar os métodos que não são genéricos.

4.4 Desafios e Aprendizados

O estagiário teve dificuldades no momento de entendimento do código devido a vários outros desenvolvedores já terem atuado, pois como diversas pessoas já codificaram no ambiente mostra que existem diversos padrões e maneiras diferentes de produzir um código. Como meta de superar o obstáculo da dificuldade da leitura de código, o estagiário fez alguns resumos das aplicações e durante a leitura de documentações, junto com a leitura do código, a eficiência na leitura dos códigos cresceu cada vez mais. É importante também dizer que a experiência adquirida nos meses de trabalho também ajudou muito em seu crescimento profissional.

A empresa possui uma regra de negócio complicada e complexa. A Equals é uma *Fintech*, ou seja, uma empresa de tecnologia do ramo de finanças e isso trás uma alta necessidade de mapeamento de informações. No dia a dia do estagiário, toda cautela era necessária, pois os dados que eram manipulados refletia diretamente no dinheiro do cliente, métricas da empresa e funcionamento das aplicações da empresa e também todas as ações executadas nos dados em produção precisavam ser auditadas. Baseado na complexa regra, a documentação era importante para entender as regras existentes, além de perguntas para funcionários mais experientes, a leitura dessa documentação era necessária em toda nova atividade e com o amadurecimento na empresa o estagiário já passava a dar auxílio para os demais funcionários.

5 CONCLUSÃO

Este documento relata as atividades realizadas durante o estágio supervisionado do discente Guilherme Henrique de Melo, na empresa Equals, uma fintech do grupo Stone, no período de Fevereiro de 2020 até Setembro de 2021. Durante o período, o estagiário foi responsável por exercer funções que abrangiam: o desenvolvimento de um web site interno para o monitoramento do recebimento de arquivos; automações de serviços, e suporte a outras equipes.

O período de estágio foi muito construtivo para a carreira e crescimento pessoal. No período de estágio habilidades foram aprendidas, como :Trabalho em equipe; Proatividade; Resiliência; Ser autodidata; Assumir responsabilidades; Comunicação; Organização. Uma das habilidades mais importantes que foi aprendida é o senso analítico, pois era muito necessário pensar em todos os impactos no que estava sendo desenvolvido e analisar cenário com total cautela nos impactos que podem surgir no funcionamento da empresa.

A necessidade de aprendizado constante de novas tecnologias na empresa ajudou muito na formação do estagiário. A convivência diária com trechos de códigos e análises de melhores cenários para resolução de problemas agregaram muito para a habilidade de resolução de problemas, facilitando a resolução de trabalhos e atividades da faculdade.

O curso de computação na Faculdade foi de total importância para o estágio. Analisando pelo lado de tecnologias que foram aprendidas e utilizadas, o estagiário já tinha um pré conhecimento de todos os assuntos. Outro ponto muito fundamental foi a capacidade lógica e resolução de problemas adquirida no curso, principalmente na matéria de Estrutura de Dados. No curso diversos conteúdos foram aprendidos e vários deles, como Engenharia de Software, Banco de dados, foi possível observar na prática durante o período de estágio.

A faculdade foi de total importância para que o estagiário possa melhorar diversas habilidades pessoais, dentre elas: Oratória; Trabalho em Equipe; Capacidade de Resolução de Problemas; Escrita de Relatórios; Organização. Acredito que o curso de Ciência da Computação possa focar mais em tecnologias que estão em alta no mercado e também um maior foco na prática, juntamente com mais atividades que treine a oratória dos alunos.

REFERÊNCIAS

ATLASSIAN.COM. **Atlassian.Jira**. 2021. Disponível em: <<https://www.atlassian.com/br/software/jira>>. Acesso em: 28 out. 2021.

ELMASRI, R. et al. Sistemas de banco de dados. Pearson Addison Wesley São Paulo, 2005.

GITHUB.COM. **Github**. 2021. Disponível em: <<https://github.com/>>. Acesso em: 28 out. 2021.

GIT.SCM. **Git**. 2021. Disponível em: <<https://git-scm.com/>>. Acesso em: 28 out. 2021.

NODEJS.ORG. **NodeJs**. 2021. Disponível em: <<https://nodejs.org/>>. Acesso em: 28 out. 2021.

REACTJS.ORG. **Reactjs**. 2021. Disponível em: <<https://pt-br.reactjs.org/>>. Acesso em: 27 out. 2021.

SPRING.IO. **Spring**. 2021. Disponível em: <<https://spring.io/why-spring>>. Acesso em: 27 out. 2021.

VALENTE, M. T. D. O. et al. Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade. Universidade Federal de Minas Gerais, 2020.