



**VINÍCIUS TEIXEIRA NEIVA**

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O  
ESTÁDIO GOVERNADOR MAGALHÃES PINTO**

**LAVRAS - MG**

**2023**

**VINÍCIUS TEIXEIRA NEIVA**

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA O ESTÁDIO  
GOVERNADOR MAGALHÃES PINTO**

Relatório de estágio apresentado à Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação, para a obtenção do título de Bacharel.

Prof. Dr. Ramon Gomes Costa  
Orientador

**LAVRAS - MG**

**2023**

## **AGRADECIMENTOS**

Dedico toda essa trajetória aos meus pais Camilo e Maristânia, ao meu irmão Matheus, grande estimulador do estudo e da vida acadêmica. À toda minha família, e à minha namorada Giovanna, obrigado por acreditarem no meu potencial. Sei que o trajeto não foi nada fácil, sem a ajuda e confiança de vocês, nada disso seria possível. Agradeço também à todos com quem pude aprender, professores, desde a educação infantil até a superior, pelos ensinamentos e pela paciência.

*“Milagres só acontecem com as pessoas que não desistem.”*

*(Emporio Ivankov - One Piece)*

## RESUMO

Diante do crescente cenário de digitalização, as empresas têm se adaptado e concentrado seus esforços na incorporação de seus produtos e serviços no meio digital. A competição do mercado exige que empresas dos mais variados setores busquem constantes inovações, visando simplificar, automatizar e integrar seus processos de forma abrangente. Essa tendência é impulsionada pelas vantagens de redução de custos, facilitadas pela atual simplicidade no desenvolvimento de aplicações Web. Além disso, a profusão de profissionais qualificados na área e a escalabilidade oferecida permitem a manutenção eficiente de tais sistemas. Este relatório de estágio apresenta as atividades realizadas e as experiências adquiridas durante o período de estágio na empresa Gafit Soluções em Automação. O documento explora a interação do estagiário com metodologias ágeis, como Scrum e Kanban, bem como descreve o processo de criação, estruturação e desenvolvimento de uma aplicação Web. Essa aplicação está dividida em duas partes distintas: a seção administrativa e a parte pública. Para o desenvolvimento, foram utilizadas as tecnologias: *Laravel*, um framework da comunidade *PHP*. Como banco de dados relacional, *MySQL*, *JQuery*, *Docker* para a containerização do projeto, permitindo padronizar e uniformizar a execução em diversos ambientes de trabalho, beneficiando todos os colaboradores envolvidos. Durante o desenvolvimento do sistema, houve a oportunidade de absorver e assimilar os conhecimentos adquiridos ao longo do percurso acadêmico na UFLA. Além disso, foi possível contribuir de maneira significativa para a equipe da Gafit, enriquecendo sua bagagem profissional. O período descrito neste trabalho ocorreu entre junho de 2022 a março de 2023, representando uma valiosa oportunidade para a aplicação prática dos conhecimentos teóricos, bem como o desenvolvimento de habilidades pertinentes em meio ao cenário de acelerada digitalização.

**Palavras-chave:** desenvolvimento web; banco de dados relacional; Laravel; PHP; Scrum; MySQL

## LISTA DE FIGURAS

Figura 2.1 – Mapa de Atuação . . . . .	9
Figura 2.2 – Metodologia GAFIT . . . . .	10
Figura 2.3 – Mineirão Público . . . . .	11
Figura 3.1 – Dashboard ClickUp . . . . .	23
Figura 3.2 – Gather Town . . . . .	24
Figura 3.3 – Arquitetura MVC Web . . . . .	25
Figura 4.1 – Fluxograma da Hierarquia da Configuração de Usuários - Mineirão . . . . .	29
Figura 4.2 – Parte administrativa Mineirão . . . . .	30
Figura 4.3 – Modelagem E-R de nível lógico de usuários e autenticação . . . . .	31
Figura 4.4 – Tela de seleção de ingresso . . . . .	32
Figura 4.5 – Interação de leitura dos mapas de assentos . . . . .	33
Figura 4.6 – Ingresso selecionado . . . . .	33
Figura 4.7 – Checkout . . . . .	34
Figura 4.8 – Tela exemplo do Checkout Pro . . . . .	36
Figura 4.9 – Exemplar do Modelo de Ingresso . . . . .	38

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>7</b>
<b>2</b>	<b>Contextualização da empresa e mercado atuante</b>	<b>8</b>
<b>2.1</b>	<b>GAFIT</b>	<b>8</b>
<b>2.2</b>	<b>Mercado</b>	<b>8</b>
<b>2.3</b>	<b>Recrutamento e Treinamento</b>	<b>9</b>
<b>2.4</b>	<b>Participações em Projetos</b>	<b>11</b>
<b>3</b>	<b>Tecnologias utilizadas</b>	<b>12</b>
<b>3.1</b>	<b>Ferramentas de Desenvolvimento</b>	<b>12</b>
<b>3.1.1</b>	<b>HTML/CSS</b>	<b>12</b>
<b>3.1.2</b>	<b>JavaScript</b>	<b>12</b>
<b>3.1.3</b>	<b>PHP</b>	<b>13</b>
<b>3.1.4</b>	<b>Laravel</b>	<b>13</b>
<b>3.1.5</b>	<b>Docker</b>	<b>14</b>
<b>3.1.6</b>	<b>Laravel Sail</b>	<b>14</b>
<b>3.1.7</b>	<b>MySQL</b>	<b>15</b>
<b>3.1.8</b>	<b>Webpack</b>	<b>15</b>
<b>3.1.9</b>	<b>EloquentORM</b>	<b>16</b>
<b>3.1.10</b>	<b>Laravel Blade</b>	<b>16</b>
<b>3.1.11</b>	<b>jQuery</b>	<b>16</b>
<b>3.1.12</b>	<b>Bootstrap</b>	<b>17</b>
<b>3.1.13</b>	<b>PhpStorm</b>	<b>17</b>
<b>3.1.14</b>	<b>WSL</b>	<b>18</b>
<b>3.2</b>	<b>Ferramentas de Gerenciamento de Projeto</b>	<b>18</b>
<b>3.2.1</b>	<b>Git/GitLab</b>	<b>18</b>
<b>3.2.2</b>	<b>Scrum</b>	<b>19</b>
<b>3.2.3</b>	<b>Kanban</b>	<b>21</b>
<b>3.2.4</b>	<b>Clickup</b>	<b>22</b>
<b>3.2.5</b>	<b>Gather</b>	<b>23</b>
<b>3.3</b>	<b>Definições</b>	<b>24</b>
<b>3.3.1</b>	<b>Arquitetura MVC</b>	<b>24</b>
<b>3.3.2</b>	<b>CSRF</b>	<b>25</b>

<b>4</b>	<b>Atividades desenvolvidas</b>	<b>26</b>
<b>4.1</b>	<b>Configuração do Ambiente</b>	<b>26</b>
<b>4.1.1</b>	<b>Instalação do Laravel Sail</b>	<b>27</b>
<b>4.1.2</b>	<b>Gerenciamento dos Pacotes</b>	<b>27</b>
<b>4.1.3</b>	<b>Configuração do Banco de Dados</b>	<b>27</b>
<b>4.2</b>	<b>Desenvolvimento do projeto</b>	<b>28</b>
<b>4.2.1</b>	<b>Sistema de Autenticação e Autorização</b>	<b>29</b>
<b>4.2.2</b>	<b>Criação de Telas</b>	<b>31</b>
<b>4.2.3</b>	<b>Sistema de Pagamento</b>	<b>35</b>
<b>4.2.4</b>	<b>Geração do Pagamento</b>	<b>35</b>
<b>4.2.5</b>	<b>Processamento do Pagamento</b>	<b>36</b>
<b>4.2.6</b>	<b>Geração dos Ingressos</b>	<b>38</b>
<b>4.3</b>	<b>Vivência com Metodologias Ágeis</b>	<b>38</b>
<b>4.3.1</b>	<b>Scrum</b>	<b>39</b>
<b>4.3.2</b>	<b>Kanban</b>	<b>40</b>
<b>4.4</b>	<b>Relação entre teoria e prática</b>	<b>41</b>
<b>5</b>	<b>Considerações Finais</b>	<b>43</b>
	<b>REFERÊNCIAS</b>	<b>45</b>



## 1 INTRODUÇÃO

Este documento apresenta as experiências vivenciadas pelo aluno durante o período de estágio na GAFIT Soluções em Automação, empresa sediada em Belo Horizonte, Minas Gerais, e que também conta com uma base em São Paulo, capital. O estágio, ocorreu entre 26/03/2022 a 01/03/2023 com duração de 30 horas semanais, totalizando 1044 horas. A atuação foi como desenvolvedor *full-stack*, que é um desenvolvedor que atua tanto na camada de apresentação, que consiste no desenvolvimento de telas, interações e todo aparato que exibe uma interface para o usuário final de um sistema, quanto na camada de negócio, que contempla a lógica da aplicação, seus comportamentos de entrada e saída, rotinas, tarefas em segundo plano.

A empresa oferta serviços voltados às plataformas Web e *mobile*, ambas experienciadas pelo estagiário durante o período. Para a plataforma Web, um sistema para o Estádio Governador Magalhães Pinto, comercialmente conhecido como Mineirão, dividindo o sistema em duas partes, uma com acesso para o módulo administrativo mediante a autenticação e autorização, e outro, camada pública com exibição de eventos, compra de ingressos e notícias. O sistema foi criado utilizando o *framework* PHP chamado *Laravel*. A organização também potencializou as experiências adquiridas proporcionando ao estagiário a oportunidade de participar de todos os processos que a empresa vivia no seu dia a dia, como colaborar e aplicar as práticas das metodologias ágeis, reuniões com cliente para definições de requisitos, suporte técnico e até de experiências gerenciais como a liderança de projetos.

Toda *expertise* adquirida neste processo foi fundamental para alavancar a formação profissional e acadêmica do aluno, visto que, tanto com a prática durante o exercício do estágio, quanto na assimilação com a vida acadêmica foram essenciais e complementares entre si.

## 2 CONTEXTUALIZAÇÃO DA EMPRESA E MERCADO ATUANTE

Este capítulo tem como objetivo introduzir o contexto da empresa onde o estágio foi realizado, trazendo também informações sobre o histórico, produtos, nicho de mercado, pré e pós pandemia, entre outros.

### 2.1 GAFIT

A GAFIT Soluções em Automação, fundada em 2012, é uma empresa de desenvolvimento de software, Web e aplicativos para plataformas móveis (Android e iOS), que tem como objetivo, auxiliar na consolidação da ideia e no desenvolvimento da área de tecnologia da informação. Sediada em Belo Horizonte, localizada no centro da cidade, a empresa até 2019 tinha um endereço físico, que com a pandemia da Covid-19, a mesma teve que se adequar às normas de distanciamento, o que acarretou na adequação ao modelo *remote-first*. Este modelo consiste em a empresa operar e modificar seus processos de forma a atender suas necessidades remotamente. Apesar do fim da pandemia e das restrições sanitárias de distanciamento, a GAFIT se manteve no modelo *home-office*, o endereço físico foi dispensado e atualmente a organização aproveita da oportunidade de alocação de uma equipe distribuída por todo o país, tendo colaboradores, na região sudeste e nordeste do país. Recentemente se expandiu ao adquirir um endereço na capital de São Paulo para impor presença nas ferramentas de busca na região paulista, onde se concentra o maior número de *startups* do país, sendo um ponto estratégico de negócio.

### 2.2 Mercado

A GAFIT conta com um portfólio de clientes extenso, muitos não exibidos na página de clientes da empresa (Figura 2.1), mas ainda sim alguns destaques como Prêmio Journalism Awards nos EUA como melhor site informativo da campanha presidencial de 2016 no projeto Campaign 2016 e Homologação do sistema da Secretaria de Turismo do Governo de MG para uso nacional. Porém, ao longo dos anos, a empresa teve grandes casos de sucesso, com clientes na América do Norte, Europa, Ásia e Oceania (GAFIT, 2023).

A organização também tem seu modelo de implementação da execução de um projeto, operando com metodologia própria, baseada nas metodologias ágeis do *Scrum* e *Kanban*. Este método relaciona o software ao crescimento em espiral. Isso significa que o software é cons-

tuído por meio de formas de ciclos (chamadas Sprints) e, a cada ciclo, uma parte executável do software é entregue. Na Gafit, o ciclo é dividido em quatro partes: inicia-se com a apresentação da *Sprint*, apresentando o escopo do que será entregue ao cliente no ciclo de desenvolvimento e definição de sub-tarefas a serem divididas e passadas para a equipe de desenvolvimento. Na sequência, o segundo passo é a criação do *Backlog*, com a quantificação da complexidade e tempo de desenvolvimento das sub-tarefas (chamadas estórias) junto à equipe e definição do prazo de entrega da *Sprint*. Com o *Backlog* definido, no passo três cada sub-tarefa possui um desenvolvedor responsável que atua e atualiza o líder do projeto diariamente com o seu relatório de atividade realizada. Por fim, teste e entrega das sub-tarefas (estórias) e finalização da *Sprint* (parte entregável do software). Após este ciclo, o Passo 1 é iniciado novamente até que o software tenha sido entregue completamente (Figura 2.2).

Figura 2.1 – Mapa de Atuação

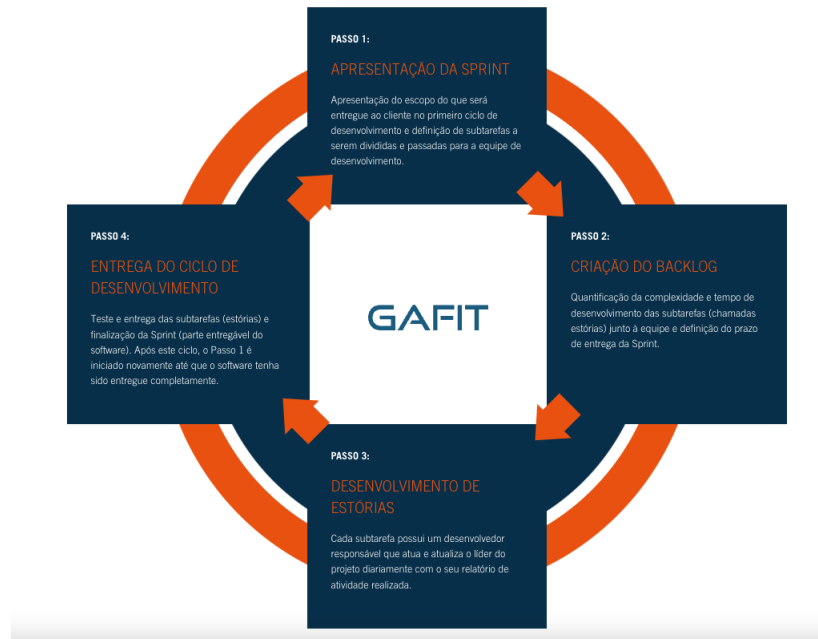


Fonte: (GAFIT, 2023)

### 2.3 Recrutamento e Treinamento

As vagas ofertadas pela empresa surgem em conjunto com novos projetos ou contratos de prestação de serviço. A quantidade de vagas e cargos são definidos de acordo com a necessidade de cada projeto e na sequência são divulgadas. No caso do estagiário, o processo seletivo se iniciou através de uma divulgação entre os grupos de *WhatsApp* da universidade. Em torno de cinco vagas foram disponibilizadas com alguns requisitos básicos para aplicação. Mesmo a vaga sendo para estágio, que comumente não se espera experiência técnica, conhecimento em

Figura 2.2 – Metodologia GAFIT



Fonte: (GAFIT, 2023)

programação orientada à objetos foi exigido como requisito obrigatório e conhecimentos sobre PHP e Laravel como requisito desejável.

Apesar dos requisitos, o processo seletivo foi de pouca dificuldade para o estagiário, não somente pelo fato de já ter experiências prévias no desenvolvimento de aplicações Web, mas também a habilidade comunicativa foram pontos fundamentais para o sucesso. O processo seletivo inicialmente exige o envio do currículo para o e-mail de recursos humanos da empresa. Com a avaliação do recrutador, foi necessário apenas uma reunião agendada para realizar a entrevista técnica que se encerra com o *feedback* do recrutador seguido da proposta de valores e benefícios.

As primeira semanas do estágio são direcionadas somente a treinamentos técnicos e dos processos da empresa. Os entrantes recebem conteúdo de estudo disponibilizados em formato de vídeo aulas sobre HTML, CSS, JavaScript, PHP, Laravel e SQL nas plataformas *YouTube*<sup>1</sup> e *Alura*<sup>2</sup>. Com o treinamento das tecnologias finalizado, em seguida o estagiário recebeu treinamento focado na compreensão do funcionamento do código base da empresa, que consiste em classes, componentes, funções e serviços que são utilizados para atividades de desenvolvimento.

<sup>1</sup> <https://www.youtube.com.br/>

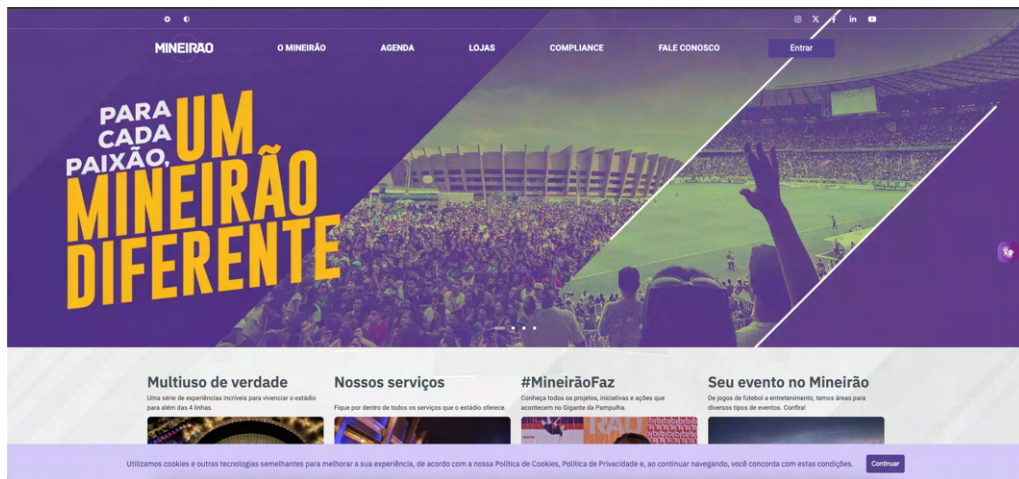
<sup>2</sup> <https://www.alura.com.br/>

## 2.4 Participações em Projetos

Com o processo de treinamento concluído, ao todo, o estagiário participou de dois projetos, ambos desde a concepção e em um deles até a entrega final. Houve participações rápidas em outros projetos, mas apenas tarefas simples ou correções esporádicas.

No projeto para o Mineirão, o estagiário atuou no desenvolvimento somente para a plataforma Web. O projeto foi dividido em duas partes, uma delas sendo a parte administrativa, que confere funcionalidades das mais diversas necessidades que o estádio tem, como questões jurídicas, envolvendo termos desde a Lei Geral de Proteção de Dados até a confecção de termos para cada partida de futebol sediada no estádio, administração de eventos, gestão de usuários administradores e clientes, setor financeiro, acessos ao estádio, partidas esportivas, postagem de notícias, entre outras atividades. Para a outra parte do projeto, contemplando a parte pública (Figura 2.3), é apresentada uma interface direcionada ao cliente final, exibindo os eventos agendados como shows e partidas de futebol, com a possibilidade de compra de ingressos, acesso às notícias e agenda dos eventos públicos e gratuitos que o estádio proporciona.

Figura 2.3 – Mineirão Público



Fonte: O Mineirão

### 3 TECNOLOGIAS UTILIZADAS

Neste capítulo são apresentadas as tecnologias utilizadas durante o período de estágio na empresa GAFIT.

#### 3.1 Ferramentas de Desenvolvimento

##### 3.1.1 HTML/CSS<sup>1</sup>

O Hypertext Markup Language (HTML) é uma linguagem de marcação simples e versátil, utilizada para criar documentos hipertexto que não dependem de uma plataforma específica. Documentos HTML são, na verdade, documentos SGML (Standard Generalized Markup Language) com semântica genérica que podem representar uma ampla variedade de informações de diferentes domínios. O HTML é capaz de representar notícias em formato hipertexto, mensagens de e-mail, documentação, multimídia hipertextual, menus de opções, resultados de consultas de banco de dados e documentos estruturados com gráficos incorporados. Além disso, ele permite a criação de visualizações hipertextuais de informações existentes. Desde 1990, o HTML tem sido fundamental na iniciativa global de informação da World Wide Web (WWW), e sua especificação abrange as capacidades em uso comum até junho de 1994, (W3C, 1995).

CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) é uma linguagem de marcação destinada à estilização usada para descrever a apresentação de um documento escrito em HTML, e XML. O CSS descreve como os elementos são mostrados na tela, na fala, no papel ou em outras mídias, (FOUNDATION, 2023b).

O HTML e CSS auxiliaram na criação das telas, modificação de fontes, definição de cores, além de utilização e customização de elementos pré-definidos.

##### 3.1.2 JavaScript<sup>2</sup>

JavaScript, ocasionalmente abreviado como JS, é uma linguagem de programação leve, interpretada, que se baseia em objetos. Ela é mais amplamente reconhecida como a linguagem de script usada em páginas da Web, mas também é aplicada em diversos outros contextos além do navegador, como no ambiente Node.js e Adobe Acrobat. O JavaScript é uma linguagem versátil, sendo baseada em protótipos, adotando vários paradigmas e oferecendo suporte a dife-

<sup>1</sup> <https://developer.mozilla.org/pt-BR/docs/Web/>

<sup>2</sup> <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

rentes estilos de programação, incluindo orientação a objetos, imperativa e declarativa, como a programação funcional (FOUNDATION, 2023c).

O JavaScript foi utilizado para criar e controlar as interações da interface gráfica da aplicação.

### 3.1.3 PHP<sup>3</sup>

O PHP é uma linguagem de *script*, multipropósito, interpretada, extensamente utilizada no desenvolvimento para a Web, e pode ser integrada facilmente ao código HTML. A sintaxe da linguagem lembra C, Java e Perl. Embora seu foco principal seja permitir que desenvolvedores criem páginas geradas dinamicamente de forma eficiente, o PHP oferece possibilidades muito além desse propósito inicial (GROUP, 2023).

O PHP é *OpenSource* e gratuito, foi desenvolvida para a Web, fazendo com que sua integração com os mais diversos servidores seja simples, facilitando a sua manutenção e diminuindo barreiras de entradas de novos produtos e serviços no mercado. Outro ponto é que a linguagem tem curva de aprendizado suave, comparada com outras linguagens. Isso possibilita que muitas pessoas aprendam o básico e consigam por em prática projetos simples sem custos. Estes projetos podem evoluir por sua vez, sem que a linguagem seja um ponto limitante. A linguagem também abstrai muitas das rotinas mais comuns da comunicação entre clientes (navegadores) e servidores, deixando o caminho livre para que o desenvolvedor possa se preocupar somente em focar na lógica do que está sendo construído e não em como processar uma requisição HTTP (BENTO, 2021).

### 3.1.4 Laravel<sup>4</sup>

“Em engenharia e programação de software, uma *framework* é uma coleção de componentes de software reutilizáveis que tornam mais eficiente o desenvolvimento de novas aplicações” (AMAZON, 2023b). De acordo com a própria documentação (LARAVEL, 2022), o Laravel é um *framework* feito para aplicações Web com sintaxe expressiva e elegante, fornecendo uma estrutura e um ponto de partida para a criação da aplicação. O *framework* tem como foco oferecer uma experiência agradável para o desenvolvedor ao mesmo tempo em que oferece

---

<sup>3</sup> <https://php.net>

<sup>4</sup> <https://laravel.com/>

recursos como injeção completa de dependência, uma camada expressiva de abstração de banco de dados, filas e trabalhos agendados, testes unitários e de integração, entre outros.

Segundo (STAUFFER, 2023) o Laravel é um *framework* de desenvolvimento de aplicativos rápidos. Isso significa que ele se concentra em uma curva de aprendizado rasa (fácil) e em minimizar as etapas entre o início de uma nova aplicação e sua publicação. Todas as tarefas mais comuns na construção de aplicativos da web, desde interações com banco de dados até autenticação, filas, e-mail e armazenamento em cache, são simplificadas pelos componentes que o Laravel oferece. No entanto, os componentes do Laravel não são ótimos apenas por si mesmos; eles fornecem uma API consistente e estruturas previsíveis em todo o *framework*.

### 3.1.5 Docker<sup>5</sup>

(DOCKER, 2023) é uma plataforma aberta para desenvolver, distribuir e executar aplicações. O Docker permite separar a camada de aplicação da infraestrutura, permitindo publicar o software rapidamente. O Docker pode gerenciar a infraestrutura da mesma maneira que um desenvolvedor gerencia sua aplicação. Ao aproveitar as metodologias do Docker para distribuir, testar e implantar código, é possível reduzir significativamente o intervalo entre escrever código e executá-lo em produção.

O Docker oferece a capacidade de empacotar e executar um aplicativo em um ambiente isolado e flexível chamado de contêiner. O isolamento e a segurança permitem executar vários contêineres simultaneamente em um mesmo servidor. Os contêineres são leves e contêm tudo o que é necessário para executar o aplicativo, eliminando a necessidade de depender do que está instalado no servidor. É possível compartilhar contêineres enquanto trabalha e garantir que todos com quem compartilha recebam o mesmo contêiner que funciona da mesma maneira (DOCKER, 2023).

### 3.1.6 Laravel Sail<sup>6</sup>

O *Laravel Sail* utiliza de contêineres Docker para comportar o ambiente de desenvolvimento. Este, é uma interface de linha de comando destinada a interagir com o ambiente de desenvolvimento Docker, padrão do Laravel. O Sail oferece um ponto de partida para a

---

<sup>5</sup> <https://www.docker.com/>

<sup>6</sup> <https://laravel.com/docs/9.x#docker-installation-using-sail>



construção de uma aplicação Laravel utilizando PHP, MySQL e Redis, sem a necessidade de experiência prévia com Docker (LARAVEL, 2023c).

Esta ferramenta surgiu como uma grande oportunidade para evolução dos ambientes de projetos que utilizavam o *Vagrant Homestead* como solução de ambientação e containerização dos projetos da empresa.

### 3.1.7 MySQL<sup>7</sup>

Um banco de dados é uma coleção estruturada de dados, desde uma simples lista de compras até uma galeria de imagens ou as vastas quantidades de informações em uma rede corporativa. Uma forma para adicionar, acessar e processar dados armazenados em um banco de dados de computador, é utilizando um sistema de gerenciamento de banco de dados, como o MySQL. Como os computadores são muito bons em lidar com grandes quantidades de dados, os sistemas de gerenciamento de banco de dados desempenham um papel central na computação, seja como utilitários independentes ou como parte de outras aplicações (MYSQL, 2023).

O MySQL é um sistema gerenciador de banco de dados que é distribuído e mantido pela Oracle Corporation. O MySQL é o sistema de código aberto mais amplamente utilizado, ele oferece um servidor de banco de dados SQL (Structured Query Language) rápido, multithreaded, multiusuário e robusto. O MySQL é projetado para sistemas de produção de carga pesada e crítica, bem como para ser incorporado em software de implantação em massa (MYSQL, 2023).

O MySQL foi utilizado para persistir, dividir e organizar os dados da aplicação.

### 3.1.8 Webpack<sup>8</sup>

Webpack (2023) é um agrupador de módulos cujo principal objetivo é empacotar arquivos JavaScript para uso em um navegador, mas também é capaz de transformar, agrupar ou empacotar praticamente qualquer recurso ou ativo. A ferramenta pode criar um único pacote ou vários pedaços que são carregados de forma assíncrona em tempo de execução a fim de reduzir o tempo de carregamento inicial. As dependências são resolvidas durante a compilação, reduzindo o tamanho em tempo de execução.

---

<sup>7</sup> <https://www.mysql.com/>

<sup>8</sup> <https://webpack.js.org/>

O Webpack foi fundamental para reduzir o tempo de carregamento das páginas do projeto, pois a grande maioria das telas contam com inúmeros arquivos JavaScript e imagens responsáveis pela estilização das páginas.

### 3.1.9 EloquentORM<sup>9</sup>

Segundo Prisma (2023), ORM ou Mapeador Objeto-Relacional, é um software projetado para traduzir entre as representações de dados usadas por bancos de dados e aquelas usadas em programação orientada a objetos. Usando o EloquentORM, cada tabela do banco de dados é representada por uma “*Model*” que é a forma de interagir com a camada de dados no Laravel, facilitando criação de relacionamentos e consultas (LARAVEL, 2022).

O uso do EloquentORM facilitou a confecção de consultas, que por muitas vezes, exigia cálculos, funções e junções entre tabelas. A ferramenta entrega uma sintaxe efetiva e de baixa verbosidade para criação de consultas.

### 3.1.10 Laravel Blade<sup>10</sup>

O Blade utiliza da construção do HTML unido à sintaxe PHP. É o motor de *template* incluído no Laravel. Ao contrário de alguns motores de *template* em PHP, o Blade não impede o uso de código PHP puro em seus *templates*. Na verdade, todos os *templates* Blade são compilados em código PHP puro e armazenados em cache até que sejam modificados, o que significa que o Blade adiciona essencialmente nenhum *overhead* à sua aplicação (LARAVEL, 2022).

O Blade auxiliou na componentização e reuso de componentes HTML, além de reduzir a verbosidade implicada por linguagens declarativas, a ferramenta expõe funcionalidades integradas com a camada de dados, fornecendo uma interação mais rica para o usuário.

### 3.1.11 jQuery<sup>11</sup>

De acordo com a própria documentação (FOUNDATION, 2023d), o jQuery é uma biblioteca JavaScript ágil, compacta e repleta de recursos. Simplifica tarefas como a manipulação e travessia de documentos HTML, o gerenciamento de eventos, animações e o uso de Ajax, oferecendo uma API fácil de usar que funciona em uma ampla variedade de navegadores. Com sua

---

<sup>9</sup> <https://laravel.com/docs/9.x/eloquent>

<sup>10</sup> <https://laravel.com/docs/9.x/blade>

<sup>11</sup> <https://jquery.com/>

versatilidade e capacidade de extensão, jQuery revolucionou a maneira como inúmeras pessoas escrevem JavaScript.

Apesar do JavaScript moderno oferecer as mesmas ferramentas de manipulação que o jQuery entrega, ainda assim, a biblioteca conta com uma comunidade muito grande, oferecendo alternativas rápidas e práticas para o desenvolvimento da interface da aplicação, como por exemplo a geração de tabelas dinâmicas, muito usadas nas telas de relatório dos projetos da empresa, a biblioteca *jquery-datatable* por exemplo, que será melhor explicada nos capítulos adiante.

### 3.1.12 Bootstrap<sup>12</sup>

O Bootstrap tem como slogan, "Desenvolva sites rápidos e responsivos". Um conjunto de ferramentas poderosa e repleta de recursos para o *frontend* (TEAM, 2023).

O Bootstrap é um framework de *frontend* de código aberto e gratuito que oferece uma ampla gama de componentes prontos para serem incorporados aos seus projetos. Ele facilita a criação e personalização de sites responsivos, adaptáveis à dispositivos móveis, desktops e laptops, usando componentes pré-projetados e plugins JavaScript robustos. Isso é possível porque todo o esqueleto do HTML, CSS e JavaScript já está pronto, exigindo apenas que os desenvolvedores importem essas dependências em seus próprios projetos (GUEDES, 2020).

A ferramenta foi fundamental para a customização dos estilos da interface de usuário e responsividade de telas entre as diversas dimensões dos dispositivos do usuário final.

### 3.1.13 PhpStorm<sup>13</sup>

O PhpStorm é um ambiente integrado de desenvolvimento, voltado ao PHP que realmente captura seu código. Tem suporte para PHP da versão 5.3 até 8.1, oferece prevenção de erros em tempo real, complementação automática e refatoração de código, depuração com configuração zero e um editor estendido de HTML, CSS e JavaScript, além de amparar todo o processo rotineiro de um desenvolvedor, com apoio que trespassa o código escrito (JETBRAINS, 2023).

O PhpStorm como ambiente de desenvolvimento integrado, fornece diversas ferramentas de apoio ao desenvolvimento de aplicações PHP.

---

<sup>12</sup> <https://getbootstrap.com/>

<sup>13</sup> <https://www.jetbrains.com/pt-br/phpstorm/features/>

### 3.1.14 WSL2<sup>14</sup>

Segundo (AWARI, 2023), o Windows Subsystem for Linux 2 (WSL2) é uma ferramenta desenvolvida pela Microsoft que possibilita que usuários do Windows executem aplicativos e comandos Linux em um ambiente nativamente Windows. Com o WSL2, desenvolvedores podem desfrutar de uma experiência de desenvolvimento mais suave e eficaz, unindo as vantagens dos sistemas operacionais em questão. Esta tecnologia representa uma significativa melhoria em relação a sua versão anterior, trazendo melhorias substanciais em termos de desempenho, confiabilidade e funcionalidade.

Ao adotar o WSL2, desenvolvedores podem tirar proveito do ecossistema Linux e do Windows de forma simultânea. Isso significa que os usuários podem acessar aplicativos e ferramentas do Linux diretamente em seus sistemas operacionais Windows, eliminando a necessidade de instalar máquinas virtuais ou sistemas operacionais adicionais. O WSL2 é especialmente valioso para desenvolvedores que precisam trabalhar com ambientes de desenvolvimento integrado e ferramentas de linha de comando em projetos de programação (AWARI, 2023).

## 3.2 Ferramentas de Gerenciamento de Projeto

### 3.2.1 Git<sup>15</sup>/GitLab<sup>16</sup>

Desenvolvimento colaborativo é fundamental para o sucesso de projetos de software. Isso se deve graças aos sistemas de controle de versão, os quais permitem o trabalho simultâneo de desenvolvedores nas equipes de projetos. Um dos mais populares é o Git, que permite que os desenvolvedores compartilhem código de uma forma gerenciada sem fazer uso de um repositório principal (CUNHA, 2018).

Segundo Silverman (2013), Git é uma ferramenta para acompanhar as alterações feitas em um conjunto de arquivos ao longo do tempo, uma tarefa tradicionalmente conhecida como 'controle de versão'. Embora seja mais frequentemente usado por programadores para coordenar as mudanças no código-fonte de software, é especialmente eficaz nisso, rastrear arquivos de texto. Qualquer conjunto de arquivos relacionados que evolua ao longo do tempo, pode se beneficiar do uso do Git.

---

<sup>14</sup> <https://learn.microsoft.com/pt-br/windows/wsl/>

<sup>15</sup> <https://git-scm.com/>

<sup>16</sup> <https://about.gitlab.com/>

GitLab é uma plataforma Web de hospedagem de código e rastreamento de problemas baseada no sistema de controle de versão Git. Lançado pela primeira vez em 2011, continua a crescer e evoluir ao longo dos anos, adicionando novos recursos e capacidades, se tornando uma ferramenta completa para uma equipe de trabalho ágil. Embora seja de propriedade e gerenciamento da GitLab Inc., que direciona o rumo do projeto, o núcleo do GitLab é um software de código aberto, contando com mais de 2.000 colaboradores individuais até o momento (O'GRADY, 2018).

O Git foi utilizado juntamente com o GitLab para versionar o projeto, além de atividades voltadas a integração e entrega contínua.

### 3.2.2 Scrum<sup>17</sup>

O Scrum é um framework de gerenciamento que as equipes aplicam nos processos rotineiros de uma organização com o intuito de se auto-organizar e trabalhar em direção a um objetivo em comum. Extensamente utilizado no contexto do desenvolvimento de software, porém não se limita a este contexto. A estrutura descreve uma remodelagem nos processos trazendo um conjunto de ferramentas, processos e funções para uma entrega ágil e eficiente de projetos (AMAZON, 2023a). A essência do Scrum é uma equipe auto-gerenciada que produz entregas de valor ao cliente e partes interessadas num período de tempo definido chamado *Sprint*. O Scrum define artefatos, funções e eventos associados a cada Sprint. Abaixo são apresentados cada um em detalhes (AMAZON, 2023a).

As equipes usam de aparatos que apoiam o processo Scrum para lidar com as adversidades encontradas no projeto e gerenciar de forma dinâmica e ágil, também em busca de um maior controle da execução do projeto.

“Há três artefatos principais:

- **Product backlog**  
O backlog de produto é uma lista em constante evolução que contém recursos, requisitos, melhorias e correções necessários para o sucesso do projeto. Basicamente, corresponde à lista de tarefas da equipe, que é continuamente revista e reavaliada em termos de prioridade para se adaptar às mudanças nas necessidades do cliente. O responsável pelo produto é responsável por manter e atualizar essa lista, retirando itens que não são mais relevantes e incluindo novas solicitações dos clientes.
- **Sprint backlog**  
O backlog da Sprint é a lista de itens que devem ser concluídos pela equipe de desenvolvimento durante o ciclo atual da Sprint. Antes de iniciar cada Sprint,

---

<sup>17</sup> <https://www.scrum.org/>

a equipe decide quais itens do backlog de produto serão trabalhados. O backlog da Sprint é flexível e pode sofrer alterações ao longo da Sprint, conforme necessário.

- **Incremento**

O incremento é um passo em direção a uma meta ou visão. Representa o produto final utilizável que é produzido ao final de uma Sprint. As equipes podem adotar diferentes métodos para definir e apresentar suas metas para a Sprint. No entanto, a meta central da Sprint, ou seja, o que a equipe pretende alcançar na Sprint atual, não pode ser comprometida.

A metodologia Scrum é caracterizada por certos princípios e valores:

- **Transparência**

Em um ambiente de trabalho, a consciência dos desafios enfrentados pelos outros membros da equipe é promovida. Conversas regulares são conduzidas entre os membros da equipe multifuncional e os responsáveis pelo projeto, com o objetivo de evitar falhas de comunicação e a ocorrência de gargalos de informações.

- **Reflexão**

São incorporados pontos de reflexão frequentes ao framework, permitindo que o próprio progresso seja revisado pelos membros da equipe. Os *insights* provenientes dessas reuniões de revisão são utilizados pelos gerentes de projeto para fazer estimativas e planejar o futuro. Como resultado, os projetos podem ser executados de maneira mais eficiente, dentro do orçamento e dentro do prazo.

- **Adaptação**

A priorização das tarefas pode ser redefinida pelos membros da equipe conforme as mudanças nos requisitos do cliente. Eles têm a autonomia de decidir quais tarefas serão concluídas primeiro e quais serão revisitadas em momentos futuros.

A equipe Scrum precisa de três funções específicas: um responsável pelo produto, um líder Scrum e uma equipe de desenvolvimento.

- **Product Owner (PO)**

A ênfase é colocada na garantia de que o máximo valor seja entregue aos negócios pela equipe de desenvolvimento, com uma compreensão e priorização das necessidades em constante evolução dos usuários finais e clientes. A equipe de desenvolvimento é orientada de forma clara sobre quais recursos devem ser entregues a seguir, servindo como uma ponte entre as expectativas da empresa e a compreensão da equipe, e determinando a frequência das entregas.

- **Líder Scrum**

A representação do Scrum nas equipes é assegurada pelos líderes Scrum, que têm a responsabilidade de otimizar a eficácia da equipe Scrum. Eles orientam as equipes, os responsáveis pelo produto e a empresa na melhoria dos processos Scrum. Além disso, os líderes Scrum planejam os recursos necessários para cada Sprint, promovem eventos e reuniões da equipe, lideram a transformação digital na equipe, facilitam o treinamento da equipe em novas tecnologias e lidam com desafios externos que possam afetar a equipe como um todo.

- **Development Team**

A equipe Scrum é composta por testadores, designers, especialistas em experiência do usuário, engenheiros de operações e desenvolvedores. Eles possuem habilidades diversas e colaboram entre si para garantir a conclusão eficiente do trabalho. A equipe de desenvolvimento Scrum trabalha de forma auto-organizada, defende práticas de desenvolvimento sustentável e orienta o planejamento e a estimativa da quantidade de trabalho que podem concluir em cada Sprint, demonstrando um forte senso de coletividade”(AMAZON, 2023a).

O Scrum é executado através de ritos ou cerimônias que consistem em um conjunto de reuniões sequenciais que as equipes Scrum realizam regularmente. Alguns eventos Scrum incluem:

- ***Sprint Planning***  
Nesse evento, a equipe estima o trabalho a ser concluído no próximo Sprint. Os membros definem metas de Sprint que sejam específicas, mensuráveis e atingíveis. Ao final da reunião de planejamento, cada membro Scrum sabe como cada incremento pode ser apresentado no Sprint.
- ***Sprint***  
O Sprint é o período de tempo real em que a equipe Scrum trabalhará em conjunto para concluir um incremento. Normalmente, um Sprint dura duas semanas, mas pode variar de acordo com as necessidades do projeto e da equipe. Quanto mais complexo for o trabalho e mais incógnitas houver, mais curto deve ser o Sprint.
- ***Daily***  
A *Daily* é uma reunião curta em que os membros da equipe se apresentam e planejam o dia. Eles relatam o trabalho concluído e expressam os desafios no cumprimento das metas do Sprint. Chama-se stand-up porque o objetivo é fazer a reunião o mais breve possível, como quando todos estão de pé.
- ***Sprint Review***  
Ao final do Sprint, a equipe se reúne para uma sessão informal para revisar o trabalho concluído e apresentá-lo às partes interessadas. O responsável pelo produto também pode retrabalhar o product backlog com base no Sprint atual.
- ***Sprint Retrospective***  
A equipe se reúne para documentar e discutir o que funcionou e o que não funcionou durante o Sprint. As ideias geradas são usadas para melhorar Sprints futuros (AMAZON, 2023a).

A empresa aplicou modificações na execução do Scrum para adequar a sua realidade e contexto inserido, estas modificações são explicadas na seção 4.3.1.

### 3.2.3 Kanban<sup>18</sup>

De acordo com (UNIVERSITY, 2021), utilizar o método Kanban implica adotar uma abordagem abrangente para analisar os serviços da organização, com o objetivo de aprimorá-los tendo foco principal a perspectiva dos clientes. Com o Método Kanban, existe um ganho de visibilidade sobre o trabalho de conhecimento que normalmente fica oculto e como ele flui através de um processo de trabalho. Isso é fundamental para identificar, operar e monitorar atividades do negócio de maneira eficiente, o que inclui compreender e gerenciar os riscos envolvidos na entrega de serviços aos clientes. Ao implementar o Kanban, a empresa deve desenvolver a capacidade de se adaptar ao longo do tempo, permitindo a resposta de maneira mais ágil e eficaz às mudanças nas necessidades e expectativas de seus clientes ou do próprio negócio. Esta metodologia é amplamente conhecida e utilizada dentro dos times, para aliviar a sobrecarga e administrar o trabalho feito. O Kanban é implementado com o ClickUp que fornece uma visão geral das tarefas escaladas para o desenvolvimento com seus respectivos estados definindo o andamento da tarefa. Além de expor de forma intuitiva as informações do

<sup>18</sup> <https://kanban.university/kanban-guide/>

estado atual do projeto como um todo, é possível também controlar e limitar as atividades em progresso de cada colaborador.

Segundo Yeret, Vacanti e Scrum.org (2021), o Scrum determina que o *Backlog* da Sprint seja transparente, mas provê instruções limitadas sobre como isso deve ser feito. Ele também não define como alcançar a transparência explícita do fluxo de trabalho no *Backlog* do Produto ou do fluxo do *Backlog* do Produto até o *Backlog* da Sprint e o que quer que aconteça com o trabalho depois que ele se torna um incremento “Pronto”. Ao visualizar o trabalho de novas formas, um Time Scrum pode aplicar o conjunto de práticas descritas neste guia para otimizar o valor da entrega de maneira mais efetiva. Essas práticas se apoiam e expandem os princípios do pensamento *Lean*, fluxo de desenvolvimento de produtos e teoria das filas.

### 3.2.4 ClickUp<sup>19</sup>

O ClickUp é uma ferramenta de apoio para a aplicação das metodologias ágeis. Segundo (GONÇALVES, 2019), o ClickUp aprimora os fluxos de trabalho e traz benefícios incalculáveis para a gestão de projetos, facilita o aumento da produtividade de uma equipe, sem a necessidade de investimentos significativos em infraestrutura física ou tecnológica. Além de se tornar uma ferramenta indispensável para a execução de projetos em geral, o ClickUp executa um papel crucial na geração de análise, relatórios entre outras informações gerais. Isso contribui para a constante melhoria das operações diárias, monitoramento da produtividade e colaboração nas decisões estratégicas. Apesar de ser um recurso pago, com um custo mensal acessível de R\$21 por pessoa, é simples e não representa um grande ônus financeiro para uma organização. Além disso, o envolvimento inicial de apenas 5 membros-chave da equipe não afeta negativamente o desempenho da produção. O ClickUp foi utilizado como ferramenta de apoio ao Scrum e Kanban, exibindo além da descrição, o andamento das tarefas com possibilidade de definição de prazos, prioridades e responsável pela entrega.

---

<sup>19</sup> <https://clickup.com/>



Figura 3.1 – Dashboard ClickUp

#	Supplier	Status	Due Date	Amount	Priority
<b>PAID</b>					
1	Acme Inc.	PAID	10/31/22	\$2,000	
2	Plan for next year	PAID	11/4/22	\$500	
<b>READY TO PAY</b>					
1	Coco Bongo Club	READY TO PAY	9/12/22	\$1500	
2	Ipkiss Electric	READY TO PAY	9/12/22	\$450	
<b>UNPAID</b>					
1	Wolf Cola	UNPAID	9/9/22	\$1000	
2	Invigaron	UNPAID	10/26/22	\$600	
2	Kitten Mittens	UNPAID	9/12/22	\$450	

Fonte: ClickUp

### 3.2.5 Gather<sup>20</sup>

O Gather<sup>21</sup> é uma plataforma inovadora de ambiente de trabalho virtual que permite aos usuários criar e personalizar escritórios virtuais para equipes remotas. A plataforma oferece uma ampla gama de recursos destinados a melhorar a produtividade, a colaboração e a comunicação em equipes dispersas geograficamente (GATHER, 2023).

Com o Gather, os usuários podem projetar seus próprios escritórios virtuais, customizar seus personagens, oferecendo uma imersão satisfatória ao usuário. A plataforma oferece espaços dedicados para reuniões, trabalho individual e colaboração, permitindo que as equipes se adaptem às suas preferências e estilos de trabalho (GATHER, 2023).

Uma característica distintiva do Gather é a capacidade de realizar conversas em tempo real em um ambiente visual, o que facilita a tomada de decisões rápidas e o desenvolvimento de ideias criativas. A integração com calendários, como o Outlook e o Google Agenda, simplifica o agendamento de reuniões, enquanto as conversas espontâneas são facilitadas pela interface fluida (GATHER, 2023). O Gather foi o espaço utilizado para a execução das reuniões e todo o ambiente de trabalho, de forma virtual. Possibilitando a execução dos processos internos

<sup>20</sup> <https://pt-br.gather.town/>

<sup>21</sup> <https://gather.town/>

do time sem a necessidade de um espaço físico, além de momentos de descontração que a plataforma proporciona.

Figura 3.2 – Gather Town



Fonte: (GATHER, 2023)

### 3.3 Definições

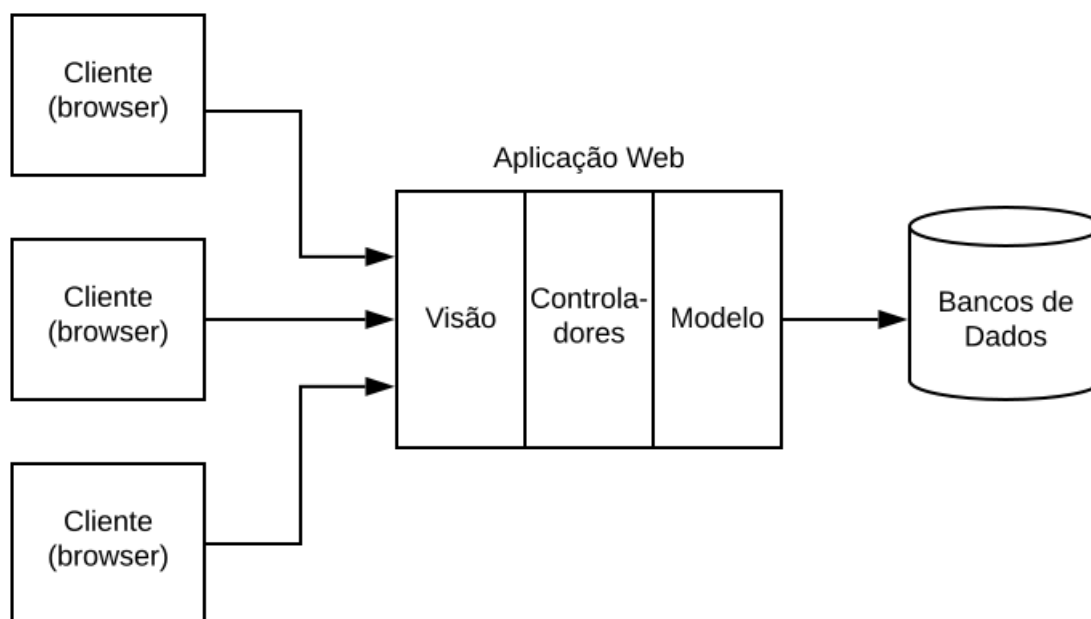
#### 3.3.1 Arquitetura MVC

Uma arquitetura que define que as classes de um sistema devem ser organizadas em três grupos (VALENTE, 2020):

- **Visão:** classes responsáveis pela apresentação da interface gráfica do sistema, incluindo janelas, botões, menus, barras de rolagem, entre outros.
- **Controladoras:** classes que tratam e interpretam eventos gerados por dispositivos de entrada, como mouse e teclado. Como resultado de tais eventos, Controladoras podem solicitar uma alteração no estado do Modelo ou da Visão.

- **Modelo:** classes que armazenam os dados manipulados pela aplicação e que têm a ver com o domínio do sistema em construção. Assim, classes de modelo não têm qualquer conhecimento ou dependência para classes de Visão e Controladoras. Além de dados, classes de modelo podem conter métodos que alteram o estado dos objetos de domínio.

Figura 3.3 – Arquitetura MVC Web



Fonte: (VALENTE, 2020)

### 3.3.2 CSRF

CSRF ou (Cross-site Request Forgery), que segundo a (IBM, 2023), consiste em um ataque que envia as informações submetidas de um usuário de um site seguro para o atacante. O ataque utiliza a confiança que um site tenha no navegador de um usuário autenticado para ataques maliciosos. O CSRF utiliza links ou *scripts* para enviar solicitações de HTTP involuntárias para um site de destino onde o usuário está autenticado. Segundo OWASP (2023), com auxílio de engenharia social (como enviar um link por e-mail ou chat), um atacante pode enganar os usuários de uma aplicação web para executar ações de escolha do atacante. Se a vítima for um usuário comum, um ataque bem-sucedido de CSRF pode forçar o usuário a realizar solicitações que alteram o estado, como transferir fundos, alterar o endereço de e-mail, entre outras coisas. Se a vítima for uma conta administrativa, o CSRF pode comprometer toda a aplicação web.

## 4 ATIVIDADES DESENVOLVIDAS

Nesse capítulo, são apresentadas algumas das atividades desenvolvidas durante o período de estágio na GAFIT Soluções em Automação. O projeto do Sistema Web do Estádio Governador Magalhães Pinto, o Mineirão, será o projeto relatado neste documento. Ao decorrer do mesmo, as atividades realizadas também serão relacionadas ao conteúdo visto em sala de aula.

### 4.1 Configuração do Ambiente

Após o período de treinamento, a empresa passava por dificuldades com a estabilidade do ambiente de desenvolvimento. Executado em sistemas operacionais *Windows* e para a execução de trabalhos de desenvolvimento foi utilizado o WSL, um recurso opcional disponibilizado em sistemas operacionais *Windows* utilizado para executar aplicações Linux de forma nativa. Os projetos desenvolvidos pela empresa são majoritariamente Laravel. A solução utilizada era o Vagrant<sup>1</sup> para a criação de ambientes virtualizados com o Laravel na versão 7, versão até então utilizada na GAFIT, oferece a solução *Homestead*<sup>2</sup>, que consiste em uma imagem Vagrant oficial e pré-configurada que oferece um ambiente de desenvolvimento completo sem exigir instalar o PHP, um servidor Web ou qualquer outro serviço replicado em uma máquina local, não sendo necessário se preocupar com conflitos no ambiente do sistema operacional. As instâncias Vagrant são completamente descartáveis. Em caso de qualquer eventualidade, é possível proceder com a eliminação e subsequente reconstrução da instância em questão de poucos minutos (LARAVEL, 2023a).

Dada a mudança para o Laravel 9, surge então a oportunidade de substituir o uso do Vagrant, e aplicar a solução do *Laravel Sail*. O que mais foi ponderado na justificativa da mudança, foi a simplicidade que o Docker oferecia, encorajado pela integração com o WSL2 e, visto a dificuldade e instabilidade experienciada com o Vagrant.

Sendo assim, e com a sugestão do estagiário, apoiado às vantagens o projeto foi criado utilizando o Laravel Sail.

---

<sup>1</sup> <https://www.vagrantup.com/>

<sup>2</sup> <https://laravel.com/docs/7.x/homestead>

### 4.1.1 Instalação do Laravel Sail

Para criar um projeto Laravel, por padrão o projeto já configurado com o Docker, a partir da versão 8. Para criar um projeto Laravel, basta ter o cURL<sup>3</sup> instalado em um terminal *bash* e executar o comando apresentado no Código 4.1.

Código 4.1 – Criação do projeto em Laravel

```
1 curl -s "https://laravel.build/example-app" | bash
```

Com o projeto instalado, foi necessário realizar a instalação das dependências que o projeto utiliza. Como o Laravel é um framework *full-stack* o mesmo gerencia tanto o *frontend* quanto o *backend*. Sendo assim, para o backend, as dependências do projeto utilizam o *Composer*<sup>4</sup>, um gerenciador de pacotes para o PHP. Para o frontend, utiliza-se o *npm*<sup>5</sup>, gerenciador de pacotes Node.js.

### 4.1.2 Gerenciamento dos Pacotes

Para gerenciar os pacotes, o Laravel, até as versões finais da versão 9, ainda utilizava o *Webpack*<sup>6</sup>, que trata-se de uma ferramenta de código aberto amplamente usada no desenvolvimento Web para empacotar *assets*, que consistem em arquivos JavaScript, CSS, e imagens, em um único arquivo ou em vários arquivos menores, de forma otimizada para implantação em um ambiente de produção.

Durante este processo, foi necessário para o estagiário declarar e definir os pacotes que seriam devidamente empacotados para a entrega minificada ao cliente. Minificação é o processo de redução do código e marcação em suas páginas da Web e arquivos de *script*. Este, é um dos principais métodos utilizados para diminuir os tempos de carregamento e o uso de largura de banda em sites. A minificação melhora drasticamente a velocidade do site e a acessibilidade, traduzindo diretamente em uma experiência do usuário aprimorada (IMPERVA, 2023).

### 4.1.3 Configuração do Banco de Dados

O banco de dados definido para o projeto foi o MySQL, que por padrão é o banco instalado nos projetos Laravel. Ao gerar o projeto, graças a configuração gerada automaticamente

---

<sup>3</sup> <https://curl.se/>

<sup>4</sup> <https://getcomposer.org/>

<sup>5</sup> <https://www.npmjs.com/>

<sup>6</sup> <https://webpack.js.org/>

no arquivo `docker-compose.yml`, arquivo este que contém algumas configurações para a geração do contêiner Docker e adiciona o MySQL como uma imagem pertencente ao contêiner do projeto.

Como ferramenta de interface para manipular o sistema gerenciador de banco de dados, o estagiário utilizou o MySQL Workbench<sup>7</sup>. Além de ter fácil instalação, o mesmo pertence a criadora e mantenedora do MySQL, entregando maior credibilidade.

## 4.2 Desenvolvimento do projeto

Após o processo de configuração e definição inicial do projeto, o mesmo passou por um período de adaptação e adequação para a inserção de uma interface customizada que utiliza de um *template* pago, específico para projetos Laravel. Um *template* consiste em customizações pré-definidas de estilizações, convenções, e outras ferramentas que auxiliam no desenvolvimento das telas do projeto. Além disso, o Laravel sendo um framework *full-stack* oferece outras ferramentas para amparar tanto no desenvolvimento das telas da aplicação, quanto na concepção lógica das regras de negócio e modelagem de dados.

O Laravel, por padrão utiliza da arquitetura MVC (Model-View-Controller), mas apesar do seu uso, o Laravel não evangeliza nem estimula a arquitetura, oferecendo a opção de implementação de qualquer outro padrão arquitetural. O projeto em questão utiliza do padrão MVC (Figura 3.3), porém com algumas peculiaridades, para organizar e distribuir melhor as atribuições de cada camada do modelo MVC. A convenção utilizada se chama “*Fat Model, Skinny Controller*”, que tem como objetivo, reduzir ao máximo qualquer tipo de ação, processamento ou computação na camada do controlador, este padrão leva toda a manipulação e processamento para camada de modelo. Esta convenção é oriunda de projetos *Ruby on Rails*, um *framework* para a linguagem *Ruby*, o *On Rails* se assemelha bastante com o Laravel, tanto por utilizar de Orientação a Objetos, MVC, e até mesmo pelas estruturas de pastas serem muito semelhantes. A convenção descrita acima raramente é aplicada em projetos Laravel. Em geral, até mesmo na grande maioria do conteúdo didático disponível para o framework, estimula-se a implementação do processamento na camada de controladores. Isso se deve pelo fato de que a própria arquitetura utilizada tem como boa prática manter as regras de negócio pertencentes a camada de Model e a camada Controller apenas com responsabilidades de transporte e comunicação entre camadas. O período de adaptação a modificação do MVC por parte do estagiário

---

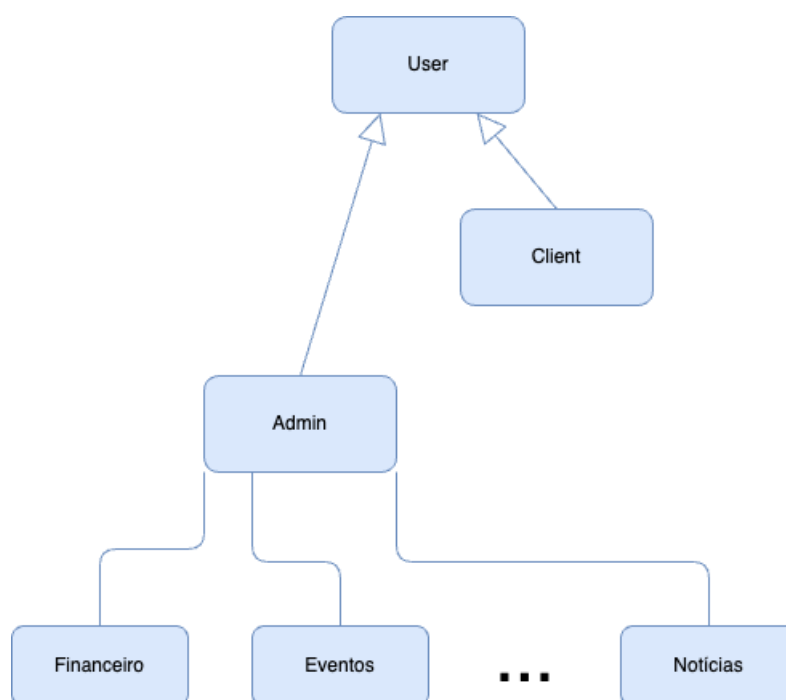
<sup>7</sup> <https://www.mysql.com/products/workbench/>

durou poucas semanas, visto que a modificação não adiciona camadas, apenas reorganiza as responsabilidades das camadas.

#### 4.2.1 Sistema de Autenticação e Autorização

Como o sistema contempla vários cenários possíveis para um usuário, o sistema de autenticação precisou de uma atenção maior no desenvolvimento. Um usuário neste sistema pode ser um colaborador do Mineirão ou um cliente final que esteja interessado nos produtos e serviços que o estádio oferece. Sendo assim, o estagiário necessitou desenvolver uma lógica para lidar com a autenticação e autorização de forma customizada e genérica. A (Figura 4.1) ilustra a hierarquia de especialização da configuração dos usuário. Todo usuário é um *User* que pode ser ou um administrador, ou um cliente. No caso dos administradores, existe uma camada a mais para definir qual é o tipo do usuário.

Figura 4.1 – Fluxograma da Hierarquia da Configuração de Usuários - Mineirão

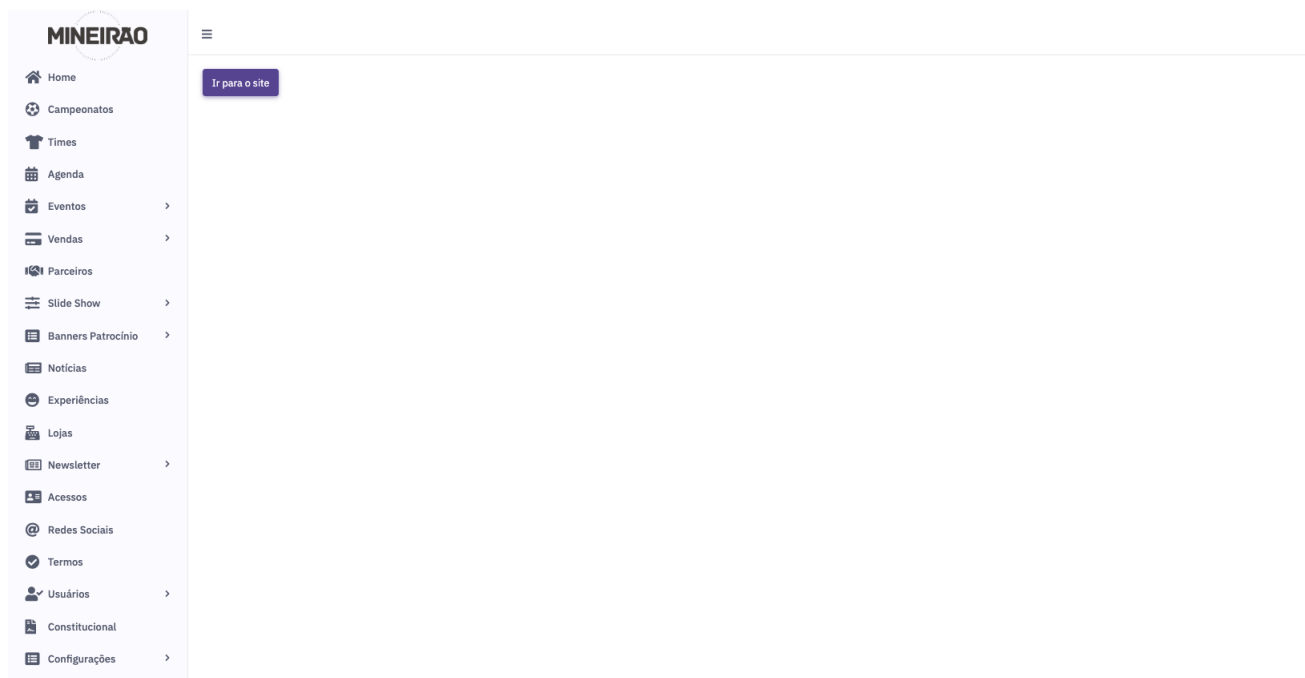


Fonte: Do autor (2023)

Partindo do pressuposto que os usuários administradores se diferem completamente dos usuários clientes em seus propósitos, foi necessário de início para o estagiário, separar ambas as vertentes. Para o usuário cliente, o acesso está limitado apenas às páginas públicas; já o usuário administrador, foi necessário um tratamento específico. Devido a grande diversidade de cargos e atribuições dentro das operações de um estádio de futebol, foi necessário restringir e limitar

o acesso administrativo em “classe” de permissão. O sistema todo foi particionado em áreas, separando cada seção do menu administrativo (Figura 4.2).

Figura 4.2 – Parte administrativa Mineirão



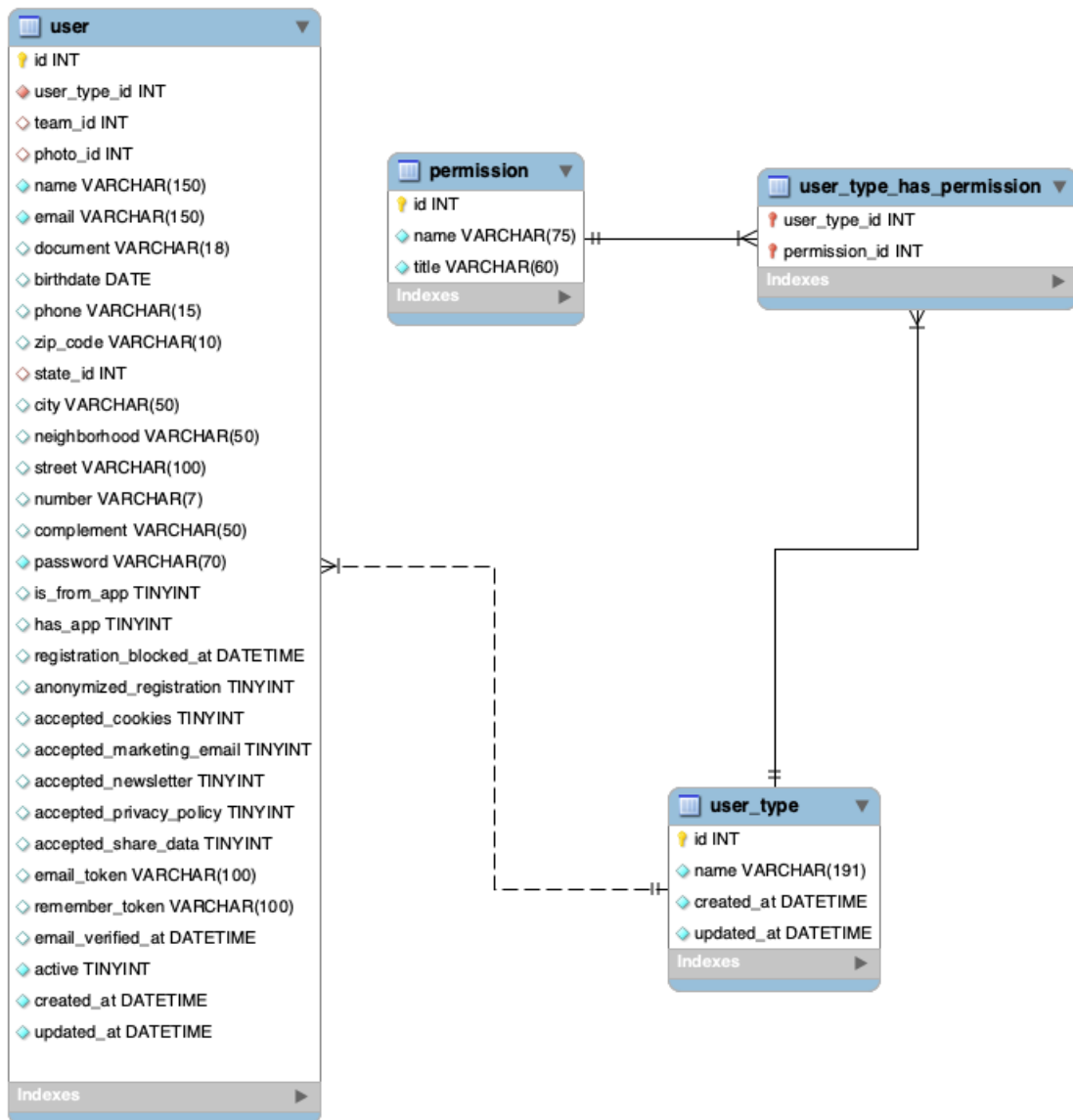
Fonte: O Mineirão (2022)

Para facilitar o controle e gestão dos usuários administrativos, as permissões de acesso às áreas, isto é, às páginas administrativas que o usuário poderia acessar, foram encapsuladas em perfis de usuário, para padronizar as permissões e autorizações que cada usuário tem, baseado em suas competências, além de facilitar o rastreamento e reconhecimento de um perfil de usuário através de um nome descritivo. Um perfil de usuário contém um conjunto de permissões que definem qual área do sistema tem o direito de acessar. Por sua vez, cada usuário administrador está vinculado a um perfil de usuário. Cada rota de acesso do sistema foi agrupada por área, onde cada área contém o código da permissão que refere a ela mesma, sendo assim, cada página contém o código da respectiva área.

Além da modelagem E-R de nível lógico (Figura 4.3), para auxiliar em todo o controle de permissões, o Laravel oferece um sistema de *middleware*, que oferece um mecanismo para inspecionar e filtrar as solicitações HTTP que entram na aplicação a fim de controlar o acesso e o comportamento da sua aplicação com base em determinadas condições (LARAVEL, 2023b). Unido à marcação das rotas, foi possível desenvolver um algoritmo capaz de discernir a permissão dos usuários. Toda a configuração de usuários e permissões se aplicam após a persistência do tipo de usuário e a associação do usuário ao tipo no banco de dados.



Figura 4.3 – Modelagem E-R de nível lógico de usuários e autenticação



Fonte: Do autor (2022)

Neste caso, o *middleware* desenvolvido pelo estagiário foi utilizado para validar se o usuário tem permissão de obter o resultado da requisição solicitada, verificando qual o grupo do usuário, e se este grupo contém a permissão referente à requisição.

#### 4.2.2 Criação de Telas

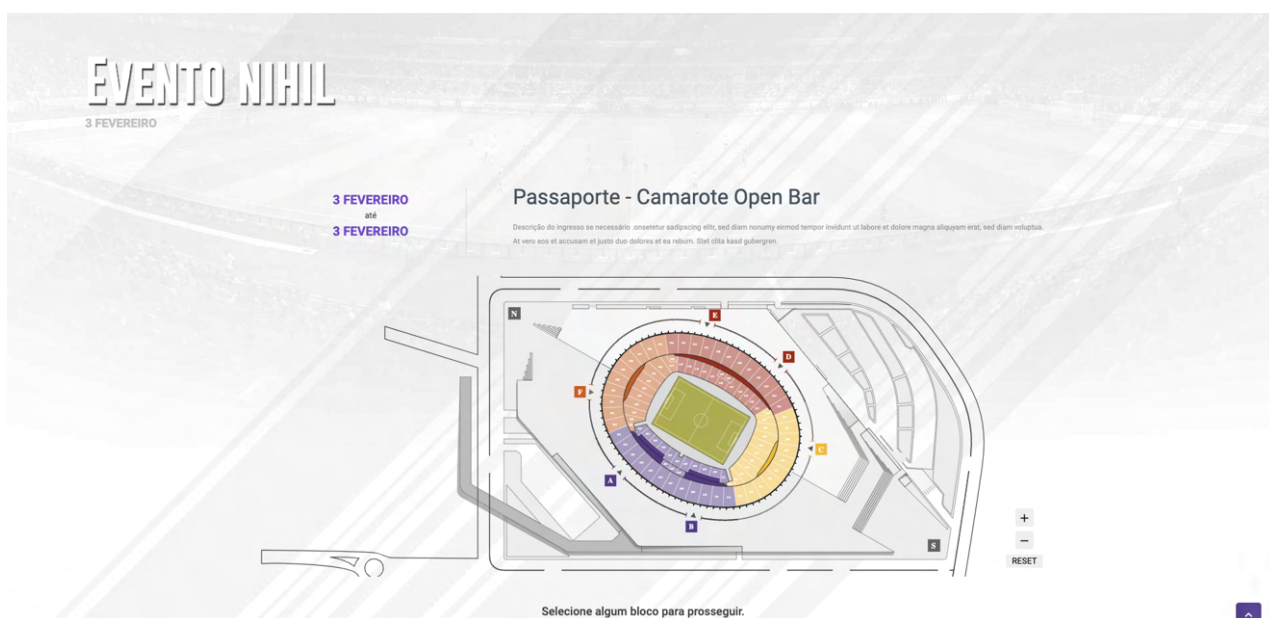
Para criação das telas, o Laravel oferece o Laravel Blade<sup>8</sup>, que é um motor gerador de páginas HTML com uma sintaxe PHP embutida. O Blade oferece algumas funcionalidades de apoio para a confecção de telas, como componentização, possibilitando o reuso e abstração de

<sup>8</sup> <https://laravel.com/docs/9.x/blade>

implementações que são utilizadas em várias telas, estruturas de repetição, condicionais, entre outras.

O desenvolvimento das telas de seleção de assentos (Figura 4.4) para a compra dos ingressos de evento de futebol, foi uma das telas que mais propôs desafios ao estagiário. A tela consiste na exibição de uma imagem interativa, que apresenta o mapa de blocos do estádio, com sua respectiva seção, numeração, valor e quantidade de assentos disponíveis. Cada bloco varia de quantidade, de acordo com a capacidade real do estádio, variando a quantidade de blocos, entre 20 e 700 assentos, totalizando cerca de 65.000 assentos.

Figura 4.4 – Tela de seleção de ingresso



Fonte: Do autor (2022)

O funcionamento por trás da tela consiste em um arquivo SVG (Scalable Vector Graphics) com a ilustração de todo o esquema de assentos do estádio, com elementos selecionáveis, o mesmo é embutido na página HTML, que ao renderizar todos os elementos, a página executa uma busca no banco de dados, a fim de encontrar as informações sobre a disponibilidade dos assentos, quantidades e valores. Ao fim do carregamento, cada célula numerada será modificada de acordo com os resultados retornados na busca. Caso a célula esteja disponível, ao manter o ponteiro do mouse sobre a célula, um *pop-up* é exibido trazendo as informações sobre o bloco em questão (Figura 4.5).

Ao selecionar o ingresso, a cor da célula é exibida de maneira vibrante para destacar a seleção, além disso, o valor do ingresso, nome do bloco e quantidade são listados abaixo do mapa, conforme a seleção. Caso a quantidade reduza a zero ou um dos blocos, for previamente

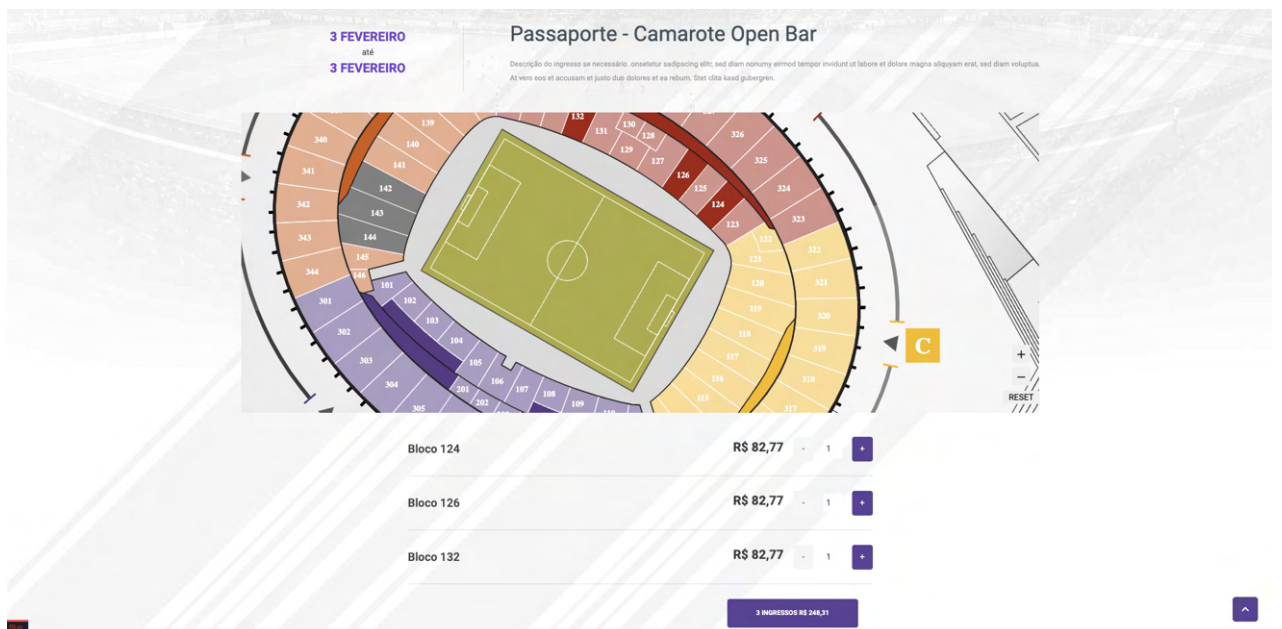
removido dos disponíveis para aquele evento, a célula será exibida em cinza, com o evento de clique desabilitado (Figura 4.6).

Figura 4.5 – Interação de leitura dos mapas de assentos



Fonte: Do autor (2022)

Figura 4.6 – Ingresso selecionado



Fonte: Do autor (2022)

Além disso, para oferecer mais conforto ao usuário durante a navegação, e também pensando no acesso de usuários via dispositivos móveis, o mapa de assentos desenvolvido pelo

estagiário oferece a funcionalidade de *zoom*, arrastar e soltar, além dos controles por botão, por questões de acessibilidade.

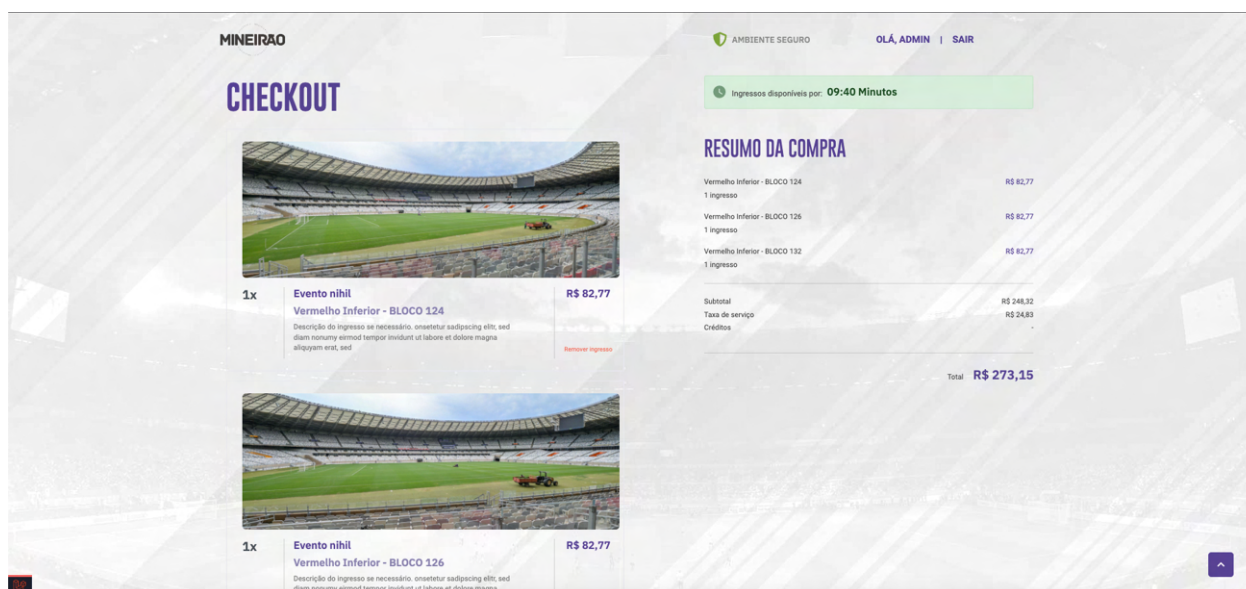
A página apresentou uma grande complexidade durante sua construção devido à falta de ferramentas gratuitas para este tipo de cenário, o estagiário ao buscar modelos de mapa de seleção de assentos, não obteve sucesso em encontrar algo que suprisse as necessidades do cliente.

Sendo assim, para solucionar o problema, foi necessário modificar as linhas de código do arquivo vetorial SVG, aplicando classes HTML para que fosse possível iterar pelo elemento utilizando JavaScript. Além disso, ao inicializar a página, foi necessário buscar estes elementos e aplicar ações durante os eventos de sobrepor/remover o mouse no elemento, clique, e para os dispositivos mobile, os eventos de toque, que são divergentes aos de clique (FOUNDATION, 2023a).

Todo o processo de seleção do assento é envolvido em um formulário, que persiste a solicitação de compra na sessão do usuário, até a próxima página (Figura 4.7).

A página de checkout contém uma lógica específica para lidar com a concorrência durante a compra dos ingressos. Para cada evento, um usuário administrador com permissão para acessar a configuração dos eventos, pode delimitar um tempo máximo para o cliente executar a compra do ingresso, e também cadastrar cupons de desconto informando o código e a porcentagem de desconto, para o evento em questão.

Figura 4.7 – Checkout



Fonte: Do autor (2022)

### 4.2.3 Sistema de Pagamento

Qualquer processo que envolva pagamentos exige uma grande atenção e impõe responsabilidades ao desenvolvedor, não só por questões fiscais, mas também pela necessidade intrínseca de integridade do processo. A venda de ingressos, da mesma forma exige estas questões com o adicional do aumento da complexidade devido a quantidade de itens à venda, o caráter único de cada ingresso, entre outros fatores que contribuem para tornar uma tarefa muito mais complexa. O estagiário sendo o responsável por este desenvolvimento, foi orientado pelo próprio cliente que ofereceu um grande esclarecimento dos processos, além dos pontos fracos e dificuldades vivenciadas na venda de ingressos.

O sistema de pagamento inicia o seu fluxo já no momento de criação dos eventos que se diferem em comerciais, nos quais não se utilizam dos assentos da arquibancada para a realização destes. Portanto o controle de quantidade de ingressos se simplifica apenas à quantidade disponibilizada definida pelo organizador. Os tipificados como futebol envolvem uma lógica extensa para gerenciar os mais de 65.000 assentos, separados em 9 seções, divididos entre 93 blocos enumerados de 101 a 344. A definição da quantidade disponível em cada bloco é inicializada por padrão com a capacidade total de assentos, podendo o administrador limitá-la. O valor dos ingressos pode ser definido no nível da seção, propagando o preço para todos os blocos pertencentes à seção ou ao nível do bloco, tendo este maior prioridade sobre o valor. A venda física, feita presencialmente no estádio, também deve ser cadastrada e computada para garantir uma fidelidade e controle entre as duas frentes de venda.

Com o evento criado e configurado com seus respectivos valores, este é agendado para a exibição na camada pública do sistema para que dê início às vendas.

### 4.2.4 Geração do Pagamento

A compra é gerenciada pelo MercadoPago<sup>9</sup>. Todo o processo de preenchimento das informações do usuário é controlado por ele. Somente ao finalizar o processo da compra, o usuário é redirecionado para a página de compra finalizada. Página esta, que informa a situação da compra. Caso a compra efetuada seja bem sucedida, um processamento dos dados é feito e a quantidade total de assentos disponíveis por bloco é reduzida, de acordo com a venda.

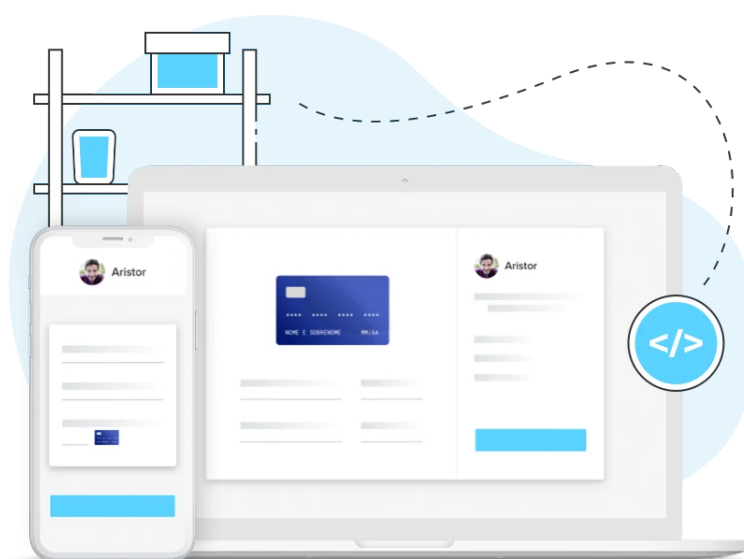
---

<sup>9</sup> <https://www.mercadopago.com.br/>



A solução aplicada para o pagamento foi o Checkout Pro<sup>10</sup>, uma solução simples em que o cliente realiza a compra em uma página própria do MercadoPago com todos os métodos de pagamentos disponíveis na plataforma (Figura 4.8). O papel do estagiário foi de utilizar o MercadoPago SDK PHP, que é um conjunto de classes e implementações prontas para realizar a integração. Para o modelo do Checkout Pro, basta gerar uma *Preference*, que consiste em um conjunto de informações que são necessárias para a geração da compra, como por exemplo, o que está sendo comprado, o valor, quantidade, informações do usuário, entre outras.

Figura 4.8 – Tela exemplo do Checkout Pro



Fonte: Mercado Pago (2023)

#### 4.2.5 Processamento do Pagamento

O processamento do pagamento e a notificação da situação da venda funcionam através de *webhooks*, que consistem em mensagens pequenas que são enviadas automaticamente entre servidores, com intuito de notificação. A documentação de Webhooks do MercadoPago instruiu ao estagiário apenas como configurar o recebimento da mensagem. Muitos dos *frameworks* Web de propósito geral, implementam ferramentas de segurança para oferecer maior confiabilidade nos sistemas desenvolvidos. Um ataque muito comum que o Laravel trata nativamente é o CSRF. Sendo assim, foi necessário adicionar a rota de recebimento do webhook na lista de exceções à validação CSRF<sup>11</sup> para remover a verificação de segurança além do recebimento pleno das notificações de pagamento. Todavia nenhuma rota deve ficar exposta sem a mínima

<sup>10</sup> <https://www.mercadopago.com.br/developers/pt/docs/checkout-pro/landing>

<sup>11</sup> <https://laravel.com/docs/9.x/csrf>

proteção, e com o intuito de solucionar o problema, o estagiário criou uma regra de validação a fim de garantir que a requisição seja oriunda do MercadoPago. A validação afere as seguintes questões para permitir o acesso e processar a notificação:

- O endereço IP do solicitante deve pertencer à lista de IPs do MercadoPago.
- A requisição deve trazer os dados de acordo com o modelo padrão do MercadoPago.
- O *user-agent* do MercadoPago tem nome próprio, diferenciando de navegadores normais.

Com os dados validados, a informação do webhook é persistida no banco de dados. A partir deste ponto o estagiário utilizou da funcionalidade própria do Laravel chamada Observers<sup>12</sup> que utiliza do Eloquent, facilitando a integração e entregando uma sintaxe de uso de baixa verbosidade, executando tarefas complexas com poucas linhas de código.

Os Observers, utilizam do ecossistema construído com o Eloquent para aplicar o padrão de projeto Observer, de forma que as modificações feitas nas Models são refletidas e notificadas nas classes observadoras. O estagiário optou pelo uso desta implementação ao invés do uso de *Cron Jobs*, que consiste na execução de tarefas rotineiras que são executadas de maneira cíclica num intervalo de tempo definido.

O Cron não caberia nesta solução, pois existem pontos de desvantagem que afetariam o desempenho de processamento dos eventos de venda. Como é uma tarefa cíclica e não reativa, um dos piores cenários e mais previsíveis seria o tempo de espera de processamento de uma venda ser igual ao tempo do ciclo Cron. Além disso, em situações de pico, com alto número de requisições geradas, processar um lote de informações acaba se tornando uma tarefa que exige esforço maior do servidor.

Já com o uso do Observer, o consumo das notificações se faz de imediato, no momento em que o banco de dados persiste a notificação do webhook, uma transação é criada caso ela não exista, contendo as informações da venda e o *status* do pagamento, se a mesma já estiver persistida no banco de dados, a transação é atualizada com seu devido *status*. Da mesma forma que o Observer foi utilizado para os webhooks, ele é aplicado durante as modificações nas transações. Após uma transação sair do estado de pendência e o processamento do pagamento for concluído, a transação será finalizada e a quantidade total de ingressos disponíveis será atualizada. Por fim, com o pagamento finalizado, o cliente receberá um e-mail com o detalhamento da compra.

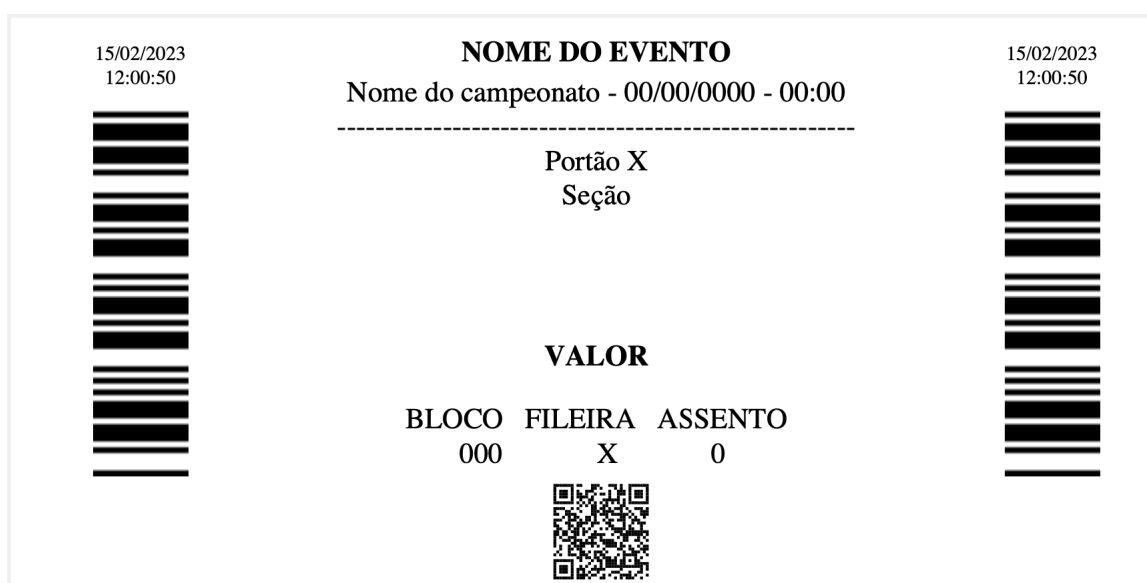
---

<sup>12</sup> <https://laravel.com/docs/9.x/eloquent>

#### 4.2.6 Geração dos Ingressos

O processo de geração de ingressos utiliza de duas bibliotecas PHP específicas, uma para renderização de arquivos no formato PDF e uma para geração de código de barras e QRCode. A geração de documentos PDF tem uma peculiaridade para sua implementação que consiste no uso do HTML, na versão 4, se diferenciando muito do HTML 5, a versão moderna. A versão anterior não oferece os recursos de estilização extensamente utilizados na versão superior, o que dificultou a confecção do ingresso por parte do estagiário.

Figura 4.9 – Exemplar do Modelo de Ingresso



Fonte: Do autor (2022)

O ingresso conta com um dimensionamento peculiar de página (Figura 4.9) com uma disposição dos códigos de barra de maneira vertical. Toda esta disposição do ingresso se deve à compatibilidade com a catraca de entrada do estádio. A máquina leitora do código de barras utiliza o padrão PDF417. O conteúdo do código de barras envolve uma lógica particular e contém um identificador que representa o evento, a data, seção, bloco, assento entre outras validações. O QRCode valida o ingresso em tempo real, verificando a procedência, além de apresentar informações sobre o evento. Cada QRCode contém o identificador único da venda, sendo discernido de outros ingressos.

#### 4.3 Vivência com Metodologias Ágeis

A vivência com as metodologias ágeis foi algo inédito para o estagiário, porém não houve problema durante o processo de adaptação, graças ao treinamento oferecido pela GAFIT



e também devido às modificações nos processos que a empresa utiliza. Toda a relação e vivências diárias são compartilhadas através da plataforma *Gather*, um espaço de conversa criado especialmente para lidar com atividades remotas. Com uma aparência temática inspirada em jogos de arquitetura 16-bit com possibilidade de customizações e extensões, também oferece ferramentas de apoio à comunicação, como chamadas de voz compartilhadas ou privadas, *chats* de texto, exibição de tela ou *webcam*.

### 4.3.1 Scrum

A metodologia Scrum sofreu modificações sob a forma em que é executada, com o intuito de simplificar os processos, visto que grande parte da equipe é de estagiários, que por sua vez possuem pouco contato com métodos ágeis e rotinas de desenvolvimento de software. Os processos foram modificados com intuito de adaptar a realidade da empresa:

- As reuniões diárias, comumente chamadas Daily Meetings, são mais longas que o comum, com duração entre 30 e 40 minutos, com todos os times presentes. O tempo de duração da reunião se difere do padrão com duração entre 10 a 15 minutos, isso se deve pois a proposta da reunião também inclui um período de descontração entre a equipe.
- O rito de *Retrospective* não é implementado nos processos.
- Não há *Product Owner* no projeto, sendo assim todo o time tem contato com o cliente, participando de reuniões de montagem de *backlogs*, definição de requisitos e apresentação de lançamentos.
- O rito de *Sprint Review* é executado juntamente com o cliente, realizado após a homologação das entregas.

O Scrum apesar de sofrer estas modificações não apresentou empecilhos para o estagiário por em prática; na verdade ele otimiza o tempo e a objetividade de executar o processo de desenvolvimento de *software* de uma forma simplista. A execução do projeto suportou a imprevistos, mudanças de planejamento, situações inesperadas, gerando atrasos mínimos na entrega, demonstrando que para estes casos, simplificar o processo de acordo com a realidade da organização apresenta fatores positivos.

### 4.3.2 Kanban

O Kanban foi utilizado como ferramenta para auxiliar o andamento e execução das histórias geradas em cada *sprint*. A Gafit utiliza o ClickUp, uma ferramenta de apoio para gerenciamento de projetos. O ClickUp oferece ferramentas para gestão de projetos como agendamento de tarefas, definição de prioridades, cadastro de pessoas e atribuição de tarefas à equipe. Todo o processo de confecção dos cards para o Kanban é feito juntamente com o desenvolvedor responsável pelo projeto, definindo prazos, objetificando e listando os requisitos a serem cumpridos em cada tarefa.

O desenvolvedor alocado para a tarefa, deve então, no momento em que for orientado a iniciá-la, precisará executar as especificações definidas, avaliar a correteza das funcionalidades desenvolvidas e por fim sinalizar a tarefa como pendente à revisão ou concluída.

O responsável pela gerência do projeto, por sua vez, também avalia o resultado final da tarefa e garante o cumprimento dos requisitos, caso contrário, a tarefa é reaberta e escalada para ser corrigida. Existem também cenários impeditivos que obstruem a conclusão da tarefa, neste caso ela deve ser marcada como bloqueada, demonstrando que existe algum item bloqueante para a conclusão do ciclo.

A Gafit se destaca com o cliente devido a sua transparência com a exibição do projeto, oferecendo acesso à plataforma ClickUp para que o cliente tenha, de maneira clara e transparente, informações sobre o projeto em tempo real, além de expor claramente a saúde e andamento do projeto.

#### 4.4 Relação entre teoria e prática

Nesta seção é descrita a relação das disciplinas cursadas durante a graduação que estão ligadas com as atividades práticas realizadas no estágio.

- GCC188 Engenharia de Software: Engenharia de software foi muito importante devido à visão geral apresentada para o desenvolvimento completo de uma aplicação, facilitando o entendimento das motivações de uso de padrões, arquiteturas, ferramentas e aproximando-se muito da experiência prática. A disciplina que tem como trabalho final, o desenvolvimento de um sistema em seu fluxo completo, concepção da camada lógica, engenharia de requisitos, desenvolvimento do software e testes unitários. O trabalho final foi executado com certa facilidade pelo estagiário, por ter optado pelo Laravel como tecnologia de desenvolvimento, o que acelerou o processo de desenvolvimento, além de explorar outras ferramentas antes não vistas no estágio.
- GCC122 Linguagens Formais e Autômatos: Apesar da disciplina não ter envolvimento direto com aplicações Web, ela exercita o pensamento limpo e o raciocínio lógico, sendo fundamental para o desenvolvimento do estagiário. A disciplina explora do aluno a capacidade de pensar e resolver os problemas impostos, que por muitas vezes não é perceptível a utilidade direta do conteúdo. No entanto, a habilidade de resolver problemas é muito valiosa para a carreira de um desenvolvedor de maneira geral.
- GCC219 Interação Humano-Computador: Importante para visualizar e entender questões que geralmente são implícitas para desenvolvedores iniciantes e com pouca experiência de mercado, auxiliando a desenvolver aplicações, pensando no comportamento do usuário final, como prioridade. Toda a camada pública recebeu uma atenção maior voltada a parte de Interação Humano-Computador, pois um dos requisitos funcionais das telas é a acessibilidade, oferecendo alternativas de contrastes, aumento e redução de fontes e navegação assistida. A disciplina propõe um trabalho prático que utiliza de ferramentas de avaliação da acessibilidade do site, facilitando a concepção e refinamento da exibição acessível.
- GCC244 Processos de Software: Fundamental para o entendimento da vida cotidiana no âmbito do desenvolvimento de *software*, auxiliando na compreensão das necessidades de uma organização, amparando-se em fluxos e atividades rotineiras, que apesar de nem

sempre serem iguais para todas as empresas, existem semelhanças, bem como o objetivo comum, a entrega do produto final.

- GCC263 Introdução a Sistemas de Banco de Dados: Apesar do estagiário já ter experimentado o uso de banco de dados relacionais, conceitos como normalização e o uso de chaves estrangeiras eram pouco claros e confusos. Além do entendimento de modelos entidade-relacional, que são muito utilizados. A disciplina apesar de ter um vasto conteúdo disponibilizado, os tópicos apresentados são não triviais e de difícil aprendizado. A parte prática da disciplina é voltada somente a bancos de dados relacionais, o que dificultou para o estagiário no momento de recomendar o uso de um banco não relacional para otimizar a persistência das informações de blocos e valores de ingressos. A empresa também já utilizava o *MySQL Workbench*, o que facilitou a gestão do banco de dados e na geração de Modelos Relacionais das tabelas.
- GCC175 Sistemas Gerenciadores de Banco de Dados: Diferentemente da disciplina introdutória a bancos de dados, esta disciplina apresenta conceitos fundamentais para lidar com complexidades de bancos de dados, lidar com a integridade de transações, performance e otimização. O conteúdo da disciplina é rico de informações que auxiliaram na construção da camada de dados do projeto, principalmente com o uso de consultas transacionais. Contudo, a aplicação dos conceitos poderia ser acompanhada de alguma implementação prática para melhor aderência do conteúdo.

Existem outras disciplinas não listadas que foram essenciais para além de enriquecer o conteúdo aprendido, são fundamentos da computação que são diferenciais para desenvolvedores de início de carreira. Conceitos de Sistemas Distribuídos, Redes de Computadores, *DevOps* na Prática e Complexidade de Projetos de Algoritmos foram disciplinas desafiadoras mas que também ofereceram grandes resultados como um todo.

## 5 CONSIDERAÇÕES FINAIS

O estágio realizado na empresa Gafit foi um período extremamente valioso para o amadurecimento profissional do estagiário. Nos dias atuais, apesar da alta da oferta de vagas voltadas para as tecnologias de informação, as empresas têm buscado profissionais com experiência até mesmo para cargos iniciais. Isso demonstra que o mercado vem cada vez mais se tornando um ambiente altamente disputado. Os conhecimentos adquiridos durante o estágio, assimilados com a vivência acadêmica, serão importantes para a evolução como profissional, além de oferecer visão crítica e analítica para a solução de problemas computacionais.

Vale evidenciar as dificuldades vivenciadas no estágio, principalmente no início, que foi de adequar com as peculiaridades da arquitetura de software definida, além de algumas tarefas desenvolvidas necessitarem de soluções inovadoras e que forcem o uso da criatividade para se obter resultados. Em especial, o desenvolvimento do mapa de assentos do estádio que foi uma tarefa de alta complexidade executada sem auxílio técnico de outros colaboradores com mais experiência, que demandou esforços para transformar uma imagem SVG de um esquema de blocos do estádio, em um mapa interativo com funcionalidades que se modificam em tempo real.

Destacam-se pontos de observação analisados durante o desenvolvimento do projeto, que ao reavaliar as decisões técnicas, seria possível otimizar e melhorar o produto final a ser desenvolvido:

- A solução utilizada para a gestão e controle dos assentos vendidos em cada evento, não se encaixa bem para bancos relacionais, não só pelo volume de requisições esperados, mas também devido a perda de desempenho ao cruzar informações de cada evento, com seu respectivo preço para todos 65.000 assentos, de forma separada. Para tratar estas informações, o uso de um Sistema No-SQL como o *Redis*, trataria melhor o cruzamento destas informações, reduzindo muito a complexidade de consulta e busca.
- Apesar da empresa evangelizar boas práticas durante a confecção de linhas de código, os testes de *software* são fundamentais para garantir a saúde e o bom funcionamento de qualquer *software*.
- O *framework* oferece uma gama imensa de recursos para as mais diversas atividades, como o uso de Broadcasting, Observers, Queues, recursos que não foram apresentados

durante o período de treinamento e não postos em prática. Artifícios, que podem resolver problemas de forma otimizada e moderna.

- A prática de revisão de código através dos *Pull Requests* é uma prática simples, e fácil de ser implementada, que reduz a inserção de códigos pouco performáticos ou com a presença de erros. Assim, além de impulsionar o aprendizado de quem produziu o código, através de um *feedback* rápido e claro, põe em prática a habilidade de entendimento e visão analítica sobre o código para o revisor.

Por fim, o percurso acadêmico, anterior ao período de estágio foi desafiador, principalmente pelo fato da dificuldade de entender como os conceitos se aplicariam na prática. O estágio foi o catalisador para a evolução do discente, estimulando a prática do raciocínio lógico e facilitando não só o entendimento das disciplinas, mas também, aplicando uma visão crítica e analítica.

## REFERÊNCIAS

- AMAZON. **O que é o Scrum?** 2023. (Acessado em 09/09/2023). Disponível em: <<https://aws.amazon.com/pt/what-is/scrum/#:~:text=adotar%20o%20DevOps%3F-,O%20que%20%C3%A9%20o%20Scrum%3F,uma%20entrega%20eficiente%20de%20projetos.>>
- AMAZON. **O que é uma framework em programação e engenharia?** 2023. (Acessado em 13/12/2023). Disponível em: <<https://aws.amazon.com/pt/what-is/framework/>>.
- AWARI. **O que é o WSL2 e como ele pode melhorar sua experiência.** 2023. (Acessado em 09/09/2023). Disponível em: <<https://awari.com.br/wsl2/>>.
- BENTO, E. J. **Desenvolvimento web com PHP e MySQL.** [S.l.]: Casa Do Código, 2021.
- CUNHA, M. B. **Entendendo o Uso do Git em Equipes de Desenvolvimento de Software.** 2018. (Acessado em 09/09/2023). Disponível em: <[https://www.cin.ufpe.br/~tg/2018-2/TG\\_EC/tg-mbc3.pdf](https://www.cin.ufpe.br/~tg/2018-2/TG_EC/tg-mbc3.pdf)>.
- DOCKER. **Docker Documentation.** 2023. (Acessado em 08/09/2023). Disponível em: <<https://docs.docker.com/get-started/overview/>>.
- FOUNDATION, M. **JavaScript Touch Events Documentation.** 2023. (Acessado em 16/09/2023). Disponível em: <[https://developer.mozilla.org/en-US/docs/Web/API/Touch\\_events](https://developer.mozilla.org/en-US/docs/Web/API/Touch_events)>.
- FOUNDATION, M. **Mozilla Foundation, CSS Reference.** 2023. (Acessado em 08/09/2023). Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>.
- FOUNDATION, M. **MySQL Documentation.** 2023. (Acessado em 08/09/2023). Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>.
- FOUNDATION, O. **jQuery Documentation.** 2023. (Acessado em 08/09/2023). Disponível em: <<https://api.jquery.com/>>.
- GAFIT. **GAFIT.** 2023. (Acessado em 27/08/2023). Disponível em: <<https://gafit.com.br/>>.
- GATHER. **Gather.** 2023. (Acessado em 09/09/2023). Disponível em: <<https://pt-br.gather.town/features>>.
- GONÇALVES, F. P. **Produtividade: software como aliado no crescimento.** 2019. (Acessado em 28/05/2019). Disponível em: <<https://repositorio.animaeducacao.com.br/handle/ANIMA/9032>>.
- GROUP, T. P. **PHP Documentation.** 2023. (Acessado em 07/09/2023). Disponível em: <[https://www.php.net/manual/pt\\_BR/](https://www.php.net/manual/pt_BR/)>.
- GUEDES, M. **O que é Bootstrap?** 2020. (Acessado em 09/09/2023). Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-bootstrap>>.
- IBM. **Prevenção de ataques Cross-site Request Forgery (CSRF).** 2023. (Acessado em 16/09/2023). Disponível em: <<https://www.ibm.com/docs/pt-br/sva/10.0.6?topic=configuration-prevention-cross-site-request-forgery-csrf-attacks>>.

IMPERVA. **What is Minification | Why minify JS, HTML, CSS files** |. 2023. (Acessado em 09/09/2023). Disponível em: <<https://www.imperva.com/learn/performance/minification/#:~:text=Minification%20is%20the%20process%20of,into%20a%20better%20user%20experience.>>

JETBRAINS. **Recursos - PhpStorm**. 2023. (Acessado em 09/09/2023). Disponível em: <<https://www.jetbrains.com/pt-br/phpstorm/features/>>.

LARAVEL. **Laravel Documentation**. 2022. (Acessado em 08/09/2023). Disponível em: <<https://laravel.com/docs/9.x>>.

LARAVEL. **Laravel Homestead Documentation**. 2023. (Acessado em 09/09/2023). Disponível em: <<https://laravel.com/docs/7.x/homestead>>.

LARAVEL. **Laravel Middleware Documentation**. 2023. (Acessado em 09/09/2023). Disponível em: <<https://laravel.com/docs/10.x/middleware>>.

LARAVEL. **Laravel Sail Documentation**. 2023. (Acessado em 09/09/2023). Disponível em: <<https://laravel.com/docs/10.x/sail>>.

MYSQL. **MySQL Documentation**. 2023. (Acessado em 08/09/2023). Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>>.

O'GRADY, A. **GitLab Quick Start Guide: Migrate to GitLab for all your repository management solutions**. [S.l.]: Packt Publishing, 2018.

OWASP. **Cross Site Request Forgery (CSRF)**. 2023. (Acessado em 09/12/2023). Disponível em: <<https://owasp.org/www-community/attacks/csrf>>.

PRISMA. **What is an ORM?** 2023. (Acessado em 12/12/2023). Disponível em: <<https://www.prisma.io/dataguide/types/relational/what-is-an-orm#:~:text=An%20ORM%2C%20or%20Object%20Relational,used%20in%20object%2Doriented%20programming.>>

SILVERMAN, R. E. **Git Pocket Guide: A Working Introduction**. [S.l.]: Casa Do Código, 2013.

STAUFFER, M. **Laravel Up & Running Third Edition**. CA, United States of America: O'Reilly, 2023.

TEAM, B. **Bootstrap Documentation**. 2023. (Acessado em 09/09/2023). Disponível em: <<https://getbootstrap.com/>>.

UNIVERSITY, K. **O guia oficial do método Kanban**. 2021. (Acessado em 09/09/2023). Disponível em: <[https://kanban.university/wp-content/uploads/2021/04/The-Official-Kanban-Guide\\_Portuguese\\_A4.pdf](https://kanban.university/wp-content/uploads/2021/04/The-Official-Kanban-Guide_Portuguese_A4.pdf)>.

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

W3C. **RFC 1866 - Hypertext Markup Language - 2.0**. 1995. (Acessado em 08/09/2023). Disponível em: <<https://datatracker.ietf.org/doc/html/rfc1866>>.

WEBPACK. **Concepts Webpack**. 2023. (Acessado em 16/12/2023). Disponível em: <<https://webpack.js.org/concepts/>>.



YERET, Y.; VACANTI, D.; SCRUM.ORG. **O Guia Kanban para Scrum Teams**. 2021. (Acessado em 09/09/2023). Disponível em: <<https://scrumorg-website-prod.s3.amazonaws.com/drupal/2021-02/2021-Kanban-Guide-Portuguese-Brazilian-2.0.pdf>>.