



**JOÃO VITOR COSTA DE GÓES**

**RELATÓRIO DE ESTÁGIO - DESENVOLVIMENTO DE  
SOFTWARE NA EMPRESA TECHNOLOG**

**LAVRAS – MG**

**2022**

**JOÃO VITOR COSTA DE GÓES**

**RELATÓRIO DE ESTÁGIO - DESENVOLVIMENTO DE  
SOFTWARE NA EMPRESA TECHNOLOG**

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

Prof. Paulo Afonso Parreira Júnior  
Orientador

**LAVRAS - MG  
2022**

Dedico aos meus pais Raimundo e Niceia, que sempre me apoiaram nas minhas escolhas e forneceram tudo que foi necessário para chegar até aqui.

## **AGRADECIMENTOS**

Agradeço aos meus pais por me proporcionarem condições para dar sequência nos estudos e por me apoiarem em todas as decisões que fiz no caminho. Também agradeço especialmente à Technolog, por me dar a oportunidade de exercer os aprendizados adquiridos pela graduação na prática e adquirir muitos conhecimentos sobre como funciona o mercado de tecnologia da informação.

Além disso, agradeço as pessoas que estiveram envolvidas neste processo comigo, como os funcionários do Departamento de Ciência da Computação, os amigos que dividiram momentos dessa jornada e fizeram com que tudo ficasse mais fácil e alegre.

Agradeço também a Gabriela Lemos da Silva, por todo o companheirismo e por estar presente em toda essa trajetória comigo, seu apoio foi fundamental. Por último, queria agradecer ao professor Paulo Afonso Parreira Júnior pela excelente orientação deste trabalho, pela paciência e disposição em ajudar.

## RESUMO

Este relatório de estágio supervisionado tem como objetivo descrever as atividades exercidas na empresa Technolog, que atua no ramo de logística e gestão de transportadoras. Existem inúmeros sistemas que compõem a organização, como o sistema Especialistas, o sistema Darwin e o sistema Manager. Foram realizadas algumas melhorias e manutenções em todos estes sistemas, trazendo para o estagiário o aprendizado em diversas tecnologias diferentes e a melhor aplicabilidade de cada uma delas. Neste relatório, são apresentadas as descrições dessas atividades, assim como as tecnologias empregadas nelas.

**Palavras-chave:** Desenvolvimento Web, Manutenção de software, Estágio supervisionado, Gestão de Logística.

## **ABSTRACT**

This supervised internship report aims to describe the activities carried out in the Technolog company, which acts in logistics and transport management fields. There are several software systems in this organization, such as the “Especialista”, “Darwin” and “Manager”. Some improvements and maintenance were carried out in all of them, bringing to the intern the learning in several different technologies and the best applicability of each one of them. This report presents descriptions of these activities, as well as the technologies used.

**Keywords:** Web Development, Software Maintenance, Supervised Internship

## LISTA DE FIGURAS

Figura 1 - Exemplo de Quadro Kanban. ....	14
Figura 2 - Gestão de Projetos Plan.io ...	17
Figura 3 - Página Web sem aplicação de CSS .....	20
Figura 4 - Página Web com CSS .....	22
Figura 5 - Alerta criado com Javascript.. .....	23
Figura 6 - Exemplo de página com PHP. ....	25
Figura 7 - Página “Onde Está o Técnico”. .....	27
Figura 8 - Código para gerar o Mapa .....	28
Figura 9 - Mensagem de Erro Múltiplos Acessos .....	30
Figura 10 - Portal Especialistas .....	31
Figura 11 - Cadastro Instalação Especialistas .....	32
Figura 12 - Resultado Instalação Especialistas .....	33

## LISTA DE QUADROS

Quadro 1 - Endpoint usado API Technolog . . . . .	16
Quadro 2 - Exemplo de SQL . . . . .	18
Quadro 3 - Estrutura Básica de uma Página HTML . . . . .	19
Quadro 4 - Estrutura Básica de uso do CSS . . . . .	20
Quadro 5 - Código de Página com CSS. . . . .	21
Quadro 6 - Exemplo de Código em JS . . . . .	23
Quadro 7 - Exemplo de Código em PHP . . . . .	24
Quadro 8 - Exemplo SQL Onde Está o Técnico . . . . .	27



## LISTA DE SIGLAS

API Application Programming Interface

SQL Standard Query Language

HTML HyperText Markup Language

CSS Cascade Style Sheets

JS JavaScript

PHP Hypertext Preprocessor

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>13</b>
<b>2.1</b>	<b>Kanban</b>	<b>13</b>
<b>2.2</b>	<b>API (<i>Application Programming Interface</i>)</b>	<b>15</b>
<b>2.3</b>	<b>Gestão de Projetos com <i>Plan.io</i></b>	<b>16</b>
<b>2.4</b>	<b>Banco de Dados</b>	<b>17</b>
<b>2.5</b>	<b>Tecnologias web</b>	<b>18</b>
<b>2.5.1</b>	<b>HTML - <i>HyperText Markup Language</i></b>	<b>18</b>
<b>2.5.2</b>	<b>CSS - <i>Cascade Style Sheets</i></b>	<b>20</b>
<b>2.5.3</b>	<b>JavaScript</b>	<b>22</b>
<b>2.5.4</b>	<b>PHP - <i>Hypertext Preprocessor</i></b>	<b>24</b>
<b>3</b>	<b>ATIVIDADES DESENVOLVIDAS</b>	<b>26</b>
<b>3.1</b>	<b>Onde Está o Técnico?</b>	<b>26</b>
<b>3.2</b>	<b>Múltiplos <i>Logins</i></b>	<b>29</b>
<b>3.3</b>	<b>Cadastro de Serviço de Instalação</b>	<b>31</b>
<b>4</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>34</b>
<b>5</b>	<b>REFERÊNCIAS</b>	<b>36</b>

## 1 INTRODUÇÃO

Neste capítulo, é apresentada uma contextualização do estágio supervisionado, bem como uma introdução sobre a empresa *Technolog*. Além disso, são descritos, sucintamente, alguns dos projetos existentes nesta empresa e as atividades realizadas pelo estagiário nos mesmos.

Dos sistemas logísticos existentes no país, o principal deles é o transporte rodoviário, sendo responsável por mais de 60% de todas as cargas movimentadas no território nacional (BOLETIM CNT, 2019). Além de ser a maior forma de escoamento de cargas, também é a mais cara, devido ao valor do frete, dos combustíveis e da manutenção dos veículos (INFOESCOLA, 2015). Isso se deve, principalmente, ao fato que somente 12% de todas as estradas existentes no país são pavimentadas (ANUÁRIO CNT, 2021), ou seja, a maioria é de baixa qualidade, gerando maior custo para as empresas que utilizam essa forma de transporte. Com isso, a cada dia, muitas empresas pensam em estratégias para reduzir o custo total da manutenção de suas frotas e, conseqüentemente, aumentar seu lucro.

Na cidade de Lavras/MG, existem algumas empresas que atuam com foco no desenvolvimento de sistemas de software que ofereçam serviços a empresas transportadoras; uma destas empresas é a *Technolog*, empresa na qual foi realizado o estágio supervisionado descrito neste relatório. A *Technolog* possui sede na cidade de Lavras/MG e conta com aproximadamente 60 (sessenta) funcionários, sendo 10 (dez) funcionários que trabalham na área de Tecnologia da Informação (TI). Os demais funcionários são compostos por analistas de logística, bem como técnicos responsáveis pela manutenção dos sensores e dos demais equipamentos que são instalados nos caminhões dos clientes da empresa.

O principal produto da empresa é o sistema “Darwin”, utilizado para controlar informações referentes à frota de cada cliente, como por exemplo o consumo de combustível, a localização e velocidade dos veículos, dentre outras coisas. Outro produto disponibilizado pela empresa é o sistema “Especialistas”, por meio do qual é possível realizar o agendamento de visitas técnicas com os profissionais da *Technolog*, bem como ter acesso a todo histórico de manutenções, instalações e retirada de equipamentos dos clientes da empresa.

Na *Technolog*, existe a necessidade de se realizar manutenção corretiva nos sistemas comentados anteriormente, bem como desenvolver novas funcionalidades para os mesmos. Neste contexto, o estagiário realizou ambos os tipos de atividades nos dois principais sistemas da empresa, o *Darwin* e o *Especialistas*. Tais atividades foram feitas tanto no back-end como no front-end e são sucintamente descritas a seguir. Mais detalhes sobre elas podem ser vistos no Capítulo 3 deste relatório.

**Darwin.** Um exemplo de nova funcionalidade desenvolvida pelo estagiário foi a criação de um sistema de controle de *login* de usuários na plataforma, a fim de permitir múltiplos acessos por meio de diferentes dispositivos.

**Especialistas.** Neste sistema, foram implementadas a lógica de negócio e as telas de interface com o usuário para se cadastrar diferentes tipos de serviços prestados pelos técnicos, tais como como instalações e manutenções dos veículos. Além disso, foi desenvolvida também uma tela, utilizando o Google Maps, com informações sobre a localização, os horários de atendimento e informações pessoais para contato dos técnicos da empresa.

O restante deste trabalho está organizado da seguinte forma: no Capítulo 2 são apresentados alguns conceitos a respeito das tecnologias utilizadas ao longo do estágio; no Capítulo 3, são apresentadas as atividades desenvolvidas, assim como os resultados alcançados a partir delas; por fim, no Capítulo 4 são apresentadas as considerações finais.

## 2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados alguns conceitos a respeito das tecnologias utilizadas no dia-a-dia de trabalho dentro da *Technolog*, os quais foram fundamentais para o desenvolvimento das atividades descritas no Capítulo 3 deste relatório.

### 2.1 Kanban

As organizações sempre estão buscando formas de aumentar e melhorar a produtividade de seus colaboradores. Uma das opções que podem ser adotadas neste sentido é a utilização de metodologias ágeis para gestão de projetos. Essas metodologias são um conjunto de práticas que visam aumentar a produtividade de uma equipe, particionando as entregas em determinados ciclos, a fim de garantir uma maior qualidade do projeto final (LUMIS, 2021). A metodologia ágil adotada pela empresa *Technolog* é o Kanban.

Segundo o “Guia Kanban” (Kanban Guide for Scrum Teams, 2021), Kanban é uma metodologia para otimizar o fluxo de valor, por meio de um processo que utiliza facilitação visual e limitação de *work-in-progress* (trabalho em progresso). Ou seja, é um sistema visual que utiliza um quadro com cartões, geralmente *post-its*, para conduzir cada tarefa durante o fluxo de trabalho, sendo assim possível identificar gargalos e corrigi-los para manter a fluidez das atividades da empresa.

Este quadro, que pode ser físico ou digital, é composto por colunas. Como é possível observar na Figura 1, existem três colunas principais (*to do*, *doing* e *done*), mas pode-se criar outras, de acordo com a necessidade da organização. A princípio, a primeira coluna apresenta os cartões das tarefas que devem ser realizadas pela equipe; na segunda encontram-se as tarefas que estão sendo realizadas no momento; por fim, na última coluna, encontram-se as tarefas finalizadas pela equipe.

No caso da *Technolog*, o quadro era composto, primeiramente, por uma coluna na esquerda com um *avatar* que identificava cada colaborador da equipe de desenvolvimento. Depois, eram divididos dois setores de trabalho, o de desenvolvimento (*development*) e o de testes (*test*).

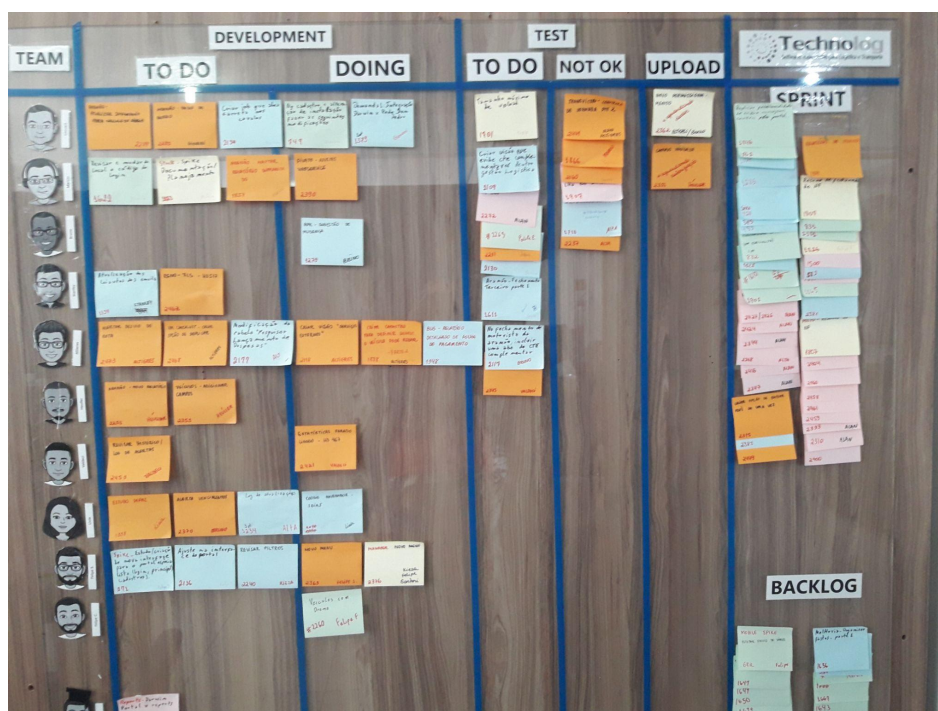
O setor de desenvolvimento ainda era dividido em duas colunas, *to do* e *doing*, nos quais existia, respectivamente, uma fila de demandas a serem realizadas

e uma das que já estavam em execução, separadas por colaboradores.

Assim que as atividades eram finalizadas, elas migravam para a coluna de testes. Nesta coluna, existia também a mesma divisão citada anteriormente, porém com a adição de mais uma coluna, *Upload*, que significava que ao fim dos testes de uma demanda, se tudo estivesse certo, era liberado para ser disponibilizado para o cliente.

Por último, existia uma coluna final, que era dividida em *Sprint* e *Backlog*. Em *Backlog*, estavam todas as demandas que necessitavam ser realizadas com o objetivo de se chegar ao produto final. Em *Sprint*, estavam as demandas e os requisitos que deveriam ser concluídos até o fim de um “ciclo” (um intervalo de tempo que, normalmente, durava algumas semanas). Nos *post-its* das demandas, existiam informações importantes, dentre elas o número que a identificava na ferramenta de gestão *Plan.io* (Seção 2.3). Assim, era possível obter mais detalhes sobre determinada demanda, como o que o solicitante desejava, qual era o prazo máximo de término, qual era seu tipo de prioridade, seus requisitos, entre outros.

**Figura 1** - Exemplo de quadro Kanban usado na *Technolog*



Fonte: Technolog (2022)

O uso de tal metodologia foi muito importante para que o estagiário conseguisse visualizar todas as atividades que estavam sendo realizadas pela equipe de desenvolvimento e quem era responsável por cada uma delas. Além disso, era possível saber quais eram as próximas tarefas que deveriam ser realizadas pelo estagiário.

O fluxo de trabalho na *Technolog* era intenso. Nas primeiras semanas, era dado um treinamento para os estagiários e, depois, eram disponibilizadas tarefas para treinar o que foi aprendido. As demandas que deveriam ser feitas chegavam ao estagiário por meio do seu superior. Pelo fato de o autor não ter muita experiência com as tecnologias usadas, ele não poderia pegar de forma independente as tarefas e executá-las. Todas as instruções do que deveria ser feito na demanda estavam contidas com detalhes no *Plan.io* (Seção 2.3). Existiam alguns funcionários no setor responsáveis por fazer os testes das demandas, a fim de validar e verificar o que foi feito, antes de fazer a replicação no sistema de produção da empresa.

## **2.2 API (*Application Programming Interface*)**

Uma API, do inglês “*Application Programming Interface*”, é um conjunto de padrões que facilitam a troca de informações entre sistemas. Por meio de APIs, soluções ou serviços de software podem se comunicar uns com os outros, sem precisar saber como eles foram implementados. Isso simplifica o desenvolvimento de aplicações, gerando economia de tempo e dinheiro (REDHAT, 2017). Metaforicamente, pode-se imaginar uma API como o ambiente de um restaurante. O cliente realiza um pedido e o garçom o envia para a cozinha; um tempo depois, ele retorna com o que foi solicitado. Contudo, não é necessário que o cliente saiba como o alimento foi preparado.

Um conceito importante sobre APIs é o de *endpoints*, que são a forma pela qual um cliente/usuário consegue criar uma comunicação com um destino, normalmente um servidor. Essa comunicação se dá pelo endereço (URL - *Uniform Resource Locator*) informado pelo cliente para encontrar esse destino.

No Quadro 1, tem-se um exemplo de *endpoint* disponibilizado pela *Technolog*, no qual é possível acessar a funcionalidade *Onde Está o Técnico* (mais detalhes na seção 3.1).

#### **Quadro 1** - Exemplo Endpoint do uso da API na *Technolog*

1 <a href="http://solutions.technolog.com.br/ondeestaotecnico/">http://solutions.technolog.com.br/ondeestaotecnico/</a>
---

Fonte: Autor (2022)

O conceito de API foi importante no desenvolvimento de algumas tarefas na *Technolog*, principalmente em relação àquelas relacionadas à geolocalização. Por se tratar de uma empresa de logística e trabalhar com diversos veículos, uma das atividades realizadas pelo estagiário foi utilizar a API do Google Maps para se criar uma tela de interface com usuário, na qual era possível observar e analisar onde se encontram alguns técnicos para realizar a manutenção nos veículos dos clientes. Mais detalhes sobre isso podem ser vistos na Seção 3.1.

### **2.3 Gestão de Projetos com *Plan.io***

Para manter o controle de cada demanda a ser realizada, em auxílio com a metodologia Kanban, a *Technolog* utiliza um sistema de gestão de projetos denominado *Plan.io*<sup>1</sup>. Ele permite a estruturação do projeto em etapas, armazena e dissemina informações entre os membros da equipe, além de cruzar os dados para gerar relatórios sobre o andamento das atividades (PROJECTBUILDER, 2019). Nesse sistema, é possível ainda visualizar as características de cada solicitação, tais como nome, tipo, tempo, *status*, responsável por realizá-la, data de início e data de conclusão, dentre outros.

Na *Technolog*, o software *Plan.io* foi utilizado, principalmente, para a gestão das atividades desenvolvidas, ao mesmo tempo em que oferecia uma forma de documentação dos sistemas de software da empresa. Devido a entrada/saída de colaboradores, é necessário que se tenha as informações sobre as atividades

---

<sup>1</sup> <https://plan.io/>



realizadas bem descritas e detalhadas, a fim de se facilitar futuras manutenções. Na Figura 2, há o exemplo de uma tela de usuário do *Plan.io*, na qual é possível observar as demandas que foram solicitadas e algumas de suas características.

**Figura 2** - Tela do software para gestão de projetos *Plan.io*

ID	PROJETO	TIPO	ESTADO	PRIORIDADE	ASSUNTO	VERSÃO	ATRIBUÍDO A	ALTERADO	DATA DE FIM
6956	Darvin Portal	Verificar	Fechada	Alta	Demanda via HelpDesk - HD 2022137059	Darvin Portal - Banco de dados	João Vitor Costa de Góes	03/03/2022 09:00	02/03/2022
6938	Darvin Lite	Verificar	Fechada	Alta	Demanda via HelpDesk - HD 2022137037	Darvin Lite - Lite dev	João Vitor Costa de Góes	04/03/2022 09:47	11/03/2022
6851	Darvin Portal	Verificar	Fechada	Alta	Demanda via HelpDesk - HD 2022137022		João Vitor Costa de Góes	28/02/2022 12:02	25/02/2022
6837	Darvin Portal	BUG	Fechada	Normal	Portal New - Controle de Pneus - PHP 7.0	Darvin Portal - PHP 7	João Vitor Costa de Góes	23/02/2022 14:07	17/03/2022
6829	Darvin Portal	Nova Funcionalidade	Fechada	Normal	Portal New - Relatório Sumarizado - PHP 7.0	Darvin Portal - PHP 7	João Vitor Costa de Góes	24/02/2022 08:41	17/03/2022
6828	Darvin Portal	Nova Funcionalidade	Fechada	Normal	PORTAL NEW - PROCESSAMENTO DE FOLHA - PHP 7.0	Darvin Portal - PHP 7	João Vitor Costa de Góes	24/02/2022 09:27	17/03/2022

Fonte: Technolog (2022)

## 2.4 Banco de Dados

Um banco de dados é um conjunto de dados com um significado implícito, que representa uma porção do mundo real (ALVES, 2014). Os bancos de dados podem ser classificados quanto ao modelo de dados adotado por eles, sendo os principais, o modelo relacional e o modelo não-relacional. Durante o estágio, apenas o modelo relacional foi utilizado.

Segundo Alves (2014), um banco de dados relacional se caracteriza pelo fato de organizar os dados em tabelas (ou relações), formadas por linhas e colunas. Assim, essas tabelas são similares a conjuntos de elementos ou objetos, uma vez que relacionam as informações referentes a um mesmo assunto de modo organizado. Para manipular tais informações, é utilizada uma linguagem de consulta estruturada, chamada SQL (*Standard Query Language*), por meio da qual é possível

consultar, remover, adicionar, alterar e realizar diversas operações sobre os dados do banco.

No Quadro 2, encontra-se um exemplo de uma pesquisa em uma tabela chamada “Cad\_Clientes”, na qual é retornado o nome dos clientes cadastrados no estado de São Paulo e que possuem limites de compra acima de R \$3.000,00.

#### Quadro 2 - Exemplo de consulta SQL

```
1 SELECT
2     NOME_CLIENTE
3 FROM
4     CAD_CLIENTES
5 WHERE
5     ESTADO = 'SP' AND LIMITECOMPRA > 3000;
```

Fonte: ALVES (pág 130, 2014)

O uso da linguagem SQL ao longo do estágio se fez necessária para realização de algumas atividades, tais como a inserção, alteração e recuperação dos dados dos clientes nos bancos de dados da empresa *Technolog*. Também foi importante para verificar se determinados usuários cadastrados possuíam ou não a permissão para logar no sistema de forma múltipla (mais detalhes sobre essa funcionalidade podem ser vistos na Seção 3.2).

## 2.5 Tecnologias web

Nesta seção, são apresentadas as tecnologias web utilizadas durante o estágio, a saber, HTML (*HyperText Markup Language*), CSS (*Cascade Style Sheets*), JavaScript e PHP (*Hypertext Preprocessor*).

### 2.5.1 HTML - *HyperText Markup Language*

HTML (*HyperText Markup Language*) é uma linguagem de marcação de hipertexto e é o componente principal das páginas web existentes na Internet. Utilizando estruturas de marcação, também chamadas de *tags*, consegue-se organizar a informação de modo que os navegadores saibam a forma adequada de

apresentá-la ao usuário final.

As *tags* HTML servem para indicar marcações específicas dentro do código, como por exemplo, um título de um artigo, uma imagem, dentre outras (TABLELESS, 2011). Toda *tag* necessita de uma abertura e um fechamento e, com isso, é possível limitar cada bloco de informação, facilitando para o navegador distinguir uma imagem de um formulário, por exemplo. No Quadro 3, tem-se um exemplo de uma estrutura básica de uma página HTML.

### Quadro 3 - Estrutura básica de uma página HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Página Teste </title>
5     <meta charset='utf-8'>
6   </head>
7   <body>
8     <h1> Título Exemplo </h1>
9     <p> Este é um Parágrafo </p>
10  </body>
11 </html>
```

Fonte: Autor (2022)

No exemplo citado, na primeira linha de código, tem-se uma instrução para indicar aos navegadores que se trata de um arquivo escrito em HTML. Na sequência, é aberta a tag *html*, indicando que tudo que se encontra dentro dela está escrito em HTML. Existem duas *tags* importantes chamadas *head* e *body*. A tag *head* é responsável por definir certos parâmetros do documento, tais como a codificação utilizada sobre os caracteres, o título da página, entre outros. A tag *body*, por sua vez, delimita toda a informação que será apresentada diretamente ao usuário, utilizando outros tipos de *tags*.

A linguagem HTML foi utilizada no estágio, principalmente na criação de novas telas de usuário, assim como para reconstrução de algumas telas já existentes. O principal componente do HTML utilizado foi o formulário (*form*), necessário para manipular dados que os usuários forneciam ao fazer uso dos sistemas mantidos pela *Technolog*.

### 2.5.2 CSS - *Cascade Style Sheets*

CSS (*Cascading Style Sheets*) é uma linguagem de estilo utilizada para descrever a apresentação de um documento escrito em HTML. Com CSS, é possível descrever como os elementos são mostrados e formatados na tela (MOZILLA, 2021). Algumas alterações possíveis são tamanho, estilo de fonte, cor de fundo, animações e espaçamento de vários elementos distintos.

Uma estrutura de arquivo CSS básica é composta por um ou mais seletores e propriedades, tal como apresentado no Quadro 4.

**Quadro 4** - Estrutura básica de um código CSS

```
1 seletor {  
2     propriedade: valor;  
3 }
```

Fonte: Autor (2022)

Como é possível observar no código do Quadro 3 (Seção 2.5.1), não existe qualquer informação de CSS na página. Assim, o resultado final exibido no navegador do usuário seria equivalente ao da Figura 3.

**Figura 3** - Exemplo de página web sem aplicação de CSS



Fonte: Autor (2022)

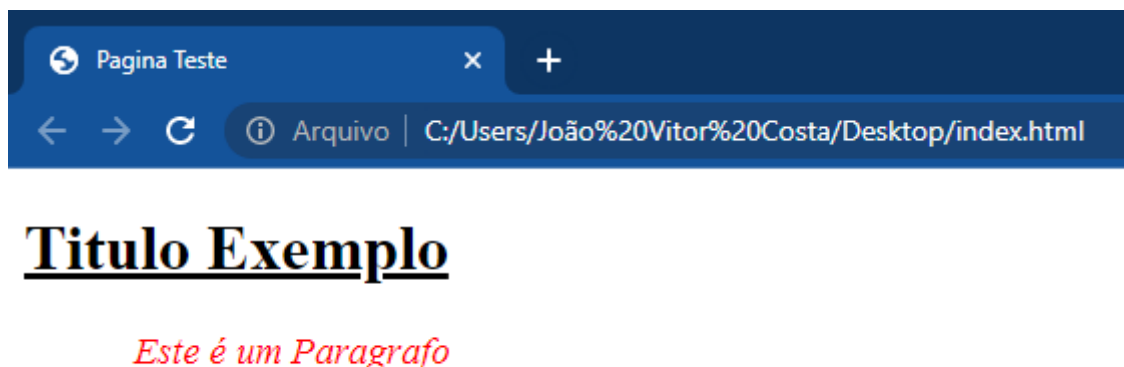
No Quadro 5, é possível ver o mesmo exemplo citado, porém com o uso de CSS, sendo aplicado na *tag p* com o aumento da fonte (*font-size*), definindo o estilo itálico (*font-style*), movendo o texto para a direita, ao aumentar a sua margem à esquerda (*margin-left*) e mudando sua cor para vermelho (*color*). Já na *tag h1*, é aplicada uma linha sublinhada com a propriedade *text-decoration*, que é responsável por decorar o texto com linhas. Na Figura 4, tem-se o resultado dessas estilizações aplicadas pelo código do Quadro 5.

Apesar de, no exemplo citado, o código CSS estar embutido no HTML, é possível separá-lo em um arquivo independente e depois referenciá-lo por meio da tag `<link href="estilos/estilos.css" rel="stylesheet">`, onde *href* determina o caminho onde o arquivo está localizado e *rel* o tipo de arquivo. Assim, é possível obter uma melhor modularização do conteúdo existente, facilitando a manutenção do código no futuro.

#### Quadro 5 - Exemplo de código de HTML com CSS embutido

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Pagina Teste </title>
5     <link href="estilos/estilos.css" rel="stylesheet" />
6     <meta charset='utf-8'>
7     <style>
8       p {
9         font-size: 19px;
10        font-style: italic;
11        margin-left: 55px;
12        color: red;
13      }
14
15      h1{
16        text-decoration: underline;
17      }
18    </style>
19  </head>
20  <body>
21    <h1> Titulo Exemplo </h1>
22    <p> Este é um Parágrafo </p>
23  </body>
24 </html>
```

Fonte: Autor (2022)

**Figura 4** - Exemplo de página com css

Fonte: Autor (2022)

O uso da linguagem CSS foi útil no desenvolvimento de novas telas de usuário para os sistemas da *Technolog*, assim como para as manutenções realizadas. Com CSS, era possível estilizar as páginas web dos sistemas da empresa, para que elas possuíssem a identidade visual da própria organização.

### 2.5.3 JavaScript

JavaScript (às vezes, denominada apenas JS) é uma linguagem leve, interpretada e baseada em objetos, mais conhecida como a linguagem de *script* para a web. JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando paradigmas tais como orientação a objetos, imperativo e declarativo (como por exemplo a programação funcional) (MOZZILA, 2021). É a linguagem responsável por aplicar comportamento dinâmico às páginas web, complementando as duas tecnologias anteriores (HTML e CSS) e, fazendo assim, com que haja maior interatividade com o usuário.

Para inserir JS nos documentos HTML, é necessário usar a *tag script*. O exemplo apresentado no Quadro 6 demonstra um uso simples de alerta ao usuário, com um código embutido no documento HTML. Na Figura 5, é possível notar o resultado de tal código no navegador do usuário. Analogamente ao que foi comentado sobre os arquivos CSS e sua possibilidade de serem importados, os

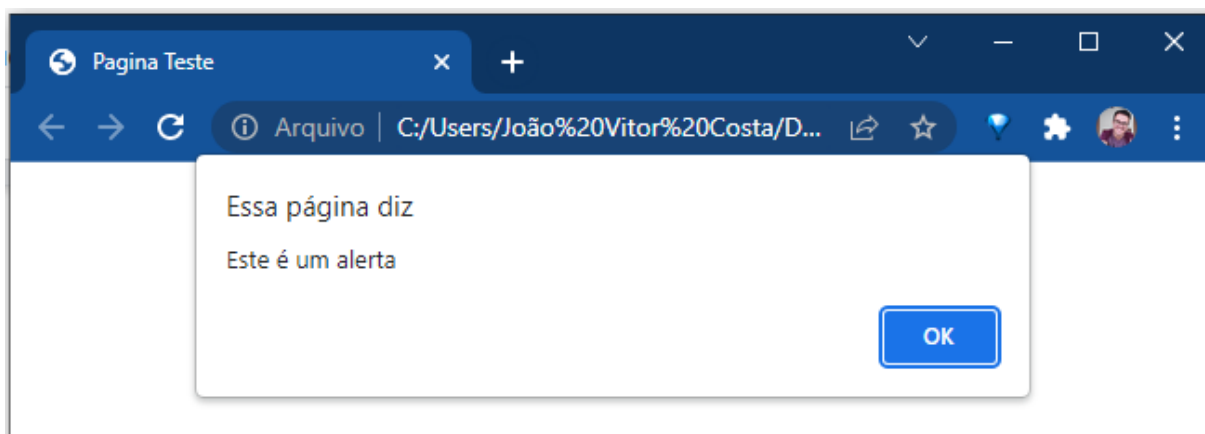
arquivos JavaScript também podem ser escritos de forma independente em arquivos separados e serem referenciados usando a *tag* `<script language="JavaScript" src="javascript/arquivo.js"></script>`, na qual se especifica a linguagem do arquivo com *language* e o caminho onde ele se encontra com o *src*.

#### Quadro 6 - Exemplo de código com JS

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Pagina Teste </title>
5
6     <meta charset='utf-8'>
7   </head>
8   <body>
21    <script type="text/javascript">
22      alert('Este é um alerta!')
23    </script>
24  </body>
25 </html>
```

Fonte: Autor (2022)

#### Figura 5 - Exemplo de alerta criado com JS



Fonte: Autor (2022)

A linguagem JavaScript foi utilizada ao longo do estágio, principalmente, nas atividades que faziam uso de APIs, pois era necessário ter uma lógica de programação para invocar a funcionalidade dessas APIs. Além disso, também era

necessário JS para verificar a consistência dos dados informados pelos usuários nos formulários web antes de enviar as informações para o servidor e serem salvas no banco de dados. É importante mencionar que existia também uma validação no *back-end* (servidor), a fim de verificar se os dados recebidos eram válidos e se poderiam ser salvos no banco de dados.

#### 2.5.4 PHP - *Hypertext PreProcessor*

PHP é uma linguagem de programação de ampla utilização, interpretada, que é especialmente utilizada para desenvolvimento web do lado do servidor (*back-end*). O objetivo principal da linguagem é permitir aos desenvolvedores escreverem páginas web, cujo conteúdo será gerado dinamicamente (PHP, 2021). O funcionamento dessa linguagem acontece no que é chamado *back-end*, ou seja, no computador servidor e não na máquina atual do usuário. Em uma página web, o código PHP é delimitado pelas instruções de início `<?php` e fim `?>`. Tudo que está contido entre essas *tags* será executado do lado do servidor, como por exemplo, uma consulta ao banco de dados.

Segundo Hostinger (2022), as principais vantagens de se utilizar PHP são a facilidade de aprendizado, comunidade ativa e ampla, o fato de ser *open-source*, dentre outros. No Quadro 7, é possível observar um exemplo de código PHP, cujo resultado, após executado, encontra-se na Figura 6.

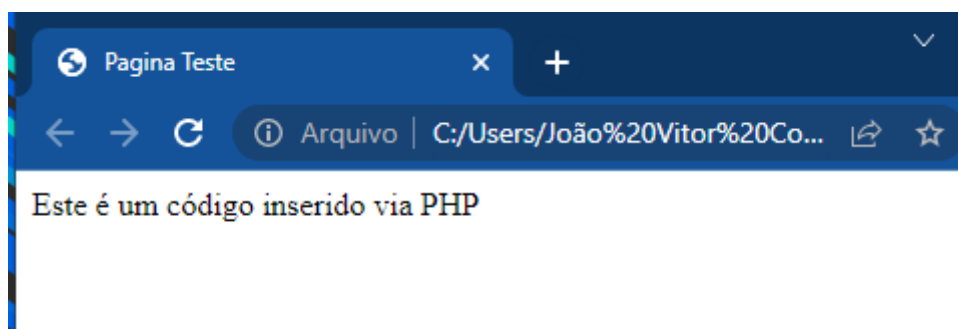
**Quadro 7** - Exemplo de código PHP

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Pagina Teste </title>
5     <meta charset='utf-8'>
6   </head>
20  <body>
21    <?php
22      echo "<p> Este é um código inserido via PHP </p>";
23    ?>
24  </body>
25 </html>
```

Fonte: Autor (2022)



**Figura 6** - Exemplo de página com PHP



Fonte: Autor (2022)

O PHP foi utilizado durante o estágio, principalmente, para realizar requisições de dados no banco de dados da empresa e para geração de relatórios. Mais a frente neste relatório, será detalhado o seu uso na atividade de validação de *login* para acessar o sistema Darwin.

### 3 ATIVIDADES DESENVOLVIDAS

Neste capítulo são descritas algumas das atividades que foram realizadas durante o estágio.

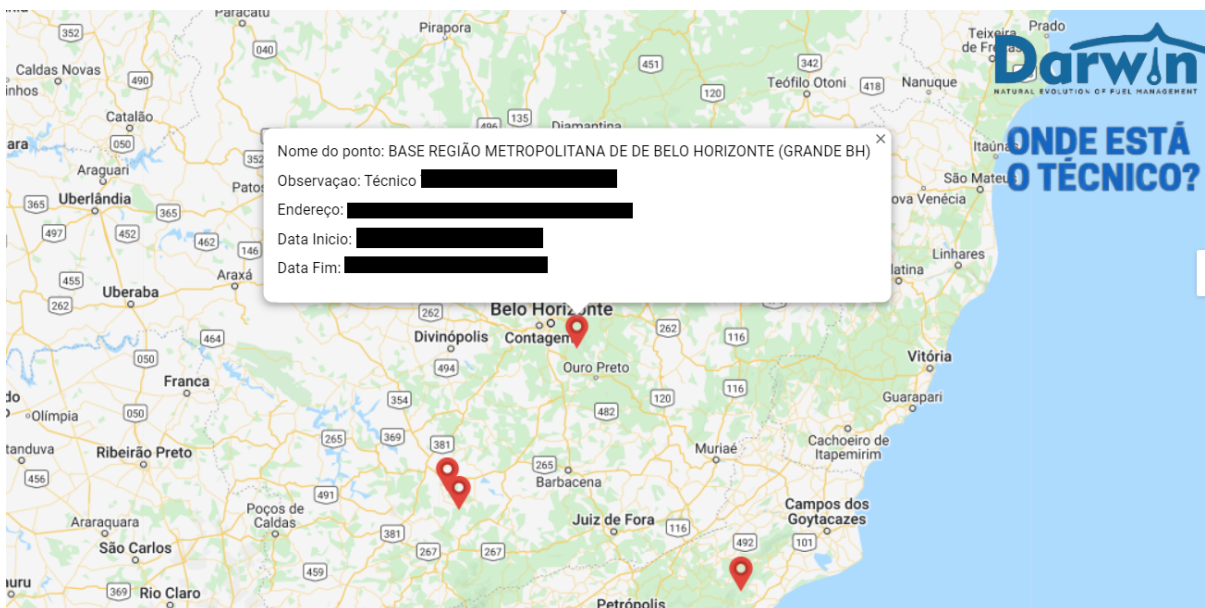
#### 3.1 Onde Está o Técnico?

Como a *Technolog* atende clientes no Brasil inteiro e possui um conjunto de técnicos especializados para fazer a manutenção dos equipamentos, é importante que, ao surgir algum problema, os motoristas saibam onde podem encontrar algum técnico disponível para resolvê-lo. Para isso, foi desenvolvida uma nova funcionalidade no sistema *Darwin*, denominada “Onde está o Técnico?”, que permite aos motoristas e aos gestores visualizarem a localização dos técnicos e as informações sobre como contatá-los.

Os técnicos não ficam em lugares fixos, exceto se é um ponto onde existe a movimentação de muitos caminhões de clientes da empresa. Os responsáveis por definir onde estarão os técnicos são os supervisores de gestão logística da empresa, que trabalham junto com os supervisores da área de manutenção e definem onde e por quanto tempo cada técnico deve atuar.

Os dados dos técnicos são cadastrados no banco de dados da empresa, em uma tabela com algumas informações essenciais, tais como detalhes referentes ao técnico e sua atual localização (longitude e latitude). Na Figura 7, há um exemplo de como é sinalizada a posição de um técnico no mapa, do ponto de vista do motorista. Para cada ponto, tem-se o nome do técnico, seu telefone, o nome daquele ponto e o período (data de início e fim) em que o técnico estará disponível naquele local. Algumas informações sigilosas referentes aos técnicos e seus serviços foram ocultadas a fim de preservar sua identidade.

**Figura 7** - Exemplo da página “Onde Está o Técnico?”



Fonte: Technolog (2022)

Para implementar tal funcionalidade, durante o estágio, foi utilizada a linguagem PHP para se desenvolver a tela do usuário com os códigos HTML embutidos no PHP, assim como, as requisições ao banco de dados usando SQL. No Quadro 8 é possível observar o código responsável pela consulta dos dados necessários no banco de dados.

**Quadro 8** - Exemplo de SQL Onde Está o Técnico

```

1  SELECT
2      NOME_POSTO, NOME_TECNICO, ENDERECO,
3      LATITUDE, LONGITUDE, DATA_INICIO, DATA_FINAL
4  FROM
5      TABELA_TECNICOS

```

Fonte: Autor (2022)

De posse das informações retornadas pelo banco de dados, na sequência, era usada a API do Google Maps, para que, através dos pontos de latitude e longitude, fosse inserido um “Pin” no mapa (um ícone que, ao ser clicado, traz todas as informações sobre aquele determinado local). Na Figura 8, encontra-se um trecho

de código responsável por criar o mapa e colocá-lo na tela para o usuário final.

**Figura 8** - Exemplo de código que gera o mapa “onde está o técnico?”

```

<script type="text/javascript">
  try
  {
    // define um ponto central ao iniciar o mapa e as opções que serão exibidas
    // como o tipo de mapa: Imagem Satélite, Relevo, etc.
    var latlng = new google.maps.LatLng(-15.777631, -47.930535);
    var myOptions = {
      zoom: 5,
      center: latlng,
      mapTypeId: google.maps.MapTypeId.ROADMAP
      //mapTypeId: google.maps.MapTypeId.TERRAIN
    };

    //cria o mapa e o atribui a um elemento do html
    var mapa = new google.maps.Map( document.getElementById("mapa2") , myOptions );

    // é chamada uma função que retorna os dados do banco
    var requisicaoDados = getTécnicos();

    // Para cada tecnico encontrado é criado um ícone no mapa
    for (i = 0; i < requisicaoDados.length; i++) {
      // este ícone é atribuído com base na latitude e longitude que é retornado do banco de dados
      const marker = new google.maps.Marker({
        position: new google.maps.LatLng(requisicaoDados[i].latitude, requisicaoDados[i].longitude),
        title: requisicaoDados[i].nome,
        map: mapa
      });

      // é criada uma "caixinha" de informações que contem os dados sobre o técnico e a localização do mesmo
      const conteudo = "<div style='color: black'> <p style='font-weight: bolder'> Nome do ponto: " +
        requisicaoDados[i].nome_ponto + " </p><p style='font-weight: bolder'> Observação: Técnico " +
        requisicaoDados[i].nome_tecnico + " </p><p style='font-weight: bolder'> Endereço: " + requisicaoDados[i].
        endereco + " </p><p style='font-weight: bolder'> Data Inicio: " + requisicaoDados[i].data_inicio + "
        </p><p style='font-weight: bolder'> Data Fim: " + requisicaoDados[i].data_final + " </p>";

      //é associado a caixinha de informações para aparecer sempre que se interagir com o marcador específico
      const infowindow = new google.maps.InfoWindow({
        content: conteudo
      });
    }
  }
}

```

Fonte: Technolog (2022)

Como é possível observar, é instanciada uma variável chamada `latlng` que recebe um par de coordenadas que será o ponto central do mapa. Outras configurações são definidas na variável `myOptions`, como o zoom do ponto inicializado anteriormente e o tipo de mapa que será mostrado. Na sequência, é criada uma variável `mapa`, e também é feita a requisição dos dados com a função `getTécnicos`. Em um laço de repetição para cada dado retornado da requisição anterior, é criado um marcador com as informações daquele laço e o mesmo é inserido no mapa.

A principal dificuldade encontrada durante a realização desta atividade foi descobrir como manusear o mapa e suas apresentações, que eram executadas através da linguagem JavaScript. Tal conhecimento foi sendo adquirido durante o tempo em que foi sendo construída a funcionalidade, assim como outras atividades no dia a dia da empresa.

### 3.2 Múltiplos Logins

Como o sistema Darwin opera com dados sigilosos dos clientes, dos motoristas e dos contratos que são fechados, é importante que se tenha um controle do que as contas de usuários podem fazer. Cada usuário tem uma conta cadastrada no sistema e cada conta tem suas permissões sobre o que a mesma pode executar, visualizar e realizar no Darwin.

Somada a essas permissões, a *Technolog* decidiu implementar no sistema Darwin uma forma de uma mesma conta não se autenticar no sistema mais de uma vez em um determinado período de tempo. Sendo assim, se um acesso já foi realizado, não é possível acessar o sistema através de outro computador ou dispositivo, simultaneamente.

No entanto, as contas de usuários responsáveis pela manutenção e desenvolvimento do sistema Darwin deveriam poder se logar de forma múltipla no sistema, devido a testes que devem ser executados. Logar de forma múltipla se refere à possibilidade de um mesmo login/usuário, utilizar os sistemas da *Technolog* em mais de um dispositivo ao mesmo tempo.

Para implementar tal funcionalidade foram criadas, na tabela de informações dos usuários cadastrados no banco de dados, duas colunas:

- Uma delas é referente à possibilidade de o usuário poder se conectar em vários dispositivos ao mesmo tempo. Trata-se de um valor para sinalizar se uma conta em específico pode fazer múltiplos *logins* ou não.
- A outra coluna seria corresponde a um *token* de identificação do último *login* de determinada conta. Por meio desse identificador, o sistema pode controlar e saber quando uma nova tentativa de autenticação é realizada.

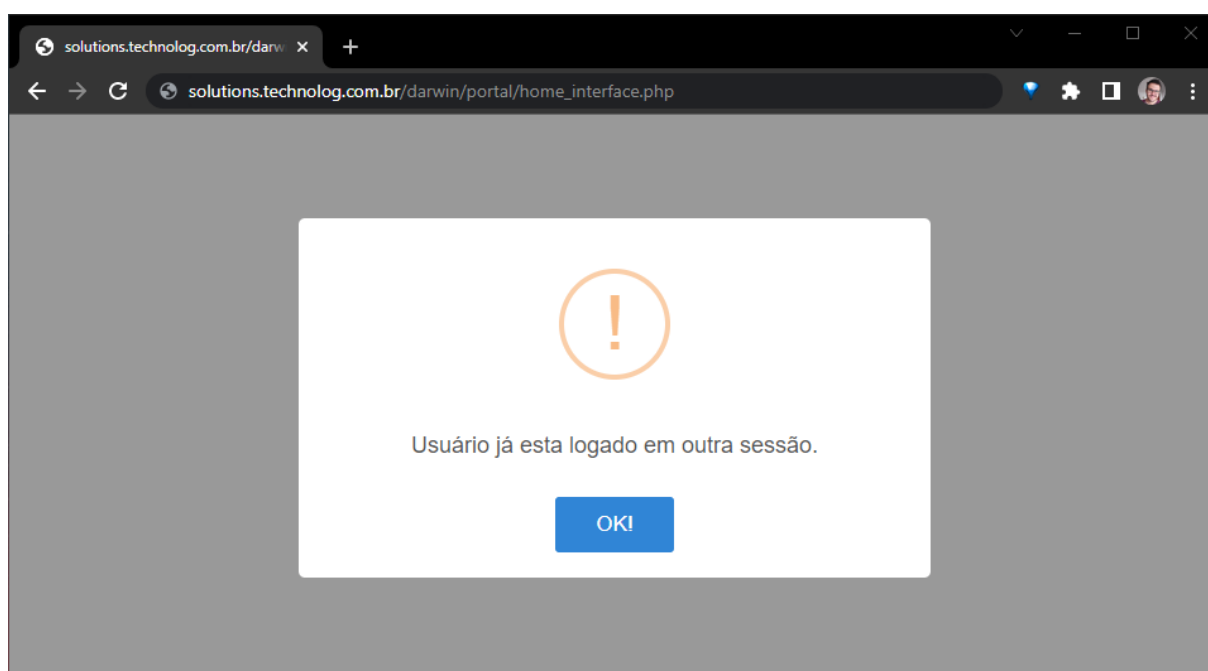
A cada nova tentativa de autenticação é gerado pelo navegador um número que identifica aquela sessão de acesso ao sistema. Esse identificador é, então, armazenado no banco de dados, sendo vinculado ao usuário que está tentando fazer o *login* no sistema.

Ao tentar logar no sistema por meio de outro dispositivo, um novo número é criado e, por meio dele, é feita a análise das permissões daquele usuário (por

exemplo, se a conta tem a permissão de múltiplos *logins*). Então, é verificado o *token* da nova tentativa com o armazenado no banco para “derrubar” ou não o *login* que se encontra ativo.

Na Figura 9, é possível observar a mensagem que é retornada a um usuário que não possui tal permissão para múltiplos *logins* e, ao entrar no sistema pela segunda vez de um outro dispositivo, o *login* que já se encontrava aberto é derrubado pelo sistema desabilitando, forçando o logoff/saída do sistema.

**Figura 9** - Mensagem de erro ao tentar múltiplos *logins* no sistema



Fonte: Technolog (2022)

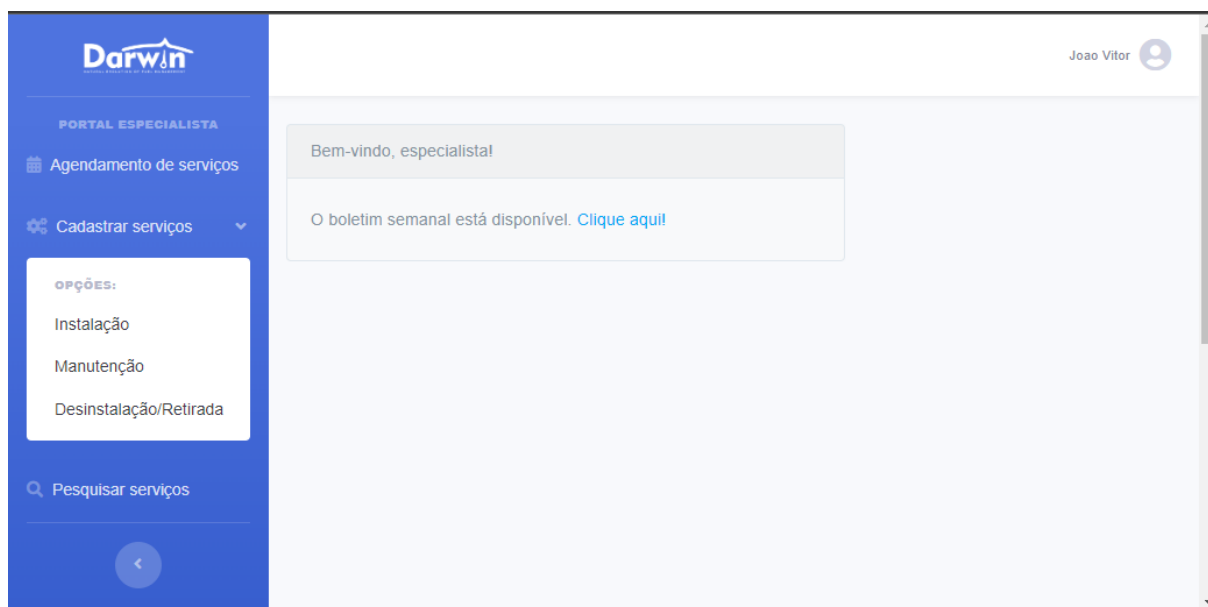
Um dos desafios de se implementar essa funcionalidade no sistema, foi o de decidir quais as maneiras adequadas de se criar um controle de acesso que fosse seguro, e que também garantisse com que o objetivo fosse alcançado.

### 3.3 Cadastro de Serviço de Instalação

O sistema Especialistas foi criado com o intuito de ajudar os técnicos com a questão da organização dos serviços que devem ser prestados e quanto ao histórico dos serviços que já foram realizados. Sua interface pode ser observada na Figura 10, contando com funcionalidades, tais como pesquisar os agendamentos de serviços, cadastrar serviços e pesquisar serviços.

Existem três tipos de cadastros que podem ser realizados no sistema: (i) *serviço de instalação*, por meio do qual os veículos são levados até os técnicos e eles adicionam as sondas e sensores nos motores, de modo que toda essa parte de equipamentos se conecte ao sistema Darwin; (ii) *serviço de desinstalação*: que ocorre quando uma empresa deixa de ser cliente da *Technolog*; e (iii) *serviço de manutenção*, que ocorre quando os aparelhos apresentam problemas e os dados não são enviados corretamente para o servidor (geralmente isso ocorre devido à queima de algum circuito e é necessário realizar a troca).

**Figura 10** - Portal Especialistas



Fonte: Technolog (2022)

A tarefa realizada pelo estagiário neste sistema consistiu na implementação da funcionalidade de cadastro de um serviço de instalação. Foi utilizada a linguagem PHP, juntamente com HTML e CSS, para as operações de captura e validação dos valores digitados pelo usuário no sistema, e a linguagem SQL, para armazenamento dos dados no banco de dados. É possível visualizar a tela de cadastro de um serviço na Figura 11, já com alguns campos preenchidos. O resultado desse cadastro de serviço pode ser visto na Figura 12.

**Figura 11** - Portal Especialistas Instalação

**PORTAL ESPECIALISTA**

Agendamento de serviços

Cadastrar serviços

Pesquisar serviços

## Instalação

Possibilita cadastrar e editar a instalação.

**Instalação**

Ordem serviço:\*  
000000229

Cliente:\*  
EMPRESA TESTE

Placa:\*  
TST7357

Local:\*  
LAVRAS

Data:\*  
06/05/2022

Possui:\*

<input checked="" type="checkbox"/> ALTO-FALANTE	Data DD/MM/AAAA	Quantidade 1
<input checked="" type="checkbox"/> ANÁLISE TÉCNICA	Data DD/MM/AAAA	Quantidade 1
<input checked="" type="checkbox"/> ATUALIZAÇÃO DO FIRMWARE DO COLETOR	Data DD/MM/AAAA	Quantidade 1

Fonte: Technolog (2022)



**Figura 12 - Portal Especialistas Instalação Resultado**

Data inicial: 06/05/2022 Data final: 06/05/2022

Dados atualizados em 06/05/2022

10 resultados por página

Cod.	Data serviço	Cliente	Local	Placa	Serviço	Instaladores	Última alteração
3699	06/05/2022	EMPRESA TESTE	TESTE	AHU7735	INSTALAÇÃO	TESTE ESPECIALISTA	06/05/2022 09:33:46
3700	06/05/2022	EMPRESA TESTE	TESTE7384	FQL4413	MANUTENÇÃO	TESTE ESPECIALISTA	06/05/2022 10:12:47
3705	06/05/2022	EMPRESA TESTE	LAVRAS	TST7357	INSTALAÇÃO	TESTE ESPECIALISTA	06/05/2022 16:24:41
Cod.	Data serviço	Cliente	Local	Placa	Serviço	Instaladores	Última alteração

Mostrando de 1 até 3 de 3 registros

Fonte: Technolog (2022)

Algumas dificuldades encontradas na realização dessa atividade foi como manter a integridade das informações antes de salvá-las no banco de dados, ou seja, verificar se todos os dados informados pelo usuário estavam preenchidos corretamente, e também entender a lógica de negócio por trás dos agendamentos para que todos os técnicos cumprissem com suas agendas e nenhum veículo ficasse carente de algum serviço.

#### 4 CONSIDERAÇÕES FINAIS

A realização do estágio supervisionado proporcionou ao autor inúmeras atividades que contribuíram para o seu crescimento profissional e pessoal. Dentre essas atividades, algumas foram o desenvolvimento do raciocínio lógico e a capacidade de entender e resolver problemas.

A experiência de atuar em uma empresa como a *Technolog* abriu os olhos do estagiário sobre como o mercado de trabalho da área de tecnologia funciona. Também quais características são essenciais para conseguir ter um bom desempenho e exercer de maneira excelente as habilidades que foram adquiridas em sala de aula. Disciplinas como “Introdução aos Algoritmos” e “Estruturas de Dados” foram essenciais para o desenvolvimento da capacidade de pensar em soluções para os problemas que surgiam; “Introdução a Banco de Dados” e “Sistemas Gerenciadores de Banco de Dados” ajudaram a entender o funcionamento de como eram armazenados os dados da empresa e a melhor forma de operá-los; “Engenharia de Software” e “Programação Orientada a Objetos” auxiliaram a aplicar e entender as melhores práticas de desenvolvimento. Todas as disciplinas citadas foram de suma importância para que o aprendizado fosse melhor aproveitado, visto que foi colocado em prática toda a teoria que foi aprendida na universidade.

Destaca-se ainda o desenvolvimento do estagiário não apenas no quesito técnico, mas também a respeito das chamadas *Soft Skills*, que são as habilidades pessoais, sociais e emocionais, que dizem respeito ao comportamento e à capacidade de interagir com pessoas. As atividades que foram desenvolvidas no estágio eram baseadas em outras tarefas que a equipe também realizava, assim, o trabalho de um afetava o outro diretamente. Logo, uma boa comunicação era necessária para que todos trabalhassem bem e soubessem o que acontecia na empresa. Devido a uma recente mudança na gestão do setor de desenvolvimento, começou a ser implantada novas metodologias ágeis, com isso diariamente eram feitas reuniões para se debater o que estava sendo realizado por cada desenvolvedor, assim como os *feedbacks* que eram dados a respeito do desempenho dos funcionários, as informações quanto aos prazos e ao ritmo de

trabalho foram importantes para melhorar a atuação da equipe.

Além de tudo, o estágio possibilitou o contato com novas tecnologias que não foram abordadas na graduação, levando assim, a uma maior qualificação do autor.

## REFERÊNCIAS

ALVES. Banco de dados. 2014. Disponível em:

<[ALVES, William Pereira. Banco de dados. São Paulo Erica 2014](#)>. Acesso em: 06 mar. 2022

ANUARIO CNT .Anuário CNT do Transporte 2021. Disponível em:

<<https://anuariodotransporte.cnt.org.br/2021/>>. Acesso em: 10 mar. 2022

BOLETIM CNT .Boletim Estatístico CNT 2019. Disponível em:

<<https://www.cnt.org.br/boletins>>. Acesso em: 10 mar. 2022

HOSTINGER.COM. O que é o PHP. 2022. Disponível em:

<<https://www.hostinger.com.br/tutoriais/o-que-e-php-guia-basico>>. Acesso em: 04 mar. 2022

INFOESCOLA .Transporte Rodoviário 2015. Disponível em:

<<https://www.infoescola.com/geografia/transporte-rodoviario/>>. Acesso em: 10 mar. 2022

LUMIS.Metodologias Ágeis. 2021 Disponível em:

<<https://www.lumis.com.br/a-lumis/blog/metodos-ageis.htm>>. Acesso em: 30 mar. 2022

MOZILLA.ORG. CSS. 2021. Disponível em:

<<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 04 mar. 2022

PHP.NET. Prefácio PHP. 2021. Disponível em:

<[https://www.php.net/manual/pt\\_BR/preface.php](https://www.php.net/manual/pt_BR/preface.php)>. Acesso em: 06 mar. 2022

PROJECTBUILDER. Sistema de Gestão de Projetos. 2019. Disponível em:

<<https://www.projectbuilder.com.br/blog/conheca-6-funcionalidades-essenciais-de-um-sistema-de-gestao-de-projetos/>>. Acesso em: 07 mar. 2022

REDHAT.COM. O que é uma API? 2017. Disponível em:

<<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>> Acesso em 25 fev. 2022.

SCRUM.ORG. Kanban Guide for Scrum Teams. 2021. Disponível em:

<<https://www.scrum.org/resources/kanban-guide-scrum-teams>>. Acesso em: 25 fev. 2022

TABLELESS.COM. O básico: O que é HTML? 2011. Disponível em:

<<https://tableless.com.br/o-que-html-basico/>> Acesso em 05 mar. 2022.