



BRUNO CARLOS DE MESQUITA NAVES

**RELATÓRIO DE ESTÁGIO EM DESENVOLVIMENTO
BACKEND NA EMPRESA DELTA GLOBAL S.A.**

LAVRAS-MG

2023

BRUNO CARLOS DE MESQUITA NAVES

**RELATÓRIO DE ESTÁGIO EM DESENVOLVIMENTO BACKEND NA
EMPRESA DELTA GLOBAL S.A.**

Relatório de estágio supervisionado 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. André Pimenta Freire
Orientador

LAVRAS-MG

2023

BRUNO CARLOS DE MESQUITA NAVES

**RELATÓRIO DE ESTÁGIO EM DESENVOLVIMENTO BACKEND NA
EMPRESA DELTA GLOBAL S.A.**

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

APROVADO em 20 de julho de 2023

Prof. Dr. Renata Teles Moreira (UFLA)

Me. Samuel Terra Vieira (Delta Global S.A.)

Prof. Dr. André Pimenta Freire

Orientador

LAVRAS-MG

2023

Dedico este trabalho a todos que confiaram em meu potencial, esposa, família e amigos.

Todo esforço e dedicação tem a sua recompensa.

AGRADECIMENTOS

A minha amada esposa Bruna que esteve sempre ao meu lado, pela paciência e apoio nos momentos mais difíceis. Ao meu pai Antônio Carlos e minha mãe Maria Teresinha pela educação e carinho, sem eles nada disso seria possível. Aos professores que tive nessa jornada, em especial a professora Rosário do ensino fundamental, sem a sua perseverança não teria chegado até aqui. A Delta Global S.A. pela confiança em meu trabalho. A Universidade Federal de Lavras e a todos os professores pelos conhecimentos transmitido. E a todos que de alguma forma contribuíram para minha formação pessoal e profissional.

RESUMO

O presente relatório descreve as experiências nas atividades desenvolvidas durante o período de estágio na empresa Delta Global S.A. na área de desenvolvimento de aplicações web como desenvolvedor *backend*, atuando diretamente com a área comercial e usuário final. A sede empresarial e operacional da empresa está situada na cidade de Porto Alegre-RS e sua filial de desenvolvimento na cidade de Lavras-MG. Sua atuação está relacionada à prestação de serviços de assistência e monitoramento no ramo de transportes por meio de tecnologias de geolocalização embarcada e *software*. As principais tecnologias utilizadas no estágio em desenvolvimento de aplicações *web* foram: PHP, MySQL, Javascript, HTML, CSS, a utilização do *framework* Codeigniter, arquitetura MVC e o Padrão de Camadas, Git. O estágio consistiu na implementação *backend*, utilizando banco de dados e a construção de algoritmos eficientes a fim de propor soluções, melhorias e novas funcionalidades aos sistemas da empresa.

Palavras-chave: PHP. Desenvolvimento de Aplicações Web. Back-end. Algoritmos.

ABSTRACT

This report describes the experiences during an internship period at Delta Global S.A. as a back-end developer in the web application development field, working directly with the commercial and end-user areas. The company's headquarters and operations are located in Porto Alegre-RS, with a development branch in Lavras-MG. Its activities are related to providing assistance and monitoring services in the transportation sector through embedded geolocation technologies and software. The main technologies used during the internship in web application development were PHP, MySQL, Javascript, HTML, CSS, Codeigniter framework, MVC architecture, Pattern Layer, and Git version control system. The internship involved back-end implementation, using database and efficient algorithm construction in order to propose solutions, improvements, and new features to the company's systems.

Keywords: PHP. Web Application Development. Back-end. Algorithm.

SUMÁRIO

1	INTRODUÇÃO	9
2	LOCAL DO ESTÁGIO	10
2.1	Produtos	10
2.2	Fábrica de software.....	12
3	ARQUITETURA DE SOFTWARE E TECNOLOGIAS UTILIZADAS.....	14
3.1	Arquitetura MVC.....	14
3.2	HTML.....	16
3.3	CSS.....	16
3.4	Javascript	18
3.5	PHP.....	19
3.6	MySQL.....	21
3.7	Framework CodeIgniter.....	21
3.8	IDE PHPStorm	22
4	ATIVIDADES DESENVOLVIDAS	23
4.1	Atualização de Base de Veículos.....	23
4.2	Cotação no sistema	26
4.3	Implementação de Tabelas Fixas.....	27
4.4	Refatoração e implementação de Camada de Service	28
5	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	33

1 INTRODUÇÃO

O estágio apresentado neste relatório foi realizado na empresa Delta Global S.A., tendo iniciado em 22 de novembro de 2022, seguindo um modelo híbrido de trabalho, com 3 dias de trabalho remoto e 2 dias presenciais. O objetivo principal do estágio foi o desenvolvimento de sistemas utilizando a linguagem de programação PHP. Durante esse período, o aluno teve a oportunidade de aplicar os conhecimentos adquiridos em sala de aula, como gestão de projetos, desenvolvimento e arquitetura.

O desenvolvimento *backend* é a etapa do processo de desenvolvimento de software que se dedica à criação de aplicativos web e sistemas que operam na camada do servidor, ou seja, na "parte de trás" do site ou aplicativo. Essa fase envolve a implementação e manutenção de bancos de dados, servidores, aplicativos e lógica de negócios que são executados no servidor e são responsáveis por gerar as respostas apresentadas aos usuários na interface do aplicativo.

Na posição de desenvolvedor *backend*, o estagiário desempenhou suas atividades com o objetivo de propor melhorias e soluções para o crescimento da empresa. Todo o processo foi acompanhado por um desenvolvedor sênior, e o estagiário participou frequentemente de reuniões para discutir os processos de desenvolvimento e análise dos requisitos.

Este documento está estruturado em capítulos. O Capítulo 2 descreve o local de trabalho e seus principais produtos, além de contextualizar sobre a Fábrica *Software* e o ambiente de trabalho. O Capítulo 3 apresenta as tecnologias utilizadas no desenvolvimento das atividades. O Capítulo 4 descreve as atividades desenvolvidas pelo estagiário durante o período. Por fim, o Capítulo 5 traz as considerações finais sobre o estágio, destacando o aprendizado adquirido na empresa Delta Global S.A. e a conexão entre os conhecimentos obtidos no curso de Sistemas de Informação na Universidade Federal de Lavras e as práticas desenvolvidas.

2 LOCAL DO ESTÁGIO

A Delta Global S.A. é uma empresa renomada no setor de transportes, destacando-se por seus serviços e tecnologia. Seu principal enfoque é o monitoramento e assistência veicular (DELTA, 2023a). Mais recentemente, a empresa expandiu sua atuação para o ramo de assistência a sinistros, atendendo as principais seguradoras e locadoras do país. A sede administrativa, base de operações e fábrica de hardware estão localizadas na cidade de Porto Alegre-RS, enquanto a fábrica de *software* está situada em Lavras, MG.

Fundada em 2015, a empresa rapidamente conquistou grande prestígio no cenário empresarial, contando com clientes de destaque, tais como JBS, Randon, Sampo Seguros, Unidas, entre outros (DELTA, 2023b). Atualmente, a Delta Global S.A. atende aproximadamente 500 mil veículos em todo o território nacional, com a previsão de alcançar a marca de 1 milhão de veículos atendidos até o final de 2023.

Entre os serviços oferecidos pela empresa, destaca-se a comercialização de planos de assistência para veículos. A Delta Global S.A. possui uma ampla base de dados de prestadores de serviços em todo o território nacional, o que permite atender de maneira ágil e com qualidade seus clientes. Esses planos estão disponíveis tanto para pessoas jurídicas quanto para pessoas físicas, e são comercializados por meio de corretores internos e externos. Além disso, para pessoas físicas, há a opção de contratar o serviço diretamente, sem a necessidade de intermediação de corretores, por meio do sistema Delta 24horas.

2.1 Produtos

A empresa Delta Global S.A. oferece uma variedade de produtos relacionados à assistência veicular, os quais são divididos em plataformas separadas para atender de forma mais eficiente seu público-alvo.

O principal sistema da empresa é chamado de Delta Assist, nele são gerenciados todos os aspectos relacionados aos clientes, contratos, veículos e aspectos financeiros da empresa. O sistema é composto por camadas de acesso, que incluem as áreas administrativa, operacional, de clientes e de prestadores de serviços. Os produtos

oferecidos no Delta Assist são direcionados a clientes com grandes frotas, como transportadoras, locadoras de veículos, cooperativas e seguradoras.

O sistema é completamente integrado. O cliente, por meio de telefone ou WhatsApp, utilizando a tecnologia de *chatbot*, entra em contato com a central de operações. A central valida os dados informados pelo cliente e, por meio da coleta de informações, realiza uma geolocalização para identificar o local do acionamento. Dessa forma, o operador pode localizar o prestador de serviços mais próximo e iniciar as negociações para a realização do serviço.

Conforme mencionado anteriormente, o sistema da Delta Global S.A. é totalmente integrado. No momento do acionamento, tanto o cliente quanto o prestador de serviço são informados sobre a ocorrência. A empresa valoriza um atendimento de excelência e o tempo é cuidadosamente cronometrado, desde o início até a conclusão do serviço, gerando informações que são utilizadas para medir a qualidade do atendimento.

Além do sistema Assist, a Delta Global S.A. também oferece o Delta 24Horas, que disponibiliza serviços de assistência. A diferença está no tamanho da frota atendida. O Delta 24Horas é direcionado a pessoas físicas e permite a aquisição dos serviços de forma rápida e simplificada por meio da plataforma (DELTA24HORAS, 2023).

O Delta Fleet, por sua vez, é a maior e mais abrangente plataforma de gestão de frotas da América Latina. Além de fornecer assistência 24 horas (DELTA, 2023c), essa plataforma possui rastreamento inteligente com tecnologia proprietária e monitoramento em tempo real. Além disso, oferece telemetria, que fornece informações completas sobre o desempenho do veículo, maior controle das manutenções, abastecimento e desempenho dos motoristas.

Desenvolvido com tecnologia nacional, o Delta Safe é um equipamento personalizável que atende às necessidades específicas do cliente. Com até 40 sensores instalados no veículo, é possível obter informações precisas em tempo real e detectar possíveis tombamentos do veículo, graças ao sensor de tombamento desenvolvido pela Delta (DELTA, 2023d).

A plataforma Historicar é responsável pela busca de dados públicos e privados de veículos, sendo o maior e mais completo sistema de busca do segmento (HISTORICAR, 2023). Esse sistema é capaz de obter informações sobre veículos sinistrados e remontados, multas e infrações, se o veículo foi vendido em leilão, além dos dados do proprietário atual.

2.2 Fábrica de software

Ao ingressar na empresa Delta Global S.A., o colaborador passa por um processo de integração conhecido como *Onboarding*. Esse processo tem a finalidade de integrar o novo colaborador à equipe. Durante dois dias, são apresentados todos os setores da empresa e suas respectivas formas de trabalho. Nessa apresentação, são abordadas as metas, objetivos e processos da empresa como um todo.

A fábrica de *software* da empresa é dividida em *Squads* ou seja, grupos de colaboradores responsáveis por realizar as atividades, que incluem as áreas de Operações, Comercial, Financeiro, *Customer Success*, Manutenção e Plataformas. Cada *Squad* possui um *Tech Lead* (Líder Técnico) e desenvolvedores com diferentes níveis de experiência, como júnior, pleno e sênior.

Segundo Sommerville (2011, p. 3), "um sistema de *software* desenvolvido profissionalmente é, frequentemente, mais do que apenas um programa". Inicialmente, ele depende de um propósito, e em seguida são necessários requisitos bem definidos para garantir um desenvolvimento de qualidade. Nesse contexto, a empresa adota uma etapa de trabalho chamada maternidade, na qual surgem ideias para a implementação de funcionalidades ou criação de novos produtos.

A Delta Global utiliza o modelo kanban para gerenciar seus projetos na fábrica de software. O modelo kanban é um sistema visual de gerenciamento do fluxo de trabalho, dividido em etapas, como "a fazer", "em desenvolvimento" e "feito". Conforme o projeto avança, ele passa de uma etapa para outra. Conforme Anderson (2011, p. 7) afirma, o kanban permite a melhoria contínua do processo específico com o mínimo de resistência à mudança, mantendo um ritmo sustentável para os trabalhadores.

A análise de sistemas é a investigação de um problema que a empresa tenta resolver por meio de um sistema de informação (LAUDON, J. P & LAUDON, K. C, 2023, p. 498). Na Delta Global S.A.. Após a etapa da maternidade, segue-se para a priorização, que consiste em definir o esforço e a equipe necessários para o desenvolvimento. A etapa seguinte é o desenvolvimento, na qual se inicia o processo de programação, contabilizando o tempo desde o início até a entrega. Esse tempo é utilizado para métricas de qualidade e futuras auditorias internas. Após a etapa de desenvolvimento, avança-se para os testes e, uma vez aprovado pelos analistas, o desenvolvimento é liberado para produção.

Quando o desenvolvimento é liberado para produção, é criada a chamada PR (*Pull Request*), que passa por outro processo chamado revisão. Nessa etapa, os desenvolvedores sêniores avaliam a qualidade do código, além de analisarem possíveis pontos críticos que possam gerar falhas em desenvolvimentos paralelos.

Após todo o processo de verificação, esse desenvolvimento está apto para ser integrado ao produto final, seja para corrigir uma falha ou implementar um novo produto. Em seguida, avança-se para a validação em produção e o desenvolvimento é arquivado.

Todo o processo de desenvolvimento é gerenciado utilizando a ferramenta GIT, que é um sistema de controle de versão distribuído, usado para rastrear mudanças em arquivos e coordenar o trabalho colaborativo. O estagiário ao iniciar seus trabalhos buscou maior conhecimento sobre a utilização do GIT, por meio de cursos oferecidos pela Universidade além claro, do convívio com os colaboradores do setor de desenvolvimento, do qual obteve grande conhecimento.

O dia a dia na fábrica de *software* consiste em uma reunião diária matinal, na qual são discutidos todos os desenvolvimentos em andamento e finalizados. Além disso, é um momento importante para esclarecer dúvidas e promover o intercâmbio de conhecimento. Essa reunião é realizada separadamente em cada *Squad*, em horário previamente agendado, para permitir a participação do gestor de projetos, do gerente de tecnologia e do analista responsável pelo *Squad*.

O ambiente de trabalho na Delta Global é caracterizado por uma atmosfera de amizade e cooperação entre os desenvolvedores. São três dias em modelo home office e dois dias presenciais, sendo estes últimos uma oportunidade de aprendizado, pois a interação com desenvolvedores experientes contribui para o crescimento pessoal e profissional.

A Delta Global S.A. possui um fluxo constante de melhorias devido à sua imaturidade em processos de tecnologia. Por ser uma empresa jovem, os métodos de gestão de projetos estão em estágio inicial. Recentemente, foi implantada a metodologia *Sprint*, que consiste em um processo ágil que divide grandes projetos em etapas menores (RUNRUN, 2023). Essa metodologia facilita o acompanhamento das atividades, permitindo estimar prazos e promover a colaboração entre os *Squads*.

Com a implantação dessa metodologia, é possível manter um ritmo de atualizações mais frequente, em vez de períodos pré-determinados.

3 ARQUITETURA DE SOFTWARE E TECNOLOGIAS UTILIZADAS

Neste capítulo, serão apresentadas as tecnologias utilizadas durante o período de estágio, as quais a empresa Delta Global adota em seus sistemas de forma geral. Serão abordadas informações sobre a linguagem de programação utilizada, o banco de dados e a arquitetura de *software* empregados pela empresa.

3.1 Arquitetura MVC

O padrão MVC (*Model-View-Controller*) é uma arquitetura de software amplamente adotada que tem como objetivo principal separar as responsabilidades de um sistema em três componentes distintos: A *Model*, a *View* e a *Controller*. Essa separação permite uma organização eficiente das tarefas do sistema, possibilitando o desenvolvimento, teste e manutenção independentes de cada componente.

A *Model*, como componente central, é responsável pela manipulação dos dados e pelas regras de negócio do sistema. Ele representa o estado atual do sistema e desempenha um papel fundamental na garantia da consistência dos dados e na integridade das informações. Importante ressaltar que a *Model* é independente da camada de apresentação e pode ser utilizado em diferentes interfaces de usuário.

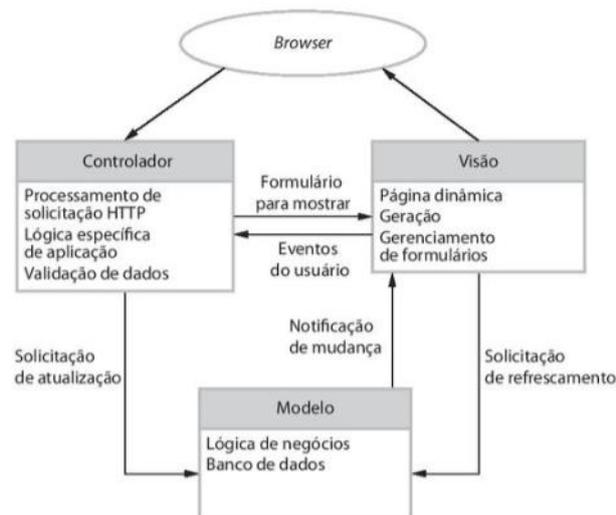
A *View*, por sua vez, é a camada de apresentação do sistema, sendo responsável pela exibição dos dados da *Model* para o usuário. Sua principal função é proporcionar uma apresentação visualmente atraente e organizada das informações. Vale destacar que a *View* é independente da lógica de negócio e pode ser modificada sem afetar o restante do sistema.

Já a *Controller* é o componente responsável por estabelecer a conexão entre a *Model* e a *View*, gerenciando as interações do usuário com a interface gráfica. Ele recebe as requisições do usuário, processa as informações e atualiza o estado da *Model* conforme necessário. A *Controller* também é responsável por garantir a integridade e a consistência dos dados, comunicando as alterações ao componente de *View* para que as informações atualizadas sejam exibidas adequadamente.

Ao adotar o padrão MVC, as empresas podem obter diversos benefícios, tais como uma maior modularidade do sistema, facilidade na manutenção e reutilização de componentes, além de uma separação clara entre as responsabilidades de cada parte do

sistema. Isso contribui para o desenvolvimento de software de qualidade, permitindo uma evolução mais eficiente e flexível ao longo do tempo. A Figura 1, mostra o funcionamento da estrutura MVC.

Figura 1: Arquitetura de aplicação Web usando o padrão MVC



Fonte: SOMMERVILLE, 2011, p. 110.

A arquitetura MVC é amplamente adotada por diversos frameworks PHP, tais como Laravel, Symfony e CodeIgniter, entre outros. Esses frameworks já incorporam o padrão MVC em sua estrutura, o que facilita sua utilização. No entanto, também é possível desenvolver o padrão MVC diretamente pelo desenvolvedor. Entre as vantagens dessa arquitetura, destacam-se a agilidade na atualização da interface, a facilidade de manutenção do código e a possibilidade de implementar camadas de segurança, o que aumenta a confiabilidade do sistema.

Apesar das vantagens mencionadas, o padrão MVC apresenta algumas desvantagens. Uma delas é a clara separação entre as camadas, que nem sempre é observada em sistemas que implementam esse padrão. Por exemplo, é comum encontrar a camada *controller* exercendo funções da camada *model*, o que pode comprometer a estrutura do sistema. Além disso, o desempenho pode ser afetado devido aos múltiplos níveis de processamento e interpretação exigidos por cada serviço solicitado nas diferentes camadas do padrão MVC.

3.2 HTML

HTML (*HyperText Markup Language*) Linguagem de Marcação de hipertexto, é uma linguagem utilizada em sistemas web e também em sistemas *desktop* graças ao avanço das tecnologias mobile. Hipertexto é um documento ou sistema formado por distintos blocos de informação (dados, textos, imagens, vídeos, sons) interligados por elos de associação (FLATSCHART, 2011, p. 17). Ela define a estrutura e o significado dos elementos do conteúdo, permitindo que os navegadores possam renderizá-los visualmente para o usuário. O HTML é uma linguagem baseada em *tags*, onde cada *tag* define um elemento específico do conteúdo.

A estrutura básica de um documento HTML vista na figura 2, consiste em um elemento `html` que contém dois elementos principais: *head* e *body*. O elemento *head* contém informações importantes sobre o documento, como o título da página, *links* para arquivos CSS e JavaScript, *meta tags*, entre outras informações. O elemento *body* contém o conteúdo principal do documento, como texto, imagens, vídeos e outros elementos.

Figura 2: Estrutura de uma página HTML

```
<html>
  <head>
    <title>Minha página</title>
  </head>
  <body>
    <h1>Bem-vindo à minha página</h1>
    <p>Esta é uma página de exemplo.</p>
  </body>
</html>
```

Bem-vindo à minha página

Esta é uma página de exemplo.

Fonte: Do autor (2023)

3.3 CSS

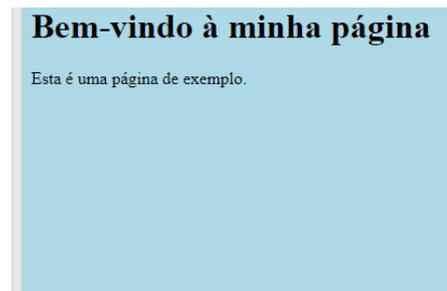
CSS (*Cascading Style Sheets*) é uma linguagem de estilo utilizada para controlar a apresentação visual de documentos HTML. Ela permite separar a estrutura e o conteúdo do documento da sua apresentação visual, possibilitando a criação de sites mais atraentes e funcionais. O CSS oferece diversos recursos para controlar cores, fontes, tamanhos, margens, bordas, espaçamentos, posicionamento, animações, entre outros aspectos visuais de uma página *web*.

Uma das grandes vantagens do CSS é que ele permite a criação de estilos que podem ser aplicados a várias páginas do site, tornando o desenvolvimento mais eficiente e facilitando a manutenção do código. Com o CSS, é possível criar *layouts* fluidos e responsivos que se adaptam a diferentes dispositivos e tamanhos de tela, tornando o site mais acessível e amigável ao usuário.

O CSS é uma linguagem de fácil aprendizado, especialmente quando se tem um conhecimento prévio de HTML. Há diversas ferramentas e frameworks disponíveis que podem ajudar a simplificar o processo de criação de estilos, além de muitos recursos online que podem auxiliar no aprendizado e solução de problemas. Ao dominar o CSS, é possível criar sites visualmente atraentes e profissionais que oferecem uma boa experiência ao usuário.

Figura 3: Utilização de CSS

```
<html>
<head>
  <title>Minha página</title>
  <style>
    body {
      background-color: lightblue;
    }
  </style>
</head>
<body>
  <h1>Bem-vindo à minha página</h1>
  <p>Esta é uma página de exemplo.</p>
</body>
</html>
```



Fonte: Do autor (2023)

Para definir uma cor de fundo para o corpo da página, podemos adicionar um bloco de estilo CSS dentro da *tag* <head> do documento, como mostrado na figura 3, o bloco de estilo define que o elemento body (corpo da página) terá uma cor de fundo *lightblue*. Ao atualizar a página no navegador, o fundo da página será exibido com a cor especificada. Esse é um exemplo simples, mas mostra como o CSS pode ser utilizado para controlar a aparência visual de uma página HTML.

Apesar de estar na posição de desenvolvedor *backend*, no período de estágio foi necessário utilizar de conhecimentos de *frontend* para a confecção dos trabalhos desenvolvidos e explicados no capítulo 4. As criações de telas foram necessárias para o aceleramento do processo para entrega ao cliente.

3.4 Javascript

JavaScript é uma linguagem de programação que permite adicionar interatividade e dinamismo a páginas *web* (MOZILLA, 2023a). Ela é utilizada principalmente em navegadores *web* para criar scripts que manipulam elementos HTML, respondem a eventos do usuário e realizam chamadas a servidores *web* para buscar ou enviar informações. O JavaScript é uma linguagem interpretada, o que significa que ela é executada diretamente no navegador do usuário sem a necessidade de compilação.

Figura 4: Código Javascript

```
const para = document.querySelector('p');

para.addEventListener('click', atualizarNome);

function atualizarNome() {
  var nome = prompt('Insira um novo nome');
  para.textContent = 'Jogador 1: ' + nome;
}
```

Fonte: MOZILLA, 2023b

Uma das características mais proeminentes do JavaScript é a sua capacidade de manipular a estrutura e o conteúdo de uma página HTML em tempo real. Essa habilidade possibilita a criação de interações e animações complexas diretamente no navegador, sem a necessidade de recarregar a página. Além disso, o JavaScript é amplamente utilizado em conjunto com outras tecnologias, como CSS e HTML5, para desenvolver jogos, aplicativos *web* e outras aplicações interativas.

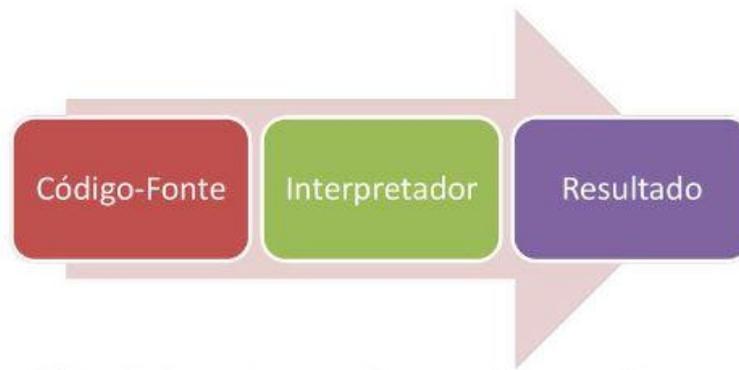
Além do uso no cliente, a linguagem JavaScript também é empregada em servidores web, por meio de frameworks como o Node.js. Essa abordagem permite que o código JavaScript seja executado tanto no cliente quanto no servidor, o que viabiliza a criação de aplicações web escaláveis e de alto desempenho.

A utilização do Javascript foi a que mais agregou conhecimento ao estagiário, devido a sua pouca experiência com a linguagem. Todos os trabalhos desenvolvidos tiveram sua parcela, feita com a linguagem javascript, isto tornou a aplicação mais dinâmica e foi decisivo na solução dada ao trabalho de cotação no sistema, proposto no capítulo 4.

3.5 PHP

O PHP (PHP: *Hypertext Preprocessor*) é uma linguagem de programação interpretada, o que significa que, durante a execução do programa, ocorre a validação da codificação pelo interpretador (FERREIRA, 2020, p. 9). A Figura 5 ilustra os estágios de uma linguagem interpretada, em que o script é executado e o interpretador realiza a validação da codificação, retornando o resultado correspondente.

Figura 5: Linguagem Interpretada



Fonte: FERREIRA D. R., 2020, p. 9.

A linguagem PHP é amplamente utilizada no desenvolvimento web devido à sua simplicidade. Sua sintaxe lembra as linguagens C, Java e Perl, o que contribui para uma curva de aprendizado mais rápida. O PHP permite mesclar scripts dentro do código HTML, sendo necessário utilizar a extensão ".php" para os arquivos, por exemplo, "index.php". Desde a sua versão 5, o PHP passou por uma evolução significativa com a implementação do paradigma de Orientação a Objetos. Na versão 8, é possível construir aplicações robustas e confiáveis, comparáveis às linguagens mais amplamente adotadas, como o Java.

Os scripts PHP são delimitados pelas *tags* "<?php" para indicar o início e "?>" para indicar o fim. Conforme ilustrado na figura 6, o código PHP é utilizado dentro do código HTML, sendo que as variáveis são representadas pelo símbolo de cifrão (\$), seguido pelo nome da variável.

Figura 6: Utilizando PHP e HTML

```

<!DOCTYPE html>
<html>
<body>

<?php

$nome_instituicao = "Universidade Federal de Lavras";
$nome_aluno = "Bruno Carlos";
$curso = "Sistemas de Informação"

?>

<h1>Utilizando PHP e HTML</h1>
<ul>
  <li>Instituição de Ensino: <?php echo $nome_instituicao; ?></li>
  <li>Aluno: <?php echo $nome_aluno; ?></li>
  <li>Curso: <?php echo $curso; ?></li>
</ul>

</body>
</html>

```

Utilizando PHP e HTML

- Instituição de Ensino: Universidade Federal de Lavras
- Aluno: Bruno Carlos
- Curso: Sistemas de Informação

Fonte: Do autor (2023)

A empresa Delta Global atualmente utiliza a versão 7.4 da linguagem PHP em seus sistemas. Essa versão oferece recursos mais robustos para trabalhar com orientação a objetos. É possível realizar a tipagem de variáveis e parâmetros nos métodos das classes, além de especificar o tipo de retorno de um método.

Um exemplo simples de uma classe é apresentado na Figura 7. A classe Pessoa possui duas variáveis privadas, uma do tipo *string* e outra do tipo *int*. No construtor da classe, os atributos são definidos como parâmetros, especificando o tipo de cada um. Dessa forma, se um tipo diferente do especificado for passado, ocorrerá um erro de tipo. A evolução da linguagem PHP tem sido significativa, em grande parte devido ao fato de ser uma linguagem de código aberto (*Open Source*).

Figura 7: Classe PHP

```

<?php
class Pessoa {
    private string $nome;
    private int $idade;

    public function __construct(string $nome, int $idade)
    {
        $this->nome = $nome;
        $this->idade = $idade;
    }

    public function getNome() :string
    {
        return $this->nome;
    }

    public function getIdade() :int
    {
        return $this->idade;
    }
}

$ Pessoa = new Pessoa("Bruno", 37);

echo $ Pessoa->getNome();
echo $ Pessoa->getIdade();

```

Fonte: Do autor (2023)

3.6 MySQL

O MySQL é um sistema gerenciador de banco de dados amplamente utilizado na indústria. Ele se destaca por utilizar a linguagem SQL, que é uma linguagem abrangente para definição de dados, consultas e atualizações em bancos de dados (ELMASRI & NAVATHE, 2005, p. 149).

De acordo com a documentação oficial, o MySQL é reconhecido como um banco de dados de código aberto que oferece velocidade, confiabilidade e escalabilidade (MYSQL, 2023a). Essas características tornam o MySQL uma escolha popular para aplicativos web que demandam alto desempenho, confiabilidade e capacidade de escalonamento.

O MySQL foi originalmente desenvolvido pela empresa sueca MySQL AB, que posteriormente foi adquirida pela Sun Microsystems e, posteriormente, pela Oracle Corporation. Atualmente, a Oracle é responsável pela manutenção e desenvolvimento contínuo do MySQL, que é licenciado sob a Licença Pública Geral GNU (GPL).

Dentre as principais características do MySQL, destacam-se: suporte a transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), replicação de banco de dados, gerenciamento de usuários e permissões, suporte a índices, suporte a múltiplas linguagens de programação e API's, entre outros recursos.

O MySQL é amplamente adotado por empresas de todos os portes e setores, desde startups até grandes corporações como Facebook, Twitter, Airbnb, Netflix e Uber (MYSQL, 2023b). Ele desempenha um papel fundamental na infraestrutura de muitos aplicativos web e sistemas de gerenciamento de conteúdo, sendo reconhecido por sua confiabilidade, escalabilidade e desempenho.

3.7 Framework CodeIgniter

O CodeIgniter é um *framework* de desenvolvimento web de código aberto, escrito em PHP, que proporciona uma abordagem facilitada para a criação de aplicativos *web* robustos, escaláveis e seguros. Seu principal objetivo é oferecer simplicidade e facilidade de uso, sem impor muitas restrições ao desenvolvedor.

O framework adota uma arquitetura baseada no padrão MVC (*Model-View-Controller*), que permite a separação clara das responsabilidades de apresentação, lógica

de negócios e acesso a dados. Essa abordagem proporciona uma estrutura organizada para o desenvolvimento e facilita a manutenção do código. Além disso, ela permite que diferentes partes do aplicativo sejam desenvolvidas e testadas de forma independente.

O CodeIgniter oferece uma ampla gama de recursos e bibliotecas úteis, incluindo manipulação de sessão, geração de URL's amigáveis, validação de formulários e proteção contra ataques de segurança (CODEIGNITER, 2023). O *framework* é altamente configurável e personalizável, permitindo que o desenvolvedor selecione apenas as bibliotecas e recursos necessários para o projeto em questão.

Além disso, o CodeIgniter possui uma comunidade ativa de desenvolvedores, o que significa que existem muitos recursos e documentação disponíveis para ajudar os desenvolvedores a superar desafios e solucionar problemas específicos. Isso contribui para a ampla adoção e popularidade do *framework*.

3.8 IDE PHPStorm

O PHPStorm é uma IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado) desenvolvida pela empresa JetBrains. Essa ferramenta oferece suporte para os principais frameworks do mercado, como Symfony, Laravel, Drupal, WordPress, entre outros (JETBRAINS, 2023a).

A interface de desenvolvimento do PHPStorm possui diversas ferramentas incorporadas que são especialmente úteis para os desenvolvedores. Uma das ferramentas destacadas é a integração com diversos bancos de dados, permitindo a realização de operações de inclusão, edição e consultas diretamente na interface da IDE. Essa funcionalidade contribui para aumentar a produtividade durante o desenvolvimento.

Outra característica relevante do PHPStorm é o recurso *Live Edit* (JETBRAINS, 2023b), que possibilita trabalhar com tecnologias de *frontend*, como HTML, CSS, Sass, Less, TypeScript e JavaScript. Com o *Live Edit*, qualquer alteração realizada no código é renderizada instantaneamente no navegador, proporcionando uma experiência de desenvolvimento ágil e interativa.

4 ATIVIDADES DESENVOLVIDAS

O estágio foi realizado exclusivamente na área de desenvolvimento de software, mais especificamente no *Squad* Comercial. O estagiário integrou uma equipe dedicada e focada na resolução de bugs e implementação de novos projetos. Após um período nesse *Squad*, surgiu a oportunidade e o desafio de atuar no *Squad Customer Success*.

Nesse contexto, o presente capítulo tem como objetivo detalhar as principais atividades desempenhadas durante o período de estágio, abordando o fluxo do dia a dia e os trabalhos realizados ao longo desse período, na plataforma Delta Assist.

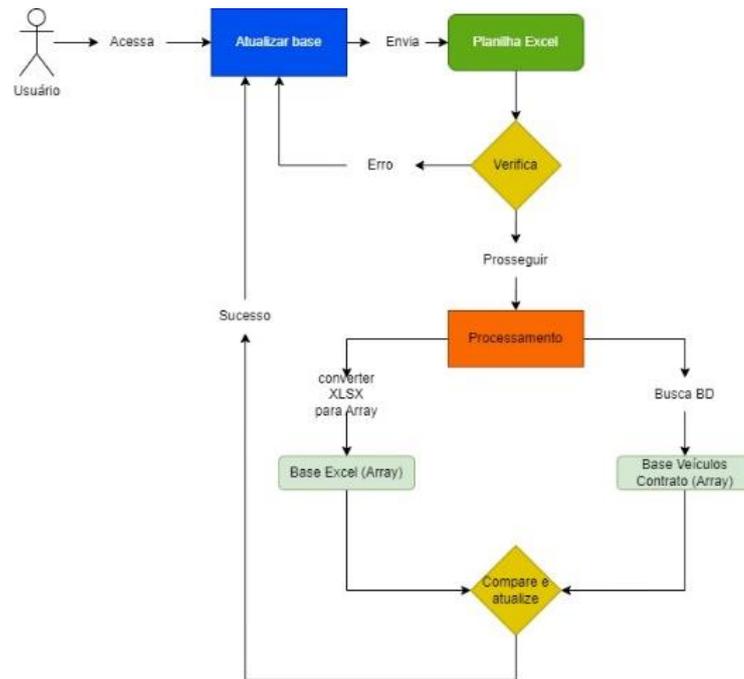
4.1 Atualização de Base de Veículos

Cada cliente possui um ou mais veículos em um contrato e, ao longo da vigência, surgem necessidades de inclusão ou exclusão desses veículos. No entanto, o processo para realizar essas alterações era oneroso, envolvendo a criação de propostas, inserção da base de veículos nas propostas, transformação das propostas em contratos e, em seguida, a necessidade de acessar o banco de dados para efetuar a troca de bases. Após a conclusão desse processo, o contrato recém-criado era cancelado.

Diante do crescimento da carteira de clientes da empresa, surgiu a necessidade de automatizar esse processo. O time de analistas do *Squad* Comercial foi responsável por realizar a eliciação dos requisitos e, em seguida, deu-se início ao desenvolvimento da versão 1.0 do sistema automatizado.

A Figura 8 ilustra o processo a ser desenvolvido, no qual o usuário precisa acessar uma página específica. Essa página não apenas exibe os dados existentes no contrato, mas também possui um botão para baixar o modelo e outro botão para realizar o upload do arquivo. Para implementar essa funcionalidade, foi necessário criar um método na *Controller* responsável por redirecionar o usuário para a página, bem como métodos para salvar os dados e métodos auxiliares para executar o processo.

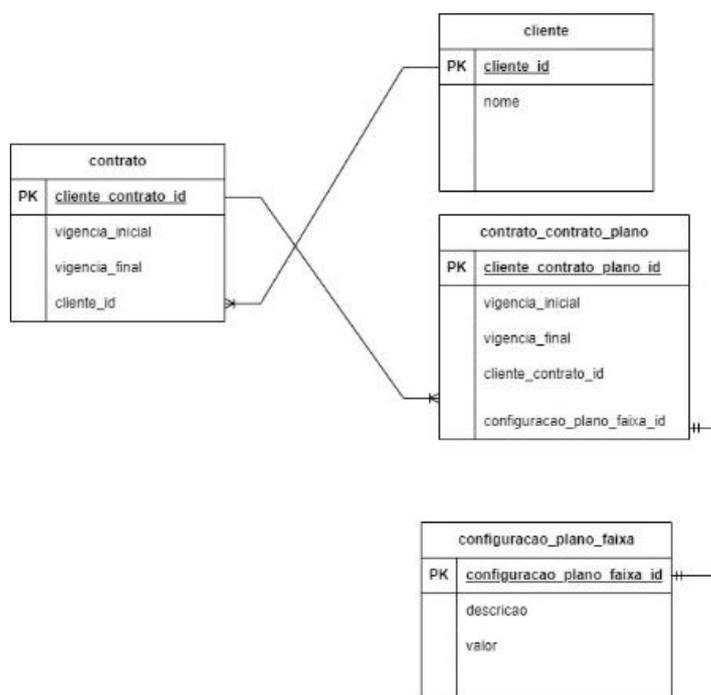
Figura 8: Mapeamento da Atualização de Base



Fonte: Do autor (2023).

A tela principal da atualização de bases exibe os seguintes dados para visualização: número do contrato, nome do cliente, vigência do contrato, nome do plano e a quantidade de veículos por plano. Esses dados são obtidos por meio de uma consulta, na qual uma query é utilizada para descrever as recuperações do banco de dados (ELMASRI & NAVATHE, 2005, p. 25). Essa consulta envolve três tabelas, que são relacionadas por meio de um processo de junção. Para executar essa query, foi necessário criar um método na *Model*. A Figura 9 apresenta o diagrama ER utilizado para a consulta dos dados exibidos ao usuário, ela nos mostra a relação utilizada no processo, cada contrato possui 1 (um) cliente e N itens/veículos (cliente_contrato_plano), cada item/veículo possui 1 (um) plano.

Figura 9: Diagrama Relacional de Entidade (ER) da query



Fonte: Do autor (2023).

O processo de atualização de bases envolve o envio de uma planilha no formato Excel (.xlsx) contendo um padrão de cabeçalho predefinido. Esse padrão de cabeçalho facilita a conversão da planilha em uma estrutura de *Array* no PHP.

Um array no PHP é na verdade um mapa ordenado. Um mapa é um tipo que relaciona valores a chaves. Este tipo é otimizado para vários usos diferentes: ele pode ser tratado como um array, uma lista (vetor), hashtable (que é uma implementação de mapa), dicionário, coleção, pilha, fila e provavelmente mais. (PHP. 2023)

Após o envio do arquivo e sua conversão, o algoritmo realiza a busca dos veículos presentes no contrato para realizar uma comparação entre os resultados. Inicialmente, esse processo era realizado utilizando um laço de repetição FOR dentro de outro laço, considerando as seguintes possibilidades: se um veículo está presente na planilha, mas não está presente na base de dados do contrato, o veículo é inserido no banco de dados; se um veículo está presente tanto na planilha quanto na base de dados do contrato, são verificadas possíveis alterações, como mudanças de planos, e, se necessário, as alterações são aplicadas.

Após a conclusão da primeira verificação e processamento, é realizada uma nova busca para identificar a seguinte variante: se um veículo não está presente na planilha, mas está presente na base de dados do contrato, o veículo é removido do contrato. Essa

primeira versão do algoritmo foi suficiente para automatizar um processo que antes demandava tempo do desenvolvedor, resultando em um deslocamento de suas funções para o setor de Pesquisa e Desenvolvimento (P&D) da empresa.

No entanto, à medida que o número de veículos dos clientes aumentava, foi observado que o processo se tornava lento. Quando um cliente possuía cerca de 700 veículos, o tempo de execução do algoritmo ultrapassava o limite de espera do servidor, resultando em erros e interrupção do processamento.

Após análise completa do processo, foi identificado que a utilização de um laço dentro de outro laço, juntamente com acessos frequentes ao banco de dados, resultava em um tempo de execução exponencial. A análise de complexidade de um algoritmo consiste em avaliar seu tempo de execução e uso de espaço, permitindo a comparação da eficiência de algoritmos diferentes para um mesmo problema (OLIVEIRA, 2020, p.23).

A solução adotada para otimizar o algoritmo foi separar as operações de inserção, edição e exclusão em estruturas de *arrays*. Essa abordagem eliminou a necessidade de acessar o banco de dados dentro dos laços, reduzindo o tempo de execução. Como resultado, contratos com aproximadamente 10 mil veículos puderam ser atualizados em um tempo significativamente menor. Essa implementação exigiu o emprego dos conceitos aprendidos na disciplina de Banco de Dados e Estrutura de Dados, evidenciando a aplicação prática desses conhecimentos.

4.2 Cotação no sistema

Para que um cliente estabeleça um contrato com a Delta, são necessárias negociações entre as partes envolvidas. Durante esse processo, são considerados diversos fatores-chave para a definição do tipo de contrato e seus valores, como o tamanho da frota, idade dos veículos, localização e tipo de carga transportada. Esses aspectos têm influência direta no acordo firmado. O processo tem início no *Kanban* de Frotas, onde o analista de negócios da empresa entra em contato com o cliente e dá início à negociação, por meio do envio de formulários para avaliar a frota do cliente.

Com o objetivo de aprimorar o processo de negociação, a Delta propôs o desenvolvimento automatizado desse procedimento na plataforma Delta Assist. Dessa forma, o corretor de clientes poderia realizar todo o *checklist* necessário dentro da

plataforma Delta Assist. Essa implementação demandou um esforço significativo na criação da interface visual, seguindo os padrões da empresa, e no desenvolvimento do *backend*, considerando que o tempo de execução da importação de uma base de veículos não poderia ser excessivamente longo.

Para o cliente, foi criada uma visualização específica, na qual todas as requisições de novos contratos com seus respectivos status seriam concentradas. Além disso, foi inserido um botão na tela de acompanhamento que permitia a criação de uma nova cotação. Nessa cotação, o cliente preencheria todo o *checklist* necessário e enviaria a frota de veículos. Após o envio da cotação, a proposta seria apresentada no *Kanban* de Frotas, onde o analista avaliaria a proposta e apresentaria os valores ao cliente. Caso o cliente aceitasse os termos e condições apresentados, o contrato seria gerado automaticamente.

A tela de criação de propostas apresentava um formulário extenso, cuja submissão excedia a quantidade máxima de dados enviados por meio da entrada padrão. Em outras palavras, o número de inputs enviados via método POST ultrapassava o limite permitido pelo sistema. Para solucionar esse problema, foi proposto o envio dos dados por meio de *JavaScript*, utilizando a biblioteca *AJAX*. Os inputs de cada etapa do *checklist* foram inseridos em um objeto *JSON*, resultando em uma redução do número de inputs a serem submetidos.

No lado do servidor, esses dados foram recebidos e convertidos em um objeto *PHP* utilizando a função *json_decode()*, própria da linguagem. Em seguida, os dados do *checklist* foram inseridos nas tabelas "kanban", "kanban_historico", "proposta" e "proposta_veiculo".

4.3 Implementação de Tabelas Fixas

A empresa Delta Global S.A. tem se destacado como uma referência nacional na prestação de serviços de assistência veicular. Com o crescimento contínuo de sua carteira de clientes, a empresa precisa realizar ajustes e implementações para se manter competitiva no mercado. Recentemente, a Delta Global expandiu seus serviços para seguradoras e locadoras, atuando no atendimento de sinistros. Essa nova modalidade exigiu adaptações no processo de abertura de ocorrências e faturamento.

As principais empresas desse setor utilizam o modelo de Tabela Fixa, que é personalizada para cada cliente. Essa tabela contém os valores para cada tipo de serviço, permitindo que, ao final do atendimento, os valores sejam repassados ao cliente, como o cliente "A" da Delta Global, e também sejam cobrados do atendido pelo cliente, com base nos valores da tabela fixa. Para lidar com essas demandas, foram formados três *Squads* com responsabilidades distintas: o *Squad* de Operações ficou encarregado de adequar a abertura de ocorrências, o *Squad* Financeiro cuidou do ajuste no faturamento e o *Squad* CS foi responsável pela criação de toda a estrutura da tabela fixa.

O *Squad* CS desenvolveu as operações básicas de cadastro, conhecidas como CRUD (*Create, Read, Update e Delete*), que são as quatro operações fundamentais para o armazenamento persistente de informações (MOZILLA, 2023c). Para esse fim, foi criada uma camada de serviço específica denominada "TabelaFixa". Essa classe agora se dedica exclusivamente ao seu propósito, sendo separada da camada de *controller*.

A camada de serviço possui métodos básicos para salvar, editar e remover uma tabela, além de funções específicas, como identificar o tipo de tabela utilizada e calcular os valores da ocorrência com base em cada tabela proposta. Essa abordagem de serviço foi adotada com base na sua utilização em outras áreas, como a abertura de ocorrências e o setor financeiro.

O *Squad* CS também realizou ajustes nas visualizações do usuário, promovendo mudanças nas regras de negócio para adequar os cálculos dos valores exibidos. Foram criados métodos para facilitar a manutenção futura dos cálculos, separando as regras existentes em estruturas modulares, o que permite a inclusão de novas regras, se necessário.

Esse trabalho exigiu um esforço conjunto entre os desenvolvedores, analistas e o time de negócios da empresa. A implementação está atualmente em processo piloto, sendo constantemente ajustada. Essas atividades requerem sólidos conhecimentos em operações de banco de dados e habilidades para desenvolver soluções algorítmicas de forma ágil.

4.4 Refatoração e implementação de Camada de Service

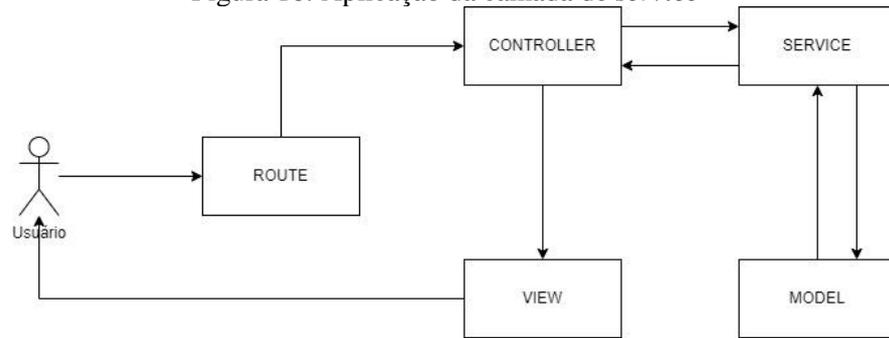
Um dos pontos fundamentais para o sucesso de uma empresa que trabalha com tecnologias é manter-se atualizado em relação às tecnologias utilizadas em seus

sistemas. Nesse sentido, a equipe de desenvolvedores da empresa identificou a necessidade de atualização do *framework* PHP e do banco de dados MySQL. Em busca de melhorias, a equipe estabeleceu um projeto de refatoração do sistema em conjunto com a diretoria.

Para iniciar o processo de atualização, foram conduzidos estudos para identificar os elementos mais importantes a serem atualizados. Após essa análise, foram definidas as prioridades e a ordem das atualizações. A primeira etapa foi atualizar a versão do PHP, passando da versão 7.1 para a 7.4. Todo o procedimento foi devidamente documentado e repassado aos desenvolvedores para que pudessem realizar a atualização em seus ambientes de desenvolvimento. Após a atualização do PHP, a equipe concentrou-se no *framework* CodeIgniter. Por meio do estudo da documentação, traçou-se um plano para atualizá-lo, seguindo as versões subsequentes, partindo da versão 2.1.3 para a versão 2.2.6.

Segundo a documentação do CodeIgniter, a última versão possível de atualização a partir do modelo antigo do *framework* seria a versão 3.0.x. Conscientes desse desafio, os desenvolvedores iniciaram a aplicação de um padrão de projeto que permitisse a reestruturação do *framework*, considerando a atualização para a versão 4. Essa atualização exigiria a reescrita completa do código em um novo projeto, devido às mudanças drásticas nos métodos, bibliotecas e outras dependências do *framework*. Diante disso, foi proposta a implementação de uma camada de serviço seguindo o padrão *Model-View-Controller* (MVC).

Essa camada de serviço no padrão MVC possibilitará uma estrutura mais flexível e adaptável às mudanças do *framework*, o que facilitará futuras atualizações e manutenções do sistema. Dessa forma, a equipe busca garantir que o sistema esteja em conformidade com as tecnologias mais recentes, proporcionando maior eficiência, segurança e escalabilidade ao longo do tempo. Esse processo de atualização requer um esforço cuidadoso e aprofundado, pois envolve a revisão e aprimoramento de diversos aspectos do sistema, mas é essencial para manter a empresa competitiva e atualizada no mercado em constante evolução de tecnologias.

Figura 10: Aplicação da camada de *service*

Fonte: Do autor (2023)

No sistema, foi implementada a camada de *Service*, que tem a responsabilidade de abrigar toda a lógica de negócio, sem depender da estrutura atual do framework CodeIgniter. Dessa forma, é possível transferir o código que estava acoplado à *Controller* para essa camada de *Service*, e no futuro será viável migrar essa camada para o framework atual sem consequências negativas. Na Figura 10, é possível observar a modificação do padrão MVC, onde a *Controller* deixa de exercer a responsabilidade de regra de negócio, passando essa responsabilidade para a camada de *Service*. Portanto, o *Service* é encarregado de se comunicar com o *Model* e retornar o valor tratado para a *Controller*.

O próximo passo consiste na criação da camada de Repositório, que tem como função desacoplar a camada *Model*. Ao mesmo tempo, a interface gráfica poderá ser transferida para o novo CodeIgniter ou qualquer outro framework que seja utilizado. Essas melhorias proporcionam uma maior flexibilidade no desenvolvimento das soluções. Utilizando a camada de *Service*, é possível aplicar o padrão de projeto *Abstract Factory* na solução de busca de oficinas próximas à ocorrência, levando em consideração as particularidades de cada cliente na árvore de decisão para a coleta do veículo. O padrão *Abstract Factory* fornece uma interface para criar famílias de objetos relacionados ou dependentes sem especificar suas classes concretas (FREEMAN, E. et al, p. 137).

Essas melhorias, ao adicionar mais camadas e aplicar padrões de projeto, tendem a trazer benefícios significativos em relação à qualidade do software. Isso implica em melhorias na manutenibilidade do código e no desacoplamento do *framework* utilizado, o que contribui para aprimorar a qualidade e a robustez do sistema.

5 CONSIDERAÇÕES FINAIS

O estágio em desenvolvimento *backend* na empresa Delta Global S.A. trouxe um aprofundamento de técnicas aprendidas em sala de aula de forma prática, no estágio foi possível realizar na prática conceitos de disciplinas de Engenharia de Software, Banco de Dados, Estruturas de Dados e Complexidade além claro de Interação Humano Computador.

Traçando um paralelo em cada uma dessas disciplinas citadas, conceitos de engenharia de software como a elicitação de requisitos, gestão de desenvolvimento foram vistos constantemente durante o período de estágio, foi possível entender a importância dos requisitos em um desenvolvimento de software, sem um gerenciamento, o desenvolvimento acaba sendo oneroso tanto no fator de tempo quanto no financeiro, é preciso entender o que o cliente necessita, documentar, quantificar e após isso desenvolver.

Dentre as disciplinas a qual tive maior experiência em campo, destaco a Estrutura de Dados e Complexidade e Análise de Algoritmo, trabalhando diretamente do lado servidor, ou seja, no *backend*, constantemente tive contato com essas disciplinas, seja na concepção de algoritmo ou em estruturas já existentes no sistema, graças ao ótimo ensino e professores preparados consegui desempenhar com satisfação todos os desafios a mim conferidos. Estagiando na empresa Delta Global tive a total liberdade em opinar e oferecer soluções com base no que estudei na Universidade Federal de Lavras.

O estágio pôde proporcionar uma visão mais concreta de todo o processo que envolve a criação e manutenção de um software. Além do conhecimento prático em desenvolvimento *backend* adquirido estagiando na empresa Delta Global S.A., tive a oportunidade de conhecer um pouco do mundo dos negócios, da administração de uma empresa, isso graças à proximidade com os diretores da empresa.

Todos os trabalhos realizados pelo estagiário foram de grande importância para a empresa, a atualização de bases foi peça fundamental no gargalo de onerar o tempo do desenvolvedor, a cotação no sistema trouxe maior agilidade para o setor comercial da empresa trazendo maior economia de tempo, a refatoração do sistema em geral trará à Delta Global S.A. uma maior confiabilidade do sistema Delta Assist além de um ganho em *performance* geral da plataforma.

Portanto, colaborar em uma empresa de tecnologia com outros profissionais trocando experiências, reforçou a aspiração em continuar a estudar e evoluir na profissão, cujo os ensinamentos aprendidos no curso de Sistemas de Informação foram decisivos nas sólidas tomadas de decisões, além de destacar o estagiário na empresa cujo o mesmo recebeu a sua promoção.

REFERÊNCIAS

ANDRESON, J. R.. **Kanban: Mudança evolucionária de sucesso para seu negócio de tecnologia**. Tradução de Andrea Pinto. Sequim. Blue Hole Press, 2011. p. 288.

CODEIGNITER. **Welcome to Codeigniter**. 2023. Disponível em https://codeigniter.com/user_guide/intro/index.html. Acesso em 2023-05-06.

DELTA. **O que fazemos**. 2023a. Disponível em: <https://www.deltaglobal.com.br/>. Acesso em 2023-04-29.

DELTA. **Nossos cliente**. 2023b. Disponível em: <https://www.deltaglobal.com.br/>. Acesso em 2023-04-29.

DELTA. **Delta Fleet**. 2023c. Disponível em: <https://www.deltaglobal.com.br/fleet>. Acesso em 2023-04-29.

DELTA. **Delta Safe**. 2023d. Disponível em: <https://www.deltaglobal.com.br/fleet>. Acesso em 2023-04-29.

DELTA24HORAS. **Quem somos**. 2023. Disponível em: <https://delta24h.com.br/#institucional>. Acesso em 2023-04-29.

ELMASRI, Ramez, NAVATHE, Shamkant B.. **Sistemas de Bancos de Dados**. 4. ed. São Paulo: Pearson Education, 2005. 724 p.

FERREIRA, D. R.. **Linguagem de Programação**. Curitiba: Contentus, 2020. 56 p.

FREEMAN, E. et al. **Use a cabeça : padrões e projetos**. Rio De Janeiro: Alta Books, 2009. 489 p.

HISTORICAR. **Quem Somos**. 2023. Disponível em: https://historicar.com.br/quem_somos. Acesso em 2023-04-2029.

GANZAGA DE OLIVEIRA, S. L. **Algoritmos e seus fundamentos**. 2. ed. rev. Lavras: Editora UFLA, 2020, 341 p.

JETBRAINS. **Suporte para as principais estruturas**. 2023a. Disponível em <https://www.jetbrains.com/pt-br/phpstorm/>. Acesso em 2023-05-06.

JETBRAINS. **Tecnologias de front-end incluídas**. 2023b. Disponível em <https://www.jetbrains.com/pt-br/phpstorm/>. Acesso em 2023-05-06.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais: administrando a empresa digital**. 17. ed. São Paulo. Pearson_Grupoa, 2023. p. 602.

MOZILLA. **O que é Javascript.** 2023a. Disponível em https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Acesso em 2023-05-07.

MOZILLA. **O que é Javascript.** 2023b. Disponível em https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Acesso em 2023-05-07. Acesso em 2023-05-07.

MOZILLA. **CRUD.** 2023c. Disponível em <https://developer.mozilla.org/pt-BR/docs/Glossary/CRUD>. Acesso em 2023-05-07.

SOMMERVILLE, Ian. **Engenharia de Software.** 9. Ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.

MYSQL. **Documentação Oficial.** 2023a. Disponível em <https://dev.mysql.com/doc/>. Acesso em 2023-05-06.

MYSQL. **Some o four customers.** 2023b. Disponível em <https://dev.mysql.com>. Acesso em 2023-05-06.

PHP. **Arrays.** 2023. Disponível em https://www.php.net/manual/pt_BR/language.types.array.php. Acesso em 2023-05-06.

RUNRUN. **O que são sprints.** 2022. Disponível em <https://blog.runrun.it/o-que-sao-sprints/>. Acesso em 2023-05-06.