



**LEONARDO AMORIM DE SENA**

**RELATÓRIO DE ESTÁGIO - REESTRUTURAÇÃO DE  
SISTEMA ESCRITO EM VB.NET PARA C# CORE E  
DESENVOLVIMENTO DE POCS EM VUE.JS E ANGULAR**

**LAVRAS – MG**

**2022**

**LEONARDO AMORIM DE SENA**

**RELATÓRIO DE ESTÁGIO - REESTRUTURAÇÃO DE SISTEMA ESCRITO EM  
VB.NET PARA C# CORE E DESENVOLVIMENTO DE POCS EM VUE.JS E  
ANGULAR**

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

Prof. Dr. Bruno de Abreu Silva

Orientador

**LAVRAS – MG**

**2022**

**LEONARDO AMORIM DE SENA**

**RELATÓRIO DE ESTÁGIO - REESTRUTURAÇÃO DE SISTEMA ESCRITO EM  
VB.NET PARA C# CORE E DESENVOLVIMENTO DE POCS EM VUE.JS E  
ANGULAR**

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

APROVADA em 08 de setembro de 2022.

Prof. Dr. Bruno de Abreu Silva	UFLA
Prof. Dr. Mauricio Ronny De Almeida Souza	UFLA
Prof. Dr. Rafael Serapilha Durelli	UFLA

Prof. Dr. Bruno de Abreu Silva  
Orientador

**LAVRAS – MG  
2022**

*Dedico ao meu pai, André de Sena Ribeiro e a minha mãe, Cláudia Renata Amorim.*

## **AGRADECIMENTOS**

Agradeço imensamente ao meu pai, André, que sempre foi meu pilar e forneceu apoio durante toda a minha jornada. Agradeço também a minha mãe, que certamente está olhando por mim do alto, me protegendo e se orgulhando a cada conquista.

Agradeço também a minha tia Maria pelo apoio e conselhos dados durante os deliciosos almoços de domingo em sua casa.

Agradeço a todos meus amigos, que me acompanharam e auxiliaram em todo o processo. Em especial, destaco cinco deles: Amanda, por todo apoio mesmo estando longe; Alice, pelo companheirismo e incontáveis caronas; Luiz, por embarcar comigo nesta jornada, me acompanhando desde o ensino médio; Álvaro por ser amigo de todas as horas e me acompanhar na paixão pelo futebol e pelo Cruzeiro; e Aline por toda a ajuda em noites a fio de estudos "velados".

A Maria Fernanda por todo o suporte afetivo e emocional durante todos os momentos durante os últimos dois anos.

Também agradeço a Danilo por todo o suporte e ensinamentos nesse início de jornada no mercado de trabalho

Agradeço ao Prof. Dr. Bruno de Abreu Silva por aceitar me orientar e me ajudar a escrever este documento.

Agradeço, por fim, à Universidade Federal de Lavras, e a todos seus professores e funcionários que, de uma forma ou outra, contribuíram para meu desenvolvimento acadêmico e pessoal.

*Você não consegue ligar os pontos olhando pra frente; você só consegue ligá-los olhando pra trás.”  
(Steve Jobs)*

## RESUMO

A dificuldade em encontrar profissionais capacitados para trabalhar com determinadas tecnologias pode impor que empresas de desenvolvimento de software realizem a reestruturação de seus sistemas. O presente trabalho tem por objetivo relatar, do ponto de vista do desenvolvedor, como foi a participação do estagiário na reestruturação de um sistema escrito em VB.net para C# Core bem como a criação de provas de conceito visando comprovar a viabilidade técnica de reestruturar o frontend escrito em ASP.NET Web Forms para Vue.js e Angular. O sistema em questão trata-se de um software chamado iTransport, utilizado para gestão de transporte e rota e desenvolvido pela empresa HTP, local do estágio. Espera-se que, ao realizar a reestruturação, a empresa seja capaz de captar mais profissionais aptos a trabalhar com as novas tecnologias adotadas.

**Palavras-chave:** Rotas. Frota. Framework Scrum. Vue.js. C#. Sql.

## **ABSTRACT**

The difficulty in finding qualified professionals to work with certain technologies may impose that software development companies carry out the restructuring of their systems. The present work aims to report, from the developer's point of view, how the trainee participated in the restructuring of a system written in VB.net for C# Core as well as the creation of proofs of concept in order to prove the technical feasibility of restructure the frontend written in ASP.NET Web Forms for Vue.js and Angular. The system in question is a software called iTransport, used for transport and route management and developed by the company HTP, location of the internship. It is expected that, when carrying out the restructuring, the company will be able to attract more professionals able to work with the new technologies adopted.

**Keywords:** Routes. Fleet. Scrum Framework. Vue.js. C#. Sql.



## LISTA DE FIGURAS

Figura 2.1 – Tabela de Alunos em SQL. . . . .	14
Figura 2.2 – Trecho de código em JavaScript. . . . .	15
Figura 2.3 – Exemplo de componente de arquivo único Vue. . . . .	16
Figura 2.4 – Exemplo de componente Angular. . . . .	17
Figura 2.5 – Código HTML exemplificando a utilização da tag iFrame. . . . .	18
Figura 2.6 – Classe Aluno escrita em VB.NET. . . . .	19
Figura 2.7 – Web Forms e código de servidor, escrito em VB.NET, associado. . . . .	20
Figura 2.8 – Resultado de Web Forms em um navegador. . . . .	20
Figura 2.9 – Classe Aluno escrita em C#. . . . .	21
Figura 2.10 – Exemplo de expressão regular e texto capturado por ela. . . . .	24
Figura 3.1 – Captura de tela do sistema iTransport. Ao centro a aba "Mapa" em gestão de rotas. . . . .	27
Figura 3.2 – Captura de tela do sistema iTransport. . . . .	28
Figura 3.3 – Aplicação de expressão regular para captura de nome dos campos de uma classe. . . . .	31

## LISTA DE QUADROS

Quadro 2.1 – Exemplo de representação de um banco de dados. . . . .	12
Quadro 3.1 – Comparativo entre Vue.JS e Angular no contexto de utilização no estágio na empresa HTP. . . . .	29

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Empresa</b>	<b>10</b>
<b>1.2</b>	<b>Objetivos</b>	<b>11</b>
<b>2</b>	<b>Conceitos e Tecnologias utilizadas</b>	<b>12</b>
<b>2.1</b>	<b>Prova de Conceito</b>	<b>12</b>
<b>2.2</b>	<b>Banco de Dados</b>	<b>12</b>
<b>2.2.1</b>	<b>SQL</b>	<b>12</b>
<b>2.2.2</b>	<b>T-SQL</b>	<b>13</b>
<b>2.3</b>	<b>Programação WEB</b>	<b>13</b>
<b>2.3.1</b>	<b>JavaScript</b>	<b>14</b>
<b>2.3.2</b>	<b>Frameworks Javascript</b>	<b>15</b>
<b>2.3.2.1</b>	<b>Componentes</b>	<b>15</b>
<b>2.3.2.2</b>	<b>VUE.js</b>	<b>15</b>
<b>2.3.2.3</b>	<b>Angular</b>	<b>17</b>
<b>2.3.3</b>	<b>iFrame</b>	<b>18</b>
<b>2.3.4</b>	<b>Visual Basic.NET</b>	<b>18</b>
<b>2.3.5</b>	<b>Web Forms</b>	<b>19</b>
<b>2.3.6</b>	<b>C#</b>	<b>20</b>
<b>2.4</b>	<b>Métodos ágeis</b>	<b>21</b>
<b>2.4.1</b>	<b><i>Framework Scrum</i></b>	<b>21</b>
<b>2.5</b>	<b>Expressão Regular - RegEx</b>	<b>23</b>
<b>3</b>	<b>Atividades desenvolvidas</b>	<b>25</b>
<b>3.1</b>	<b>iTransport</b>	<b>26</b>
<b>3.2</b>	<b>Provas de conceito</b>	<b>26</b>
<b>3.3</b>	<b>Reestruturação Back-end</b>	<b>29</b>
<b>3.4</b>	<b>Analizador de código a nível léxico</b>	<b>30</b>
<b>3.5</b>	<b>Desafios</b>	<b>32</b>
<b>4</b>	<b>CONCLUSÃO</b>	<b>33</b>
	<b>REFERÊNCIAS</b>	<b>35</b>

## 1 INTRODUÇÃO

Empresas que oferecem fretamento de funcionários lidam com muitos desafios. Dentre eles destacam-se o fato de muitas vezes existirem veículos lotados ou com muito espaço livre, embarque e desembarque de funcionários em rotas incorretas e atrasos nos pontos de embarque e desembarque. Além disso, desafios como uso indevido do ônibus por pessoas não autorizadas e embarque de colaboradores sem identificação também não são raros na tarefa de gerir o transporte de funcionários. (TRACKJOB, 2018)

Para lidar com desafios tão complexos, o serviço de transporte dos funcionários de uma empresa demanda cuidados específicos. Mesmo que o transporte seja realizado de maneira terceirizada, um sistema de gestão de rotas que, além de auxiliar na gestão do transporte fretado de funcionários ajuda na gestão de frotas e na demanda por táxi corporativo e carro compartilhado pode facilitar e organizar bastante todo o trabalho.

O iTransport é um *software as a service*, desenvolvido pela empresa HTP, que visa gerenciar todo o processo de transporte de funcionários. Contudo, por ter sido desenvolvido em 2007, o sistema utiliza tecnologias que dificultaram a contratação de novos profissionais para evoluir o software. Pensando nisso, a HTP decidiu reestruturar todo o seu sistema utilizando tecnologias em voga no mercado atualmente.

### 1.1 Empresa

O estágio foi realizado na HTP Solution S/A<sup>1</sup>, empresa sediada em Belo Horizonte que tem como ramo de atuação a tecnologia da informação desenvolvendo soluções de software e hardware. O foco da empresa dá-se no mercado de mobilidade urbana, sendo, atualmente, a plataforma iTransport seu principal produto, permitindo a seus usuários que gerenciem todo o serviço de transporte de pessoas, equipamentos e cargas.

A HTP é uma empresa que está a mais de 25 anos no mercado, tendo iniciado suas atividades em 1996 fabricando softwares sob encomenda e posteriormente em 2011 iniciado o desenvolvimento de seu principal produto. Já foi contemplada com diversos prêmios, dentre eles, o MPE Brasil Competitividade na edição estadual de Minas Gerais em 2010 e o Prêmio Regional da Qualidade em 2012.

---

<sup>1</sup> <<https://itransport.com.br/sobre-nos/>>

A empresa é dividida em dois times principais: time de logística, que lida com atividades relacionadas a implantação de hardware e suporte ao cliente e time de desenvolvimento, responsável por desenvolver novas tecnologias em software e realizar manutenção nas existentes. O time de desenvolvimento, em que deu-se o estágio, utiliza a Metodologia de Desenvolvimento Ágil, a partir do Framework Scrum.

A plataforma iTransport contempla hardwares e software. Os equipamentos de hardware são usados para coletar dados de telemetria de veículos, a fim de prover os recursos tecnológicos necessários para que os clientes possam integrar a operação aos módulos de software. Já o software, responsável por receber e tratar os dados da telemetria, é dividido em três módulos: Leve, que gerencia demandas por táxi corporativo, carro compartilhado e reserva de frotas com seus veículos ou de terceiros; Frotas, que gerencia veículos com monitoramento on-line, telemetria, manutenção, controle de combustível, bloqueio e desbloqueio remoto; e Fretado, que gerencia transporte de funcionários em rotas de trabalho com monitoramento on-line, controle de acesso, aplicativo para passageiro e motorista.

## **1.2 Objetivos**

Este relatório tem por objetivo demonstrar o desenvolvimento e manutenção do sistema de gestão de rotas inteligentes chamado iTransport, bem como relatar as provas de conceito desenvolvidas para a reestruturação desse sistema, sendo ambos desenvolvidos para a empresa HTP. O estagiário esteve envolvido com o projeto iTransport durante todo o período de estágio, que se iniciou em 23/04/2021 e se encerrou em 10/09/2021. Do ponto de vista do crescimento do estagiário, os objetivos do estágio foram:

- Vivenciar as experiências no meio profissional;
- Aprimorar os conhecimentos técnicos adquiridos durante a formação acadêmica, estando em contato com problemas do mundo real;
- Contribuir com a empresa, buscando o crescimento profissional e pessoal.

## 2 CONCEITOS E TECNOLOGIAS UTILIZADAS

Neste Capítulo serão apresentados os conceitos, tecnologias, *frameworks* e ferramentas utilizados durante o estágio.

### 2.1 Prova de Conceito

Prova de Conceito, em inglês Proof of Concept (ou apenas PoC) refere-se à demonstração da possibilidade de validação de uma ideia ou conceito, tanto na área de TI ou na de negócios. (MENA, 2017) Em se tratando do estágio ao qual este relatório disserta, a prova de conceito foi utilizada com o objetivo de verificar a viabilidade técnica de realizar a reestruturação do front-end sem a necessidade de descartar totalmente o código fonte já existente.

### 2.2 Banco de Dados

Segundo Ramakrishnan e Gehrke (2008), pode-se definir banco de dados como sendo uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas. Em bancos de dados relacionais, as entidades são representadas por tabelas, que por sua vez são constituídas de colunas que representam as características (ou atributos) das entidades. Por exemplo, um banco de dados de uma universidade pode ter as entidades alunos, professores e cursos e, por sua vez, ter os relacionamentos matrícula de alunos nos cursos e cursos ministrados por professores. O Quadro 2.1 ilustra a representação da entidade aluno em tabela.

Quadro 2.1 – Exemplo de representação de um banco de dados.

Alunos				
Identificador	Nome	Telefone	Endereço	Matrícula
1	Alberto Silva	(11)99999-9999	Rua Um...	201810111
2	Maria Marta	(11)99999-9998	Rua Dois...	201810222
3	José Augusto	(11)99999-9997	Av. Três...	201810333
...	...	...	...	...

Fonte: Autor.

#### 2.2.1 SQL

No início dos anos de 1970 foi desenvolvida a linguagem SQL, pelo Departamento de Pesquisas da IBM, como forma de interface para o sistema de banco de dados SYSTEM R.

Inicialmente a linguagem foi chamada de SEQUEL (*Structured English QUery Language*) e teve sua nomenclatura revisada em 1977 passando a se chamar definitivamente SQL (*Structured Query Language*) (FRANCA; GOYA; PUGA, 2013).

O Instituto Nacional Americano de Padrões (*American National Standard Institute - ANSI*) em conjunto com a Organização Internacional de Padrões (*International Standards Organization - ISO*) publicou, em 1986, o padrão SQL-86 tornando o SQL linguagem padrão para bancos de dados relacionais (FRANCA; GOYA; PUGA, 2013). Já em 1999, o padrão SQL3 surgiu, ao qual se seguiram atualizações como o SQL 2003, o SQL 2006 e o 2008 que possuem recursos de XML (CARDOSO; CARDOSO, 2013).

Devido à padronização, as instruções no SQL seguem a mesma nomenclatura e formato para os diferentes tipos de SGBD, respeitando as particularidades de cada um (FRANCA; GOYA; PUGA, 2013). Além disso, a linguagem SQL divide-se em algumas categorias de instruções dentre as quais se encontram a Linguagem de Manipulação de Dados (DML - *Data Manipulation Language*) e a Linguagem de Definição de Dados (DDL - *Data Definition Language*). O subconjunto DML permite que os usuários insiram, excluam e modifiquem tuplas além de permitir a formulação de consultas. Já o subconjunto DDL suporta a criação, exclusão, e modificação de definições das tabelas e visões (RAMAKRISHNAN; GEHRKE, 2008).

A Figura 2.1 apresenta um exemplo de criação de tabela e CRUD (*Create, Read, Update, Delete*) de uma tabela de banco de dados feita em SQL utilizando os subconjuntos DDL e DML.

### 2.2.2 T-SQL

Como descrito anteriormente, instruções SQL básicas formam um padrão disponível independentemente do sistema de banco de dados relacional. Entretanto, muitos sistemas têm suas próprias extensões às instruções SQL. Essas extensões fornecem funcionalidades não abrangidas pelo padrão SQL. O SQL Server, sistema de gerenciamento de banco de dados da Microsoft, usa a extensão T-SQL, que inclui extensão de linguagem para gerenciar contas de usuário e escrever procedimentos e funções armazenados, ou seja, código de aplicativo armazenado no banco de dados (MICROSOFT, 2022a).

## 2.3 Programação WEB

A *Web* tornou-se completamente integrada à sociedade. Grande parte das empresas estão presentes na Web e não é raro se deparar com propagandas que exibam um URL. O

Figura 2.1 – Tabela de Alunos em SQL.

```

1  -- Criação de tabela usando o subconjunto de instruções DDL
2  CREATE TABLE Alunos (
3      identificador INTEGER PRIMARY KEY,
4      nome TEXT NOT NULL,
5      telefone TEXT NOT NULL,
6      endereco TEXT NOT NULL,
7      matricula INTEGER NOT NULL
8  )
9
10 -- Inserção de dados utilizando subconjunto de instruções DML
11 INSERT INTO Alunos VALUES (1, 'Alberto Silva', '(11)99999-9999', 'Rua Um...', 201810111)
12 INSERT INTO Alunos VALUES (1, 'Maria Marta', '(11)99999-9998', 'Rua Dis...', 201810222)
13 INSERT INTO Alunos VALUES (1, 'José Augusto', '(11)99999-9997', 'Rua Três...', 201810333)
14
15 -- Recuperação de dados de alunos utilizando o subconjunto de instruções DML
16 SELECT * FROM Alunos;
17
18 -- Atualização de dado de aluno utilizando subconjunto DML
19 UPDATE Alunos SET endereco = 'Rua Doze' WHERE identificador = 1;
20
21 -- Deleção de tupla de aluno utilizando subconjunto DML
22 DELETE FROM Alunos WHERE identificador = 2;

```

Fonte: Autor.

fato de que grande parte das pessoas estão familiarizadas com conceitos como URL deixa bem claro a importância da web atualmente (LEMAY; COLBURN; TYLER, 2002). Esta Seção visa apresentar as tecnologias de desenvolvimento web utilizadas durante o estágio.

### 2.3.1 JavaScript

JavaScript é a principal linguagem de programação da Web. Dentre as características do JavaScript destacam-se os fatos de ser mais próxima da linguagem humana do que da linguagem de máquina (alto nível), orientada a objetos e ser interpretada (FLANAGAN, 2011). Outra característica importante é o fato de ser uma linguagem sem tipos de dados, em que o tipo de qualquer variável é definido de acordo com a informação que está armazenada naquele momento (OLIVEIRA; ZANETTI, 2021). A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos conseguem interpretar JavaScript, sendo essa a linguagem de programação mais onipresente da história. É importante ressaltar que, apesar do nome, a única semelhança que JavaScript compartilha com a linguagem Java é uma similaridade superficial de sintaxe, sendo duas linguagens completamente diferentes (FLANAGAN, 2011).

A linguagem JavaScript apresenta-se como uma linguagem bastante popular, contando com aproximadamente 17,4 milhões de desenvolvedores ao redor do mundo, segundo a pesquisa SlashData (2020). Ainda segundo essa pesquisa, o crescimento da comunidade JavaScript tem



tido constante durante os últimos anos, tendo 5,2 milhões de desenvolvedores se juntado à comunidade nos últimos dois anos.

A Figura 2.2 mostra um trecho de código em JavaScript que tem o objetivo de converter números decimais em números binários.

Figura 2.2 – Trecho de código em JavaScript.

```
decToBin(n) {  
  const binario = Math.abs(n).toString(2);  
  
  const qtdZeros = 4 - binario.length;  
  
  const binarioComZero = "0".repeat(qtdZeros) + binario;  
  
  return (n > 0 ? "1" : "0") + binarioComZero;  
}
```

Fonte: Autor.

### 2.3.2 Frameworks Javascript

Segundo Marcolino (2021), um *framework* JavaScript objetiva, principalmente, abstrair a interação do navegador com o modelo de objeto de documento (*Document Object Model - DOM*). Substitui-se a manipulação e referência diretamente a elementos do DOM pela interação declarativa em alto nível. Muitas vezes *frameworks* são usados para agilizar o processo de desenvolvimento de aplicações, além de reduzir custos e esforços, agilizando, assim, o processo de entrega de soluções de software.

#### 2.3.2.1 Componentes

Segundo Fernandes (2017), componentes são conjuntos isolados de lógica, visualização e possível estilização. Dentre as vantagens de se utilizar componentes encontram-se a possibilidade de reutilização de trechos de código sem gerar código duplicado; o isolamento do contexto; e a melhoria na legibilidade do código. *Frameworks* modernos como Angular e Vue.JS são baseados em componentes e se aproveitam de todas as vantagens citadas anteriormente.

#### 2.3.2.2 VUE.js

Vue.js é um *framework* para a construção de interfaces de usuário. O *framework* é progressivo, ou seja, não precisa ser adotado em sua completude em um projeto, e sua biblioteca

principal é focada exclusivamente na camada visual, sendo fácil adotar e integrar outros projetos ou bibliotecas que já existem (VUE.JS, 2022b). Um ponto importante de se destacar a respeito do Vue.js é que ele é um projeto *indie*, portanto, não é mantido por grandes empresas de software, e sim pela própria comunidade, por meio de doações (MARCOLINO, 2021).

Embora seja possível criar aplicativos Vue.js usando apenas JavaScript através da tag `<script>`, pode-se obter maior potência e flexibilidade através do uso da interface de linha de comando (*command-line interface* - CLI) Vue. A CLI Vue possibilita o uso de componentes de arquivos únicos, além de ferramentas robustas para aprimorar a experiência de desenvolvimento (MICROSOFT, 2022b). Componentes de arquivo únicos, por sua vez, nada mais são que arquivos com extensão `.vue` que permitem encapsular o *template*, a lógica e a estilização de um componente Vue em um único arquivo (VUE.JS, 2022d). A Figura 2.3 demonstra um componente de arquivo único, em que os blocos `<template>`, `<script>` e `<style>` encapsulam, respectivamente, a interface (template), a lógica e a estilização.

Figura 2.3 – Exemplo de componente de arquivo único Vue.

```
<script>
export default {
  data() {
    return {
      greeting: 'Hello World!'
    }
  }
}
</script>

<template>
  <p class="greeting">{{ greeting }}</p>
</template>

<style>
.greeting {
  color: red;
  font-weight: bold;
}
</style>
```

Fonte: Vue.js (2022d).

### 2.3.2.3 Angular

Angular é um *framework* que adota um superconjunto de JavaScript, chamado TypeScript, como linguagem padrão. O TypeScript, por sua vez, inclui novos recursos e tipagens às variáveis do JavaScript (MARCOLINO, 2021). Angular é de código aberto e mantido pela Google que tem por finalidade fornecer ferramentas necessárias para deixar o desenvolvimento deste tipo de aplicações mais simples e otimizado. Além disso, permite desenvolver aplicações web voltadas tanto para desktop quanto para aplicativo *mobile* (GUEDES, 2022).

É importante ressaltar que, apesar do Angular ser uma evolução do AngularJS, tratam-se de *frameworks* distintos. Angular (também conhecido como "Angular 2") foi uma reescrita completa do código do AngularJS e tornou-se uma arquitetura completamente diferente. Nessa nova implementação, foram seguidos padrões modernos, técnicas e boas práticas aprendidas com os erros cometidos no desenvolvimento da ferramenta anterior (GUEDES, 2022).

A Figura 2.4 apresenta os três arquivos principais que definem um componente Angular, o arquivo .ts, .css e .html, respectivamente de cima para baixo.

Figura 2.4 – Exemplo de componente Angular.

```
TS app.component.ts x
1  import { Component } from "@angular/core";
2
3  @Component({
4    selector: "app-root",
5    templateUrl: "./app.component.html",
6    styleUrls: ["./app.component.css"]
7  })
8  export class AppComponent {
9    greeting = "Hello World!";
10 }

# app.component.css x
1  .greeting {
2    color: red;
3    font-weight: bold;
4  }

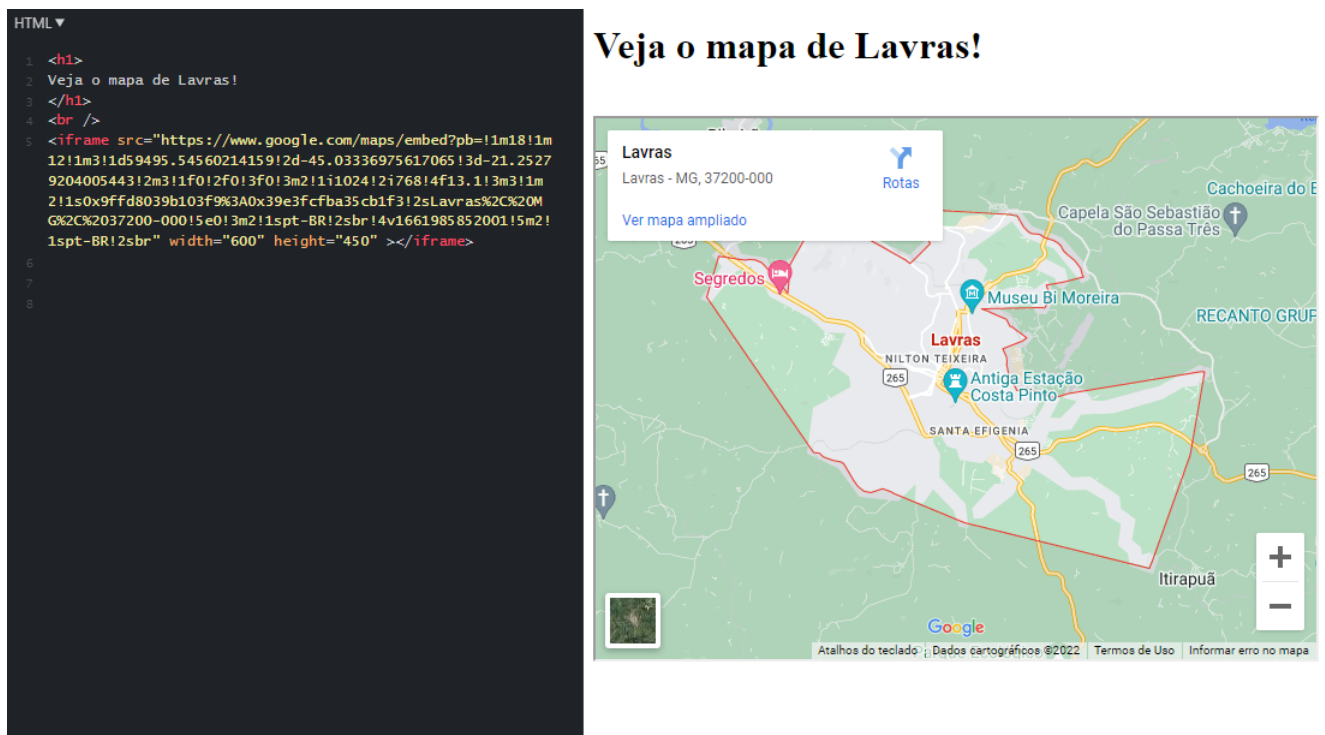
<> app.component.html x
1  <p class="greeting">
2    {{greeting}}
3  </p>
```

Fonte: Angular (2022) adaptado pelo autor.

### 2.3.3 iFrame

iFrame é uma *tag* de marcação HTML que representa um contexto de navegação aninhado, incorporando outra página HTML para a página atual (MDN, 2022). Os iFrames são bastante usados para integrar objetos de outras páginas em um site. Para utilizar uma tag iFrame, o atributo *src*, que define a URL da página a ser incorporada, deve ser utilizado. Outros atributos também podem ser definidos ao utilizar a tag, como por exemplo *height*, que indica a altura do elemento e *width* que indica a largura. A Figura 2.5 demonstra a utilização da *tag* iFrame para incorporar um mapa do serviço Google em uma página HTML simples. À esquerda da imagem está presente o código HTML e a direita o resultado do código em questão exibido em um navegador.

Figura 2.5 – Código HTML exemplificando a utilização da tag iFrame.



Fonte: Autor.

### 2.3.4 Visual Basic.NET

Visual Basic.NET (VB.NET) é uma linguagem de programação desenvolvida pela Microsoft e resultado da reescrita do Visual Basic 6.0. Para além da eliminação dos "lixos" resultantes das sucessivas atualizações do Visual Basic desde sua criação, a reescrita ao VB.NET possibilitou a adaptação da linguagem ao novo ambiente .NET, que pode ser explicado como uma

camada abaixo do software com o objetivo de fornecer um conjunto de serviços e funções básicas (MACKENZIE, 2002). Além disso, VB.NET oferece orientação a objetos aprimorada, incluindo uma biblioteca de componentes, o que permite o desenvolvimento rápido de aplicativos. O Visual Basic.NET permite também a interação via Internet através de padrões como XML (DEITEL, 2004).

De acordo com Barnes (2004), um programa escrito utilizando orientação a objetos modela alguma parte do mundo. Por sua vez, as partes das quais o modelo é construído são os objetos presentes no domínio do problema. A categorização dos objetos dá origem a classes que descrevem, de maneira abstrata, os objetos de um tipo em particular. Tais classes são formadas por propriedades, que representam as características do objeto representado.

A Figura 2.6 apresenta o código referente a uma classe aluno, com seus respectivos atributos, escrito em Visual Basic .NET.

Figura 2.6 – Classe Aluno escrita em VB.NET.

```
1. Public Class Aluno
2.     Public Identificador As Integer
3.     Public Nome As String
4.     Public Telefone As String
5.     Public Endereco As String
6.     Public Matricula As Integer
7. End Class
```

Fonte: Autor.

### 2.3.5 Web Forms

Segundo Macoratti (2022b), Web Forms são arquivos de texto que contém *tags*, similar a uma página HTML. Entretanto, existem alguns elementos que o diferem de uma simples página HTML, como por exemplo o fato de existirem blocos que são processados no servidor. Além disso, alguns eventos são levantados no cliente, mas tratados no servidor web, sendo as informações do evento transmitidas para o servidor via HTTP.

A Figura 2.7 demonstra um exemplo de código web forms bem como o código do lado do servidor associado a ele. A página em questão apresenta um simples conversor de metros para polegadas. A Figura 2.8 apresenta como a página seria exibida no navegador.

Figura 2.7 – Web Forms e código de servidor, escrito em VB.NET, associado.

```

Page Language="vb" AutoEventWireup="raise" CodeBehind="WebForm1.aspx.vb" Inherits="WebForm1.WebForm1"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<title>WebForm1</title>
<meta name="GENERATOR" content="Microsoft Visual Studio.NET 7.0">
<meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
<meta name="vs_defaultClientScript" content="JavaScript">
<meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
</HEAD>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">
<asp:Label id="Label1" style="Z-INDEX: 101; LEFT: 72px; POSITION: absolute; TOP: 58px" runat="server" Wj
<asp:TextBox id="TextBox1" style="Z-INDEX: 102; LEFT: 141px; POSITION: absolute; TOP: 54px" runat="serve
<asp:Button id="Button1" style="Z-INDEX: 103; LEFT: 144px; POSITION: absolute; TOP: 93px" runat="server"
<asp:Label id="lblresultado" style="Z-INDEX: 105; LEFT: 367px; POSITION: absolute; TOP: 60px" runat="ser
</form>
</body>
</HTML>

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim polegadas As Single
    polegadas = Val(txtmetros.Text) / 0.0256
    lblresultado.Text = polegadas & " polegadas "
End Sub

```

Fonte: Macoratti (2022a).

Figura 2.8 – Resultado de Web Forms em um navegador.

Fonte: Macoratti (2022a).

### 2.3.6 C#

A linguagem de programação C# é amplamente utilizada para criação de sistemas robustos. Foi desenvolvida pela Microsoft dentro da iniciativa .NET e tem como características ser orientada a objetos, permitindo a criação de sistemas por meio de código estruturado e de fácil manutenção (LEDUR, 2018).

Por ser uma linguagem orientada a objetos, todos os tipos de dados primitivos em C# são objetos no *namespace* do sistema. Entretanto, são fornecidos nomes curtos (ou *alias*) para cada tipo primitivo, por exemplo, *int* é o nome curto para *System.Int32* e *double* para *System.Double*.

Segundo a pesquisa SlashData (2020), C# conta com uma comunidade de aproximadamente 10 milhões de desenvolvedores apresentando-se, portanto, como uma linguagem bem

consolidada. Ainda segundo a mesma pesquisa, durante os dois últimos anos, o número de desenvolvedores que trabalham com C# cresceu em 2,2 milhões.

A Figura 2.9 apresenta a classe Aluno escrita em C#.

Figura 2.9 – Classe Aluno escrita em C#.

```
1 using System;
2
3 public class Auno
4 {
5     public int Identificador;
6     public string Nome;
7     public string Telefone;
8     public string Endereco;
9     public string Matricula;
10 }
11
```

Fonte: Autor.

## 2.4 Métodos ágeis

Segundo Pressman e Maxim (2021), um grupo de desenvolvedores de software, autores e consultores de renome assinou, em 2001, o "Manifesto para o desenvolvimento ágil de software. Esse manifesto defendia “indivíduos e interações acima de processos e ferramentas, software operacional acima de documentação completa, colaboração dos clientes acima de negociação contratual e respostas a mudanças acima de seguir um plano”. A partir das ideias fundamentais do desenvolvimento ágil, foram desenvolvidos métodos ágeis, com o objetivo de sanar fraquezas da engenharia de software tradicional.

É importante destacar que, no contexto de engenharia de software, agilidade está relacionada, principalmente, com resposta rápida a mudanças. Entretanto, agilidade também abrange a filosofia proposta no manifesto ágil e incentiva a estruturação e as atitudes em equipe que tornam a comunicação mais fácil. Além disso, agilidade também está relacionada a entrega rápida do software operacional e diminuição da importância dos artefatos intermediários (PRESSMAN; MAXIM, 2021).

### 2.4.1 Framework Scrum

*Scrum*<sup>1</sup> é um método de desenvolvimento ágil de software cujos princípios são coerentes com o manifesto ágil e que é usado para orientar as atividades de desenvolvimento dentro de

<sup>1</sup> Uma descrição completa sobre o Scrum pode ser encontrada em <<https://scrumguides.org/scrum-guide.html>>

um processo que incorpora as atividades metodológicas de requisitos, análise, projeto, evolução e entrega. As atividades realizadas em cada atividade metodológica são realizadas em janelas de tempo chamadas *sprints* (PRESSMAN; MAXIM, 2021).

O *framework Scrum* é composto por um *time Scrum* que é uma equipe interdisciplinar auto-gerenciável composta por um *product owner* (PO), um *Scrum master* e uma equipe de desenvolvimento. Além disso, o *Scrum* traz consigo alguns artefatos, dentre os quais os principais são o *backlog* do produto e o *backlog* da *sprint*. Além disso, o *Scrum* também define algumas cerimônias (ou reuniões) que ocorrem em momentos diferentes de uma *sprint*, sendo elas: planejamento da *sprint*, reunião diária, revisão da *sprint* e retrospectiva da *sprint*. Sbrocco e Macedo (2012) descrevem as funções de cada um dos papéis, artefatos e cerimônias do *Scrum*:

- *Product Owner - PO*: responsável por representar o cliente e por garantir que a equipe *Scrum* agregue valor ao negócio.
- *Scrum Master*: desempenha um papel de facilitador, responsável pela remoção de impedimentos que eventualmente surjam durante o desenvolvimento.
- Equipe de desenvolvimento: responsável pelo desenvolvimento do projeto em si, sendo composto geralmente por um grupo de cinco a nove pessoas, possuindo característica multifuncional.
- *Backlog* do produto: documento que representa a visão do produto, contendo os itens a serem desenvolvidos durante o projeto. Pode ser definido como uma lista de prioridades feitas no início do projeto. Visa esclarecer e elencar o que deve ser entregue para o cliente.
- *Backlog* da *sprint*: documento que contem as tarefas necessárias para cumprir a meta de uma determinada *sprint*.
- Planejamento da *sprint*: primeira reunião do projeto, deve contar com a participação de todos. O PO deve definir suas prioridades, selecionar os itens do *backlog* e a meta da *sprint*. A equipe, por sua vez, deve definir o *backlog* da *sprint*.
- Reunião diária: reunião em que cada membro deve responder sobre o que já fez, o que pretende fazer e relatar os impedimentos para a conclusão das tarefas em curso. Deve ser uma reunião rápida e com a presença apenas do time e do *Scrum Master*.



- Revisão da *sprint*: Reunião de balanço sobre o que foi feito durante uma *sprint*. Deve-se apresentar os resultados da *sprint* (itens que estiverem 100% prontos) para o *Product Owner*.
- Retrospectiva da *sprint*: Reunião que visa verificar o que houve de bom e o que pode ser melhorado em uma *sprint*. Devem ser avaliados aspectos relacionados ao trabalho em equipe, pontos positivos e negativos, além de serem feitas reflexões sobre estratégias de melhoria.

## 2.5 Expressão Regular - RegEx

Segundo Louden (2004), expressões regulares (*Regular expression* - ReGex) representam padrões de cadeias de caracteres. Um conjunto de caracteres com os quais uma expressão regular casa define essa expressão regular. Esse conjunto gera uma linguagem que depende do conjunto de caracteres disponível que, em geral, será o conjunto de caracteres ASCII ou algum subconjunto dele. O conjunto de caracteres legais (aceitos pela expressão regular), por sua vez, formam o alfabeto.

Além disso, uma expressão regular pode conter caracteres com significados especiais, sendo denominados metacaracteres ou meta-símbolos. Esses caracteres, portanto, de modo geral não são caracteres legais do alfabeto. Por outro lado, caso seja necessário incluir um metacartere como membro do alfabeto, usa-se um caractere de escape que "desliga" o caráter especial de um metacartere.

A expressão regular apresentada na Figura 2.10 tem por objetivo capturar o padrão de um endereço eletrônico. Apenas cadeias de caracteres que iniciam com "www.", tem apenas caracteres alfabéticos válidos (da letra a até a letra z) e terminam com ".com" são aceitos. Vale observar que o caractere de ponto final (.) é considerado um metacartere e, portanto, fez-se necessário o uso da contra barra (caractere de escape) para capturá-lo. O texto destacado em azul representa a cadeia capturada (e, portanto, válida) pela expressão regular.

Figura 2.10 – Exemplo de expressão regular e texto capturado por ela.



Fonte: Autor.

### 3 ATIVIDADES DESENVOLVIDAS

Este Capítulo explorará o desenvolvimento das Provas de Conceito para a reestruturação do sistema iTransport durante o período de estágio. Será explorada também a construção do analisador, a nível léxico, de código desenvolvido durante o mesmo período. Tais atividades supracitadas contribuíram para realização dos objetivos iniciais citados no Capítulo 1.2 ao passo que, para realizá-las foi necessário aplicar conceitos aprendidos durante o tempo de graduação do estagiário, bem como buscar novos conhecimentos ao se deparar com novos desafios.

Em um primeiro momento, foi concedido um período de estudos ao estagiário, tendo como objetivo desenvolver uma pesquisa técnica sobre as tecnologias em uso na empresa até aquele momento (VB.NET, JavaScript, SQL e T-SQL). Tendo sido finalizada a pesquisa, o time de desenvolvimento da empresa foi reunido para que o estagiário pudesse realizar uma apresentação dos resultados obtidos durante a pesquisa, bem como os demais membros do time pudessem arguir o estagiário.

Após esse período de pesquisa e apresentação, o estagiário foi alocado em uma equipe composta por 1 *Product Owner*, 3 desenvolvedores e 1 líder técnico, sendo o estagiário um dos desenvolvedores. Durante um primeiro momento, foi responsabilidade do estagiário o desenvolvimento de uma prova de conceito, utilizando Vue.JS, com o objetivo de verificar a viabilidade técnica de realizar a reestruturação do *front-end* do sistema iTransport. Os detalhes desta prova de conceito bem como o motivo pela qual Vue.JS foi preterido serão discutidos na Seção 3.2. Durante este mesmo período, os demais desenvolvedores do time ficaram alocados na efetiva reestruturação do *back-end* do sistema de VB.NET para C#.NET. Em um segundo momento, o estagiário foi responsável pelo desenvolvimento de uma prova de conceito em Angular, que também será discutida na Seção 3.2 e com objetivo similar a prova de conceito em Vue.JS. Além disso, também foi de responsabilidade do estagiário implementar um analisador de código que objetivou apontar enganos cometidos pelos desenvolvedores durante a reestruturação do *back-end* do sistema, o qual será discutido na Seção 3.4. Por fim, o estagiário juntou-se ao restante da equipe na reestruturação do *back-end* do sistema, o que será melhor discutido na Seção 3.3.

O desenvolvimento das atividades de todo o time foi realizado segundo os ritos do Scrum. As *sprints* duravam cerca de duas semanas de trabalho, sendo realizadas reuniões de planejamento de *sprint* antes do início de cada uma bem como reuniões de revisão e retrospectiva ao término de cada uma delas. Por fim, as reuniões diárias eram realizadas às 9:15.

### 3.1 iTransport

O iTransport é um produto vendido como *software as a service*, portanto, é disponibilizado através da internet como um serviço. Além disso, para que o software funcione corretamente, ele também é composto por tecnologias de geoprocessamento, navegação por GPS, rastreamento, monitoramento e controle de acesso embarcado. De modo geral, o sistema objetiva planejar, controlar, auditar e manter otimizados os serviços de transporte de outras empresas.

Apesar de ser utilizado principalmente por outras empresas em um modelo B2B (*Business to Business*), estão sob o guarda-chuva do iTransport aplicativos que são usado por motoristas e passageiros. Tais aplicativos auxiliam em funções como navegação ponto a ponto em rotas definidas, acompanhamento de veículos em tempo real para embarque em ônibus e táxi corporativo, uso de carro compartilhado, bem como visualização dos destinos de entrega e coleta de mercadorias.

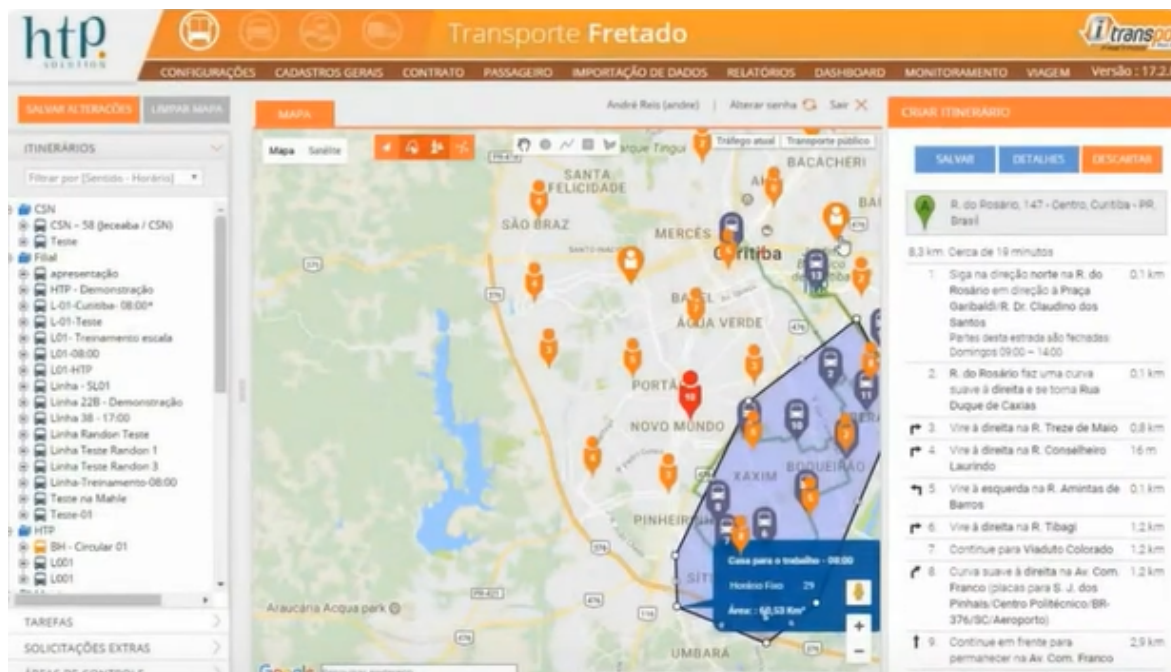
O sistema é dividido em módulos, sendo cada módulo para um tipo de demanda distinto. Dentre os principais módulos presentes no software destacam-se o fretado, o leve e o frotas. O módulo fretado é o principal módulo do sistema e objetiva gerenciar o transporte de funcionários em rotas de trabalho contando com monitoramento *online*, controle de acesso, aplicativo para passageiro e motorista. O módulo leve, por sua vez, visa realizar o gerenciamento de demandas por táxi corporativo, carro compartilhado e reserva de frotas com seus veículos ou de terceiros. Por fim, o módulo frotas visa gerenciar veículos com monitoramento *online*, telemetria, manutenção, controle de combustível, bloqueio e desbloqueio remoto.

De modo geral, o sistema web (no qual a reestruturação se concentrou) funciona através de um sistema de abas. Todas as funcionalidades disponíveis estão presentes no cabeçalho da página e, ao serem clicadas, abrem abas na parte central da página. A disposição do *layout* pode ser vista na captura de tela na Figura 3.1, em que a aba aberta "Mapa" mostra a gestão de rotas do módulo fretado.

### 3.2 Provas de conceito

A ideia de reestruturação do *front-end* do sistema surgiu através da dificuldade, relatada pelos administradores da empresa, em encontrar profissionais capacitados a trabalhar com Web Forms e JavaScript puro, tecnologias até então utilizadas pelo sistema iTransport. Apesar disso, ainda não havia sido definido qual seria a nova tecnologia empregada. Durante as reuniões

Figura 3.1 – Captura de tela do sistema iTransport. Ao centro a aba "Mapa" em gestão de rotas.



Fonte: iTransport (2017).

diárias, o estagiário relatou ter breve experiência em Vue.Js e, portanto, o líder técnico, em comum acordo com a gerência da organização, solicitou ao estagiário o desenvolvimento de uma prova de conceito utilizando o *framework*. Além da experiência do estagiário, fatores como a menor curva de aprendizado em relação a outros *frameworks* (VUE.JS, 2022a) e a existência de uma grande comunidade com possibilidade de obter ajuda em fóruns e *chats* (VUE.JS, 2022c) foram fatores determinantes para a escolha desse *framework*.

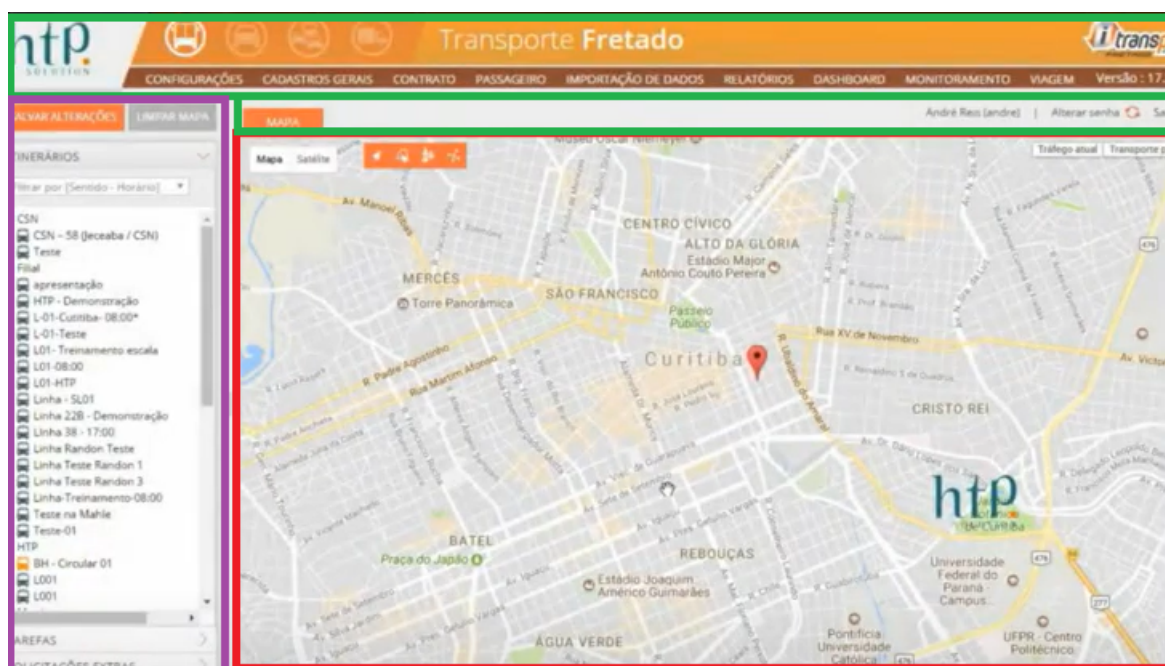
Dentre as diretrizes estabelecidas para o desenvolvimento da prova de conceito, foi especificado que a reestruturação deveria ser gradual, permitindo que os códigos já escritos fossem sendo apagados aos poucos e que não fosse necessário realizar toda a reestruturação de uma vez para o novo código funcionar. Para permitir esse comportamento, o *template* básico do sistema foi reescrito em Vue, entretanto as abas de cada uma das funcionalidades foram mantidas em VB.NET e inseridas no *template* através de *iframes*.

A Figura 3.2 detalha, por meio de uma captura de tela do sistema, todos os elementos que constituem o *template* básico do sistema:

- Verde: representa o cabeçalho, em que estão presentes informações sobre o nome do sistema, troca de módulos e menu de funcionalidades existentes. Toda essa seção foi reescrita e, portanto, reestruturada.

- Azul: representa a listagens de abas abertas. Sempre que uma funcionalidade é escolhida através do menu do cabeçalho, uma nova aba é aberta e fica listada nessa seção. Essa seção também foi reescrita e portanto era de responsabilidade da prova de conceito controlar quais abas estavam abertas e realizar o fechamento de cada uma delas.
- Vermelho: representa a área em que cada tela do sistema é aberta em forma de abas. Para possibilitar a utilização de código legado, a prova de conceito possibilitava que as telas que ainda não tivessem sido reestruturadas fossem abertas por meio de iframes. Por sua vez, telas que já tivessem sido reescritas em novos componentes eram apresentadas no mesmo espaço.
- Preto: representa a árvore de interações possíveis com o mapa. Por se tratar de um elemento complexo que deve interagir com serviços externos da Google Maps não foi reestruturado, permanecendo no template da prova de conceito através de *iframes*.

Figura 3.2 – Captura de tela do sistema iTransport.



Fonte: iTransport (2017) adaptado pelo autor.

Outra diretriz passada ao estagiário para o desenvolvimento da prova de conceito foi que deveria ser dada preferência ao estilo de codificação utilizando classes (programação orientada a objetos) na definição dos componentes. Tal diretriz foi dada pelo líder técnico e justificava-se, segundo ele, pela familiaridade dos desenvolvedores com esse estilo de programação. O estagiário realizou pesquisas e constatou que, apesar de existirem iniciativas para permitir esse

estilo de programação <sup>1</sup>, esse estilo não recebia suporte oficial pelo *core* do framework, além de receber algumas críticas<sup>2</sup> de Evan You, um dos criadores do *framework*. Por esses motivos constatou-se que Vue.Js não atenderia as demandas da empresa, dando lugar ao desenvolvimento de prova de conceito em Angular.

De forma análoga à prova de conceito em Vue, foi desenvolvida uma prova de conceito utilizando Angular. Tendo em vista que a forma padrão de declaração de componentes em Angular é através de classes, o conflito com a diretriz de utilizar o máximo de orientação a objetos foi superado. Os conceitos usados na prova de conceito em Vue foram seguidos na versão em Angular, mantendo-se a utilização de *iframes* para interoperabilidade com o código ainda não reestruturado.

Durante esta fase, o estagiário teve contato com alguns conceitos importantes para o *framework* Angular, como a biblioteca RxJS para garantir que os paradigmas de programação assíncrona e baseada em eventos fossem aplicados, por exemplo, que são característicos do *framework*.

O Quadro 3.1 apresenta um comparativo, no contexto de seu uso durante o estágio na empresa, entre os *frameworks* Vue.JS e Angular. É perceptível que, apesar de ambos *frameworks* apresentarem características positivas, algumas delas apresentaram maior peso na decisão pelo uso de Angular, destacando-se a definição de componentes através de classes.

Quadro 3.1 – Comparativo entre Vue.JS e Angular no contexto de utilização no estágio na empresa HTP.

Característica	Linguagem	
	Vue.JS	Angular
Possibilidade de integração com o código já existente	V	V
Definição de propriedades de componente utilizando-se classes (orientação a objetos) de maneira nativa no framework	X	V
Existência de uma grande organização que desenvolve novas versões	X	V
Apresenta maior curva de aprendizado (dentre os analisados)	V	X
Ampla comunidade ativa	V	V
Conhecimento prévio por parte do estagiário	V	X

Fonte: Autor.

### 3.3 Reestruturação Back-end

A reestruturação do *back-end* da aplicação teve foco no módulo fretado do iTransport devido a esse ser o módulo mais utilizado pelos clientes e usuários do sistema. A reestruturação

<sup>1</sup> <<https://github.com/vuejs/vue-class-component>>

<sup>2</sup> <<https://github.com/vuejs/vue/issues/2371#issuecomment-330068362>>

consistiu, em um primeiro momento, apenas em uma tradução simples entre as linguagens VB.NET e C#. Entretanto, existiram recomendações, providas pelo líder técnico, para que em algumas partes específicas do sistema, além de uma tradução, fosse realizada alguma melhoria, tanto de legibilidade quanto de desempenho. Apesar disso, esperava-se que, do ponto de vista de lógica de negócios, ambos os códigos fossem iguais.

De mesmo modo como ocorreu no *front-end*, os códigos do *back-end* deveriam ser traduzidos aos poucos, sendo necessária a interoperabilidade entre os códigos em VB.NET e C#. Graças ao CLS - *Common Language Specification* fornecido pelo .NET, foi possível integrar códigos C# a códigos VB.NET (que ainda não haviam sido reestruturados) e vice-versa.

### 3.4 Analisador de código a nível léxico

A tarefa de reestruturação de VB.Net para C# necessitava de extrema atenção por parte dos desenvolvedores. Muitas vezes, uma rápida distração era suficiente para trocar um caractere de lugar e resultar em um defeito no código. Algumas vezes, tais falhas eram facilmente identificadas, pois resultavam em um erro que rapidamente eram percebidos pelos programadores. Outras vezes, entretanto, defeitos só ocasionavam erros em fluxos muito específicos do software que muitas vezes não eram previstos pelos programadores e, portanto, não eram facilmente identificados. Para tentar mitigar a quantidade de defeito durante o processo de reestruturação, o estagiário propôs, durante uma reunião diária, a criação de um analisador, a nível léxico, de código que, utilizando expressões regulares, que iria comparar o código escrito em VB.NET com o já traduzido para C#.

Para realizar essa tarefa foi necessário entender alguns detalhes da sintaxe de ambas as linguagens, com o objetivo de recuperar o elementos léxicos de ambos os códigos. Em VB.NET, os campos de uma classe, por exemplo, são declarados através de um modificador de acesso ("Public", "Private", "Protected" ou "Friend"), seguido do nome do campo acompanhado da palavra reservada "As" e do tipo de dado que aquele campo guardaria. De maneira análoga, para declarar um campo de uma classe em C#, deve-se definir o modificador de acesso ("public", "private", "protected" ou "internal") seguido do tipo do dado do campo acompanhado do nome do campo. Para verificar a correspondência entre os dois códigos, portanto, o estagiário utilizou grupos de captura de expressões regulares para, por exemplo, capturar os identificadores (elemento léxico) e verificar se em ambos os códigos tais identificadores eram iguais.



A Figura 3.3 demonstra como o nome de um campo de classe escrita em VB.NET pode ser capturado. O primeiro bloco da figura destaca a expressão regular utilizada, em que o grupo de captura responsável por identificar o nome do campo é nomeado de "NomeDoCampo". O segundo bloco da figura demonstra um exemplo de trecho de código sob o qual a expressão regular iria interagir. Por fim, o último bloco da figura apresenta os grupos de captura encontrados.

Figura 3.3 – Aplicação de expressão regular para captura de nome dos campos de uma classe.

```

/(\bPublic\b|\bPrivate\b|\bProtected\b) \b(?:<NomeDoCampo>[A-z]+) \bAs\b [A-Z][a-z]+/g

```

```

Public Class Aluno
    Public Identificador As Integer
    Private
    ...
End Class

```

Correspondência completa	Public Identificador As Integer
Grupo 1	Public
Grupo "NomeDoCampo"	Identificador

Fonte: Autor.

No exemplo demonstrado na Figura 3.3, bastaria comparar o grupo "NomeDoCampo" com o grupo capturado por outra expressão regular aplicada sob a reestruturação em C#. Caso os nomes fossem diferentes, estaria identificado um defeito durante a reestruturação do código.

O processo de realizar a captura em ambos os arquivos e verificar a correspondência entre os grupos capturados foi automatizado por meio da implementação de um algoritmo em JavaScript e executado através do ambiente de execução Node.js<sup>3</sup>.

<sup>3</sup> <<https://nodejs.org/>>

### 3.5 Desafios

O estágio representou, para o estagiário, um grande desafio técnico em um primeiro momento. A necessidade de se desenvolver provas de conceito que introduzissem novas tecnologias sem a necessidade de se descartar tudo o que já estava escrito em outras linguagens foi o principal fator de tal desafio. O uso de *iframes*, por exemplo, em meio a um *framework* que prioriza componentes, demonstra como foi necessário transitar entre os mais diversos conhecimentos aplicados à tecnologia *web*, sejam eles mais antigos ou modernos.

Tal necessidade de que tecnologias modernas coabitassem um sistema com tecnologias mais antigas trouxeram alguns problemas. Um exemplo foi a necessidade de se manter o estado das abas ao transitar entre elas. Utilizando-se um sistema feito completamente utilizando *frameworks* modernos, uma simples estratégia de reuso de componentes que mantivesse o estado deles resolveria o problema. Entretanto, a estratégia de reuso consegue gerenciar apenas os elementos auto-contidos no *framework* e não um *iframe*. Para contornar essa limitação, foi necessário que todas as telas abertas em abas estivessem fora de qualquer componente, sendo apenas escondidas via estilização quando não estivessem abertas.

Outro desafio enfrentado foi a necessidade de que algumas telas do sistema se adaptassem aos diversos tamanhos de vídeo de dispositivos, que vão desde um computador de mesa até uma tela de celular. Algumas abas dos sistema, portanto, deveriam ser responsivas, para que pudessem ser acessadas de celulares e tablets.

## 4 CONCLUSÃO

O estágio foi extremamente importante para que o estagiário tivesse contato com o mundo corporativo e com outros desenvolvedores com maior experiência. O contato entre a experiência acadêmica e empresarial possibilitou ricas conversas e ensinamentos.

O ambiente de trabalho encontrado na HTP, apesar de ser na modalidade *home-office*, foi extremamente acolhedor e propício para que o estagiário se desenvolvesse profissional e pessoalmente. Os colegas de trabalho sempre se mostraram muito dispostos a escutar o que o estagiário tinha a dizer bem como repassar seus ensinamentos e técnicas.

Durante o período de estágio, poderiam ter sido colocadas em prática mais ferramentas de testes automatizados, visto que poderia aumentar a qualidade de entrega de todos os desenvolvedores. Outro ponto que poderia ser melhorado é a documentação do código, visto que, por se tratar de um sistema relativamente grande e complexo, muitas vezes era difícil compreender corretamente o que determinado trecho se propunha a fazer, o que por vezes tomava tempo excessivo.

Destaca-se que a experiência vivida, pelo estagiário, na empresa Comp Júnior (EJ) durante sua graduação foi extremamente importante para se adaptar mais rapidamente ao mundo corporativo. Além disso, a experiência técnica adquirida durante este período foi determinante para, por exemplo, realizar sugestões de linguagens que poderiam ser adotadas ou quais técnicas seriam mais adequadas, na visão do estagiário, para solucionar determinados problemas. Outro ponto importante proporcionado pela empresa Júnior, foi a habilidade de trabalhar em grupo. Anteriormente à experiência em EJ, o estagiário tinha por escassas vezes programado, em um mesmo projeto, com outras pessoas.

Deve-se destacar também a importância das disciplinas cursadas durante o curso de Ciência da Computação. A fundamentação teórica adquirida durante todo o curso mostrou-se essencial para abstrair os problemas que surgiam e pensar analítica e algoritmicamente para solucioná-los.

Dentre a gama de disciplinas cursadas durante a graduação, destaca-se a importância de algumas, como Estruturas de Dados, Linguagens Formais e Autômatos e Paradigmas de Linguagens de Programação, as quais mostraram-se de extrema importância para solucionar os problemas enfrentados ao longo do estágio. A metodologia adotada pelos professores, principalmente das disciplinas citadas, de realização de trabalhos, auxiliaram muito no desenvol-

vimento de habilidades sociais bem como na busca de soluções de problemas propostos pelos professores.

## REFERÊNCIAS

- ANGULAR. **O que é o Angular e para que serve?** 2022. Disponível em: <<https://angular.io/guide/what-is-angular>>.
- BARNES, D. J. **Programação orientada a objetos com Java: Uma Introdução Prática Usando o Blue J.** [S.l.]: Pearson Education, 2004. ISBN 9788576050124.
- CARDOSO, G. C.; CARDOSO, V. M. **Linguagem SQL, fundamentos e práticas - 1ª edição.** [S.l.]: Editora Saraiva, 2013. ISBN 9788502200463.
- DEITEL, H. M. **Visual Basic.Net: Como Programar.** [S.l.]: Makron Books, 2004. ISBN 9788534614771.
- FERNANDES, D. **iTransport - Platform as a Service.** 2017. Disponível em: <<https://blog.rocketseat.com.br/react-do-zero-componentizacao-propriedades-e-estado/>>.
- FLANAGAN, D. **JavaScript: o guia definitivo.** [S.l.]: Grupo A, 2011. ISBN 9788565837484.
- FRANCA, S.; GOYA, M.; PUGA, S. **Banco de dados: Implementação em SQL, PL/SQL e Oracle 11g.** [S.l.]: Pearson, 2013. ISBN 9788581435329.
- GUEDES, M. **O que é o Angular e para que serve?** 2022. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-o-angular-e-para-que-serve/>>.
- ITRANSPORT. **iTransport - Platform as a Service.** 2017. Disponível em: <[https://youtu.be/jtgL4\\_R5mgI](https://youtu.be/jtgL4_R5mgI)>.
- LEDUR, C. L. **Desenvolvimento de sistemas com c#.** [S.l.]: Grupo A, 2018. ISBN 9788595023147.
- LEMAY, L.; COLBURN, R.; TYLER, D. **Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias.** [S.l.]: Editora Pearson, 2002. ISBN 9788534614283.
- LOUDEN, K. C. **Compiladores: princípios e práticas.** [S.l.]: Cengage Learning Brasil, 2004. ISBN 9788522128532.
- MACKENZIE, D. **Aprenda visual basic.NET em 21 dias.** [S.l.]: Makron Books, 2002. ISBN 9788534614504.
- MACORATTI, J. C. **ASP .NET - Criando seu primeiro Web Forms.** 2022. Disponível em: <[https://www.macoratti.net/vbn\\_wbf.htm](https://www.macoratti.net/vbn_wbf.htm)>.
- MACORATTI, J. C. **ASP .NET Web Forms - Como funciona ?** 2022. Disponível em: <<https://www.macoratti.net/webfrm1.htm>>.
- MARCOLINO, A. d. S. **Frameworks Front End.** [S.l.]: Editora Saraiva, 2021. ISBN 9786589965077.
- MDN. **<iframe>.** 2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/iframe>>.
- MENA, I. **Verbete Draft: o que é Prova de Conceito.** 2017. Disponível em: <<https://www.projetodraft.com/verbete-draft-o-que-e-prova-de-conceito/>>.

MICROSOFT. **Introdução ao Transact-SQL**. 2022. Disponível em: <<https://docs.microsoft.com/pt-br/learn/modules/introduction-to-transact-sql/1-introduction>>.

MICROSOFT. **Introdução à CLI do Vue e a componentes de arquivo único no Vue.js**. 2022. Disponível em: <<https://docs.microsoft.com/pt-br/learn/modules/vue-cli-components/>>.

OLIVEIRA, C. L. V.; ZANETTI, H. A. P. **Node.js: programe de forma rápida e prática**. [S.l.]: Editora Saraiva, 2021. ISBN 9786558110217.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. [S.l.]: Grupo A, 2021. ISBN 9786558040118.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados**. [S.l.]: McGraw-Hill Medical, 2008. ISBN 9788577260270.

SBROCCO, J. H. T. d. C.; MACEDO, P. C. D. **Metodologias Ágeis - Engenharia de Software sob Medida**. [S.l.]: Editora Saraiva, 2012. ISBN 9788536519418.

SLASHDATA. **STATE OF THE DEVELOPER NATION TH 22nd EDITION**. 2020. Disponível em: <[https://developer-economics.cdn.prismic.io/developer-economics/f99dc570-f8f4-41f0-bc3b-808b2dcd6cb\\_Slashdata+-+22nd+edition+of+The+State+of+the+Developer+Nation+%28Q1+2022%29.pdf](https://developer-economics.cdn.prismic.io/developer-economics/f99dc570-f8f4-41f0-bc3b-808b2dcd6cb_Slashdata+-+22nd+edition+of+The+State+of+the+Developer+Nation+%28Q1+2022%29.pdf)>.

TRACKJOB. **Controle de fretamento de funcionários: como efetuar?** 2018. Disponível em: <<https://trackjob.com.br/controle-de-fretamento-de-funcionarios/>>.

VUE.JS. **Comparação com outros Frameworks - Vue.js**. 2022. Disponível em: <<https://br.vuejs.org/v2/guide/comparison.html#Curva-de-Aprendizado>>.

VUE.JS. **Introdução - Vue.js**. 2022. Disponível em: <<https://br.vuejs.org/v2/guide/index.html>>.

VUE.JS. **Junte-se à Comunidade Vue.js!** 2022. Disponível em: <<https://br.vuejs.org/v2/guide/join.html>>.

VUE.JS. **Single-File Components**. 2022. Disponível em: <<https://vuejs.org/guide/scaling-up/sfc.html#introduction>>.