

# RouteU: Desenvolvimento de um Sistema *Web* para roteirizar entregas

Valéria Oliveira de Paulo <sup>1</sup>, Daiane Mastrangelo Tomazeti<sup>1</sup>

<sup>1</sup>Tecnologia em Análise e Desenvolvimento de Sistemas – Instituto Federal de São Paulo –  
Câmpus Hortolândia(IFSP-HTO)

Avenida Thereza Ana Cecon Breda, 1896 - Vila São Pedro – 13183-250 – Hortolândia – SP – Brasil

valeriapaulo10@gmail.com, daianetomazeti@ifsp.edu.br

**Abstract.** *This article proposes the development of a web system for delivery routing, designed to be used by parcel deliverers. The main objective of the system is to provide an efficient and automated solution to create optimized delivery routes, saving time and resources. The methodology used involves bibliographical research to support the development, the adoption of the incremental model for flexibility in the development process, the use of Leafletjs and the Leaflet Routing Machine API for mapping and routing resources, user testing, including structured questionnaires to evaluate various aspects of the system, and the collection and analysis of data to validate the results. Despite facing setbacks, such as the unfeasibility of the Google Maps API due to costs, partial development progress appears to be viable, with the system being well received by users.*

**Resumo.** *O presente artigo tem como principal foco o desenvolvimento de um sistema web para roteirização de entregas, projetado para ser utilizado por entregadores de encomendas. O objetivo principal do sistema foi fornecer uma solução eficiente e automatizada para criar rotas de entrega otimizadas, economizando tempo e recursos. A metodologia empregada envolveu uma pesquisa bibliográfica para embasar o desenvolvimento, a adoção do modelo incremental para flexibilidade no processo de desenvolvimento, a utilização do Leafletjs e da API Leaflet Routing Machine para recursos de mapeamento e roteirização,*

*testes com usuários, incluindo questionários estruturados para avaliar diversos aspectos do sistema, e a coleta e análise de dados para validação dos resultados.*

## **1. Introdução**

O presente artigo tem como foco principal desenvolver um Sistema *Web* para entregadores de encomendas roteirizarem entregas, o qual será chamado RouteU. Para ter trabalho constante, os entregadores de encomendas precisam de clientes, empresas que solicitem seus serviços de entrega regulares, para isso é necessário um alto volume de vendas por partes destes. Quanto ao ponto chave para efetivação de uma venda, considera-se:

Na Hewlett-Packard, descobriu-se que para os clientes que colocam um pedido on-line antes da cinco da tarde o mais importante é receber o produto no dia seguinte. A hora da verdade, o parâmetro que mede a empresa, é a entrega em 24 horas (LARENTIS 2012, p.80).

Sendo assim, a eficiência e a agilidade na entrega de produtos tornaram-se aspectos cruciais nos negócios em um mundo cada vez mais orientado para o comércio eletrônico. Entregar um produto no momento certo é muitas vezes o ponto chave para garantir a satisfação do cliente.

De acordo com o cenário apresentado, o objetivo deste artigo é o desenvolvimento de uma ferramenta para auxiliar esses entregadores na otimização do seu trabalho, e aprofunda o tema nos seguintes pontos: automatização na criação de um roteiro de entregas, apresentação de um algoritmo de otimização de rotas, e testes com os entregadores para garantir que o sistema seja seguro e eficiente.

A fim de atingir o objetivo proposto, foi feita uma pesquisa bibliográfica sobre as ferramentas a serem utilizadas, utilização do modelo incremental de desenvolvimento para que se ocorra algum imprevisto, tenha flexibilidade para incorporação ou troca de ferramentas a medida que o projeto avança, e possibilitando testar partes do sistema com os usuários.

### **1.1. Justificativa**

A escolha do desenvolvimento de um Sistema *Web* para entregadores de encomendas roteirizarem entregas – RouteU, justifica-se porque não é uma tarefa fácil entregadores ma-

nualmente criarem roteiros de múltiplas entregas de maneira ágil e eficaz, o que demanda tempo e conhecimento ou estudo prévio de endereços.

Entregadores de encomendas, principalmente autônomos, vivem este desafio diariamente, com base no dinamismo diário desta atividade e podendo melhorar a qualidade do serviço oferecido. Assim, o presente trabalho partiu da necessidade de proporcionar uma solução abrangente e eficiente, capaz de simplificar e reduzir o tempo no processo de criação de um roteiro de entregas.

## **1.2. Objetivo**

### **1.2.1. Objetivo Geral**

Desenvolver um Sistema *Web* como ferramenta automatizada para auxiliar entregadores a criarem um roteiro de entregas.

### **1.2.2. Objetivos Específicos**

1. Desenvolver um sistema *web* que permita aos entregadores criar um roteiro de entregas de forma automatizada, com base em informações de localização.

2. Implementar um algoritmo de otimização de rotas que permita aos entregadores encontrar a melhor rota possível para cada entrega.

3. Testar o sistema *Web* para garantir que ele seja eficiente e seguro para uso por entregadores.

## **2. Fundamentação Teórica**

Nesta seção serão apresentados conceitos de diferentes autores sobre o tema desenvolvimento de sistema *Web*, geolocalização e algoritmo de otimização de roteiros de endereços além dos programas, ferramentas e arquiteturas que auxiliaram na criação e execução deste trabalho.

### **2.1. Desenvolvimento *Web***

O desenvolvimento *web* abrange a aplicação de várias tecnologias. Isso inclui linguagens de marcação, como HTML e CSS, para organizar e dar estilo ao conteúdo, junta-

mente com a linguagem de programação JavaScript, que permite a adição de interatividade e dinamismo às páginas *Web*. Além disso, é prática comum fazer uso de bancos de dados e servidores para armazenar e processar informações. Um dos principais objetivos do desenvolvimento *web* é a criação de *sites* e aplicativos que sejam acessíveis, funcionais e eficazes. No desenvolvimento *Web*, o cliente usa um navegador (*browser*) para acessar a aplicação, necessitando apenas a instalação do navegador em sua máquina (PACHECO 2023). Para o desenvolvimento *Web* serão utilizadas as tecnologias HTML5, TypeScript, CSS, PostgreSQL, React, além da ferramenta para edição dos códigos: Visual Studio Code.

### **2.1.1. HTML5**

O HTML é baseado no conceito de Hipertexto. Hipertexto são conjuntos de elementos – ou nós – ligados por conexões. Estes elementos podem ser palavras, imagens, vídeos, áudio, documentos, entre outros. Estes elementos conectados formam uma grande rede de informação. Eles não estão conectados linearmente como se fossem textos de um livro, onde um assunto é ligado ao outro seguidamente. A conexão feita em um hipertexto é algo imprevisível que permite a comunicação de dados, organizando conhecimentos e guardando informações relacionadas (FERREIRA e EIS 2011).

### **2.1.2. TypeScript**

TypeScript é uma linguagem de programação desenvolvida pela Microsoft que é um superconjunto de JavaScript. Isso significa que todo o código JavaScript é válido em TypeScript. No entanto, TypeScript adiciona recursos extras, especialmente relacionados à tipagem dinâmica, ou seja é possível ver erros antes da compilação diferente do Javascript onde o erro só é detectado após a compilação (ABREU 2017).

### **2.1.3. CSS**

Cascading Style Sheets<sup>TM</sup>(CSS), permite aos autores de documento especificar a apresentação dos elementos em uma página *Web*, tais como fontes, espaçamentos, cores,

entre outros, separadamente da estrutura do documento, em geral composta por cabeçalhos de seção, texto do corpo, *links*, entre outros. Essa separação entre estrutura e apresentação dos elementos simplifica a manutenção e a modificação de uma página *Web* (DEITEL e DEITEL 2008).

#### **2.1.4. React**

React é uma biblioteca JavaScript usada para renderizar interfaces de usuário (UI). Ela possibilita a construção da interface do usuário a partir de elementos individuais, como botões, texto e imagens, os quais podem ser agrupados em componentes encapsulados e reutilizáveis. Isso significa que, seja em sites ou aplicativos de telefone, todos os elementos na tela podem ser subdivididos em componentes (META OPEN SOURCE 2023).

#### **2.1.5. PostgreSQL**

O PostgreSQL é um sistema de banco de dados relacional de código aberto, desenvolvido desde 1986 como parte do projeto POSTGRES na Universidade da Califórnia em Berkeley. Com mais de 35 anos de desenvolvimento, é conhecido por sua confiabilidade, integridade de dados e conjunto robusto de recursos. Compatível com todos os principais sistemas operacionais e aderindo ao padrão ACID desde 2001, o PostgreSQL oferece extensibilidade e possui complementos poderosos, como o PostGIS para dados geoespaciais. Sua popularidade cresceu como o banco de dados relacional de código aberto preferido, proporcionando uma base sólida para uma variedade de projetos (GROUP 2023).

#### **2.1.6. Node.js**

O Node.js é uma plataforma que permite a execução de código JavaScript no lado do servidor, criada por Ryan Dahl em 2009 para preencher a lacuna de aplicativos JavaScript fora do navegador. Isso permite aos desenvolvedores utilizar uma linguagem familiar e simplificada para construir aplicativos do lado do servidor, aproveitando sua sintaxe simplificada e o modelo assíncrono de programação, ideal para aplicações em tempo real e de alta escalabilidade (BESSA 2024).

### **2.1.7. Next.js**

Next.js é uma estrutura baseada em React para a construção de aplicativos web full-stack, oferecendo recursos e otimizações adicionais. Abstrai ferramentas necessárias para React, como agrupamento e compilação, permitindo foco na construção do aplicativo. Oferece roteamento, renderização no cliente e servidor, busca de dados simplificada, suporte para vários métodos de estilo e otimizações para melhorar a experiência do usuário, incluindo suporte a TypeScript (INC 2024).

### **2.1.8. Visual Studio Code**

O Visual Studio Code (VS Code) é um editor de código, e é de código aberto, foi desenvolvido pela Microsoft, é uma poderosa ferramenta capaz de fazer várias coisas, foi feito para ser simples porém possui extensões diversas. Possui suporte a várias linguagens de programação, terminal de comandos integrado e controle de versão (HANASHIRO 2023).

### **2.1.9. API's**

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de *software* ou plataforma baseado na *Web*. A sigla API refere-se ao termo em inglês "*Application Programming Interface*" que significa em tradução para o português "Interface de Programação de Aplicativos"(CANALTECH 2024).

### **2.1.10. Leaflet JavaScript API**

É uma biblioteca JavaScript *open-source* para mapas interativos, foi projetado com simplicidade, desempenho e usabilidade, funciona de maneira eficiente em todas as principais plataformas *desktop* e móveis, pode ser estendido com vários plugins, possui uma API bonita, fácil de usar e bem documentada (LEAFLET 2024).

### **2.1.11. Leaflet Routing Machine API**

Leaflet Routing Machine é uma maneira fácil, flexível e extensível de adicionar rotas a um mapa Leaflet . Usa o padrão envolvendo apenas algumas linhas de código para adicionar roteamento totalmente funcional, mas permite ainda a personalização de quase todos os aspectos da interface do usuário e das interações, é de código aberto, permite inserção roteamento do início ao destino, possibilitando incluir pontos de parada. Vem por padrão o OSRM-OPEN SOURCE ROUTING MACHINE Api para gerar melhor caminho. que permite a importação flexível de dados do OpenStreetMap (LIEDMAN 2024).

### **2.1.12. OpenStreetMap (OSM)**

O OpenStreetMap (OSM) é uma iniciativa colaborativa de mapeamento que visa criar um mapa do mundo livre e editável. O nome traduzido para o português significa Mapa Aberto de Ruas. Este projeto disponibiliza dados para centenas de *sites* na internet, aplicativos de celular e outros dispositivos. Em julho de 2004, Steve Coast fundou o OpenStreetMap. Em abril de 2006, foi estabelecida uma fundação com o objetivo de fomentar o crescimento, desenvolvimento e distribuição de dados geoespaciais livres, disponibilizando-os para uso e compartilhamento amplo. Em dezembro de 2006, a Yahoo confirmou que o OSM tinha permissão para utilizar suas imagens aéreas como base para a produção de mapas (WIKIPEDIA 2024).

### **2.1.13. OSRM - OPEN SOURCE ROUTING MACHINE API**

O OSRM-OPEN SOURCE ROUTING MACHINE API, que permite a importação flexível de dados do OpenStreetMap e implementa o algoritmo de Dijkstra para determinar a melhor rota entre dois ou mais pontos (SCHMITT 2013).

### 2.1.14. Engenharia de Requisitos

Requisitos funcionais são declarações dos serviços que o sistema deve fornecer, descrevendo como ele deve reagir a determinadas entradas e o seu comportamento em situações específicas. Requisitos não funcionais são restrições sobre os serviços ou funções do sistema, abrangendo restrições de tempo, processo de desenvolvimento e padrões impostos (SOMMERVILLE 2018, 88-89).

<b>Requisitos Funcionais</b>
Cadastro de Entregadores: O sistema deve permitir o cadastro de usuários, nome, e-mail e senha.
Efetuar Login : Autenticação do usuário para acesso ao Sistema.
Gestão de Endereços: Os entregadores devem ser capazes de registrar endereços, especificando informações como destino.
Roteirização Automatizada: O sistema deve oferecer a funcionalidade de criar roteiros de entrega automaticamente, com base nas informações dos endereços e na localização atual dos entregadores.
Algoritmo de Otimização de Rotas: O sistema deve implementar um algoritmo de otimização de rotas que considere fatores como distância e restrições de tempo para determinar a melhor ordem de entregas.
Visualização de Rotas no Mapa: Deve haver uma interface que permita aos entregadores visualizar as rotas no mapa, facilitando a navegação.
Desempenho: Deve ser possível ver métricas de desempenho, como tempo gasto do início ao final da rota.

**Tabela 1. Requisitos Funcionais.**

<b>Requisitos não Funcionais</b>
<b>Usabilidade:</b> O sistema deve ser intuitivo e fácil de usar, mesmo por usuários com pouca experiência em tecnologia.
<b>Desempenho:</b> O sistema deve ser capaz de lidar com um grande volume de informações e calcular rotas otimizadas em tempo real.
<b>Segurança:</b> Deve haver medidas de segurança para proteger os dados dos entregadores.
<b>Disponibilidade:</b> O sistema deve estar disponível a maior parte do tempo, minimizando o tempo de inatividade.
<b>Integração com a API Leaflet JavaScript e Leaflet Routing Machine API</b> O sistema deve ser capaz de se integrar de forma eficaz com as API's Leaflet para cálculos de rotas e visualização de mapas.
<b>Eficiência:</b> O sistema deve ser eficiente na criação de roteiros de entrega, garantindo que as rotas sejam otimizadas em um tempo razoável.

**Tabela 2. Requisitos não Funcionais.**

### **3. Trabalhos Correlatos**

#### **3.1. MyRouteOnline**

A Figura 1 mostra o *software* MyRouteOnline, que é um *software* de roteamento *online* que ajuda a encontrar a melhor sequência de roteamento de entrega ao visitar mais de um endereço. A solução surgiu no final de 2009. A Baruch desenvolveu um sistema A Figura 1 mostra o *online* ideal para pequenas empresas para planejamento de entregas. MyRouteOnline é um serviço pago disponível em *Web*, Android e IOS.

MY ROUTE ONLINE

Características Preços Clientes Apoiar Login de membros

## OTIMIZAÇÃO DE ROTA

Planeje rotas com várias paradas  
Atribua motoristas ilimitados  
Navegue com Android/iOS  
Acompanhe suas entregas

Comece grátis agora!

### Otimize Suas Rotas Online

Planeje suas rotas para vários locais em apenas 3 etapas fáceis com nosso serviço de otimização de rotas

- 1 Importar**  
Carregue sua lista de endereços de uma planilha do Excel ou digite-a
- 2 Plano**  
Encontre a melhor rota e obtenha várias paradas de direção
- 3 Ir!**  
Exporte suas rotas para GPS ou celular e obtenha um mapa ao vivo com todos os destinos

Planeje minha rota!

Figura 1. MyRouteOnline.

### 3.2. RoadWarrior

A Figura 2 mostra o *software* RoadWarrior que é um sistema de roteirização adequado para a otimização de rotas com várias paradas, que busca reduzir o tempo de entrega e o consumo de combustível. É possível usar o aplicativo e na Figura 2 mostra o sistema *web* para criar, otimizar e despachar rotas para os motoristas. Os motoristas têm a possibilidade de usar o aplicativo, é um serviço pago – disponível para iOS e Android para receber e executar as tarefas.



Figura 2. RoadWarrior.

O diferencial do Sistema *Web* RouteU com relação aos trabalhos correlatos, é o preço, onde será possível oferecer uma aplicação similar com custos mais baixos.

#### 4. Metodologia

O presente estudo consiste em apresentar o desenvolvimento de um sistema *Web* para Roteirização de Entregas, que visa otimizar o planejamento e a eficiência das entregas de entregadores de encomendas. O problema identificado é a dificuldade de montar um roteiro de entregas sem uma ferramenta automatizada que possibilite a criação e otimização de rotas de entrega. Abaixo é apresentada a metodologia utilizada para o desenvolvimento do trabalho:

- **Pesquisa bibliográfica**

Como fontes de pesquisa para o referencial teórico serão utilizados livros, artigos, documentos e autores significativos, incluindo revisão bibliográfica para a apresentação e utilização das ferramentas para o desenvolvimento do Sistema *Web* Route-U.

- **Modelo Incremental**

Para o desenvolvimento do sistema, optou-se pelo modelo incremental, devido à sua flexibilidade em incorporar mudanças à medida que o projeto avança. Isso permitirá a entrega de funcionalidades de forma iterativa, possibilitando a validação precoce das soluções e a obtenção de *feedback* (O Modelo Incremental foi desenvolvido através da combinação entre os modelos linear e prototipação. O

desenvolvimento é dividido em etapas, denominadas “incrementos”, que produzem incrementalmente o sistema, até a sua versão final. Em cada incremento é realizado todo o ciclo do desenvolvimento do *software*, e planejamento dos testes do sistema já em funcionamento. Cada etapa produz um sistema totalmente funcional, apesar de ainda não cobrir todos os requisitos) (NEPOMUCENO 2012).

- **Utilização do Leafletjs e da API Leaflet Routing Machine**

o Leaflet e API Leaflet Routing Machine foram escolhida como componentes chaves para a integração de recursos de mapeamento e roteirização no sistema. Isso permitirá a visualização de mapas, cálculo de rotas otimizadas e informações em tempo real sobre o tráfego.

- **Testes com Usuários**

A fim de garantir a usabilidade e eficácia do sistema, foram realizados testes com usuários. O público-alvo incluiu motoristas de entregas e trabalhadores de logística. Os testes foram conduzidos em um ambiente controlado, onde os participantes foram convidados a interagir com o sistema para realizar tarefas de roteirização e planejamento de entregas.

- **Questionário com Perguntas Fechadas**

Após a conclusão dos testes, os participantes foram convidados a preencher um questionário com perguntas fechadas. O questionário foi estruturado para avaliar as seguintes áreas:

1. Facilidade de uso da interface do sistema.
2. Eficiência na criação e otimização de rotas.
3. Precisão das informações de roteirização.
4. Satisfação geral com o sistema.

- **Coleta e Análise de Dados**

As respostas dos questionários foram coletadas e analisadas quantitativamente. Utilizou-se uma escala de Likert (na APÊNDICE A é explicada o o que é a escala) para avaliar cada aspecto, variando de "Discordo Totalmente" a "Concordo Totalmente". Os resultados foram compilados e apresentados por meio de gráfico de barras e uma tabelas, destacando as principais conclusões.

- **Validação dos Resultados**

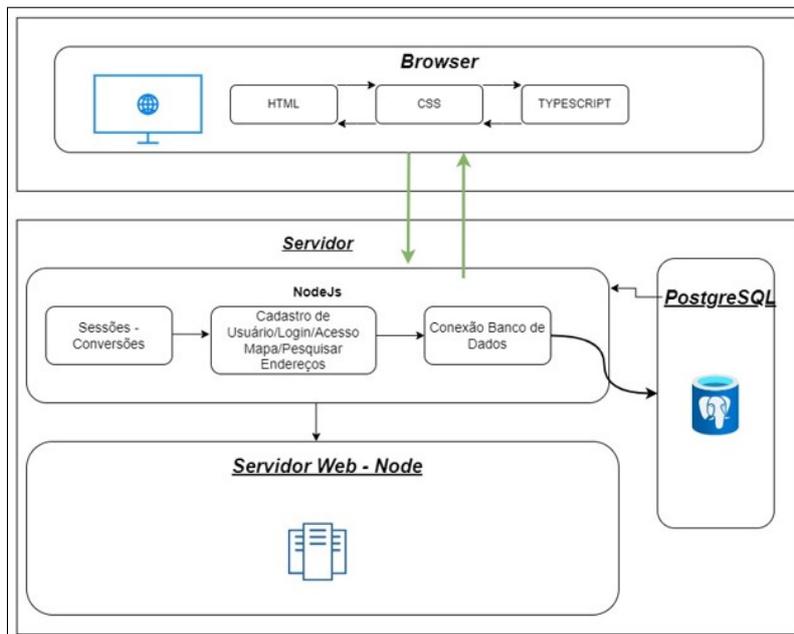
Os resultados obtidos foram validados por meio de comparações com os objetivos iniciais do sistema e as expectativas dos usuários. As conclusões dos testes e as respostas do questionário forneceram o *insights* para possíveis melhorias e ajustes no sistema.

## **5. Desenvolvimento do Sistema Web**

Foi utilizado o *Leaflet*, biblioteca de código aberto, para inserir na página Html o mapa, e para obter as localizações inseridas nas pesquisas dos endereços o *Leaflet Routing Machine*, API que utiliza o OSRM para importação flexível do OpenStreetMap. Foi aplicada responsividade ao *Site*. Implementou-se para autenticação o JWT para garantir segurança ao usuário. Nos ANEXOS A e B são mostrados partes dos código de exibição do mapa e código de roteirização.

### **5.1. Arquitetura do Sistema**

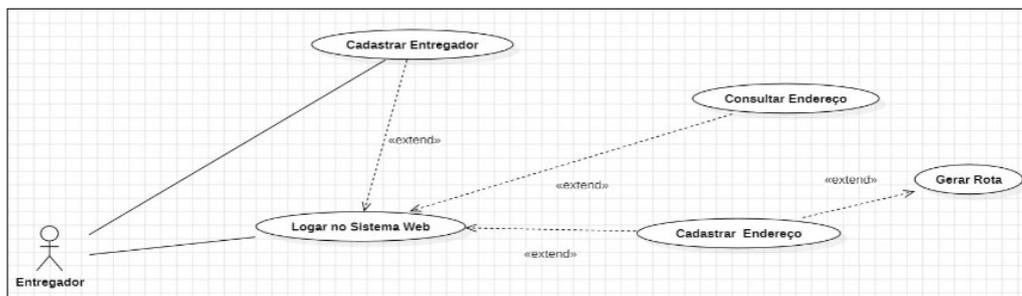
A Figura 3 apresenta a arquitetura do Sistema Web RouteU. A arquitetura do sistema consiste na sistematização ou formalização da interações entre as entidade e seu comportamento. No caso da arquitetura Sistema Web RouteU. A Figura 3, mostra que o usuário acessa *Browser* da aplicação e pode fazer o Cadastro de Usuário/*Login/Acesso* ao Mapa/Pesquisar endereços através da troca de informações entre Cliente <=> Servidor e Servidor <=> Banco de Dados.



**Figura 3. Diagrama de Arquitetura.**

## 5.2. Diagrama de Caso de Uso

A Figura 4 apresenta o Diagrama de Caso de Uso do Sistema *Web RouteU*. A Figura 4 mostra o comportamento do Sistema *Web RouteU*, que ajudou a capturar os requisitos do sistema, assim identificando as interações entre o sistema e seus usuários.



**Figura 4. Diagrama de caso de Uso.**

## 5.3. Diagrama Modelo de Banco de Dados

A Figura 5 apresenta o Diagrama Modelo de Banco de Dados do Sistema *Web RouteU*. Para modelagem do Banco de Dados foi utilizado o PostgreSQL, conforme Figura 5, foram definidos as entidades e seus relacionamentos, além dos tipos de dados que podem ser recebidos.

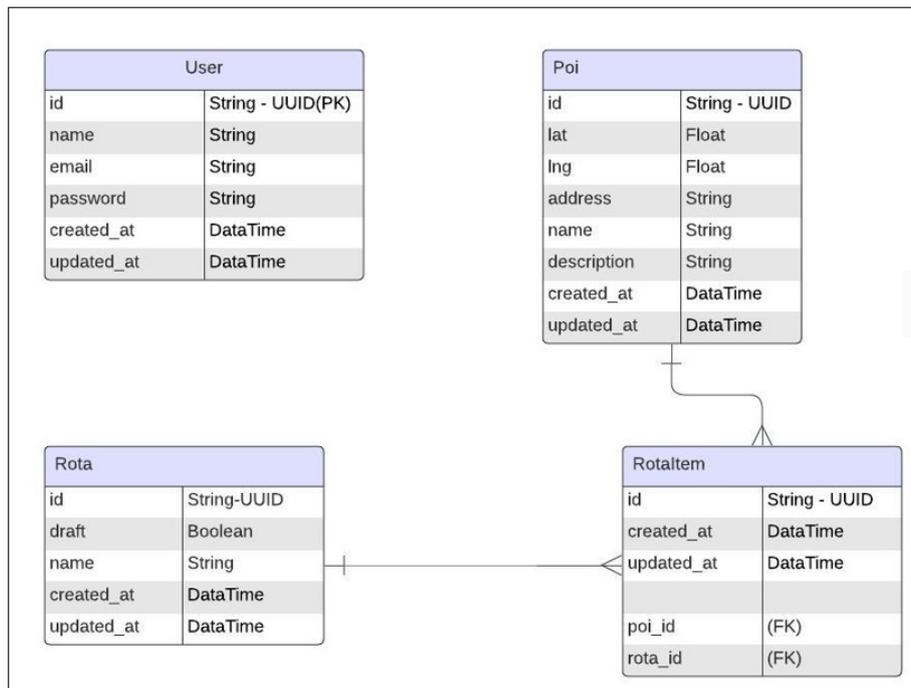


Figura 5. Diagrama Modelo de Banco de Dados.

#### 5.4. Segurança e Autenticidade

A Figura 6 mostra a representação da Autenticação com JWT. Para garantir a segurança e autenticidade utilizou-se JWT (*tokens JSON Web*):

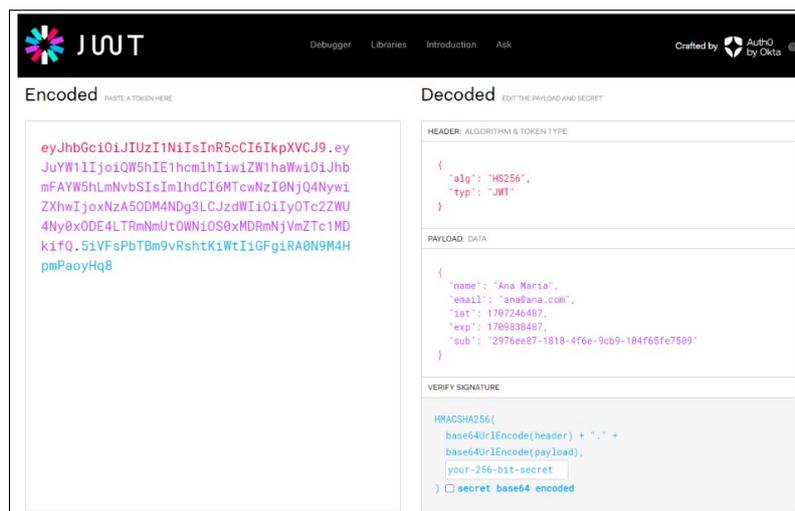


Figura 6. Autenticação com JWT.

O JWT - *JSON Web Tokens* (na APÊNDICE B é explicado o que é o JWT) é o que autoriza o acesso a aplicação, permitindo que o usuário acesse as rotas e serviços permitidos com esse *token*, o Sistema *Web RouteU* só permite acesso a tela de *Login* e

Cadastro sem o JWT, demais serviços é necessário validar o JWT através de senha.

A Figura 7 mostra a implementação no código para *login* com JWT para garantir segurança e autenticação definidos nos requisitos funcionais e não funcionais.

```
import {NextFunction, Request, Response} from 'express'
import { verify } from 'jsonwebtoken'

interface Payload{
  sub: string;
}

export function isAuthenticated(
  req: Request,
  res: Response,
  next: NextFunction
){
  //Receber Token (autenticação)
  const authToken = req.headers.authorization;

  if(!authToken){
    return res.status(401).end();
  }

  const [, token] = authToken.split(" ")

  try{
    //Validar esse token.
    const { sub } = verify(
      token,
      process.env.JWT_SECRET
    ) as Payload;

    req.user_id = sub

    return next();
  }catch(err){
    return res.status(401).end();
  }
}
```

Figura 7. Código de Autenticação com JWT no Sistema *Web RouteU*.

## 6. Utilização e Protótipos de Telas

A seguir serão apresentados a utilização e protótipos de telas.

### 6.1. Uso do Sistema RouteU

A Figura 8 apresenta a tela de login para acessar o mapa do Sistema RouteU. Para efetuar o *login* o usuário insere nome, e *password* e por fim clica no botão Acessar, assim tem acesso a tela do Mapa. Caso não possua conta pode clicar na Instrução abaixo do botão Acessar: Não possui conta? Cadastre-se, e é direcionado a tela de Cadastro.

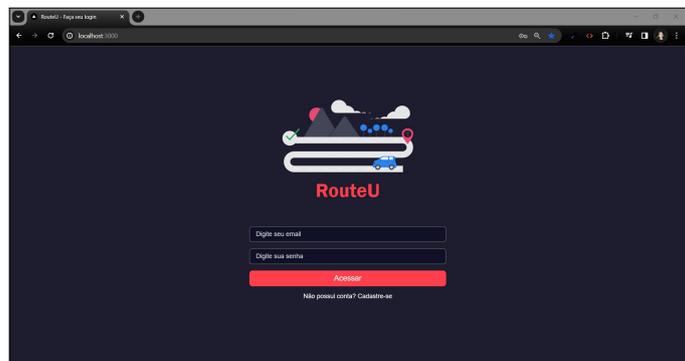


Figura 8. Tela de Login.

A Figura 9 apresenta a tela de cadastro para obter acesso ao Sistema RouteU. Para efetuar o cadastro o usuário insere nome, email, *password* e por fim clica no botão Cadastrar, após o cadastro, é direcionado para a tela de *Login*. Caso já possua conta pode clicar na Instrução abaixo do botão Cadastrar: Já possui uma conta? Faça Login, e é direcionado para tela de *Login*.

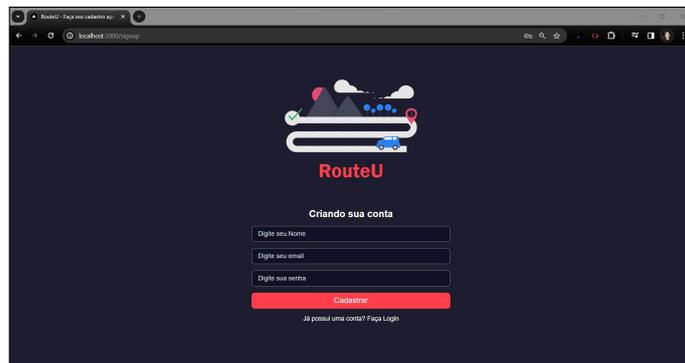
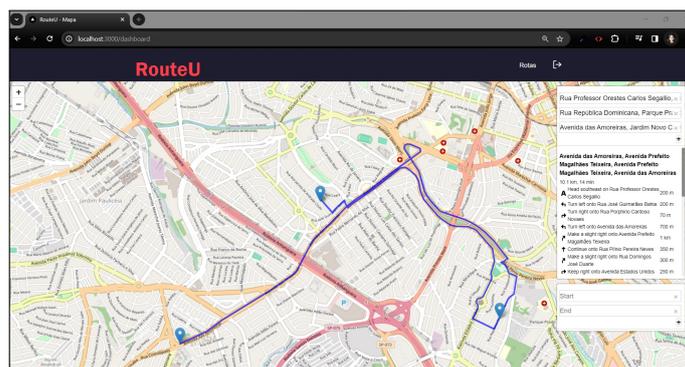


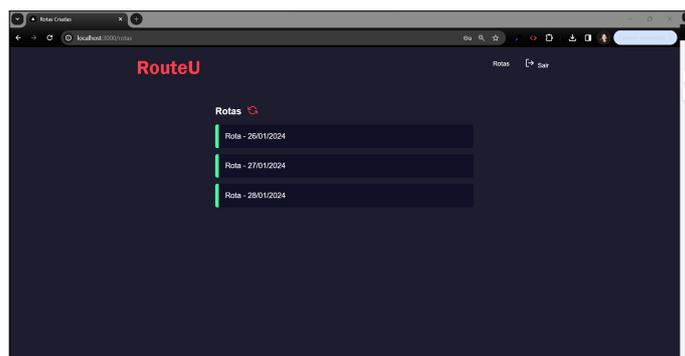
Figura 9. Tela de Cadastro.

A Figura 10 apresenta a tela com mapa e para pesquisa e adição de endereços de entregas. Após fazer o *login* o usuário acessa o Mapa e pode efetuar pesquisa de endereço e fazer a adição dos pontos no mapa. Na parte superior do lado direito existem duas opções uma para acessar as Rotas criadas (protótipo, funcionalidade a ser implementada nos trabalhos futuros) e opção para Sair, onde o usuário é desconectado do sistema.



**Figura 10. Tela Mapa.**

A Figura 11 apresenta o protótipo da tela onde pode-se consultar as rotas criadas. Após fazer o *login* o usuário pode escolher ver a lista de rotas criadas (protótipo, funcionalidade a ser implementada nos trabalhos futuros) e no topo da página no lado esquerdo tem o Logotipo RouteU, onde clicando nele é possível voltar ao mapa e ainda no lado direito tem as a opção para Sair, onde o usuário é desconectado do sistema.



**Figura 11. Tela Rotas.**

## 7. Resultados

Os requisitos funcionais propostos, cadastro de entregadores, login, gestão de endereços, reterização automatizada, utilização de algoritmo de otimização de rotas, visualização de rotas no mapa e desempenho (tempo gasto do início ao final da rota) foram plenamente

atendidos. Os requisitos não funcionais, usabilidade, desempenho, segurança, disponibilidade, integração com as API's e eficiência, conforme questionário respondido pelos usuários, foram considerados atendidos.

### 7.1. Teste de Usabilidade

Foi pedido para 3 usuários entregadores de encomendas para interagirem com o sistema e após responderem o questionário, que será mostrado em sequência.

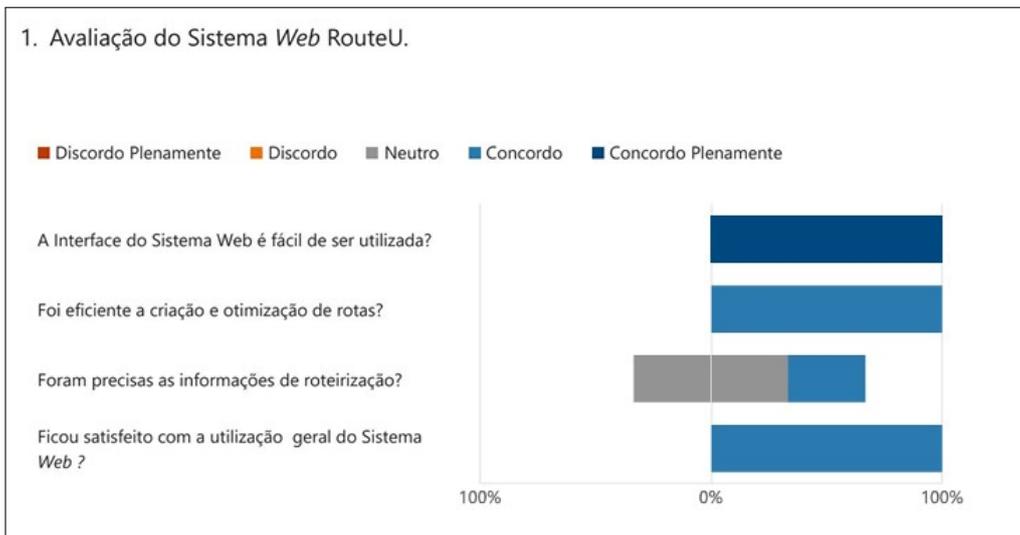
### 7.2. Resultado do Questionário

A Figura 12 apresenta o questionário aplicado aos usuários. Foram feitas 5 perguntas a respeito da usabilidade do sistema conforme 12:

1. Avaliação do Sistema <i>Web RouteU</i> .					
	Discordo Plenamente	Discordo	Neutro	Concordo	Concordo Plenamente
A Interface do Sistema Web é fácil de ser utilizada?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Foi eficiente a criação e otimização de rotas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Foram precisas as informações de roteirização?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ficou satisfeito com a utilização geral do Sistema Web ?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Figura 12. Questionário - Avaliação Sistema *Web RouteU*.**

A Figura 13 apresenta o gráfico com os resultados obtidos das respostas dos usuários. Em relação aos níveis de satisfação manifestados pelos usuários, que responderam o questionário, mais de 90 por cento da amostra se dizem satisfeitos.



**Figura 13. Gráfico - Avaliação Sistema Web RouteU.**

A Figura 14 apresenta a tabela com os resultados obtidos das respostas dos usuários. Segundo as respostas apresentadas na Figura 14, 100 por cento concordam plenamente que a interface do Sistema Web é fácil de ser utilizada, quanto a criação e otimização de rotas, 100 por cento concordam que é eficiente, 33,3 por cento concordaram com a precisão das informações de roteirização e 66,7 por cento responderam neutro. Quanto a satisfação geral do Sistema 100 por cento, responderam estarem satisfeitos.

Usuário	A Interface do Sistema Web é fácil de ser utilizada?	Foi eficiente a criação e otimização de rotas?	Foram precisas as informações de roteirização?	Ficou satisfeito com a utilização geral do Sistema Web ?
Usuário 1	Concordo Plenamente	Concordo	Concordo	Concordo
Usuário 2	Concordo Plenamente	Concordo	Neutro	Concordo
Usuário 3	Concordo Plenamente	Concordo	Neutro	Concordo

**Figura 14. Tabela - Resultado da avaliação do Sistema Web RouteU.**

## 8. Conclusão

Este trabalho teve como foco o desenvolvimento de um Sistema Web para entregadores roteirizarem entregas, pois é necessário melhorar a cada dia mais o desempenho das entregas por serem o fator mais importante para clientes, diante da decisão de efetuar uma compra ou não, e o desenvolvimento do Sistema Web RouteU mostrou-se necessário.

Foram utilizados os conhecimentos obtidos nas disciplinas Análise Orientada a Objetos, Banco de Dados, Engenharia de Software, Desenvolvimento de Sistemas Web entre outras, além de ser necessário aprender a linguagem TypeScript, React e API do Google Maps. Durante o desenvolvimento descobriu-se que a API do Google Maps não

era viável devido ao seu alto custo, na fase de testes após a implementação, sendo necessário fazer a troca de API por uma que não tivesse custos, adotou-se então as API's Leaflet. Os objetivos foram alcançados parcialmente devido a este contratempo, onde não foi possível em tempo hábil o aprendizado profundo de mais API's e aplicar para salvar as Rotas no banco de dados e ordenar os endereços otimizados. Apesar de não estar finalizado o Sistema, se torna viável já que se mostrou agradável aos usuários através das respostas desses, e possível a finalização por ter sido o modelo incremental adotado.

## **9. Trabalhos Futuros**

Além das API's Leaflet, outras opções como Mapbox, ou Bing Maps podem ser avaliadas quanto à sua adequação e custo-benefício para o projeto. Implementação de Funcionalidades Adicionais: Considerando os objetivos parcialmente alcançados, é importante planejar a implementação de funcionalidades adicionais que possam agregar valor ao sistema. Isso pode incluir a capacidade de salvar rotas no banco de dados, otimizar endereços, gerenciar preferências do usuário.

## Referências

- ABREU, L. **TypeScript-o JavaScript moderno para criação de aplicações**. São Paulo, SP: FCA, 2017.
- BESSA, A. **Node.JS: o que é, como funciona esse ambiente de execução JavaScript e um Guia para iniciar**. 2024. Disponível em: <<https://www.alura.com.br/artigos/node-js>>. Acesso em 04 mar 2024.
- CANALTECH. **O que é API?** 2024. Disponível em: <<https://canaltech.com.br/software/o-que-e-api/>>. Acesso em 21 jan 2024.
- DEITEL, P. J.; DEITEL, H. M. **Ajax, rich internet applications e desenvolvimento web para programadores**. [S.l.]: Pearson Prentice Hall, 2008.
- FERREIRA, E.; EIS, D. **HTML5: Curso W3C Escritório Brasil**. São Paulo, 2011.
- GROUP, T. P. G. D. **What is PostgreSQL?** 2023. Disponível em: <<https://www.postgresql.org/>>. Acesso em 20 dez 2023.
- HANASHIRO, A. **VS Code - O que é e por que você deve usar?** 2023. Disponível em: <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>>. Acesso em 10 out 2023.
- INC, V. **What is Next.js?** 2024. Disponível em: <<https://nextjs.org/docs>>. Acesso em 04 mar 2024.
- LARENTIS, F. **Comportamento Do Consumidor**. Iesde Brasil Sa, 2012. ISBN 9788538707134. Disponível em: <<https://books.google.com.br/books?id=sPImTMedmFIC>>.
- LEAFLET. **Leaflet**. 2024. Disponível em: <<https://leafletjs.com/>>. Acesso em 27 fev 2024.
- LIEDMAN, P. **Leaflet Routing Machine**. 2024. Disponível em: <<https://www.liedman.net/leaflet-routing-machine/>>. Acesso em 27 fev 2024.
- META OPEN SOURCE. **Construindo a UI**. 2023. Disponível em: <<https://pt-br.react.dev/learn/describing-the-ui>>. Acesso em 23 out 2023.
- NEPOMUCENO, D. **Engenharia de Software**. 2012. Disponível em: <<http://engenhariadesoftwareuesb.blogspot.com/2012/12/blog-post.html>>. Acesso em 15 abr 2023.
- PACHECO, M. **Desenvolvimento WEB: HTML, CSS E JAVASCRIPT para iniciantes**. [S.l.: s.n.], 2023. E-book Kindle.
- SCHMITT, P. R. M. **Aplicação web utilizando api google maps**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2013.
- SEGUINS, N. **<https://www.alura.com.br/artigos/o-que-e-json-web-tokens>**. 2024. Disponível em: <<https://project-osrm.org/>>. Acesso em 05 jan 2024.
- SOMMERVILLE, I. **Engenharia de software/lan Sommerville; tradução Luiz Claudio Queiroz; revisão técnica Fábio Levy Siqueira.-**. [S.l.]: São Paulo: Pearson Education do Brasil, 2018.
- SURVEYMONKEY. **O que é uma escala Likert?** 2023. Disponível em: <<https://pt.surveymonkey.com/mp/likert-scale/>>. Acesso em 23 out 2023.
- WIKIPEDIA. **OpenStreetMap**. 2024. Disponível em: <<https://pt.wikipedia.org/wiki/OpenStreetMap>>. Acesso em 27 fev 2024.

## **APÊNDICE A - Escala de Likert**

A Escala Likert, desenvolvida pelo psicólogo Rensis Likert em 1932, é uma ferramenta amplamente utilizada para medir opiniões e sentimentos em pesquisas de mercado e ciências sociais. Com ela, os pesquisadores conseguem entender melhor como as pessoas se sentem em relação a produtos, serviços, marcas e mercados. Essa escala permite que os participantes expressem seus pontos de vista em diferentes níveis, ajudando a capturar nuances e fornecendo insights valiosos para tomadas de decisões estratégicas. Sempre usada em perguntas fechadas ou afirmações, com respostas como por exemplo de Concordância: Concordo totalmente / Concordo / Neutro / Não concordo / Discordo totalmente (SURVEYMONKEY 2023).

## **APÊNDICE B - Token JWT**

Um JWT (*JSON Web Token*) é um padrão para autenticação e troca de informações, definido pela RFC7519. Ele permite armazenar de forma segura e compacta objetos JSON. Este *token* é representado por um código Base64 e pode ser assinado usando um segredo ou um par de chaves privadas/públicas. Ao utilizar um JWT, é possível garantir a integridade das informações contidas nele. Os *tokens* assinados certificam que a parte que possui a chave privada é quem de fato assinou, o que proporciona uma camada de segurança adicional. Isso contrasta com os *tokens* criptografados, que apenas ocultam as informações, sem garantir sua integridade (SEGUINS 2024).

## APÊNDICE C - Questionário

14/03/2024, 15:42

Desenvolvimento do Sistema Web Route-u - 2023/12/9 17:46:33

### Desenvolvimento do Sistema Web Route-u - 2023/12/9 17:46:33

#### 1. Avaliação do Sistema Web RouteU.

	Discordo Plenamente	Discordo	Neutro	Concordo	Concordo Plenamente
A Interface do Sistema Web é fácil de ser utilizada?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Foi eficiente a criação e otimização de rotas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Foram precisas as informações de roteirização?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ficou satisfeito com a utilização geral do Sistema Web ?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Este conteúdo não é criado nem endossado pela Microsoft. Os dados que você enviar serão enviados ao proprietário do formulário.

 Microsoft Forms

## ANEXO A - Código de exibição do mapa

```
import { MapContainer, TileLayer } from 'react-leaflet'
import { ReactDOM } from 'react';
import 'leaflet/dist/leaflet.css';
import "leaflet-control-geocoder/dist/Control.Geocoder.css"
import "leaflet-control-geocoder/dist/Control.Geocoder";
import L from "leaflet";

import LeafletGeocoder from "../LeafletGeocoder";
import LeafletRoutingMachine from "../LeafletRoutingMachine";

function Map() {
  return (
    <div id="map">
      <MapContainer
        center={[-22.9097579908644, -47.06117440162242]}
        zoom={16}
        style={{ height: '100vh', width:'100%' }}
      >
        { /*<LeafletGeocoder/> */ }
      <TileLayer
        attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
        url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
      />
      <LeafletRoutingMachine />
    </MapContainer>
  </div>
  );
}

export default Map;
```

## ANEXO B - Código de adição do Routing Machine ao mapa

```
import "leaflet-routing-machine/dist/leaflet-routing-machine.css";
import L from "leaflet";
import "leaflet-routing-machine/dist/leaflet-routing-machine.js";

import { useMap } from "react-leaflet";

const LeafletRoutingMachine = () => {
  const map = useMap();

  // Add o routing machine ao map

  useEffect(() => {
    L.Routing.control({
      waypoints: [
        L.latLng(-22.909728343259832, -47.06111002968544),
        L.latLng(-22.90869213916458, -47.05558346593604),
      ],

      lineOptions: {
        styles: [
          {
            color: "blue",
            weight: 4,
            opacity: 0.7
          },
        ],
        extendToWaypoints: false,
        missingRouteTolerance: 0
      },
      routeWhileDragging: false,
      fitSelectedRoutes: true,

      geocoder: L.Control.Geocoder.nominatim(),

    }).addTo(map);
  }, [map]);

  return null
}

export default LeafletRoutingMachine;
```

# Documento Digitalizado Restrito

## Artigo de TCC da aluna Valéria Paulo

**Assunto:** Artigo de TCC da aluna Valéria Paulo  
**Assinado por:** Daiane Tomazeti  
**Tipo do Documento:** Comprovante  
**Situação:** Finalizado  
**Nível de Acesso:** Restrito  
**Hipótese Legal:** Informação Pessoal - dados pessoais e dados pessoais sensíveis (Art. 31 da Lei nº 12.527/2011)  
**Tipo do Conferência:** Cópia Simples

Documento assinado eletronicamente por:

- Daiane Mastrangelo Tomazeti, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 18/03/2024 10:01:16.

Este documento foi armazenado no SUAP em 18/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

**Código Verificador:** 1614848

**Código de Autenticação:** 7cf2dc2833

