

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

**Daniel Cortez Bacha**

**Desenvolvimento de Sistema de Conectividade  
Wi-Fi para Veículos Elétricos e Sistema Web de  
Gerenciamento de Frota**

Florianópolis  
2017



**Daniel Cortez Bacha**

**Desenvolvimento de Sistema de Conectividade Wi-Fi  
para Veículos Elétricos e Sistema Web de  
Gerenciamento de Frota**

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação na disciplina **DAS 5511: Projeto de Fim de Curso** do curso de Graduação em Engenharia de Controle e Automação.

Orientador(a): Prof. Carlos Barros Montez

Florianópolis  
2017



**Daniel Cortez Bacha**

# **Desenvolvimento de Sistema de Conectividade Wi-Fi para Veículos Elétricos e Sistema Web de Gerenciamento de Frota**

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, 2 de agosto de 2017

## **Banca Examinadora:**

Paulo Bosquiero Zanetti  
Orientador na Empresa  
Mobilis Veículos Elétricos

Prof. Carlos Barros Montez  
Orientador no Curso  
Universidade Federal de Santa Catarina

Prof. Fernando Gonçalves  
Avaliador  
Universidade Federal de Santa Catarina

Manoel Guidi Alvares  
Debatedor  
Universidade Federal de Santa Catarina

Daniel Manzoni Seerig  
Debatedor  
Universidade Federal de Santa Catarina



## **AGRADECIMENTOS**

Gostaria de especialmente agradecer a minha família, ao meu pai Charbel, minha mãe Stella e meu irmão Tiago, por sempre apoiarem as minhas decisões e por sempre estarem presentes em todos os momentos que precisei deles, apesar de toda a distância que nos separa.

Muito obrigado por tudo pelo que já fizeram e continuam a fazer por mim, pelos conselhos, pelas broncas e pelas risadas. Vocês são a base de tudo que sou hoje.





## RESUMO

A Mobilis é uma empresa que produz veículos elétricos de vizinhança, destinados a trabalhos em ambientes fechados e sem permissão para trafegar em vias públicas. Com sua vontade de criar um diferencial para sua marca e garantir um bom acompanhamento dos seus produtos, foi desenvolvido um sistema de conectividade que envia dados do veículo a um banco de dados na internet via uma rede Wi-Fi. O sistema permite que a empresa tenha acesso aos dados dos veículos, sendo possível realizar o acompanhamento de suas performances. Para oferecer uma solução aos problemas de grandes gerentes de frota, foi desenvolvido um sistema web de gerenciamento de frota que permite a visualização dos dados enviados pelo carro. Esses dados permitem estabelecer alguns indicadores de qualidade e desempenho de cada automóvel que o cliente possui.

**Palavras-chave:** Veículos Elétricos. Conectividade. Wi-Fi. Gerenciamento de Frota. Indicadores de Desempenho.



## ABSTRACT

Mobilis is a company that produces neighborhood electric vehicles, which are destined to perform indoor labor and don't have street clearance. Eager to create a differential for your brand and assure to be able to keep track of your products, it was developed one connectivity system that sends data from the vehicle to a data base in the web via Wi-Fi. The system allows the company access the vehicle's data and evaluate its performance. To offer a solution for the big fleet managers' problems, it was developed one fleet management web system that allows the visualisation of the data sent from the vehicle presented as quality and performance indicators of all vehicles that the customer has.

**Key-words:** Electric Vehicle. Connectivity. Wi-Fi. Fleet Management. Performance Indicators.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Sequência de telas do App para primeira conexão .....	34
Figura 2 - Esquemática do banco de dados do sistema .....	39
Figura 3 - Middleware .....	40
Figura 4 - Diagrama de Sequência da função de login.....	45
Figura 5 - Diagrama de Sequência de navegação .....	46
Figura 6 - Diagrama de Sequência de login e envio de dados pelo veículo .....	46
Figura 7 - Página de Login .....	49
Figura 8 - Página Painel de Controle.....	50
Figura 9 - Página Oficina .....	51
Figura 10 - Página Log da Frota.....	52
Figura 11 - Página Log de Usuários .....	53
Figura 12 - Exemplo Zoom Gráficos .....	55
Figura 13 - Exemplo Seleção e Valor .....	56
Figura 14 - App Android .....	61



## LISTA DE TABELAS

Tabela 1 – Lista de Eventos.....	31
Tabela 2 - Estrutura do Banco de Dados .....	38
Tabela 3 - Rotas do servidor .....	43
Tabela 4 - Rotas com envio de e-mail.....	44





## LISTA DE ABREVIATURAS E SIGLAS

UFSC – Universidade Federal de Santa Catarina

INAITEC – Instituto de Apoio à Inovação

RPi – *Raspberry Pi*

MCO – Módulo de Conectividade

BT – *Bluetooth*

BD – Banco de Dados

JSON – *JavaScript Object Notation*

JWT – *JSON Web Token*

API – *Interface programming interface*

REST – *Representational State Transfer*

CAN – *Controller Area Network*

HTTP – *Hypertext Transfer Protocol*

HTTPS – *Hypertext Transfer Protocol Secure*

SSL – *Secure Sockets Layer*

SMTP – *Simple Mail Transfer Protocol*



## SUMÁRIO

1. Introdução.....	20
1.1 Justificativa.....	20
1.2 Objetivo Geral .....	21
1.3 Objetivos Específicos .....	21
1.4 Estrutura do Documento .....	21
2. Revisão Bibliográfica .....	22
2.1 Protocolos de Comunicação .....	22
2.2 Circuitos Integrados .....	23
2.3 Técnicas e Formatações .....	23
2.4 Frameworks, Bibliotecas e Ferramentas.....	24
3. Desenvolvimento .....	26
3.1 A Empresa .....	26
3.2 O Cliente .....	26
3.2 O Projeto .....	27
4. Sistema de Conectividade .....	30
4.1 Sistema Embarcado .....	30
4.2 Primeira Conexão com a Internet.....	33
5. Sistema Web .....	36
5.1 Back-End.....	37
5.1.1 Banco de Dados.....	37
5.1.2 Servidor.....	39
5.1.3 Rotas.....	42
5.2 Front-End .....	47
5.2.1 Páginas .....	48
5.2.2 Indicadores.....	56

5.2.3 Responsividade .....	60
5.2.4 App Android .....	60
6. Resultados .....	62
7. Conclusão e Perspectivas Futuras .....	64
REFERÊNCIAS .....	66

## 1. INTRODUÇÃO

### 1.1 Justificativa

Com o objetivo de projetar e desenvolver o seu próprio utilitário elétrico desde o rascunho, a Mobilis Veículos Elétricos está criando o seu primeiro protótipo de veículo elétrico de vizinhança com intenção lançamento do produto para venda em agosto de 2017.

Como há intenção de manter um acompanhamento intenso de seus produtos, foi idealizado um sistema de conectividade para um veículo que garante acesso às métricas do automóvel de maneira remota, mesmo depois da venda. O sistema tem o objetivo de permitir a análise de informações de situações reais para avaliar o desempenho do veículo em diferentes circunstâncias e identificar erros que não aparecem nos testes controlados. Ademais, como tudo isso é feito remotamente, o sistema também garante que o acompanhamento pós venda seja ágil e fácil.

Ter essa facilidade em acessar as métricas dos seus veículos é essencial no momento em que a empresa se encontra, porque assim ela é capaz de analisar as métricas do carro nas mais diferentes situações, tendo mais informações para conseguir desenvolver um produto melhor. Dessa forma, também será possível identificar falhas prematuramente, conseguindo agir de maneira rápida, garantindo a manutenção da qualidade dos seus produtos.

Com este sistema também existe a possibilidade de que o acompanhamento pós venda e atendimento ao cliente sejam feitos de maneira rápida e eficiente. Como os dados são acessados via internet, um primeiro diagnóstico pode ser feito em qualquer instante, mesmo que o carro esteja em outra cidade. Isso implica em um possível diagnóstico sem que o carro seja levado a uma oficina autorizada, agilizando o atendimento e agradando o consumidor.

A partir deste primeiro sistema de conectividade, a empresa também enxergou oportunidade de criar um sistema web que fosse capaz de apresentar as métricas do carro como indicadores de desempenho para um usuário que é dono de um carro Mobilis. Desta forma o acompanhamento do desempenho dos veículos pode ser feita de maneira fácil e rápida. O sistema também faz com que o controle e planejamento de atividades para uma frota de veículos seja feita de maneira muito

mais eficiente, sendo uma excelente ferramenta para clientes que possuem grandes quantidades de utilitários elétricos.

## **1.2 Objetivo Geral**

Os objetivos gerais deste projeto consistem em promover a integração do veículo com um banco de dados na internet. Dessa maneira, seus dados serão armazenados na nuvem e, por consequência, será possível realizar as suas análises remotamente.

Além disso, temos como objetivo geral criar um sistema web que armazene e trate os dados desejados e que seja possível plotar métricas desejadas e informações relevantes. Adicionalmente, o sistema tem como objetivo apresentar os dados através de indicadores de qualidade e desempenho que sejam relevantes para o usuário no ponto de vista gerencial de maneira intuitiva e direta.

## **1.3 Objetivos Específicos**

Os objetivos específicos são: (i) desenvolver um sistema que garanta a conectividade do sistema embarcado com a internet; (ii) projetar um Banco de Dados que suporte e organize os dados; (iii) projetar servidor que permita a aplicação das regras de negócio da empresa; (iv) desenvolver interface gráfica de apresentação; (v) criar funções que apresente as métricas do veículo para um funcionário Mobilis; e (vi) desenvolver funções onde os dados são transformados em indicadores e apresentados para o usuário final.

## **1.4 Estrutura do Documento**

Este documento está organizado da seguinte forma. No Capítulo 2 é feita uma breve revisão bibliográfica das ferramentas e conceitos utilizados no desenvolvimento do projeto. No Capítulo 3 é apresentada a empresa e também o ambiente para qual o projeto se justifica. Depois, nos Capítulos 4 e 5 são apresentados os desenvolvimentos do sistema de conectividade e o sistema web. Por fim, no Capítulo 6 são abordados os resultados obtidos e, em seguida, as conclusões e perspectivas futuras.

## 2. REVISÃO BIBLIOGRÁFICA

Neste capítulo é feita uma revisão dos conceitos e tecnologias utilizadas durante o desenvolvimento do projeto. Para facilitar o entendimento das tecnologias, elas foram separadas em diferentes partes, Protocolos de Comunicação, Circuitos Integrados, Técnicas e Formatação, e Frameworks, Bibliotecas e Ferramentas.

### 2.1 Protocolos de Comunicação

Wi-Fi é uma marca registrada que é utilizada por produtos certificados que pertencem à classe de dispositivos de rede local sem fio baseados no padrão IEEE 802.11 (LANSFORD; STEPHENS; NEVO, 2001). O termo Wi-Fi é normalmente utilizado como sinônimo para este padrão. Para ter acesso a internet através de uma rede Wi-Fi, é necessário estar dentro do raio de ação ou área de abrangência de um ponto de acesso.

Bluetooth (também conhecido como IEEE 802.15.1) é uma especificação de rede sem fio de âmbito pessoal, a qual permite conectar e trocar informações entre dispositivos como celulares, notebooks, impressoras entre outro (LANSFORD; STEPHENS; NEVO, 2001). Esse padrão apresenta um alcance menor do que uma rede Wi-Fi, porém é projetado para baixo consumo de energia.

As redes CAN (*Controller Area Network*) são protocolos baseados em mensagens desenvolvidas originalmente para multiplexação da fiação elétrica de um automóvel (VAN OSCH; SMOLKA, 2001). Este protocolo permite que micro controladores e dispositivos se comuniquem entre si sem um computador de *host*.

A aplicação web se comunica usando o *Hypertext Transfer Protocol* (HTTP), que é um protocolo da camada de aplicação utilizados para sistemas de informação. Este protocolo é a fundação da comunicação de dados na *World Wide Web* (FIELDING; RESCHKE, 2014). O HTTP funciona como um protocolo de requisição-resposta no modelo cliente-servidor. O cliente submete uma mensagem de requisição HTTP para o servidor. O servidor fornece os recursos e retorna uma mensagem resposta para o cliente.

O protocolo HTTP define uma série de métodos de requisição responsáveis por indicar a ação a ser executada na representação de um determinado recurso. Cada um deles implementa uma diferente função.

O HTTPS, *Hypertext Transfer Protocol Secure*, é a implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL (*Secure Socket Layer*) (FREIER; KARLTON; KOCHER, 2011). Esta camada permite que os dados sejam transmitidos por meio de uma conexão criptografada e que se verifique a autenticidade do servidor e do cliente por meio de certificados digitais. A existência desse certificado indica o uso do protocolo HTTPS, além de apontar que a comunicação entre navegador e servidor será de forma segura.

SMTP ou *Simple Mail Transfer Protocol* é o protocolo padrão para envio de e-mails através da internet (HOFFMAN, 2002). Ele foi padronizado pela RFC 821.

## 2.2 Circuitos Integrados

O *Raspberry Pi* é um computador do tamanho de um cartão de crédito que permite realizar muitas operações diferentes e permite também a integração com outros sistemas.

MCP25625 é um circuito integrado de controlador CAN que nos permite em fazer a interface entre barramento CAN e o protocolo SPI, que é um protocolo padrão de comunicação entre microcontroladores (LI, 2017). Este circuito integrado é essencial para o projeto pois ele traduz o barramento CAN e envia os dados para o *Raspberry Pi* via SPI.

## 2.3 Técnicas e Formatações

Técnicas *AJAX (Asynchronous Javascript and Xml)* foram utilizadas com o intuito de carregar informações de forma assíncrona para tornar as páginas mais interativas com o usuário.

A formatação *JSON (Javascript Object Notation)* foi utilizada para troca de dados e integração com o de banco de dados SQL no servidor.

JSON Web Token (JWT) é um padrão baseado em mensagens JSON para criar tokens de acesso que possuem algumas informações adicionais (JONES; BRADLEY, 2015). Os tokens foram projetados para serem compactos e de uso seguro em URL de navegadores. Eles são usados tipicamente para passar a identidade do usuário autenticado para o sistema e podem ser criptografados.

Um JWT normalmente tem três partes: o cabeçalho, o conteúdo e a assinatura. O cabeçalho identifica qual o tipo de algoritmo de criptografia é utilizado



para gerar a assinatura e pode conter algumas outras informações. A parte do conteúdo é responsável por carregar as informações sobre o usuário que gostaríamos de transmitir às outras partes do sistema. A última parte da mensagem é a assinatura, que utiliza o algoritmo escolhido junto a uma palavra chave escolhido pelo usuário. Para montar a mensagem completa, cada parte é criptografada utilizando o algoritmo de base64 e são colocadas em sequência separadas apenas por um ponto.

Para ser utilizado de maneira correta, uma autorização no cabeçalho da requisição HTTP deve ser adicionada utilizando a função “Bearer” contendo o token.

REST ou *Representational State Transfer* é uma abstração de arquitetura de aplicações web que permite flexibilidade durante o desenvolvimento e dá ao sistema melhor performance e confiabilidade (MASSE, 2011). Para uma aplicação ser considerada REST precisa obedecer algumas condições, como: separar as funções e responsabilidades entre *back* e *front-end*; o servidor não deve armazenar estado do usuário; *code-on-demand*, onde toda a lógica do sistema é executada no lado do cliente; realizar a comunicação entre cliente e servidor através de uma interface padrão HTTP.

## 2.4 Frameworks, Bibliotecas e Ferramentas

Foi utilizado o framework Front-End Materialize para melhor apresentação visual utilizando o padrão Google Material Design. Este framework também foi escolhido pela sua facilidade em desenvolver projetos web responsivos.

Para construir a lógica do servidor foi utilizado uma framework que facilita o roteamento de requisições http chamada SlimFramework. O Slim possibilita desenvolver aplicações de servidor com uma quantidade pequena de código, o que deixa mais ágil o desenvolvimento de pequenas aplicações.

A biblioteca de funções *Javascript jQuery* também foi escolhida, pois é uma biblioteca que interage com o HTML da página, facilitando a interpretação dos scripts no navegador do cliente. Sua utilização também deixa simples a seleção de elementos, criação de animações, a criação de plug-ins e o desenvolvimento de aplicações *AJAX*.



### 3. DESENVOLVIMENTO

No sentido de mostrar o contexto em que o projeto foi desenvolvido, este capítulo apresenta brevemente empresa e seu cliente.

#### 3.1 A Empresa

A Mobilis Veículos Elétricos é um startup de utilitários elétricos que oferece soluções eficientes e criativas de mobilidade sustentável. Baseado na plataforma elétrica, a empresa produz veículos elétricos de vizinhança (*Neighborhood Electric Vehicle*). Estes carros se apresentam em pequeno porte, não homologados para uso em vias públicas e destinados aos mais diversos usos como, por exemplo, na indústria, canteiros de obra, aeroportos, shoppings, parques, entre outros. Futuramente, existe a ambição de lançar um veículo elétrico homologado para vias públicas.

A referida empresa foi fundada em 2013 por ex-estudantes da Universidade Federal de Santa Catarina (UFSC) graduados em engenharias elétrica e mecânica. Encontra-se sediada no Instituto de Apoio à Inovação, Incubação e Tecnologia (INAITEC) na Cidade Criativa Pedra Branca em Palhoça, onde se localiza a micro fábrica e possui um escritório de engenharia em Florianópolis. A instituição possui a meta de apresentar o seu primeiro carro para vendas no mês de agosto de 2017. No momento, a empresa encontra-se na etapa de finalização do seu primeiro protótipo, buscando a introduzir no mercado um produto com diferentes sistemas e funcionalidades.

Como é o primeiro veículo produzido pela empresa, existe uma vontade de acompanhar os dados dos veículos mesmo depois da venda, para que seja possível fazer a análise do veículo nas mais diversas situações possíveis, podendo identificar situações que ocasionem falhas ou até mesmo problemas de produção. Desta forma, a empresa é capaz de promover uma melhoria contínua do seu produto, visando a entregar sempre um produto com a mais alta qualidade ao mercado.

#### 3.2 O Cliente

O cliente alvo da Mobilis pode não ser usual para quem não conhece este setor de mercado. Além disso, as aplicações deste produto podem ser as mais variadas possíveis. Portanto, é necessário tentar entender o contexto deste cliente.

O proprietário de um veículo elétrico de vizinhança pode ser o dono de um conjunto habitacional (condomínio) que deseja um carro para auxiliar a locomoção dos moradores em sua área interna; uma indústria que necessite de alguns veículos para locomoção de funcionários e para transporte de carga; um aeroporto que precise de veículos para meio de locomoção nas áreas internas e externas, para transporte de carga, para a brigada de incêndio, assistência médica, dentre outros. Portanto, o cliente pode vir a adquirir apenas um veículo ou dezenas deles, tendo a possibilidade de comprar uma frota inteira de carros.

O cliente que precisa de uma frota de veículos muito provavelmente não será o operador ou motorista do veículo, mas sim o gerente responsável por aqueles automóveis. Para ele, é necessário que os carros estejam em boas condições para que seus funcionários (os motoristas ou operadores) possam ter um dia de trabalho sem imprevistos técnicos. Portanto, é essencial o acompanhamento de índices de desempenho dos seus veículos como identificação precoce de falhas e alertas.

### **3.2 O Projeto**

Para atender aos objetivos da empresa foi desenvolvido um sistema de conectividade que ficará embarcado nos veículos elétricos. O sistema de conectividade é capaz de ler os dados enviados pelo seu barramento CAN, criar diferentes logs de informações e os enviar a um banco de dados na internet a partir de uma conexão Wi-Fi.

Estas informações poderão ser acessadas por um funcionário da empresa a partir de um sistema web capaz de plotar as métricas do veículo e apresentar logs de eventos, para que seja possível realizar a análise do desempenho de seu produto de maneira remota.

O sistema é capaz de plotar gráficos de informações contidas em logs do barramento CAN do veículo e logs de eventos para que possa ser posteriormente analisado por um engenheiro da Mobilis.

Para oferecer uma ferramenta indispensável para um gerente de frota, foi desenvolvido um sistema web que pode ser acessado pela internet e identifica o cliente através de login. Através de suas credenciais o sistema identifica o cliente e apresenta as informações de seus veículos em forma de indicadores relevantes, facilitando o acompanhamento do desempenho de sua frota. Com os indicadores

também fica mais fácil de identificar falhas e planejar manutenções preventivas para os veículos.

O usuário também tem acesso aos logs de eventos dos carros, que também podem ser apresentados separadamente por motorista responsável pelo evento. Esta separação permite a identificação de maus operadores de veículos, dando ao gerente o poder de trabalhar junto a este operador para evitar novas falhas.

Requisições de diagnóstico também são possíveis. O usuário tem acesso a uma seção do sistema onde pode identificar o carro com problema e especificar o que está acontecendo e fazer o pedido de diagnóstico. Assim, um e-mail contendo o pedido e as informações é enviado para um engenheiro da Mobilis que posteriormente fará a análise daquele problema específico e voltar a entrar em contato com o cliente.

O sistema também possui as mais diversas operações para garantir usabilidade como páginas de contato, perguntas frequentes, cadastro de novos usuários, cadastro de novos veículos e edição dessas informações.

Todos os veículos serão vendidos com o sistema de conectividade embarcado, precisando apenas de uma configuração inicial para conectar o carro na rede. Para facilitar, foi desenvolvido também um aplicativo para *smartphone* que cria uma conexão Bluetooth com o sistema embarcado do veículo e consegue passar os parâmetros de nome e senha da rede Wi-Fi que é desejada estabelecer a conexão. Com isso, o veículo tem acesso as informações necessárias para fazer a primeira conexão com a rede do cliente.



## 4. SISTEMA DE CONECTIVIDADE

Como foi mencionado brevemente anteriormente, a empresa se encontra na etapa de término do seu primeiro protótipo de veículo elétrico, com previsão de lançamento no mês de agosto deste ano.

Ele também é o primeiro veículo produzido pela empresa, portanto os testes e análises ainda são intensos. Esses testes são feitos contendo uma placa embarcada no carro que lê e armazena os dados do barramento CAN em sua memória para que ele possa ser analisado posteriormente por um membro da empresa. Com este log é possível analisar informações de sensores do veículo, dados como corrente e tensão no motor, corrente e tensão no inversor, velocidade, torque, entre outros.

O projeto do sistema de conectividade consiste na adaptação da placa atual para conseguir que os dados armazenados em sua memória sejam enviados via Wi-Fi para um servidor online. Deste modo é possível realizar testes de maneira mais rápida e de colher as informações de todos os seus veículos, ajudando no desenvolvimento contínuo de seu produto.

### 4.1 Sistema Embarcado

O Módulo de Conectividade (MCO) é o nome dado pela empresa ao sistema embarcado do veículo que consiste em uma placa contendo um MCP25625 acoplada a um *Raspberry Pi* (RPI). A placa também possui um hub USB integrado para permitir a conexão de outros adaptadores ao RPI.

A MCO mantém uma comunicação com o barramento CAN do veículo utilizando o seu módulo MCP25625 que recebe as informações do barramento e as converte para se comunicar com o RPI. Com o MCP também conseguimos escrever mensagens no barramento a partir do RPI.

O RPI possui scripts que disparam quando a MCO é energizada, os quais garantem a leitura do barramento e a interpretação dos seus dados, escrevendo no barramento assim que necessário. A interpretação dos dados é feita de forma simples, pois as mensagens transmitidas através do CAN são conhecidas pela empresa.

A placa tem a capacidade de criar três tipos de logs a partir das informações lidas no barramento. Um deles consiste em criar um arquivo no formato *.csv* contendo as informações do barramento CAN na íntegra, separados por colunas.

Também é capaz de criar arquivos .csv contendo o log de eventos do veículo. A partir de algumas métricas lidas no barramento, podemos acionar um evento, indicando que algo aconteceu. Os eventos não são apenas falhas e alertas, mas também podem indicar quem foi o operador que ligou o veículo, evento de conexão e desconexão do carregador. Todos os eventos são salvos com o horário em que foi ocorrido, o que ajuda nas análises posteriores. Os eventos são pré-definidos pela empresa e apresentados na tabela abaixo.

*Tabela 1 – Lista de Eventos*

<i>Classe</i>	<i>Código</i>	<i>Descrição</i>	<i>Valor Enviado</i>
Log	L001	Veículo foi ligado	ID da chave
Log	L002	Veículo foi desligado	ID da chave
Log	L003	Conexão com carregador	Carga na bateria
Log	L004	Desconexão com carregador	Carga na bateria
Log	L005	Tempo de assistente de rampa em uso	Tempo
Log	L006	Controle de cruzeiro acionado	Tempo
Log	L007	Tempo de frenagem de emergência	Tempo
Alerta	AI01	Modo de segurança em uso	Causa; Tempo
Alerta	AI02	Alarme de sobrecarga no motor	-
Alerta	AI03	Alarme de sobrecarga nos MOSFETs	-
Alerta	AI04	Alarme temperatura no motor	Temperatura
Alerta	AI05	Alarme temperatura Inversor	Temperatura
Alerta	AI06	Problemas barramento CAN	Cód. Erro
Alerta	AI07	Inversor desabilitado	-
Alerta	AI08	Dois movimentos habilitados	-
Falha	FI01	Sobre tensão no inversor	Tensão
Falha	FI02	Sobre tensão no motor	Tensão
Falha	FI03	Curto-circuito MOSFETs	Ponte em curto

Por fim, o RPi também cria um log contendo informações extras sobre o veículo, chamado pela empresa de “log permanente” já que carregam algumas informações totais do veículo, como os valores de odômetro e horímetro.

Por questões de espaço na memória, foi desenvolvida uma lógica que permite ao RPi armazenar apenas os últimos cinco minutos de Log do barramento CAN. Quando um arquivo é criado, é investigado se há um arquivo na memória. Caso não exista, o novo arquivo é adicionado; caso contrário, o arquivo antigo é sobrescrito pelo



novo. Caso um evento de falha ou alerta aconteça durante o intervalo relativo aos dados deste novo arquivo, ele é salvo na memória mesmo que lá exista outro arquivo.

Desta maneira, apenas os arquivos que não possuem falhas podem ser sobrescritos, fazendo com que o sistema armazene todos os dados de momentos de falhas para posterior análise pela empresa, já que esses são os dados mais valiosos para diagnósticos.

Este intervalo é definido para ser de um intervalo de cinco minutos para ter compatibilidade com o sistema do servidor, pois este apenas recebe mensagens de no máximo 1Mb por vez para impedir que haja uma sobrecarga da rotina de salvamento no banco.

Os logs de eventos e contendo as informações do log permanente possuem lógica mais simples devido ao tamanho reduzido. Eles se identificam por apenas um arquivo cada que é salvo na memória da placa. A cada vez que um evento acontece, uma nova linha é adicionada ao seu arquivo contendo suas informações. Quando um dado no log permanente precisa ser alterado, seu valor é sobrescrito em sua tabela.

Para a conexão *Wi-Fi* da MCO foram escolhidos adaptadores comerciais que seguem a filosofia de "plug and play", onde basta conectar os adaptadores na placa via usb e eles já estão configurados e prontos para rodar. A escolha destes dispositivos se deu pelo fato de que é necessária uma homologação da Anatel para utilizar e comercializar qualquer produto de telecomunicação no Brasil. Isso significa que caso a empresa produza uma placa que emita ondas de rádio, é necessário que esta placa seja aprovada pela Anatel para ganhar a sua homologação para venda e distribuição. E as homologações costumam ser bem custosas e trabalhosas.

Como a empresa atualmente não possui condições para homologar as placas, a solução de utilizar um adaptador comercial fica ideal, pois como já são produtos comercializados, possuem sua utilização garantida e certificada pela Anatel. Deste modo, não é necessário que nossa placa passe pela etapa de homologação para ser comercializada.

A conexão é feita utilizando scripts para o RPi que localiza uma rede *Wi-Fi* conhecida e entra com as suas credenciais, conseguindo se conectar a rede. Uma vez conectado, o RPi verifica se possui arquivos salvos. Caso tenha, envia para o servidor online do sistema todos os arquivos que possui na sua memória. Ao receber

uma confirmação de sucesso da operação de inserção dos dados no banco de dados, o RPi apaga os arquivos da sua memória, cedendo lugar para novos logs.

Ter esta abordagem para salvamento dos arquivos garante que os arquivos que são entregues ao banco de dados são sempre relevantes para análise da empresa. Garante também que não haverá perda de informação, pois caso haja erro na inserção no banco por algum motivo, os logs são mantidos até que a inserção seja completa. Além disso, como são arquivos pequenos, o veículo pode ficar um bom tempo sem se conectar a uma rede conhecida sem perder dados por falta de memória. Desta maneira garante a integridade de seus dados.

Uma vez no servidor, os logs são armazenados nos seus respectivos lugares no banco de dados do sistema. As lógicas do servidor e banco de dados serão explicadas no Capítulo 5.

## 4.2 Primeira Conexão com a Internet

Como pode ser notado no texto acima, a placa embarcada no carro se conecta a uma rede *Wi-Fi* já conhecida pelo veículo. Porém, ao vender o carro, a empresa não consegue saber qual serão os parâmetros da rede que o cliente quer se conectar de modo que consiga configurar o carro antes da venda.

Nem todos os modelos de veículos que serão vendidos pela empresa possuem algum tipo de interface para interação com o motorista, o que também dificulta a configuração do RPi para conhecer a rede. Como é de grande importância para a empresa que o cliente conecte o seu carro para conseguir enviar de fato as informações, também desenvolvemos um sistema de configuração de rede que é fácil de usar.

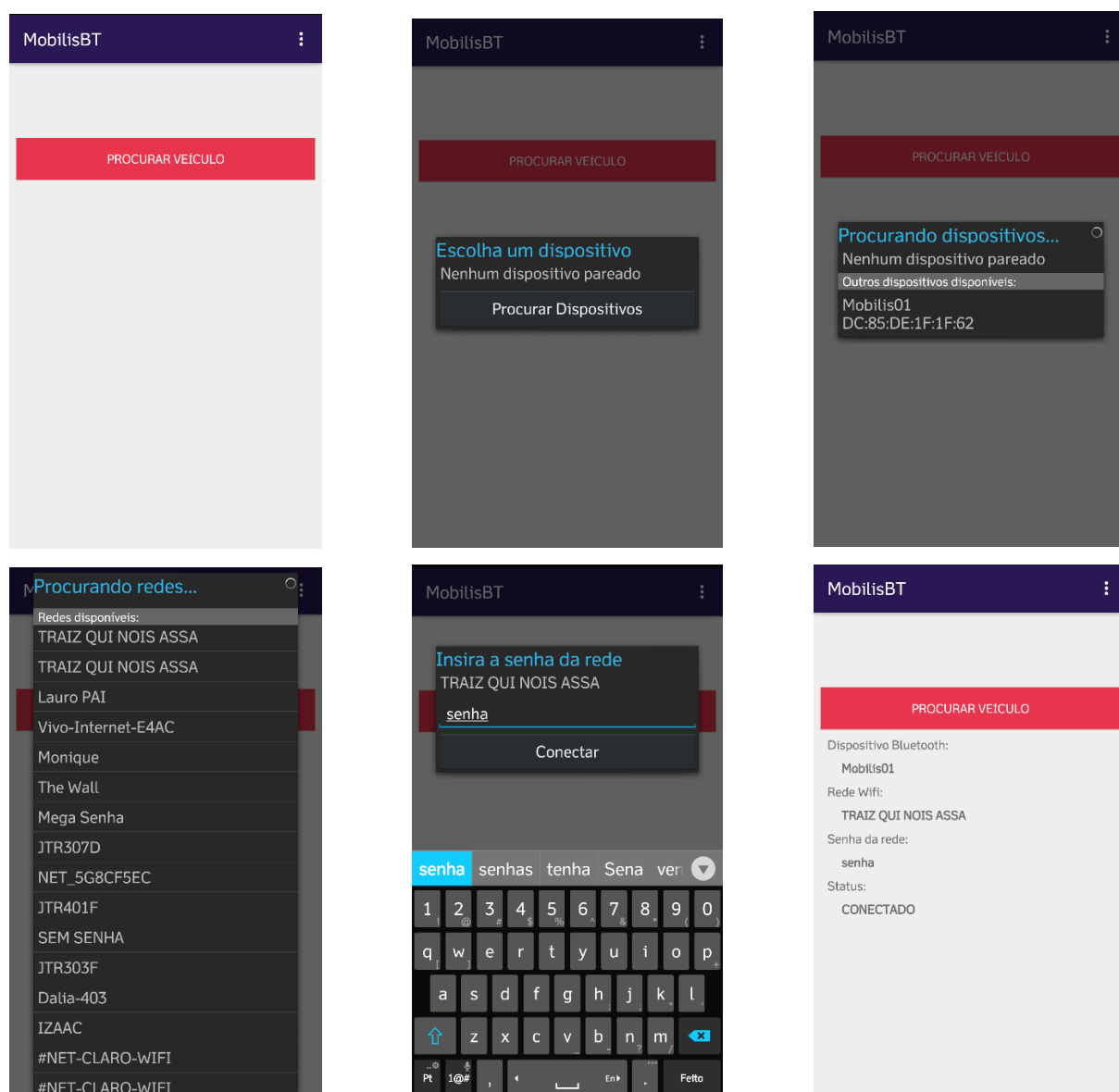
Este novo sistema se baseia em um aplicativo para dispositivos *Android* que é capaz utilizar o *Bluetooth (BT)* nativo do celular para se comunicar com a MCO do veículo. A comunicação é possível pois na placa também foi inserido um adaptador BT que possui as mesmas características que o adaptador *Wi-Fi*. Foi escolhido utilizar um adaptador comercial devido aos mesmos problemas de homologação encontradas com a tecnologia *Wi-Fi*.

Deste modo, o RPi também possui scripts que permitem a placa a se conectar a um dispositivo móvel via BT e recebe métricas do mesmo para conseguir uma conexão de rede *Wi-Fi*.

Ao iniciar, o App permite que seja listado todos os dispositivos BT ativos na proximidade. Ao selecionar o dispositivo relativo ao adaptador do veículo, é criada uma conexão. Na sequência, o aplicativo lista todas as redes *Wi-Fi* disponíveis na proximidade e pede para que o usuário selecione uma. Ao selecionar, um campo de inserção de texto aparece para que o usuário insira a senha daquela rede. Ao clicar no botão de enviar, o App envia para o veículo as informações relativas à aquela rede.

Esta sequência de acontecimentos pode ser visualizada na Figura 2, onde são apresentadas telas do sistema em sua ordem de funcionamento. A tela superior da esquerda é a tela inicial do aplicativo e a tela inferior da direita é a última.

Figura 1 - Sequência de telas do App para primeira conexão



Ao receber as informações, o RPi tenta se conectar. Se conseguir, uma mensagem de sucesso é enviada para o aplicativo e a conexão BT é encerrada. Caso contrário, uma mensagem de erro é enviada e o processo se repete até conseguir sucesso.

Este aplicativo permite que a configuração inicial de rede seja feita de maneira fácil e intuitiva, permitindo com o que o cliente consiga aproveitar os serviços oferecidos pela Mobilis facilmente.

## 5. SISTEMA WEB

Para atender todos os objetivos do projeto, foi projetado um sistema web que abranja tanto as ferramentas de análise de engenharia da empresa como as ferramentas de gerenciamento de frotas que serão disponibilizados para usuários.

O Projeto consiste em uma aplicação web que pode ser acessado por qualquer navegador acessando o link correto (mobilis.eco.br). O usuário que deseja acessar o sistema, faz o seu login (contando que já esteja cadastrado) e tem acesso as suas informações.

A solução para conseguir juntar ambas as partes do projeto em apenas um sistema, foi projetar um nível hierárquico de acesso, cuja identificação se faça a partir de login. O cliente possui um nível de acesso onde pode obter informações sobre a sua frota na forma de indicadores. Ele tem acesso ao log de eventos e também consegue obter informações sobre seus motoristas e outros usuários. Um funcionário da empresa Mobilis possui um nível de acesso que consegue obter dados dos veículos de todos os clientes como indicadores e eventos. Também consegue acesso a uma parte no sistema capaz de plotar os gráficos do barramento CAN.

Para a apresentação dos dados ao usuário, foram feitos scripts que realizam uma leitura nos dados dos logs e são capazes de calcular indicadores. Uma vez calculados e organizados, são apresentados em forma de lista ou gráficos. Os eventos são apresentados em lista e são organizados por qual carro ou por qual motorista que acionou aquele evento.

Também foram desenvolvidas páginas que permitem cadastro de novos usuários, assim como locais para alteração de dados. Essas páginas foram projetadas para ajudar na usabilidade do sistema e para dar independência ao usuário.

O sistema foi desenvolvido de maneira que armazenasse as informações dos veículos e que disponibilizasse a informação para a empresa e para qualquer usuário que tenha acesso ao sistema. Então um banco de dados adequado para a aplicação também foi projetado, garantindo a aplicação do sistema. E para controlar o acesso às informações, um servidor foi projetado.

Como é possível notar, neste projeto foi desenvolvido o *Back-End*, parcela que abriga o Banco de Dados e Servidor do sistema, tanto quanto o *Front-End*, responsável pela apresentação do conteúdo ao usuário. Neste documento estas partes serão explicadas separadamente.

Durante o desenvolvimento, o sistema foi testado utilizando um servidor local para testes, permitindo que o mesmo fosse acessado no navegador do computador onde o servidor local estava rodando. Os testes foram feitos no navegador Google Chrome utilizando as ferramentas de desenvolvedor que ele disponibiliza.

Como a empresa também está terminando de desenvolver o seu primeiro veículo, os dados inseridos no banco de dados para a análise do funcionamento do sistema são fictícios.

## **5.1 Back-End**

*Back-End* é o nome que damos à parte do sistema que é responsável pelo gerenciamento do conteúdo, regra de negócios e pela disponibilização dos serviços. Nesta parte estão presentes o Banco de Dados e o Servidor do sistema, que serão apresentados em sequência.

### 5.1.1 Banco de Dados

Para fazer que o sistema fosse viável, primeiro foi necessário projetar um banco de dados para atender a aplicação desejada. Primeiramente, foram levantadas quais eram as informações que seriam enviadas pelo veículo e quais seriam necessárias para que a aplicação atingisse o nosso objetivo. Depois disso, foi projetado o esquema do banco de dados.

O sistema deve separar de maneira correta os seus atores. Nele temos um usuário web que tem acesso ao sistema. Temos os motoristas ou operadores que terão acesso ao veículo, mas não ao sistema e o veículo em si. E para oferecer a oportunidade de se criar várias contas de usuário por cliente, também precisamos organizar o cliente. Então, para as informações de sistema, foram criadas as tabelas *grupos*, *usuarios*, *veiculos* e *motoristas*.

A tabela *grupos* separa os clientes, contendo seus identificadores e descrição. Os usuários web do sistema adquirido pelo cliente são localizados na tabela de *usuarios*. Ela possui as informações de cadastro e login de cada usuário web, contendo a identificação de seu grupo.

A tabela *veiculos* armazena os valores de cada veículo elétrico cadastrado no sistema. Contém as informações sobre o grupo o qual ele pertence, o número de série e chave de acesso. Estas duas últimas são as informações que o sistema se baseia

como login do veículo. A tabela *motoristas* contém as informações pessoais dos operadores ou motoristas autorizados a conduzir os automóveis.

O banco possui três tabelas que contém as informações do carro. Elas são *logeventos*, *logpermanente* e *logcan*, que armazenam as informações de seu respectivo log. A tabela *logeventos* apresenta uma lista com o código de erro e suas informações, assim como o horário do evento, horário de recebimento e identificações do veículo e motorista. A *logpermanente* contém informações dos veículos e são atualizadas a cada envio. *Logcan* armazena o arquivo .csv no barramento CAN na íntegra, assim como os identificadores do veículo e do horário de envio.

Por fim foi criada a tabela *error* que armazena os códigos de eventos, falhas e avisos e suas respectivas informações.

A Tabela 2 abaixo faz uma síntese da organização do banco e a Figura 1 apresenta a esquemática geral do banco de dados, conseguindo apresentar uma percepção mais completa da estrutura do banco.

*Tabela 2 - Estrutura do Banco de Dados*

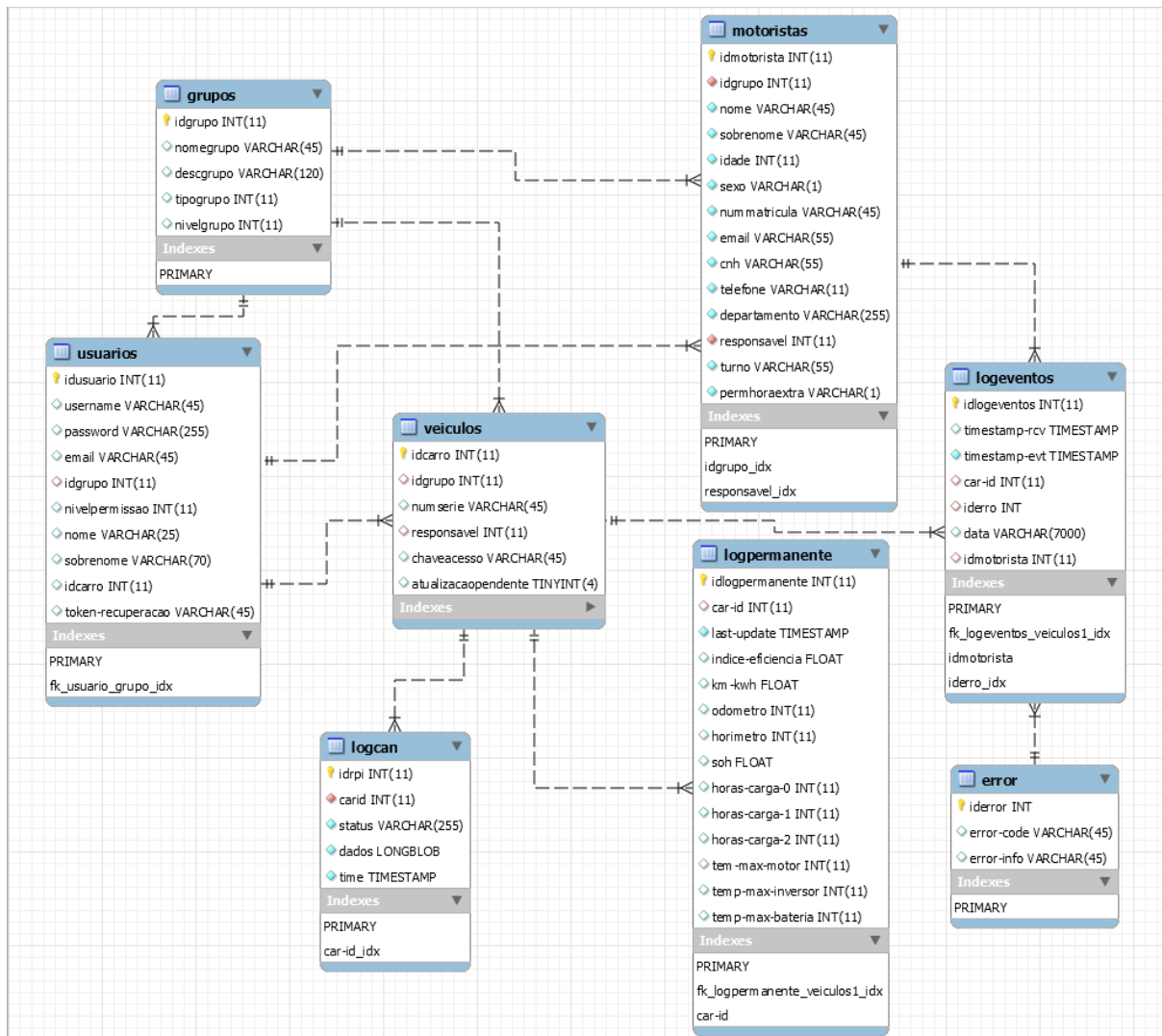
<i>Nome da tabela</i>	<i>Descrição</i>
grupos	Separa os usuários por grupo/empresa
usuarios	Armazena as informações de usuários web
veiculos	Armazena as informações dos veículos
motoristas	Armazena as informações dos motoristas/operadores
logeventos	Armazena os eventos
logpermanente	Armazena os dados permanentes do carro
logcan	Armazena o arquivo .csv do CAN do veículo
error	Armazena as informações de eventos, falhas e alertas

Para maior segurança, o banco também possui acesso bloqueado por senha. Qualquer pessoa ou requisição de sistema que queira ter acesso às informações do banco precisa possuir as informações de usuário e senha corretos, caso contrário a requisição é rejeitada.

As requisições feitas no servidor do sistema carregam essas informações e garantem que o nosso acesso seja aceito durante sua utilização. A proteção é feita para barrar acessos indevidos ao banco e proteger os dados.

Uma vez o banco projetado, ele é carregado no servidor usando a ferramenta phpmyadmin onde entramos com o código SQL para montar a lógica das tabelas que contém as informações que precisamos para o sistema.

Figura 2 - Esquemática do banco de dados do sistema



### 5.1.2 Servidor

O servidor do sistema é o responsável por atender todas as solicitações que desejam acesso aos dados do sistema, como códigos HTML para a apresentação gráfica ou como acesso aos dados contidos no banco de dados do sistema.

Para garantir um bom funcionamento da aplicação, o sistema foi baseado na arquitetura REST (Transferência de Estado Representacional) de aplicações web, onde a troca de informações entre as suas partes é baseada no protocolo HTTP. Para o seu desenvolvimento, foi utilizada a *framework* Slim para a construção de rotas mais



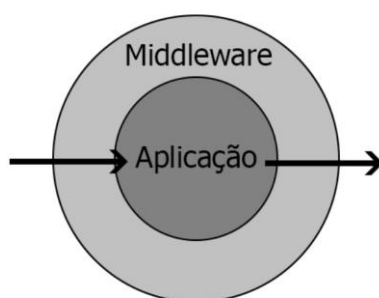
organizadas para as requisições, conseguindo organizar as funções da aplicação por rotas pré-determinadas.

Com a arquitetura REST, o servidor não deve armazenar informações de estado do usuário (como por exemplo suas credenciais), fazendo necessário que cada requisição HTTP contenha estas informações. Para conseguir transmitir as informações de usuário para o servidor e garantir que o cliente receba as informações corretas de resposta, utilizamos um Jason Web Token (JWT).

O JWT é anexado ao cabeçalho da requisição HTTP contendo as informações necessárias do usuário, como nome, e-mail e outros identificadores. Estas informações são essenciais para que as requisições ao banco de dados retornem as informações daquele usuário específico e de nenhum outro. O JWT também nos permite realizar a autenticação do usuário, o que oferece maior segurança ao sistema e integridade dos dados.

Para melhor aproveitar os recursos de utilizar JWT, foi desenvolvido um middleware de autenticação. O middleware funciona como uma máscara da aplicação, pois toda requisição ao servidor passa primeiro por ele. Neste momento, o token é verificado. Caso a autenticação falhe, o middleware nega o acesso, retornando um erro para o sistema. Caso a autenticação ocorra com sucesso, a requisição segue a sua rota normalmente. A Figura 3 representa o caminho de uma requisição no sistema, permitindo melhor visualizar a sua lógica.

*Figura 3 - Middleware*



O JWT utilizado neste sistema também possui uma informação sobre o tempo de expiração, fazendo com que cada token tenha um limite de tempo para ser utilizado, limitando o tempo de acesso único ao sistema.

O middleware, entretanto, garante passagem sem verificação de token para rotas pré-determinadas. Essas passagens sem verificação são para casos em que o

sistema ainda não gerou o token para o usuário, como operações de login (do usuário, tanto quanto do veículo) e troca de senha.

No caso do login, por exemplo, o usuário insere seus dados no sistema e os envia. O envio é feito através de uma requisição HTTP ao servidor contendo estes dados. A requisição passa pela verificação, o servidor realiza a busca ao banco de dados, e, caso os dados forem corretos, o servidor retorna uma mensagem contendo o JWT que será utilizado naquela sessão do sistema. Esta lógica de troca de mensagens também é representada pela Figura 4 na seção 5.1.4.

As Views do sistema web (ou as páginas propriamente ditas) também são separadas por rotas programadas a partir do framework. Porém, suas rotas são acessadas a partir do redirecionamento do navegador para a rota específica da página para qual queremos navegar. As rotas das views também são liberadas pelo middleware, já que resultam apenas no retorno do conteúdo HTML das páginas.

Para realizar a navegação, o link referente a próxima página é carregado na URL do navegador a partir do clique no botão referente a página desejada. O link <https://mobilis.eco.br/api/public/dashboard> aponta para a View dashboard, por exemplo, que é responsável pela renderização dos recursos HTML e CSS da página painel de controle assim como o javascript dela.

Para garantir que o sistema mantenha as informações de usuário mesmo com a troca de página, o JWT é carregado junto ao link de navegação, fazendo com que a URL do sistema fique da seguinte forma:

<https://mobilis.eco.br/api/public/dashboard?t=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJleHAiOiJlE0OTkwMjg3MzMsIm5tZSI6IiZpc2l0YW50ZSIsImVtbCI6ImFkbWluQG1vYmlsaXMiLCJpZGciOiJlIn0.pGq0xDdwpJ81uED58pZ859Ys60fU0BpJ-FpTC4UNhrw>

Assim que a página é carregada, os scripts da nova página acessam as informações contidas na URL do navegador e salvam o token como variável local. Depois disso, a página é renderizada e outros scripts fazem as requisições ao servidor, que este por sua vez faz as requisições ao banco de dados que retornam com os dados desejados. Uma vez os dados retornados, eles são então interpretados no restante do código e depois apresentados para o usuário. A lógica de navegação também é apresentada pela Figura 5 apresentada na seção 5.1.4 deste capítulo.

O servidor também está organizado para que o navegador carregue apenas o código desejado para aquela determinada parte do sistema. As pastas são organizadas de maneira que o público só tenha acesso aos códigos HTML, CSS e javascript que contêm o que é necessário para fazer a página ser montada e as informações serem apresentadas de maneira correta ao usuário. Nenhum código do servidor é carregado pelo navegador, garantindo maior proteção ao banco de dados.

Os códigos das bibliotecas e frameworks utilizados encontram-se em um diretório separado dos demais códigos e são carregados nas páginas individualmente quando requisitados.

### 5.1.3 Rotas

Qualquer requisição ao servidor segue um fluxo determinado de funções que garante o bom funcionamento do sistema. Ao começar, a requisição chega na pasta /public e por convenção, acessa o arquivo index e o arquivo .htaccess. No .htaccess estão as autorizações de requisição. É aqui que declaramos que o sistema pode ser acessado por qualquer IP de usuário e o abrimos para ser acessado de qualquer pessoa na internet.

No index a estrutura do sistema é carregada. Inicialmente cria-se a instância da nova aplicação Slim e depois as configurações de views e de jwt são setadas. Na sequência são requisitados os códigos de middleware. O middleware funciona como uma máscara do sistema. Ele é responsável por bloquear requisições indesejadas e acessos indevidos. Antes dele, qualquer requisição ao sistema seja ela qual for e da onde vem é aceita e é nele que conseguimos proteger o nosso sistema de requisições indevidas ou maliciosas. Este comportamento é desejado para garantir a integridade do banco de dados, pois uma requisição maliciosa pode ler, alterar ou deletar os dados salvos naquele banco.

Após o middleware, as rotas são declaradas no arquivo index. Essa ordem de chamada é importante porque garante que as configurações do sistema estarão sempre corretas, pois declarando primeiro, o sistema compila o código do middleware antes de criar as rotas.

Os parâmetros do banco de dados são carregados com o primeiro arquivo de rotas. É nele que está o endereço IP que aponta para o servidor onde o banco de

dados está alocado, e parâmetros de usuário, senha e nome do banco para garantir o acesso aos dados.

Cada rota é responsável por alguma função e normalmente fazem requisições ao banco de dados do sistema. Em cada uma delas, uma chamada específica em SQL é definida para realizar as ações desejadas perante ao banco.

Existem rotas que são acessadas pelo usuário no sistema web e outras que são acessadas pelo carro. Abaixo segue a tabela de rotas e uma breve descrição das mesmas. O usuário do sistema tem acesso as rotas identificadas pela rota /client, o veículo tem acesso às rotas identificadas com /car e a /plotter pode apenas ser acessado por um funcionário da Mobilis.

*Tabela 3 - Rotas do servidor*

<i>Endereço</i>	<i>Descrição</i>
/client/login	Recebe credenciais de um usuário e retorna um web token ou erro
/client/newpassword	Recebe credenciais de um usuário e retorna string de sucesso ou erro
/client/readevents	Retorna string JSON contendo os eventos ou erro
/client/readlogperm	Retorna string JSON contendo o log permanente ou erro
/client/register	Recebe credenciais de um usuário e retorna string de sucesso ou erro
/client/registercar	Recebe credenciais de um usuário e retorna string de sucesso ou erro
/client/listfleet	Retorna string JSON contendo os carros da frota ou erro
/client/listgroup	Retorna string JSON contendo as pessoas daquele grupo ou erro
/client/getInfo	Retorna string JSON contendo informações de usuário ou erro
/client/updateuser	Recebe credenciais de um usuário e retorna string de sucesso ou erro
/client/updatecar	Recebe credenciais de um veículo e retorna string de sucesso ou erro
/client/maildiagnostic	Recebe informações de um usuário e retorna string de sucesso ou erro
/cliente/mailcontact	Recebe informações de um usuário e retorna string de sucesso ou erro
/plotter	Recebe parâmetros de carro e retorna arquivo .csv de log
/car/auth	Recebe credenciais de um carro e retorna web token ou erro
/car/dados/permanente	Recebe arquivo .csv e retorna string de sucesso ou erro
/car/dados/eventos	Recebe arquivo .csv e retorna string de sucesso ou erro

O servidor também contém rotinas que enviam informações via e-mail ao destinatário desejado. Ele utiliza o protocolo SMTP de e-mails que são nativos do provedor de serviços de hospedagem e enviam e-mail assim que solicitadas. Abaixo segue uma tabela com as rotas que possuem uma função de e-mail e a descrição do conteúdo do mesmo

*Tabela 4 - Rotas com envio de e-mail*

<i>Endereço</i>	<i>Descrição</i>
/car/dados/eventos	Envia e-mail para a empresa contendo as falhas no log enviado caso encontre
/user/newpassword	Envia e-mail para usuário contendo a nova senha
/client/maildiagnostic	Envia e-mail para a empresa contendo as informações preenchidas na página pelo usuário e um e-mail de confirmação para o usuário

Para facilitar o entendimento da sequência lógica de troca de informações do sistema, foi feito três diagramas de sequência em que foram demonstradas funções principais que geram um entendimento representativo de todas as outras funções dele.

Os diagramas aqui representados são das funções de login de usuário web e seu primeiro redirecionamento, uma função de redirecionamento da página por clique e a função de login e envio de informações a partir do veículo.

É interessante notar que o veículo utiliza as suas funções sem precisar interagir com a interface gráfica do sistema, logo suas trocas de informações são diretamente com o servidor.

Figura 4 - Diagrama de Sequência da função de login

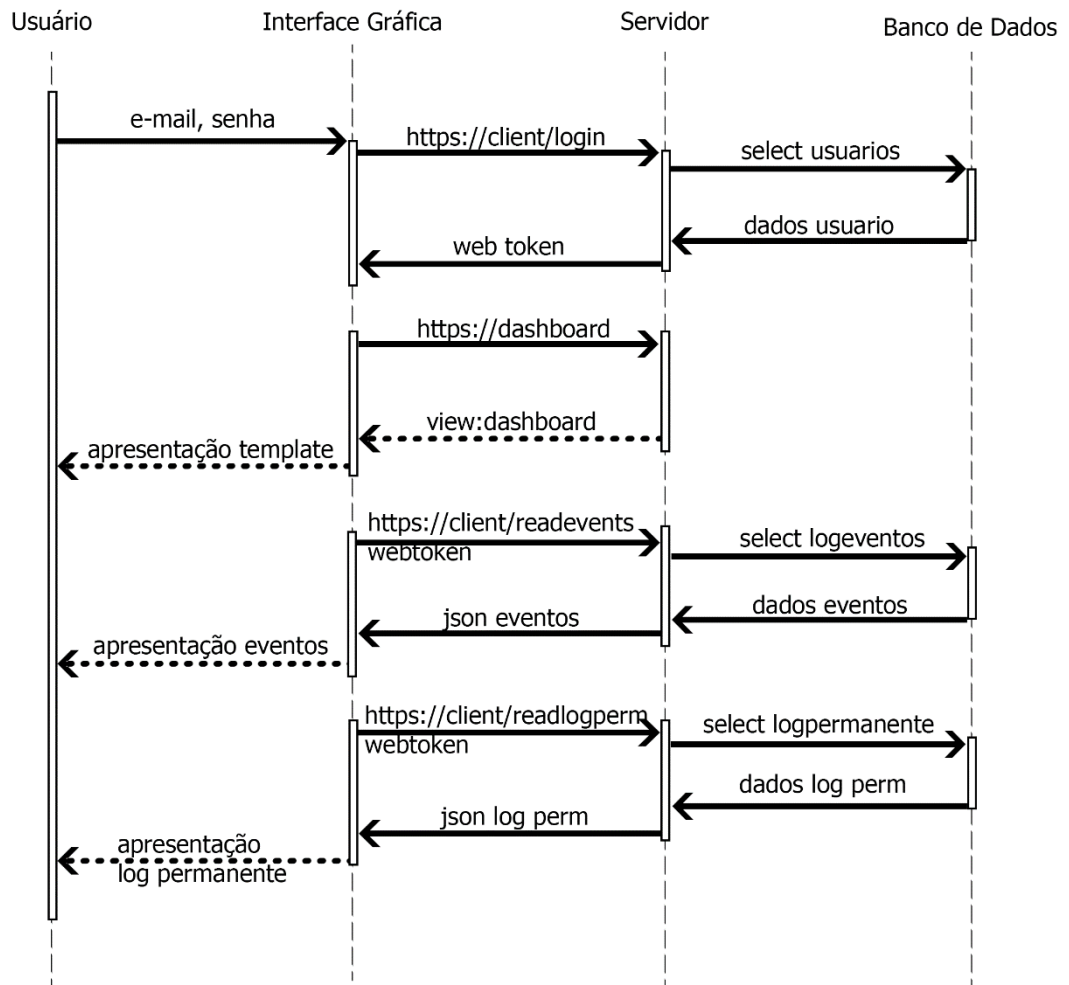


Figura 5 - Diagrama de Sequência de navegação

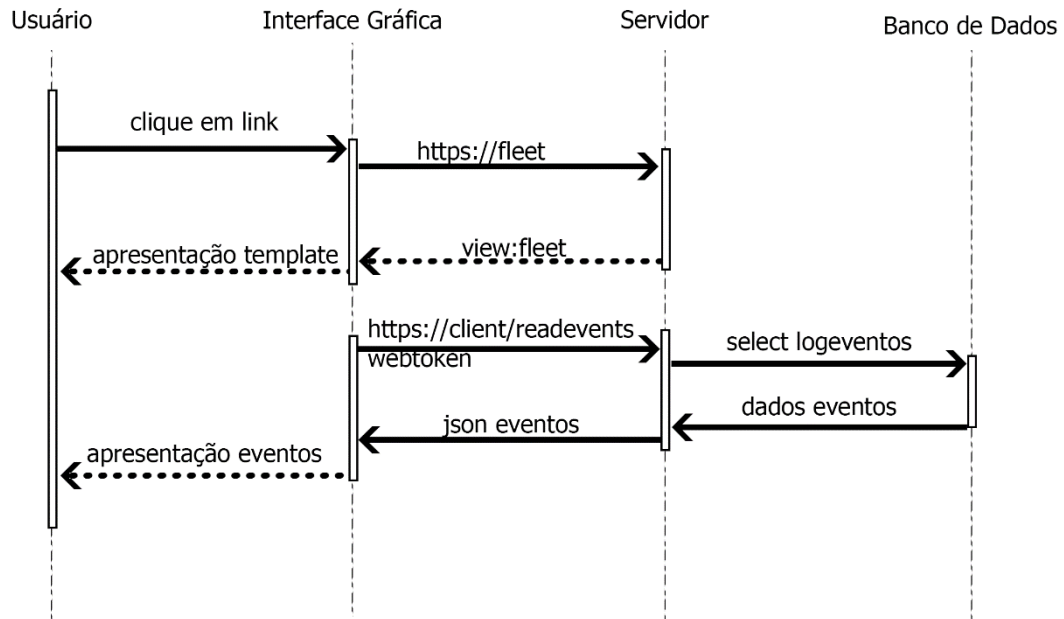
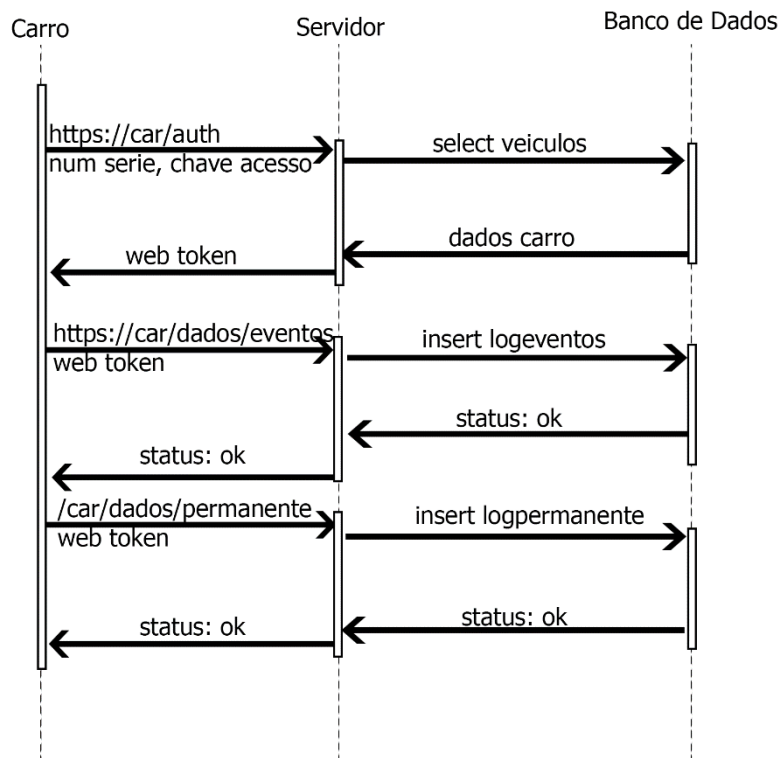


Figura 6 - Diagrama de Sequência de login e envio de dados pelo veículo



## 5.2 Front-End

O *Front-end* é responsável pela apresentação do conteúdo final ao usuário, assim como por sua interação com o sistema. É aqui que o usuário faz a inserção de dados e tem outros dados apresentados para ele.

Para a apresentação de conteúdo para o usuário final foram desenvolvidas diversas páginas com as mais variadas funcionalidades, com o objetivo de deixar a experiência do usuário a mais simples e intuitiva possível.

As páginas são: Login, Painel de Controle, Oficina, Log da Frota, Log de usuários, Cadastros (usuário web, operadores ou veículos), Diagnóstico, FAQ, Contato e Modo Engenharia.

Como esta etapa de desenvolvimento também consiste no design da página em si, uma empresa de design responsável pela identidade visual da Mobilis foi acionada para sugestões de apresentação e estilos, e o resultado desta colaboração foi o estilo presente atualmente no sistema.

Cada página carrega o seu código HTML e CSS para apresentação e o javascript para a lógica de aplicação. Toda página, com exceção da página de login, é carregada depois de um *template* que contém a identidade visual do sistema. Este *template* contém o código HTML responsável pelas barras de navegação superior e lateral, contendo os botões de navegação do sistema.

O código HTML de cada página é o responsável por fazer as requisições de scripts CSS e javascript que a página irá precisar. No script do template são feitas as chamadas dos códigos CSS que são responsáveis pelo estilo do sistema, e são os scripts HTML de cada página que chamam as funções javascript. Deste modo, apenas os scripts que serão utilizados naquele momento pelo sistema são carregados pelo navegador.

Os arquivos javascript carregados pela página que são os responsáveis por responder aos eventos de clique e foco do mouse, por fazer as requisições HTTPS ao servidor, e montar a página dinamicamente de acordo com os dados retornados nas requisições.

Abaixo segue uma explicação da lógica por trás de cada página. Os detalhes das informações que são apresentadas ao usuário são apresentados na próxima sessão.



### 5.2.1 Páginas

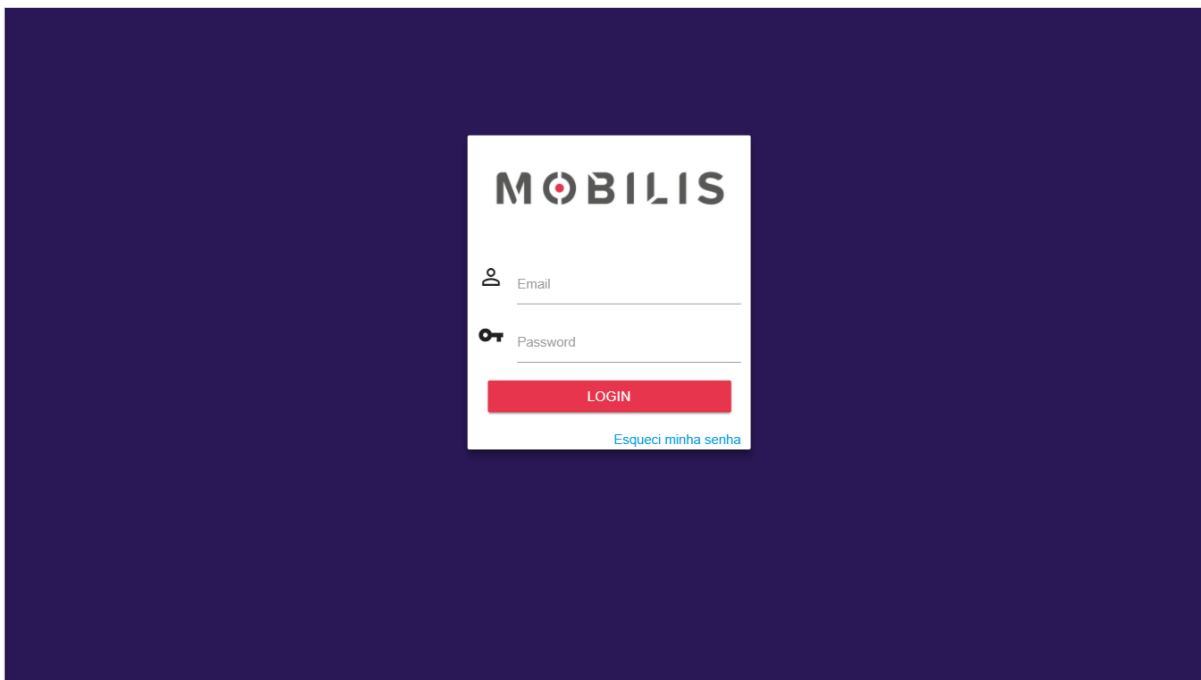
Todas as páginas possuem algumas funções que são carregadas junto a um script comum que é responsável em responder ações do usuário no template. Essas funções são a de navegação para uma página ao clique do usuário na aba correta e função de logout, que redireciona o usuário para a página inicial e apaga o jwt relativo a aquele usuário.

Cada View do sistema começa fazendo uma leitura do token e registra as informações do usuário na barra de navegação lateral para fazer a identificação do mesmo.

#### 5.2.1.1 *Login*

A página de Login (Figura 7) contém os campos de e-mail e senha para o usuário já cadastrado no sistema ter acesso às suas informações. Após o preenchimento dos dados, o sistema faz uma requisição ao BD. Se o usuário existir e a senha for a correta, o usuário é redirecionado para a página do Painel de Controle, caso o usuário ou senha forem inválidos, um aviso aparece e o usuário permanece na página de login.

Nesta página há também um campo caso o usuário tenha esquecido a sua senha. Ao clicar no link "Esqueci a senha" os campos de e-mail e senha dão lugar a campos de e-mail e primeiro nome, onde o usuário deve preencher o seu e-mail e o seu primeiro nome que está cadastrado no sistema. Caso o e-mail e nome forem válidos, um e-mail contendo uma nova senha é enviado para o usuário. Depois deste e-mail, esta nova senha já está cadastrada no sistema, então o usuário conseguirá realizar o seu novo Login com esta senha.

*Figura 7 - Página de Login*

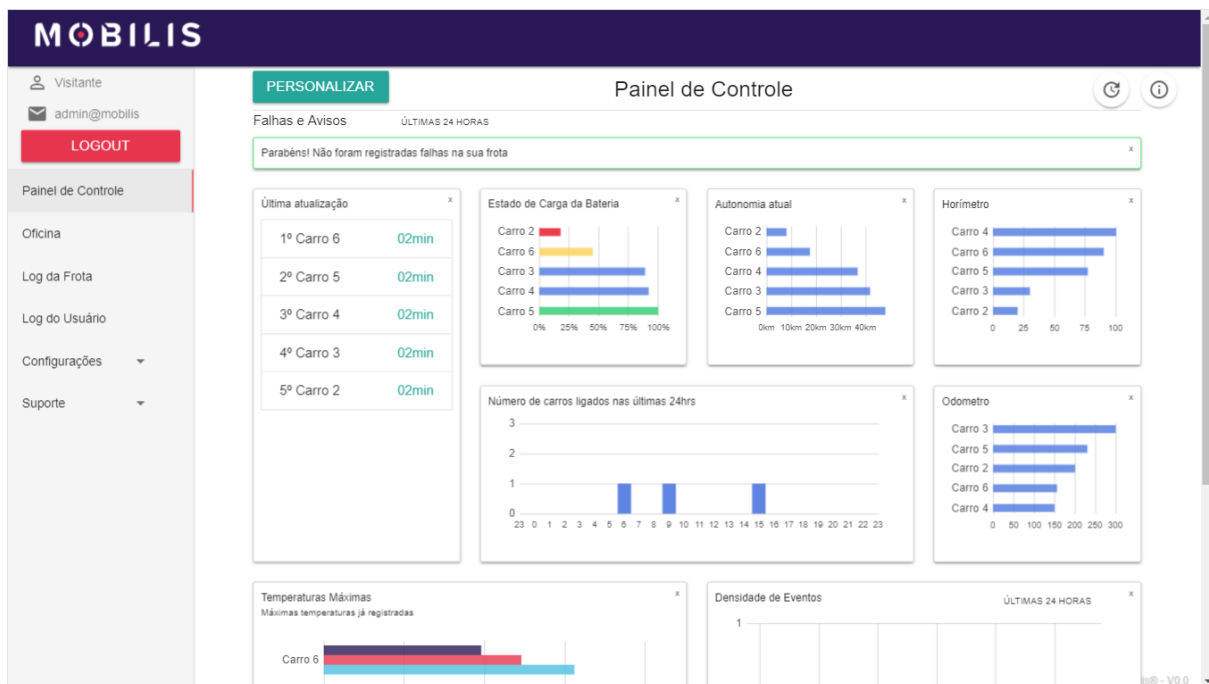
#### 5.2.1.2 Painel de Controle

No Painel de Controle são apresentadas as informações de caráter mais imediato dos veículos, assim como informações mais importantes ao gerente de frota.

Ao carregar, a página faz requisições ao banco de dados para retornar o log de eventos e o log permanente referentes a todos os veículos da frota do usuário. Após o retorno dos dados, uma análise é feita para criar os indicadores e os resultados são apresentados ao usuário.

Aqui o foco é mostrar as informações do log permanente de cada veículo. O log de eventos serve para carregar as falhas e alertas gerados por aquela frota e apresenta-las para o usuário. A Figura 8 representa a interface desta página.

Figura 8 - Página Painel de Controle

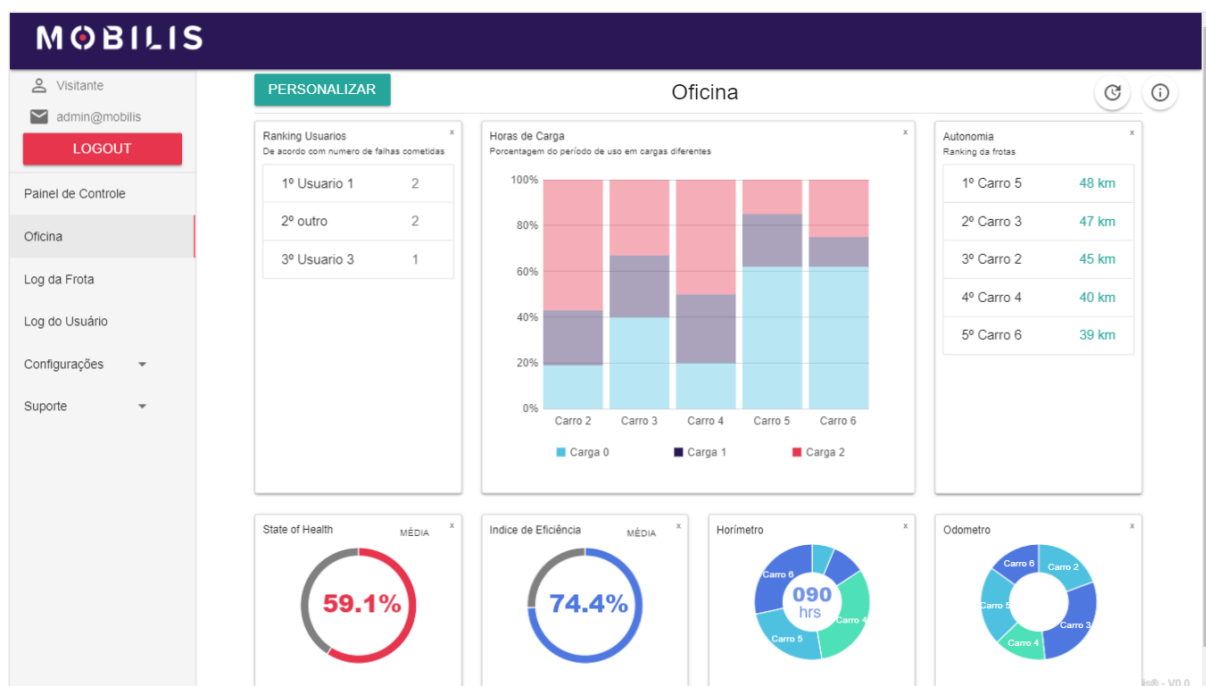


Nesta página também é possível realizar uma personalização do seu painel, permitindo que o usuário omita as informações que não deseja.

### 5.2.1.3 Oficina

Na página "Oficina" são apresentadas informações que representam um retrato da frota naquele determinado momento; são os dados relativos à saúde de cada veículo, o que auxilia na programação de uma manutenção preventiva caso seja necessário. A página está representada pela Figura 9.

Figura 9 - Página Oficina



#### 5.2.1.4 Log da Frota

Página que mostra o resumo dos eventos separados por cada veículo da frota. Ao ser carregada, a página faz requisições ao log de eventos, para a listagem do grupo e requisições do log permanente para colher algumas informações de cada veículo. Nesta página, inicialmente é apresentada uma lista dos veículos contendo alguns dados mais gerais do mesmo como: número de série, horímetro e odômetro totais. Ao clicar no veículo desejado, uma lista contendo os eventos registrados desse determinado carro é apresentada ao usuário, contendo o código do evento, hora do evento, dados e operador responsável. A página pode ser visualizada na Figura 10.

Figura 10 - Página Log da Frota

**MOBILIS**

Visitante  
admin@mobilis  
LOGOUT

Painel de Controle  
Oficina  
Log da Frota  
Log do Usuário  
Configurações  
Suporte

### Resumo da frota

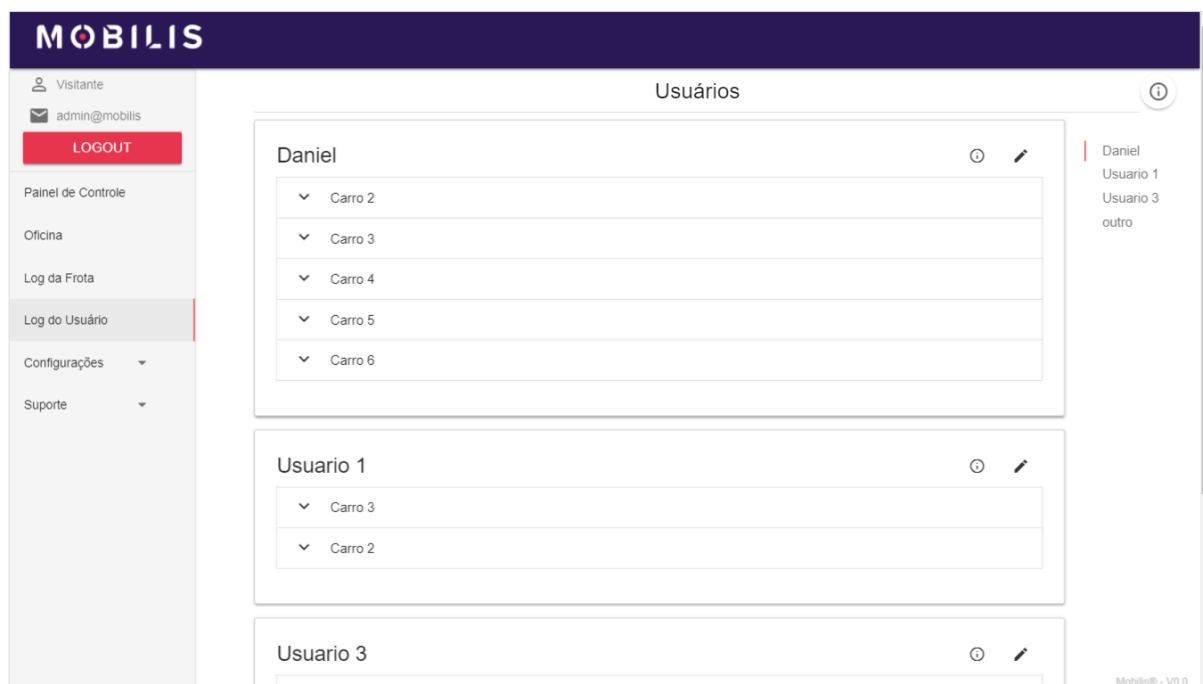
Carro	Número de série	Total de Km	Total de Horas
2	22222222	200	20
3	33333333	300	30
4	123456789	150	100
<p>L003 2017-06-06 16:22:00</p> <p>Dados: SOC Código de Erro: L003 Horário de Envio: 2017-06-19 11:44:01 Responsável: Daniel</p> <p>L001 2017-06-06 12:00:00</p> <p>EI002 2017-06-16 06:00:00</p>			
5	5559595959	230	77

Mobilis - V0.0

#### 5.2.1.5 Log de Usuários

Essa página é análoga à página de Log de Frota, porém com os eventos separados por operadores e veículos. Desta forma, o usuário pode analisar quais os veículos são utilizados por cada usuário, assim como saber quais foram os eventos acionados por ele. Os eventos nesta página e na página anterior são apresentados seguindo a sequência cronológica, onde o primeiro apresentado é o mais recente. A Figura 11 contém a representação desta página.

Figura 11 - Página Log de Usuários



#### 5.2.1.6 Cadastros

Essas são as páginas que fazem os cadastros do sistema. Temos três opções: cadastrar um novo usuário para o sistema web; um novo operador dos veículos que não terá acesso ao sistema; ou cadastrar um novo carro à sua frota.

Cada situação funciona da mesma forma: o usuário entra com os dados necessários para fazer o cadastro e os submete. Depois, o servidor recebe as informações e as salva no banco nas suas respectivas tabelas. Caso algum problema ocorra, um aviso é apresentado ao usuário e a ação de inserção no banco não é feita.

#### 5.2.1.7 Diagnóstico

Esta página foi criada para o pedido de diagnóstico de problemas de um veículo por um usuário do sistema. Nesta página o usuário identifica qual o carro que apresenta problema, seleciona alguma opção de problema que pode estar acontecendo e faz uma breve descrição. Ao solicitar o diagnóstico, um e-mail é enviado à Engenharia da Mobilis com as informações preenchidas pelo usuário e o responsável pelo diagnóstico poderá fazer o seu trabalho assim que possível.

#### 5.2.1.8 FAQ

Página contendo Perguntas e Respostas das dúvidas mais comuns dos usuários do sistema. As informações aqui contidas serão tanto da usabilidade do sistema web como do veículo.

#### 5.2.1.9 Contato

Contém as informações gerais da empresa, assim como informações de contato e espaço para enviar um e-mail com informações mais gerais.

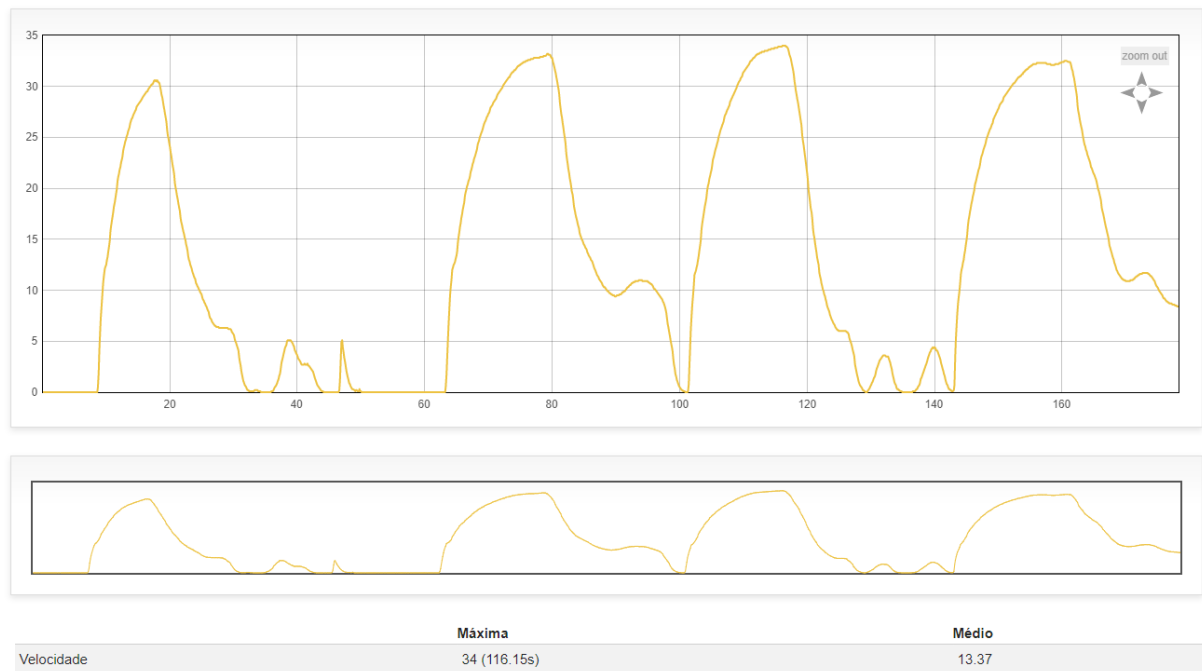
#### 5.2.1.10 Modo Engenharia

Esta página é a referente ao sistema de análise dos dados dos veículos por um engenheiro da Mobilis. Esta página nos permite escolher qual carro se quer analisar e apresenta os logs CAN salvos no banco relativos ao veículo selecionado. Nela é possível fazer as mais diversas plotagens que são relevantes para a análise do carro e cada gráfico pode ser manipulado para facilitar a análise. Aqui conseguimos analisar resultados de testes de maneira muito rápida, além de ser possível realizar o acompanhamento das métricas dos veículos remotamente.

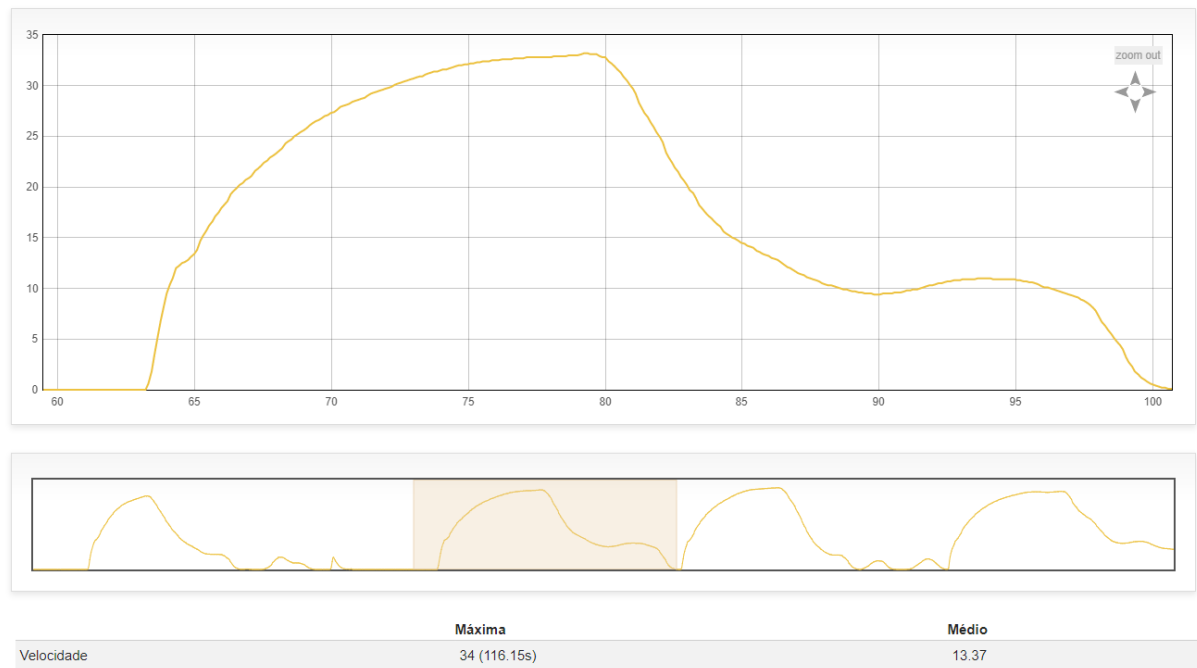
Os gráficos que são gerados nesta página também possuem muitas ferramentas que permite facilitar as análises, como: zoom, navegação e seleção de área para zoom, como apresentado na Figura 12. Os valores máximo e médio das variáveis que estão sendo representadas nos gráficos também são disponibilizados em uma tabela abaixo dos mesmos.

**Figura 12 - Exemplo Zoom Gráficos**

Velocidade X Tempo



Velocidade X Tempo

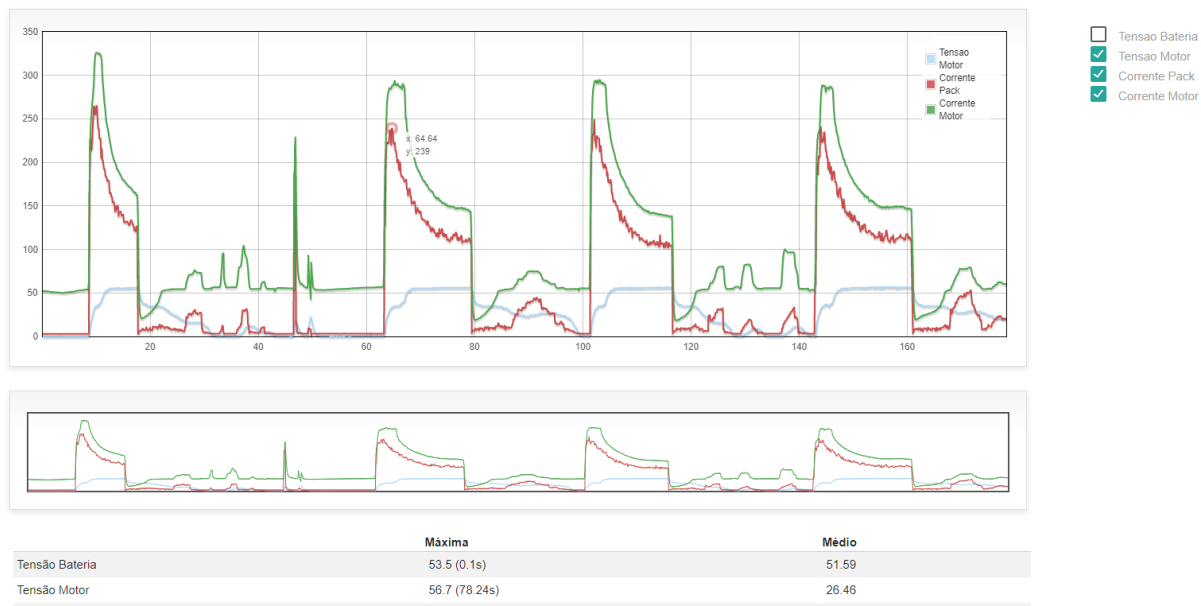


Existem gráficos que são sobrepostos e o sistema fornece a opção de omissão de variáveis que não são desejadas naquele momento, assim como apresenta o valor exato dos pontos apontados pelo ponteiro do mouse. Estas funções podem ser visualizadas na Figura 13.



Figura 13 - Exemplo Seleção e Valor

Tensão e Corrente x Tempo



### 5.2.2 Indicadores

Para que o sistema atendesse seus objetivos gerais, foi necessário elaborar maneiras de apresentar os dados armazenados pela aplicação de maneira simples e intuitiva. Portanto, para criar uma ferramenta que seja útil no dia a dia de um gerente de frotas, foi necessário criar soluções para os possíveis problemas de gerenciamento que o usuário pode estar tendo em sua frota.

Nesta etapa, contamos com colaboradores da empresa que possuem grande experiência na área e fomos capazes de identificar os principais problemas deste nosso cliente. Os problemas são: a incapacidade de identificar rapidamente as falhas e quem foram os responsáveis por elas; fazer o acompanhamento da carga atual da bateria do veículo para realizar melhores planejamentos de rotas de trabalho.

Levando em conta estes maiores problemas dos clientes e quais os dados que temos armazenados dos veículos, foram criados indicadores que são apresentados nas páginas de Painel de Controle e Oficina.

A seguir são explicados estes indicadores de maneira individual. Como as páginas de Log apresentam o log de eventos na íntegra, elas não são discutidas aqui.

### 5.2.2.1 *Painel de Controle*

O Painel de controle exibe indicadores que são de importância mais imediata para o gerente de frota. No início da página, as falhas e alertas que ocorreram nas últimas 24 horas são disponibilizados ao usuário. É importante ressaltar estas novas falhas para que o gerente saiba quando uma ocorreu e para que a frequência delas seja mais fácil de ser identificada.

Nesta etapa são disponibilizados apenas os carros em que aconteceram falhas e alertas, código e horário do evento. Caso não haja eventos de falhas nas últimas 24 horas, uma mensagem de congratulação é apresentada. O intervalo de tempo também pode ser alterado. Caso o usuário queira, ele pode selecionar a última semana, último mês ou último ano, apresentando assim as falhas e alertas destes novos intervalos de tempo.

Na sequência, é exibido a última atualização de cada veículo, pois é necessário saber se as informações estão atualizadas ou não, já que todos os dados apresentados são referentes aos dados enviados até a última atualização do veículo. E caso não esteja havendo sincronização de um ou mais veículos conseguimos saber por aqui.

As informações de estado de carga da bateria de cada carro, assim como os valores de horímetro e odômetro totais de cada veículo são plotados em gráficos de barra. Os indicadores de horímetro e odômetro são apresentados de maneira decrescente, e a informação do estado de carga é de maneira decrescente, pois desta maneira, a primeira informação contida no gráfico será sempre a mais importante para o usuário, pois identifica de maneira fácil qual o carro que mais andou e qual possui menor carga de bateria atualmente.

O gráfico de barras do estado de carga da bateria também muda de cor de acordo com o nível medido para ajudar na identificação das cargas. O gráfico fica vermelho para as cargas abaixo de 20%, amarelo para cargas abaixo de 50%, azul para cargas até 99% e verde para cargas completamente carregados. Como este indicador tem uma relevância grande para uma frota de veículos elétricos, ter a separação por cores facilita a leitura da identificação dos níveis.

Depois, temos os indicadores de autonomia atual do veículo em quilômetros e em horas. Os indicadores são medidos de acordo com os índices de autonomia que o veículo possui no banco de dados com o estado de carga atual, podendo ser escolhido em plotar o gráfico apresentando quantos quilômetros ou horas estimadas

o veículo consegue operar até que a carga acabe. Este indicador é importantíssimo para o planejamento de rotas em empresas em que o carro demora a ter acesso a uma fonte de energia. Se a operação que o veículo precisar fazer naquele dia requer no mínimo 4 horas de trabalho, é necessário escolher um veículo com essa autonomia para realizá-lo, senão o trabalho será encerrado quando a bateria do veículo acabar, gerando mais problemas de logística para o cliente.

Também há indicação de quais são os veículos ativos naquele momento e quem é o operador responsável por ele. Dessa forma, fica mais dinâmica a identificação dos operadores.

Há também um histórico de quais veículos estavam ativos nas últimas 24 horas em um gráfico número de veículos por tempo, plotando o intervalo de tempo que aqueles veículos estavam ativos, podendo filtrar a apresentação por usuário ou veículo específico. Este indicador é interessante para acompanhar a jornada de trabalho daquele usuário ou veículo nas últimas 24 horas. Também é possível mudar o intervalo de tempo de plotagem, podendo alterar para última semana ou último mês, mas aumentando o intervalo perdemos a resolução da hora exata de ativação do veículo.

Há também históricos de eventos das últimas 24 horas plotados por tipos, separados em evento normal, falha ou aviso. Indicador interessante para se identificar se existe padrão de falhas de acordo com as horas do dia. Aqui também é possível fazer a alteração do intervalo, mas novamente perdemos a resolução das horas em que o evento ocorreu.

#### 5.2.2.2 Oficina

A ideia subjacente a essa página Oficina é a de fazer um retrato da frota naquele momento, mostrando informações mais relacionados à qualidade de cada veículo. Primeiramente, é apresentado um ranking de usuários levando em conta o número de falhas e avisos que eles geraram enquanto operavam os veículos. Nele, quem cometeu mais falhas é o primeiro colocado. Este ranking é importante para identificar aqueles operadores que cometem mais falhas que os outros, promovendo ao gerente de frota a oportunidade de realizar alguma ação com aquele funcionário específico para diminuir as falhas cometidas.

Também é apresentado um gráfico que contém a porcentagem do tempo em que o veículo operou transportando uma determinada carga. Separamos três perfis de cargas: abaixo de 100kg, entre 100 e 200kg e acima de 300kg. Este indicador é interessante para identificar quais veículos estão sendo mais requisitados para fazer o transporte de cargas pesadas, pois este tipo de atividade é responsável por uma série de falhas e problemas em veículos elétricos, então fazendo a análise do gráfico é possível planejar um rodízio de veículos para transporte de cargas pesadas prezando em manter a saúde dos mesmos.

Caso seja identificado um período de mais de 50% do tempo total com cargas acima de 300kg em um veículo, um alerta aparece para o usuário.

Em sequência é apresentado um ranking da autonomia do veículo, mostrando ao gerente a autonomia com 100% de carga de cada um deles. Esse indicador é interessante para identificar o melhor e pior veículo, conseguindo fazer a análise da diferença entre eles. Caso o pior veículo esteja muito pior que os outros, alguma ação deve ser tomada.

Em seguida também é apresentado um gráfico que apresenta a média do estado de saúde da bateria do veículo. Estado de saúde é um indicador que identifica qual a porcentagem de carga efetiva que o veículo consegue carregar, pois sabemos que com o passar do tempo, a bateria vai perdendo a sua capacidade de carga, não conseguindo carregar 100% de sua carga original.

Aqui é apresentada a média da frota, podendo ser plotada por cada veículo individualmente, conseguindo nos mostrar um panorama geral da frota e identificar se existe um veículo que já precisa ter a bateria reparada.

Depois é mostrado o índice de eficiência seguindo a mesma lógica do indicador anterior, primeiro plotando a média da frota e dando liberdade para realizar as plotagens individuais. Este indicador é referente ao índice de eficiência de recarga da bateria utilizando o freio regenerativo. Com esta função, o inversor do veículo é chaveado para recarregar a bateria toda vez que o freio é acionado, transformando a ação de frenagem em energia.

Este indicador é normalmente baixo, onde uma aplicação excelente consegue em torno de 20% de eficiência. Contudo, usando este indicador é possível identificar se existe algum veículo que está muito abaixo da média da frota e tomar as ações necessárias.

### 5.2.3 Responsividade

Toda a interface do sistema web também foi programada para garantir responsividade de sua apresentação. Isso significa que os conteúdos apresentados se adaptarão ao tamanho de tela em que o sistema está sendo apresentado, seja ele um monitor de desktop, notebook, tablet ou *smartphone*. Esta programação de apresentação de conteúdo que garante ao usuário sempre uma boa usabilidade do sistema independente do dispositivo que está utilizando, o que deixa a experiência mais agradável e dá uma perspectiva de agilidade, pois quando o cliente quer acessar seus dados, mas não está no escritório, pode simplesmente abrir o navegador do seu celular e o sistema vai ser apresentado de maneira funcional.

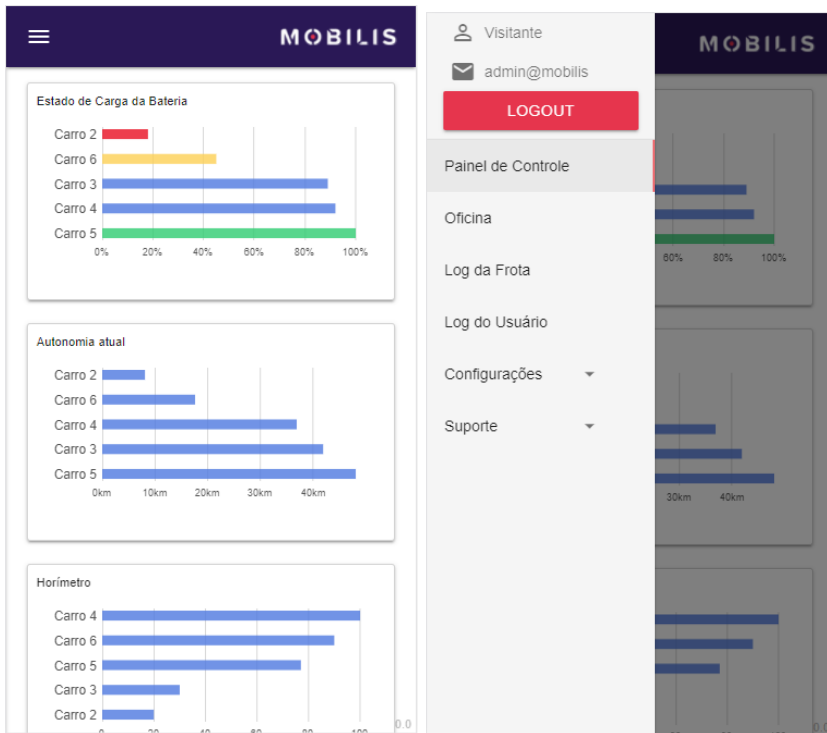
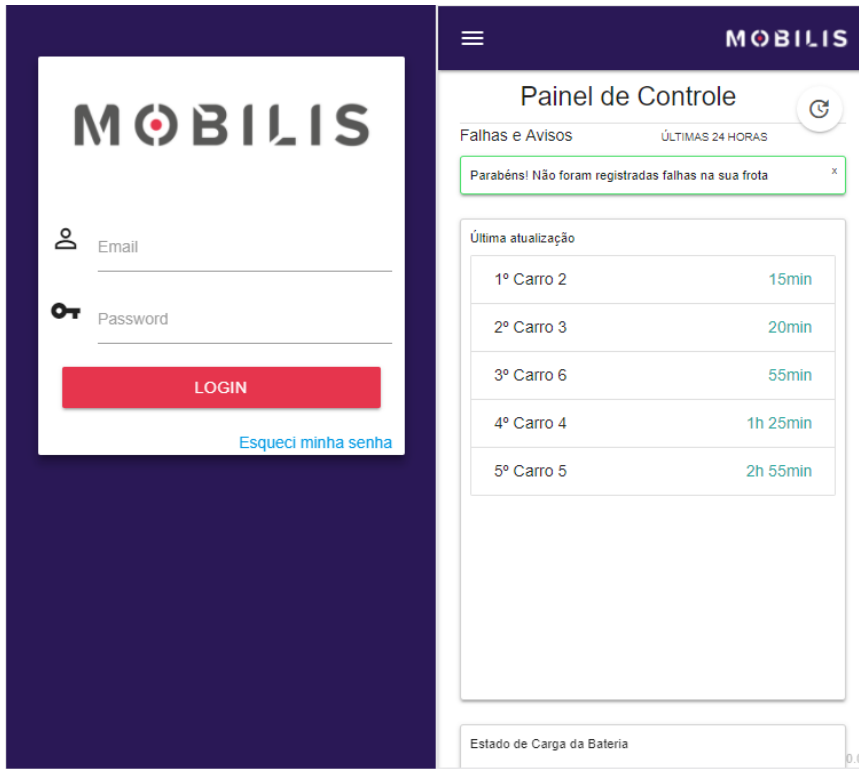
### 5.2.4 App Android

Aproveitando-se das propriedades de responsividade do sistema, também foi desenvolvido um segundo *App Android* para atender os clientes que queiram ter acesso ao sistema web em seu dispositivo mobile de maneira mais rápida.

Foi desenvolvido um aplicativo que renderiza uma *web view* na sua interface de apresentação onde carrega a URL de chamada do sistema web para o navegador padrão do dispositivo mobile e apresenta as informações sem que as barras, links e opções do navegador sejam apresentadas para o usuário, deixando a experiência igual a ao utilizar um App programado especialmente para mobile.

A apresentação do sistema na plataforma Android pode ser visualizada através da Figura 14.

Figura 14 - App Android



## 6. RESULTADOS

Após o desenvolvimento de todas as partes do sistema, conseguimos com sucesso realizar a integração dos diversos sistemas para que o nosso veículo se comunicasse com a internet de maneira fácil e que o acesso aos seus dados seja feito de maneira simples e intuitiva. Com este novo sistema, se torna fácil a análise de métricas medidas em testes do veículo e a análise de seu desempenho.

As validações do envio dos dados foram feitas a partir de testes onde o carro enviou dados gerados em um teste para o sistema com sucesso. Os gráficos apresentados na seção relativa ao Modo Engenharia foram colhidos a partir de um destes ensaios.

Como a empresa possui apenas um veículo em produção no momento, foram criados logs contendo informações fictícias de outros veículos para realizar os testes e criar lógicas e apresentações para o sistema de gerenciamento. A criação de dados fictícios também permitiu simular diferentes situações, conseguindo validar todos os comportamentos que eram esperados do sistema.

O sistema de gerenciamento de frota também foi apresentado para vários colaboradores e possíveis clientes para ter uma validação de conceito. Em todas as ocasiões, gerou impactos positivos nas pessoas e em muitas vezes foi demonstrado grande interesse na aquisição do mesmo.

Para demonstrações, o sistema já está online no endereço [www.mobilis.eco.br](http://www.mobilis.eco.br) e possuiu credenciais de login para acesso a visitantes. Esta versão de demonstração também utiliza de dados fictícios de uma frota de cinco veículos.





## 7. CONCLUSÃO E PERSPECTIVAS FUTURAS

O desenvolvimento deste projeto foi muito interessante pois ele promoveu a integração de diversas tecnologias para criar um produto inovador e com muito potencial. Foi também muito gratificante reconhecer que os conhecimentos adquiridos durante a faculdade tiveram um papel fundamental durante este desenvolvimento.

No final também conseguimos atender a todos os nossos objetivos, sendo capaz de enviar dados de um veículo elétrico utilizando adaptadores *Wi-Fi* para um servidor na internet e conseguir acesso a esses dados utilizando um portal na web. Conseguimos também acessar as métricas na íntegra, ou ter os dados apresentados como indicadores.

Há grande motivação para continuar evoluindo, aumentando a variedade e qualidade de dados lidos pelo veículo para que mais análises possam ser feitas e indicadores mais inteligentes criados. Com mais indicadores inteligentes, mais fácil se torna o gerenciamento de uma frota de veículos.

Pelo impacto positivo que este projeto gerou e potenciais compradores e conhecedores do mercado de veículos elétricos, a empresa possui planos de expandir o projeto para veículos elétricos industriais. Com esta adaptação, podemos oferecer a solução em gerenciamento de frota para qualquer indústria interessada no sistema.



## REFERÊNCIAS

FIELDING, Roy; RESCHKE, Julian. Hypertext transfer protocol (HTTP/1.1): Semantics and content. 2014.

FREIER, Alan; KARLTON, Philip; KOCHER, Paul. The secure sockets layer (SSL) protocol version 3.0. 2011.

VAN OSCH, Michiel; SMOLKA, Scott A. Finite-state analysis of the CAN bus protocol. In: **High Assurance Systems Engineering, 2001. Sixth IEEE International Symposium on**. IEEE, 2001. p. 42-52.

LI, H. Y. et al. A novel hardware architecture of CAN applications. In: **Consumer Electronics (ICCE), 2017 IEEE International Conference on**. IEEE, 2017. p. 279-280.

JONES, Michael; BRADLEY, John; SAKIMURA, Nat. **JSON web token (jwt)**. 2015.

LANSFORD, Jim; STEPHENS, Adrian; NEVO, Ron. Wi-Fi (802.11 b) and Bluetooth: enabling coexistence. **IEEE network**, v. 15, n. 5, p. 20-27, 2001.

MASSE, Mark. **REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces**. " O'Reilly Media, Inc.", 2011.

HOFFMAN, Paul. SMTP service extension for secure SMTP over transport layer security. 2002.