

**Controle do IoTDoc - documentação geral do projeto**

**Histórico de revisões**

| **Data** | **Autor** | **Versão** | **Resumo da atividade** |
| --- | --- | --- | --- |
| 17/10/2022 | Jackson Aguiar | 1.0 | Seção 1.1 - Parceiro de Negócio  Seção 1.2.1 - Problema |
| 17/10/2022 | Larissa Carvalho e André Luís | 1.1 | Seção 1.3 - Análise de Negócio  Seção 1.3.4 - Value Proposition Canvas |
| 17/10/2022 | Larissa Carvalho e André Luís | 1.2 | Seção 1.4 - Análise de Experiência do Usuário  Seção 1.4.1 - Personas |
| 18/10/2022 | Kaique Ramon | 1.3 | Seção 1.2.2 - Objetivos |
| 18/10/2022 | Larissa Carvalho e André Luís | 1.4 | Seção 1.4.2 - Jornada do Usuário (Gestor) |
| 18/10/2022 | Dayllan Alho e Gabriel Rocha | 1.5 | Seção 1.4.2 - Jornada do Usuário (Cliente) |
| 18/10/2022 | André, Larissa, Dayllan e Gabriel | 1.6 | Seção 1.4.3 - User Stories |
| 19/10/2022 | Gabriel Rocha | 1.7 | Seção 1.3.1 - Contexto da Indústria |
| 19/10/2022 | André, Dayllan, Gabriel, Jackson, Larissa e Kaique | 1.8 | Seção 1.3.2 - Análise SWOT  Seção 1.3.5 - Matriz de Riscos |
| 19/10/2022 | Larissa Carvalho e André Luís | 1.9 | Seção 1.3.3 - Planejamento Geral da Solução |
| 19/10/2022 | Dayllan Alho | 1.10 | Seção 2.1 - Arquitetura da Solução versão 1 |
| 26/10/2022 | Dayllan Alho e Larissa Carvalho | 2.0 | Seção 2.2 - Arquitetura da Solução versão 2 |
| 03/11/2022 | Dayllan Alho | 2.1 | Seção 3 - Casos de Uso 3.1.1. Entradas e Saídas por Bloco3.1.2. Entrada e Saída 13.1.3. Entrada e Saída 23.1.4. Entrada e Saída 33.1.5. Entrada e Saída 4 |
| 17/11/2022 | Gabriel Rocha | 2.2 | Seção 2.3 - Arquitetura da solução versão 3 |
| 18/11/2022 | Dayllan Alho | 2.3 | Alteração da Seção 3 |
| 02/12/2022 | Dayllan Alho | 2.4 | Toda a Seção 3 |
| 14/12/2022 | Dayllan Alho | 2.5 | Revisão completa |

**Sumário**

[**1. Definições Gerais**](#_heading=h.2et92p0) **3**

[1.1. Parceiro de Negócios (sprint 1)](#_heading=h.tyjcwt) 3

[1.2. Definição do Problema e Objetivos (sprint 1)](#_heading=h.3dy6vkm) 3

[1.2.1. Problema](#_heading=h.1t3h5sf) 3

[1.2.2. Objetivos](#_heading=h.4d34og8) 3

[1.3. Análise de Negócio](#_heading=h.2s8eyo1) 4

[1.3.1. Contexto da indústria (sprint 1)](#_heading=h.17dp8vu) 4

[1.3.2. Análise SWOT (sprint 1)](#_heading=h.3rdcrjn) 4

[1.3.3. Planejamento Geral da Solução (sprint 1)](#_heading=h.26in1rg) 4

[1.3.4. Value Proposition Canvas (sprint 1)](#_heading=h.lnxbz9) 4

[1.3.5. Matriz de Riscos (sprint 1)](#_heading=h.35nkun2) 4

[1.4. Análise de Experiência do Usuário](#_heading=h.1ksv4uv) 5

[1.4.1. Personas](#_heading=h.44sinio) 5

[1.4.2. Jornadas do Usuário e/ou Storyboard](#_heading=h.2jxsxqh) 5

[1.4.3. User Stories](#_heading=h.z337ya) 5

[1.4.4. Protótipo de interface com o usuário](#_heading=h.3j2qqm3) 6

[**2. Arquitetura da solução**](#_heading=h.4i7ojhp) **7**

[2.1. Arquitetura versão 1](#_heading=h.2xcytpi) 7E

[2.2. Arquitetura versão 2](#_heading=h.1ci93xb) 8

[2.3. Arquitetura versão 3](#_heading=h.3whwml4) 9

[**3. Casos de uso**](#_heading=h.2bn6wsx) **10**

[3.1. Entradas e Saídas por Bloco](#_heading=h.3as4poj) 10

[3.2. Interações](#_heading=h.2p2csry) 11

[**Anexos**](#_heading=h.147n2zr) **12**

# 1. Definições Gerais

## 1.1. Parceiro de Negócios

Mapear com precisão o deslocamento do carro até o cliente executando a estimativa de tempo de entrega. De modo que a solução possa transmitir informações do prisma, consequentemente do carro e do manobrista, em uma área fechada. A solução apresentará a distância do prisma estacionado até o local de busca do veículo pelo cliente, além de estimar o tempo de espera no deslocamento do automóvel.

## 

## 1.2. Definição do Problema e Objetivos

### 1.2.1. Problema

O ambiente que circunda a problemática é o posto valet, que tem a responsabilidade de receber e transitar os veículos alocados. Devido ao não uso de tecnologias de mapeamento, os stakeholders não conseguem mapear onde ele está com velocidade e precisão, além da ausência de dados como trajeto, fila de chegada, tempo de chegada, condutor e indicadores de produtividade, até mesmo se a logística adotada atende os requisitos. Dessa forma se torna complicado entender as necessidades daquele ambiente e traçar estratégias; tanto para favorecer esse tipo de negócio, quanto para ampliar outros segmentos que atraiam mais clientes para a companhia.

### 1.2.2. Objetivos

A solução proposta para o problema é desenvolver um hardware via wi-fi, que dê a estimativa de tempo e distância da vaga de estacionamento até o cliente. Além disso, nosso objetivo é verificar quem é o manobrista responsável por fazer esta manobra e produzir dashboard de produtividade de cada manobrista cadastrado em nosso sistema. Uma forma de descobrir quem é o manobrista que está estacionando o carro é dar um cartão NFC para ele e quando aproximar do prisma irá mandar para a central através do wi-fi quem é a pessoa que está manobrando o carro. Deste modo, haverá um controle por meio de um gráfico, informando os manobristas que estão pegando mais e menos carros. No momento em que o prisma não estiver sendo utilizado vai ter uma base de carregamento magnético e sem fio para não ter problemas de encaixe errado ou até mesmo de esquecer.

## 1.3. Análise de Negócio

### 1.3.1. Contexto da indústria

**Análise da Indústria:**

Embora o mercado de gestão de estacionamento tenha 15% menos frequência quando comparado aos níveis pré pandêmicos, segundo a Mordor Intelligence, esse mercado está avaliado em cerca de 3,5 bilhões de dólares e projetado para valer 6,4 bilhões de dólares em 2026. Então, ainda que não muito notório, esse é um ramo importante para economia brasileira.

**Os principais players do mercado:**

Estapar;

Indigo;

Pare Bem.

**Modelo de negócio:**

A Estapar é a principal empresa de gestão de estacionamento no Brasil. Não obstante, atua em toda área de negócio envolvendo vagas. Entre elas, concessões on-street e off-street, aluguel e gerenciamento, contratos de longa data e propriedades off-street. Além disso, a Estapar tem compromissos com a mobilidade urbana no off-street e com a conectividade automotiva.

**Tendências:**

As principais tendências do mercado de estacionamentos convergem para grande uso de tecnologia. São notórios: disponibilização de vagas com carregadores, cadastro automatizado dos carros, procura digital por vagas e parceria com empresas de aluguel diário de carros. Logo, estacionamentos conectados criarão vantagens competitivas em breve.

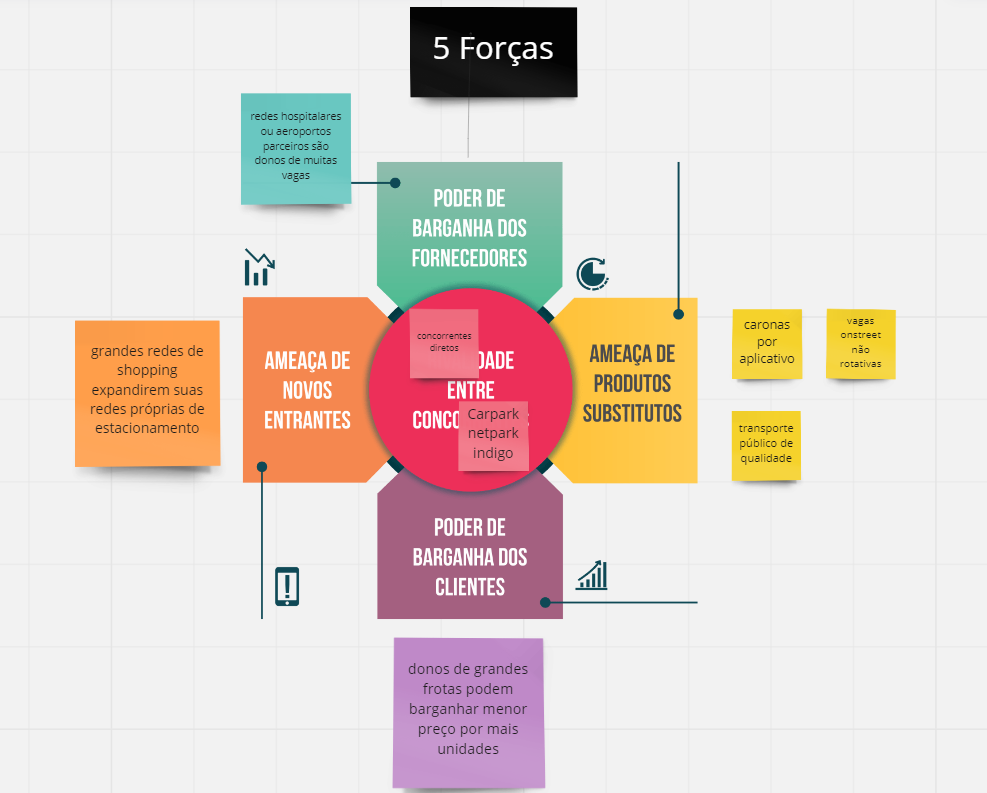


Figura 1: Modelo de 5 forças de Porter. Disponível em: <https://github.com/2022M4T4-Inteli/Projeto3/blob/main/Documentos/Outros/5%20Forcas%20Poter.pdf>

### 1.3.2. Análise SWOT



Figura 2: SWOT.

### 1.3.3. Planejamento Geral da Solução

### 1.3.3.1. Quais os objetivos da solução

A solução trata-se de um Hardware , o qual tem como objetivo o monitoramento e controle dos veículos nos estacionamentos da Estapar, mais especificamente, um sistema que possa identificar quem é o condutor que está estacionando cada veículo e o tempo da trajetória para estacionar e buscar o carro. Com isso, será possível os gestores terem uma análise estatística sobre a produtividade de cada manobrista, tendo acesso a relatórios mais precisos. Além disso, os gestores, com essa solução, poderão informar os seus clientes sobre o tempo de espera estimado dos seus veículos.

### 1.3.3.2. Quais os os dados disponíveis (fonte e conteúdo - exemplo: dados da área de Compras da empresa descrevendo seus fornecedores)

Atualmente, a Estapar possui ausência de dados como trajeto, fila de chegada, tempo de chegada, condutor e indicadores de produtividade. Logo, com o objetivo de ter-se informações mais precisas e um maior controle dos veículos nos estacionamentos, a nossa solução precisará dos seguintes dados: placa do veículo, nome do manobrista, nome do cliente, tempo de execução de estacionar e buscar um carro, status da localização do carro e a quantidade de carros estacionados por cada manobrista diariamente.

### 1.3.3.3. Qual a solução proposta (visão de negócios)

A solução proposta será um Hardware, que organizará os dados de controle e monitoramento do veículo. Assim, será viável a criação de indicadores de produtividade sobre cada manobrista, informando os gestores sobre o tempo de execução individual na realização das tarefas diárias. Além de aumentar a confiabilidade dos clientes.

**1.3.3.4. Como a solução proposta pretende ser utilizada**

A solução proposta será utilizada dentro dos estacionamentos Estapar, o Hardware ficará dentro de um prisma, o qual será colocado em todos os veículos que utilizarem o serviço. O Hardware, primeiramente, terá uma função de identificar o manobrista que conduzirá o veículo até a vaga e o mesmo procedimento na retirada. Além disso, será possível calcular o tempo do trajeto até a vaga, assim como o tempo de busca do carro para a devolução do veículo para o cliente.

Os manobristas utilizarão o Hardware diariamente para monitoramento de suas atividades, os gestores irão utilizar para acompanhar as atividades e a produtividade dos manobristas e os clientes poderão ter uma estimativa do tempo para receber o carro.

**1.3.3.5. Quais os benefícios trazidos pela solução proposta**

Os benefícios trazidos pela solução proposta são os seguintes: os gestores terão informações mais precisas sobre o tempo de execução de cada manobrista e quem é o condutor de cada veículo, através de um painel e, com isso, poderão criar um dashboard de produtividade. Além disso, outro benefício é criar uma relação de mais confiança com o cliente, que poderá acompanhar o seu veículo. Já os manobristas, terão a possibilidade de receber melhores feedbacks através dos relatórios de produtividade.

**1.3.3.6. Qual será o critério de sucesso e qual medida será utilizada para o avaliará**

Os critérios de sucesso serão estabelecidos, por parte do Hardware, a partir do cumprimento com as funções atribuídas nas User Stories e a conexão com o Software. Deve ser garantido o envio dos relatórios com os dados e deve ser acessível para os favorecidos pelo projeto; manobrista, gestor e cliente.

### 1.3.4. Value Proposition Canvas

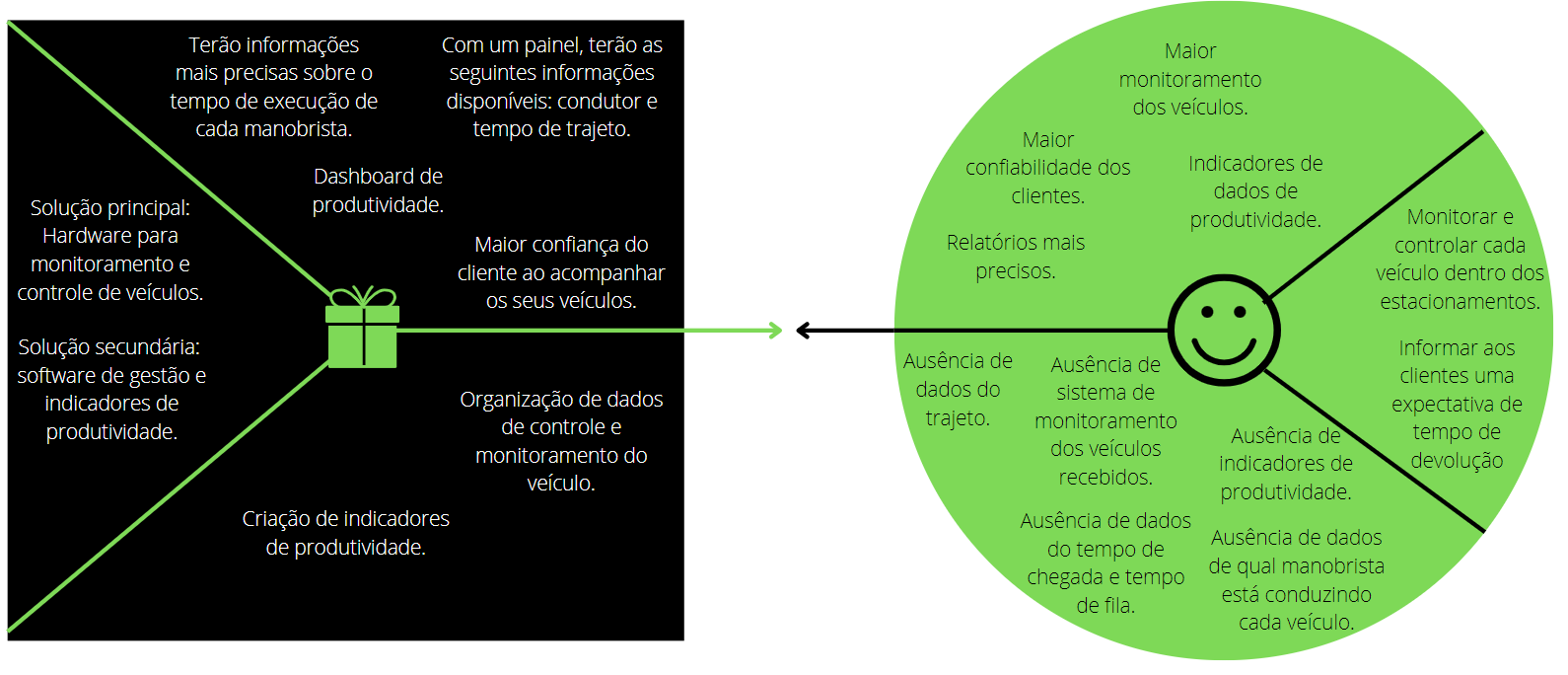


Figura 3: Value canvas proposition. Disponível em: <https://github.com/2022M4T4-Inteli/Projeto3/blob/main/Documentos/Outros/CANVAS.pdf>

### 1.3.5. Matriz de Riscos





Análise de Risco. Disponível em: <https://github.com/2022M4T4-Inteli/Projeto3/blob/main/Documentos/Outros/Matriz%20de%20Risco%20-%20IoT%20ESTAPAR%20-%20Sheet1.pdf>

## 

## 

## 

## 

## 1.4. Análise de Experiência do Usuário (sprints 1 e 2)

### 1.4.1. Personas



**Nome**: Edvaldo Braga.

**Idade**: 52 anos.

**Gênero**: Masculino.

**Ocupação**: Gestor de estacionamento.

**Resumo que define a persona:** “Sempre gostei da gestão, hoje posso gerenciar uma grande rede de estacionamento”.

**Considerações biográficas e comportamentais:**

- Nasceu em São Paulo e é formado em administração.

- Seu pai era dono de um mercado.

- Decidiu seguir a carreira de gestor, pois desde pequeno auxiliava seu pai com as planilhas e relatórios do mercado em que trabalhava.

- Adora entender e interpretar gráficos, planilhas e tabelas.

- Gosta de estar sempre informado e de ter relatórios precisos.

**Dores/motivações atuais com o problema:**

**-** Atualmente há uma ausência de sistema de controle e monitoramento dos veículos que recebem nos estacionamentos.

- Não consegue mapear onde os manobristas estão, qual a velocidade e precisão.

- Possui ausência de dados como trajeto, fila de chegada, tempo de chegada, condutor e indicadores de produtividade.

- Atualmente não possui monitoramento dos carros em tempo real, então não possui a informação do tempo de trajeto e não sabe qual manobrista está conduzindo cada veículo.

- Não consegue fazer uma análise estatística de como está a produtividade de cada manobrista.

- Incerteza ao informar os clientes sobre o tempo de chegada de seu veículo.

**Objetivos/necessidades específicas em relação ao problema:**

**-** Precisa de uma solução que monitore cada veículo.

- Precisa construir relatórios de produtividade da equipe.

- Necessita de um painel, com a informação de qual manobrista está dirigindo cada carro.

- Necessita de um painel, com a informação do tempo de execução dos veículos em movimento.



**Nome:** Adriana Silva.

**Idade:** 35 anos.

**Gênero:** Feminino.

**Ocupação:** Pediatra.

**Resumo que define a persona:** "O choro das crianças cessa por sua causa, quando você cuida delas com tanto amor.”

**Considerações biográficas e comportamentais:**

* Nascida em São Paulo e formada na USP.
* Ama trabalhar com crianças.
* Quando adolescente realizava serviços como babá.
* Tem uma jornada de trabalho massiva.

**Dores/Motivações atuais com o problema:**

* Sua jornada de trabalho é exaustiva, estando muito cansada ao ir para casa, ainda tem que esperar muito tempo pelo manobrista.
* Não possui a informação de quanto tempo levará para receber o seu carro.

**Objetivos/necessidades específicas em relação ao problema:**

* Necessita de uma solução para pedir a retirada do carro.
* Necessita saber quanto tempo tem que esperar para receber o carro.
* Precisa otimizar seu tempo na volta para casa

### 1.4.2. Jornadas do Usuário e/ou Storyboard

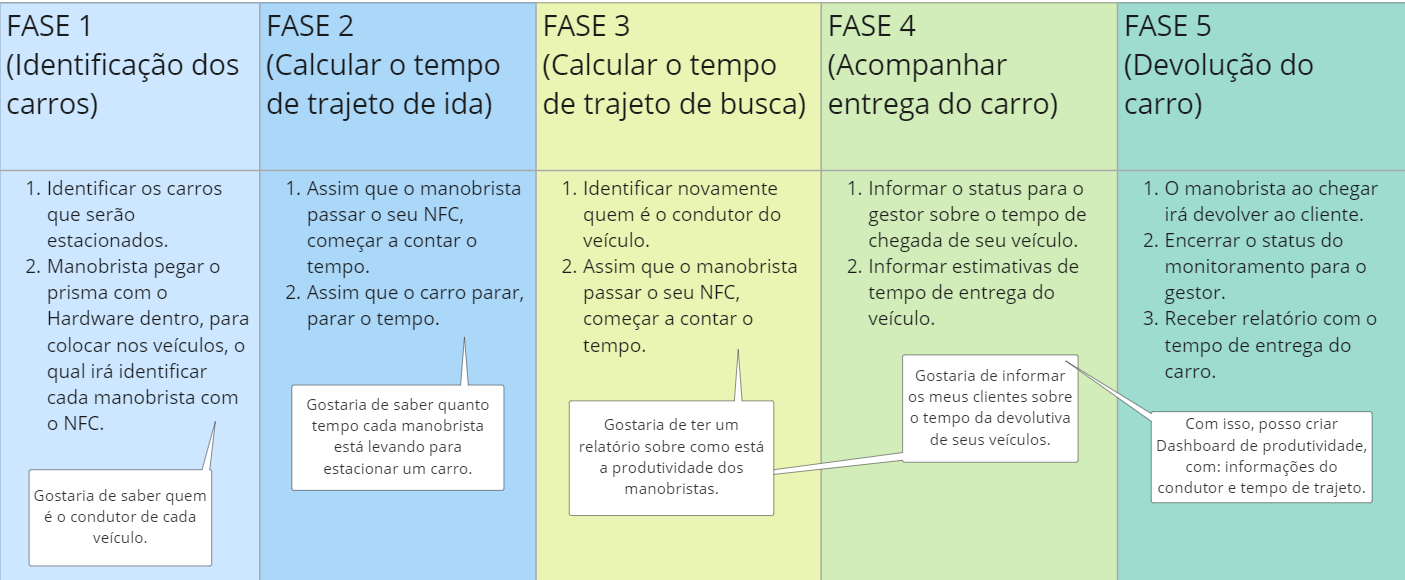
A jornada do usuário é uma forma de compreender todas as fases de interação que o cliente tem com o produto ou algum serviço, ou seja, quando se realiza o mapeamento detalhado de toda experiência possível que o usuário poderá ter durante a utilização do que está sendo proposto. Facilitando assim analisar todas as variáveis envolvidas.

Utilizamos o gestor do estacionamento Edvaldo Braga para nossa Jornada do usuário, conforme abaixo:

**Edvaldo Braga:**

**Cenário:** Edvaldo Braga precisa de uma ferramenta de tecnologia que monitore e controle cada veículo dentro dos estacionamentos, como: trajeto, fila de chegada, tempo de chegada, condutor e indicadores de produtividade.

**Expectativas:** Edvaldo espera melhorar a produtividade e ter um maior controle sobre os carros estacionados.



**Oportunidades:** As oportunidades serão que os gestores dos estacionamentos terão uma melhor análise estatística sobre o tempo de execução de tarefa de cada manobrista, além de um maior controle, tendo a informação de quem é o condutor de cada veículo. Com isso, será possível criar dashboards precisos e assim, aumentar a produtividade.

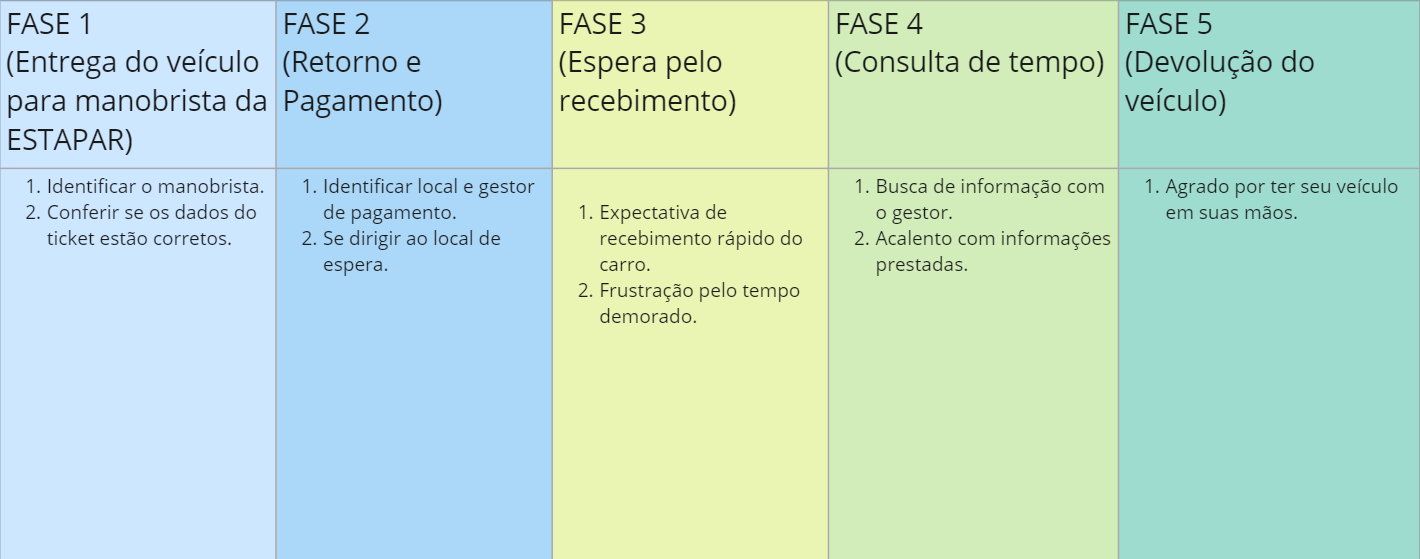
**Responsabilidades:** A responsabilidade será atribuída para os manobristas que irão utilizar o Hardware diariamente em seus serviços. Já o Hardware é responsável por realizar todo o processo de monitoramento e controle dos veículos dos estacionamentos.

Utilizamos também o usuário final para nossa Jornada do usuário, conforme abaixo:

**Adriana Silva:**

**Cenário:** Adriana Silva se sente muito desconfortável sempre que termina seu dia, pois ela sempre se depara com a incerteza do tempo que conseguirá sair do estacionamento do hospital em que trabalha.

**Expectativas:** Adriana espera receber informações, como o tempo de espera.



**Oportunidades:** As oportunidades serão que os clientes do estacionamento terão melhores informações, principalmente, sobre o tempo de espera, há a possibilidade de organizar e agendar seus compromissos e afazeres.

**Responsabilidades:** Respeitar o tempo de devolução e entender que donos de carros que estão estacionados mais perto terão que aguardar menos.

### 1.4.3. User Stories

Para conferir é uma sentença que indica as dores de negócio que estão vinculadas ao usuário final e ao usuário do sistema, seu papel é capturar a função do usuário de dimensionar o tamanho, a prioridade e o status de funcionalidades para a equipe de desenvolvimento.

| **Nº** | **DESCRIÇÃO** | **TAMANHO** | **PRIORIDADE** | **STATUS** |
| --- | --- | --- | --- | --- |
| 1 | Eu como gestor de operações do estacionamento quero criar dashboard de produtividade dos manobristas para melhorar o monitoramento de cada um na execução de suas tarefas. | média | média | concluído |
| 2 | Eu como gestor de operações, quero informar ao cliente final o tempo de espera, para melhor informá-lo sobre o status do seu veículo. | grande | alta | concluído |
| 3 | Eu como gestor de operações, quero identificar quem é o condutor de cada carro, para ter um maior controle. | grande | alta | concluído |
| 4 | Eu como gestor de operações, quero ter um relatório sobre o tempo de execução de cada manobrista, para ter um melhor monitoramento de cada um. | grande | alta | concluído |
| 5 | Eu como gestor de operações quero um hardware que seja resistente ao tempo atmosférico, para que nosso produto tenha maior durabilidade | grande | média | concluído |
| 6 | Eu como cliente, quero saber o status do meu carro para entender quanto tempo demorará para ser entregue. | média | baixa | concluído |
| 7 | Eu como cliente, quero solicitar a retirada do carro por aplicativo, para que eu não me desloque e tenha que esperar muito tempo. | média | baixa | concluído |
| 8 | Eu como cliente, quero saber quanto tempo vai demorar para ser entregue, para reduzir o tempo de espera. | média | baixa | concluído |

### 

### 1.4.4. Protótipo de interface com o usuário

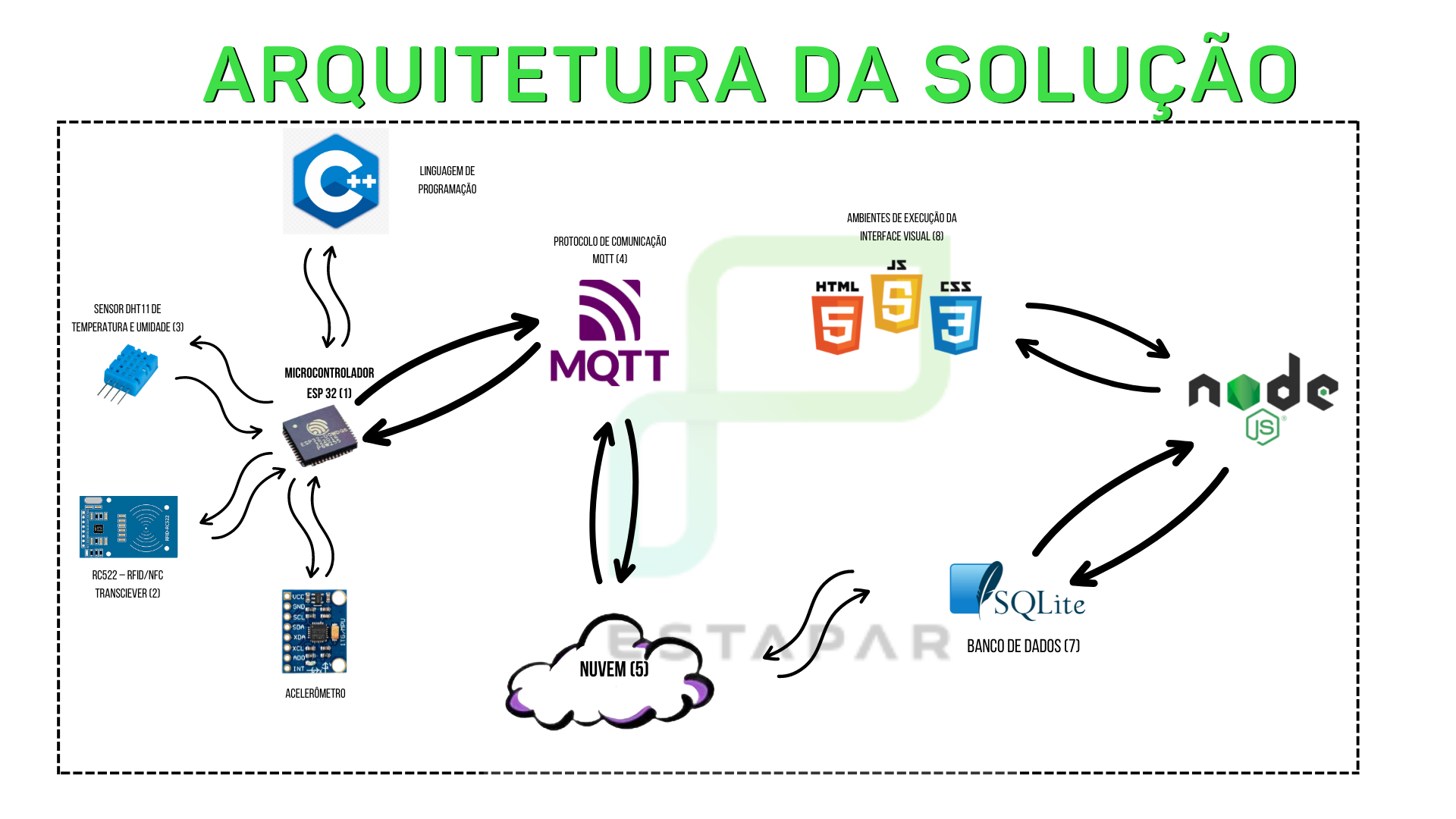
### No [link](https://www.figma.com/file/57egcBXsLLfKvsfrPdsTOE/ESTAPAR?node-id=8%3A244) é possível conferir o protótipo da interface do usuário.

# 2. Arquitetura da solução

## 2.1. Arquitetura versão 1 (sprint 1)

A arquitetura da solução é uma forma de representação gráfica que visa a criação de uma estrutura e traçar caminhos para realização da solução de um problema. Nessa estrutura, o foco é demonstrar não só as ferramentas e frameworks utilizados para o sucesso do projeto, mas também as duas interações.

Abaixo há a representação gráfica da arquitetura.



Abaixo há a representação de cada componente da arquitetura da solução, juntamente com seu nome, sua função e utilidade no projeto.

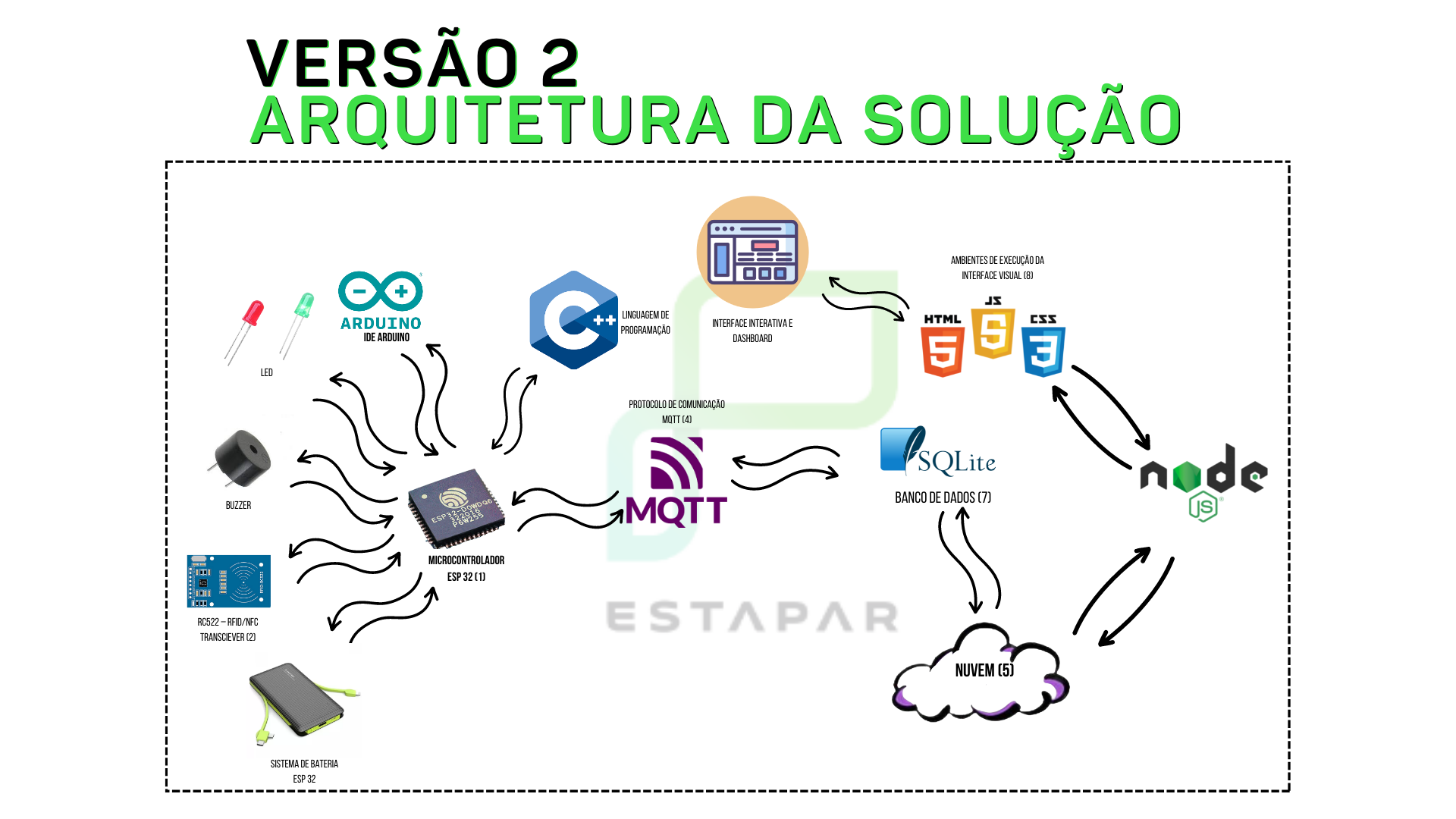
| Representação | Nome | Link | Função | Utilidade |
| --- | --- | --- | --- | --- |
|  | Microcontrolador ESP32 | N/A | O módulo ESP32 é um módulo Wi-Fi de alta performance, com um baixíssimo consumo de energia. | Integrar os sensores implantados e programados para captar dados essenciais para o negócio. |
|  | Protocolo de Comunicação | <https://mqtt.org/> | Protocolo de mensagens padrão OASIS para a Internet das Coisas (IoT). | Ele foi projetado como um transporte de mensagens de publicação/assinatura extremamente leve, ideal para conectar dispositivos remotos com um pequeno espaço de código e largura de banda de rede mínima. O MQTT hoje é usado em uma ampla variedade de indústrias, como automotiva, manufatura, telecomunicações, petróleo e gás, etc. |
|  | Html | <https://developer.mozilla.org/pt-BR/> | Linguagem de marcação utilizada para estruturar os elementos da página, como parágrafos, links, títulos, tabelas, imagens e até vídeos. | Utilizado para gerar as páginas web do site. |
|  | Css | <https://developer.mozilla.org/pt-BR/docs/Web/CSS> | Linguagem de marcação utilizada para a formatação de textos, imagens e outros tipos de arquivo. | Utilizado para editar, alinhar e personalizar os arquivos HTML utilizados ao longo da aplicação web. |
|  | JavaScript | <https://developer.mozilla.org/en-US/docs/Web/JavaScript> | Linguagem de programação amplamente utilizada para implementação de itens, atualizações e personalização de páginas web. | Utilizado para a personalização e implementação das páginas em HTML com o banco de dados. |
|  | SQLite | <https://www.sqlite.org/index.html> | Biblioteca em processo que implementa um mecanismo de banco de dados SQL transacional independente, sem servidor e sem configuração. | Utilizado como biblioteca, na qual os desenvolvedores de software incorporam seus aplicativos |
|  | Node js | <https://expressjs.com/> | Framework rápido e utilizado em conjunto com o Node. js, facilitando no desenvolvimento de aplicações back-end e até, em conjunto com sistemas de templates, aplicações full-stack. Escrito em JavaScript, o Express. | Utilizado como runtime environment. |
|  | Sensor DHT11 de umidade e temperatura | N/A | O DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%, muito usado para projetos com Arduino. | Utilizado para previsão de deterioração ao longo do tempo. |
|  | RFID/NFC | N/A | Emprega sinal de rádio para os mais variados tipos de identificação e rastreamento. | Utilizada para identificar o prisma como único. |
|  | Acelerômetro | N/A | dispositivo usado para medir a aceleração própria de um sistema. | Utilizado para medir a aceleração do veículo em todo o deslocamento. |
|  | Linguagem de Programação - C++ | <https://learn.microsoft.com/pt-br/cpp/windows/latest-supported-vc-redist?view=msvc-170> | linguagem de programação compilada multi-paradigma e de uso geral. | Utilizado como ferramenta de programação visando correta configuração do microcontrolador. |

## 

## 2.2. Arquitetura versão 2

A arquitetura da solução é uma forma de representação gráfica que visa a criação de uma estrutura e traçar caminhos para realização da solução de um problema. Nessa estrutura, o foco é demonstrar não só as ferramentas e frameworks utilizados para o sucesso do projeto, mas também as duas interações.

Abaixo há a representação gráfica da arquitetura.



Para facilitar a visualização da arquitetura da solução versão 02, é possível acessá-la no [link](https://github.com/2022M4T4-Inteli/Projeto3/blob/main/Documentos/Outros/V02%20-%20Arquitetura%20da%20Solu%C3%A7%C3%A3o.pdf).

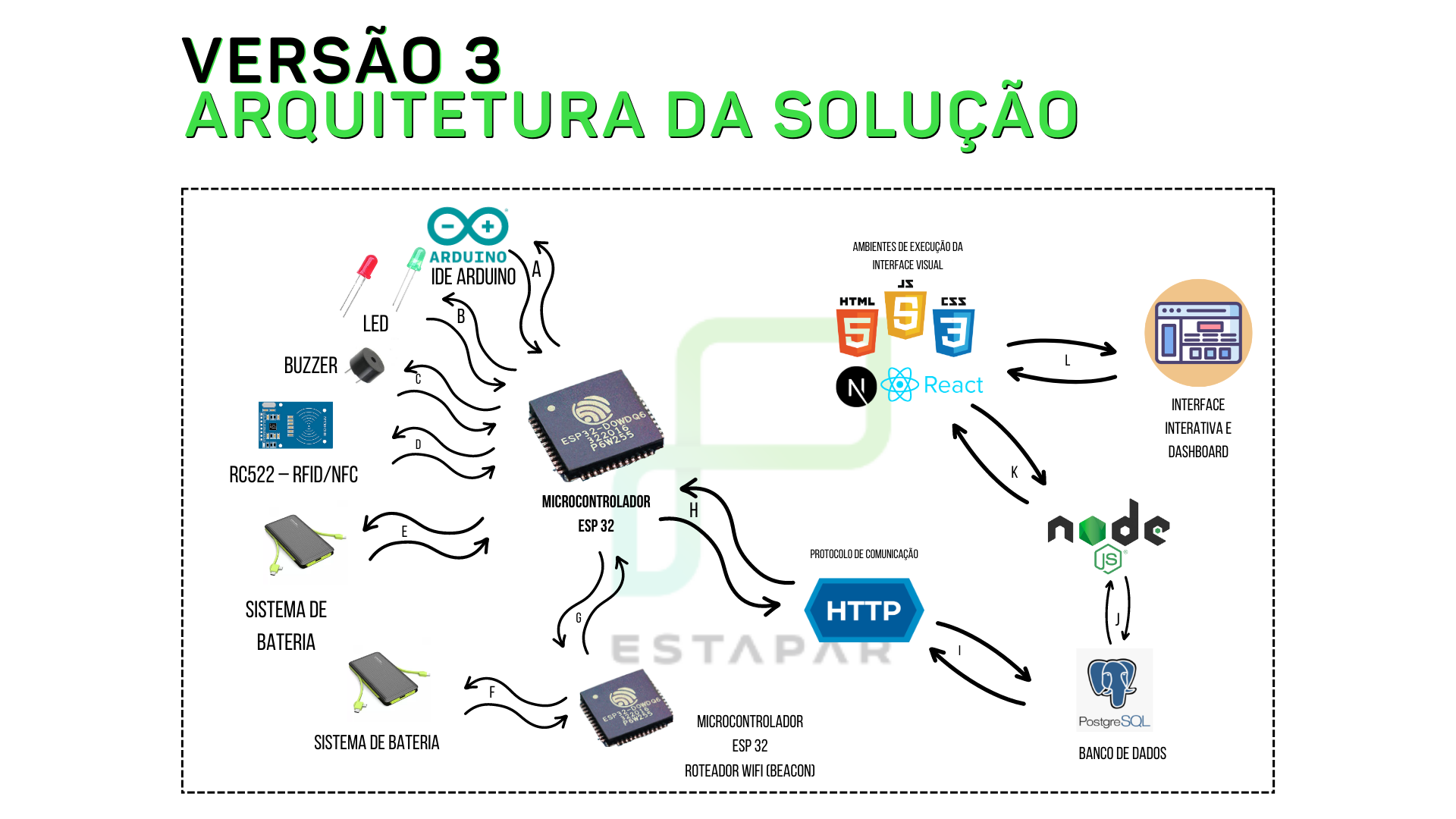
| Representação | Nome | Link | Função | Utilidade |
| --- | --- | --- | --- | --- |
|  | Microcontrolador ESP32 | N/A | O módulo ESP32 é um módulo Wi-Fi de alta performance, com um baixíssimo consumo de energia. | Integrar os sensores implantados e programados para captar dados essenciais para o negócio. |
|  | Protocolo de Comunicação | <https://mqtt.org/> | Protocolo de mensagens padrão OASIS para a Internet das Coisas (IoT). | Ele foi projetado como um transporte de mensagens de publicação/assinatura extremamente leve, ideal para conectar dispositivos remotos com um pequeno espaço de código e largura de banda de rede mínima. O MQTT hoje é usado em uma ampla variedade de indústrias, como automotiva, manufatura, telecomunicações, petróleo e gás, etc. |
|  | Html | <https://developer.mozilla.org/pt-BR/> | Linguagem de marcação utilizada para estruturar os elementos da página, como parágrafos, links, títulos, tabelas, imagens e até vídeos. | Utilizado para gerar as páginas web do site. |
|  | Css | <https://developer.mozilla.org/pt-BR/docs/Web/CSS> | Linguagem de marcação utilizada para a formatação de textos, imagens e outros tipos de arquivo | Utilizado para editar, alinhar e personalizar os arquivos HTML utilizados ao longo da aplicação web. |
|  | JavaScript | <https://developer.mozilla.org/en-US/docs/Web/JavaScript> | Linguagem de programação amplamente utilizada para implementação de itens, atualizações e personalização de páginas web. | Utilizado para a personalização e implementação das páginas em HTML com o banco de dados. |
|  | SQLite | <https://www.sqlite.org/index.html> | Biblioteca em processo que implementa um mecanismo de banco de dados SQL transacional independente, sem servidor e sem configuração. | Utilizado como biblioteca, na qual os desenvolvedores de software incorporam seus aplicativos |
|  | Node js | <https://expressjs.com/> | Framework rápido e utilizado em conjunto com o Node. js, facilitando no desenvolvimento de aplicações back-end e até, em conjunto com sistemas de templates, aplicações full-stack. Escrito em JavaScript, o Express. | Utilizado como runtime environment. |
|  | RFID/NFC | N/A | Emprega sinal de rádio para os mais variados tipos de identificação e rastreamento. | Utilizada para identificar o prisma como único. |
|  | Linguagem de Programação - C++ | <https://learn.microsoft.com/pt-br/cpp/windows/latest-supported-vc-redist?view=msvc-170> | linguagem de programação compilada multi-paradigma e de uso geral. | Utilizado como ferramenta de programação visando correta configuração do microcontrolador. |
|  | IDE Arduino | <https://www.arduino.cc/en/software> | O Arduino IDE 2.0 é uma melhoria do IDE clássico, com desempenho aprimorado, interface de usuário aprimorada e muitos novos recursos, como preenchimento automático, um depurador integrado e sincronização de esboços com o Arduino Cloud . | Utilizado para programar a parte física e lógica do projeto. |
|  | Led | N/A |  | Utilizado para melhor interação do usuário com o hardware, indicando mau funcionamento, aparelho ligado ou desligado e leitura correta dos cartões com NFC. |
|  | Interface interativa e dashboard | N/A |  | Prototipada no FIGMA, é uma interface que auxilia a ter a melhor visualização dos dados que estão sendo implementados no hardware. |
|  | PowerBank | N/A | O power Bank , é uma bateria portatil que carrega um ou mais dispositivos | Utilizado para abastecer o hardware de forma energética. Como fonte de alimentação. |
|  | Buzzer | N/A | Um buzzer é um dispositivo de sinalização de áudio. | Utilizado para indicar sinais sonoros para melhor compreensão do usuário. |

## 

## 2.3. Arquitetura versão 3

A arquitetura da solução é uma forma de representação gráfica que visa a criação de uma estrutura e traçar caminhos para realização da solução de um problema. Nessa estrutura, o foco é demonstrar não só as ferramentas e frameworks utilizados para o sucesso do projeto, mas também as duas interações.

Abaixo há a representação gráfica da arquitetura.



Para facilitar a visualização da arquitetura da solução versão 03, é possível acessá-la no [link.](https://github.com/2022M4T4-Inteli/Projeto3/blob/main/Documentos/Outros/V03_Arquitetura%20da%20Solu%C3%A7%C3%A3o.pdf)

| Representação | Nome | Link | Função | Utilidade |
| --- | --- | --- | --- | --- |
|  | IDE Arduino | <https://www.arduino.cc/en/software> | O Arduino IDE 2.0 é uma melhoria do IDE clássico, com desempenho aprimorado, interface de usuário aprimorada e muitos novos recursos, como preenchimento automático, um depurador integrado e sincronização de esboços com o Arduino Cloud . | Utilizado para programar a parte física e lógica do projeto. |
| A | Ligação A | N/A | Conectar o Arduino IDE à entrada COM do microcontrolador ESP32. | Fazer o upload do código construído no Arduino IDE. |
|  | Led | N/A |  | Utilizado para melhor interação do usuário com o hardware, indicando mau funcionamento, aparelho ligado ou desligado e leitura correta dos cartões com NFC. |
| B | Ligação B | N/A | Ligar as luzes LED conectadas ao microcontrolador ESP32. | A luz azul indica que o RFID/NFC está ligado e a luz verde acende quando o cartão for reconhecido. |
|  | Buzzer | N/A | Um buzzer é um dispositivo de sinalização de áudio. | Utilizado para indicar sinais sonoros para melhor compreensão do usuário. |
| C | Ligação C | N/A | Ligar o buzzer conectado ao microcontrolador ESP32. | O buzzer apita quando cartão é lido pelo RFID/NFC. |
|  | RFID/NFC | N/A | Emprega sinal de rádio para os mais variados tipos de identificação e rastreamento. | Utilizada para identificar o prisma como único. |
| D | Ligação D | N/A | Ligar o RFID/NFC conectado ao microcontrolador ESP32. | Ler o cartão de cada manobrista antes da manobra. |
|  | PowerBank | N/A | O power Bank , é uma bateria portatil que carrega um ou mais dispositivos | Utilizado para abastecer o hardware de forma energética. Como fonte de alimentação. |
| E | Ligação E | N/A | Conectar o sistema de bateria à entrada USB do microcontrolador ESP32. | Fornecer energia ao microcontrolador ESP32. |
| F | Ligação F | N/A | Conectar o sistema de bateria à entrada USB do beacon. | Fornecer energia ao beacon. |
|  | Microcontrolador ESP32  Roteador WiFi (beacon) | N/A | O módulo ESP32 é um módulo Wi-Fi de alta performance, com um baixíssimo consumo de energia. | Integrar os sensores implantados e programados para captar dados essenciais para o negócio. |
| G | Ligação G | N/A | Conexão wireless entre o beacon e o microcontrolador ESP32. | Fornecer o dado da distância entre o prisma e o roteador WiFi. |
|  | Microcontrolador ESP32 | N/A | O módulo ESP32 é um módulo Wi-Fi de alta performance, com um baixíssimo consumo de energia. | Integrar os sensores implantados e programados para captar dados essenciais para o negócio. |
|  | Protocolo de Comunicação | <https://mqtt.org/> | Protocolo de mensagens padrão para desenvolvimento Web. | Utilizado para envio de dados coletados via ESP para o servidor Web. |
|  | Next JS | <https://nextjs.org/> | Next.js é um framework para React. | Utilizamos para desenvolver a aplicação Web em react |
|  | React | <https://pt-br.reactjs.org/> | O React é a biblioteca mais popular do JavaScript e é usada para construir uma interface de usuário (IU). | Foi utilizada para oferecer uma resposta excelente para o usuário adicionar comandos usando um novo método de renderizar sites. |
| H | Ligação H | N/A | Conecta o microcontrolador ESP32 ao protocolo de comunicação. | Manter o padrão de fornecimento de dados dos microcontroladores presentes nos prismas. |
| I | Ligação I | N/A | Conectar protocolo de comunicação e banco de dados. | Guardar as informações recebidas em formato padrão no banco de dados. |
|  | PostgreSQL | https://www.postgresql.org/ | Biblioteca em processo que implementa um mecanismo de banco de dados SQL transacional independente, sem servidor e sem configuração. | Utilizado como biblioteca, na qual os desenvolvedores de software incorporam seus aplicativo. |
| J | Ligação J | N/A | Conectar banco de dados ao node.js. | Fornecer o banco de dados para codificação em JavaScript. |
|  | Node js | <https://expressjs.com/> | Framework rápido e utilizado em conjunto com o Node. js, facilitando no desenvolvimento de aplicações back-end e até, em conjunto com sistemas de templates, aplicações full-stack. Escrito em JavaScript, o Express. | Utilizado como runtime environment. |
| K | Ligação K | N/A | Conectar o node.js aos ambientes de execução da interface visual. | Fornecer o código dos dados à interface visual. |
|  | Html | <https://developer.mozilla.org/pt-BR/> | Linguagem de marcação utilizada para estruturar os elementos da página, como parágrafos, links, títulos, tabelas, imagens e até vídeos. | Utilizado para gerar as páginas web do site. |
|  | Css | <https://developer.mozilla.org/pt-BR/docs/Web/CSS> | Linguagem de marcação utilizada para a formatação de textos, imagens e outros tipos de arquivo | Utilizado para editar, alinhar e personalizar os arquivos HTML utilizados ao longo da aplicação web. |
|  | JavaScript | <https://developer.mozilla.org/en-US/docs/Web/JavaScript> | Linguagem de programação amplamente utilizada para implementação de itens, atualizações e personalização de páginas web. | Utilizado para a personalização e implementação das páginas em HTML com o banco de dados. |
| L | Ligação L | N/A | Conectar, já integrados, front-end e back-end à interface. | Fornecer o conteúdo final ao usuário. |
|  | Interface interativa e dashboard | N/A |  | Prototipada no FIGMA, é uma interface que auxilia a ter a melhor visualização dos dados que estão sendo implementados no hardware. |

# 3. Casos de uso

## 3.1. Entradas e Saídas por Bloco

As entradas e saídas esperadas neste momento do projeto são as leituras dos atuadores e sensores para verificar o funcionamento do sistema, sendo assim, esperamos que ao encostar o cartão do RFID e o tag, o led verde pisque para identificar o cartão presente e acende, juntamente, com o toque do buzzer ao identificar o RFID.

Abaixo explicamos passo a passo de cada entrada e saída esperada.

## 3.1.2. Entrada e Saída RFID

O RFID é um dispositivo que identifica um sinal de rádio para um cartão ou uma tag em nosso projeto. No sistema construído no bloco central versão 1, há o RFID que é a parte central da aplicação para a ESTAPAR, ele identifica o cartão aproximado e manda uma saída no console, reconhecendo que o cartão ou tag foi identificado.

## 3.1.3. Saída led verde

O led verde identifica a aproximação do cartão ou tag, piscando quando for aproximado e quando o RFID ler sua aproximação, a luz acende por alguns segundos, indicando que o código dentro do cartão ou da tag está correto e o prisma o reconheceu.

## 3.1.4. Saída led azul

O led azul é um funcionalidade implementada visando um melhor entendimento sobre o prisma e se este está ligado ou não. Quando o prisma estiver recebendo uma carga, seja da bateria quanto de um cabeamento, a luz azul imediatamente acenderá, indicando que há corrente passando pelos circuitos e orientando que todos os componentes estão funcionando perfeitamente. Além disso, apesar de ficar durante todo o tempo ligado, não há um gasto energético elevado, pois a potência do led é baixa em comparação ao da bateria utilizada.

## 3.1.5. Saída led vermelho

O led vermelho possui duas funcionalidades principais neste momento do projeto, sendo elas:

1. Indicar se há algum erro ao reconhecer os componentes no momento de ligar o equipamento, juntamente, alertar as elevadas temperaturas e alta umidade do dispositivo, com o auxílio de sensores que serão implementados futuramente;
2. Indicar sinais de erro ao se conectar via WiFi.

Assim, este led, quando estiver piscando, será de suma importância para alertar o usuário sobre o funcionamento anormal do dispositivo IoT.

## 3.1.6. Saída buzzer

O buzzer possui duas funcionalidades principais neste momento do projeto, sendo elas:

1. Indicar a leitura bem sucedida do cartão ou tag pelo RFID, disparando um alarme sonoro, indicando que a leitura está completa.
2. Indicar que o ESP(Prisma) não está conectado via WiFi.

## 

## 3.2. Tabela de casos de uso - Entrada e Saída

Abaixo há duas tabelas que melhor explicitam a configuração e a descrição das entradas e saídas do sistema. A primeira, é a do ESP(Prisma), onde é possível encontrar todos os sensores que foram implementados para melhor experiência do usuário. Já a segunda tabela explicita como funciona a comunicação entre o ESP(Prisma) e o ESP Beacon.

**Tabela do ESP(Prisma)**

| **#** | **bloco** | **Representação gráfica** | **componente de entrada** | **leitura da entrada** | **componente de saída** | **leitura da saída** | **Descrição** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | status - sucesso |  | RFID - RC522 | Lendo cartão a todo segundo | Identificação no console do IDE | Cartão lido e reconhecido no IDE | Quando o cartão encostar no RFID o programa identifica qual é este cartão e o autorizará ou não. |
| 2 |  | led verde | apagado | led verde piscando | piscando em intervalo de 0,08s | Identifica a presença do cartão. |
| 3 |  | led azul | acesso | led aceso | led aceso | Identifica que o hardware está ligado e com funcionamento normal. |
| 4 |  | led vermelho | apagado | led piscando | led piscando | Ao não se conectar no WiFi, automaticamente, o led começa a piscar indicando que o hardware não está pronto para ser utilizado. |
| 5 |  | buzzer de 0 a 4095 | mudo | toque sonoro | toque sonoro | Ao identificar o código do cartão aproximado o buzzer libera um som indicando que o hardware está pronto para ser utilizado.  Outra função do buzzer é emitir um som sonoro com intervalo de 1 segundo, ao não se conectar ao WiFi.  Indica que o hardware não deve ser utilizado. |

**Tabela de entrada e saída do sistema ESP(Prisma) e ESP Beacon**

|  | **bloco** | **Representação gráfica** | **componente de entrada** | **leitura da entrada** | **componente de saída** | **leitura da saída** | **Descrição** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | status - sucesso |  | ESP que fica dentro do Prisma | Procurando conexão WiFi | Conexão feita. | Identificação no console do programa | Quando o prisma de conectar via Wifi, o sistema começa a funcionar, ou seja, já é possível medir a distância entre ESP’s e fazer a leitura do RFID com o cartão. |
| 2 |  | ponto de acesso | Transmitindo/Roteando sinal via WiFi para conexão ESP-ESP | Transmitindo/Roteando sinal via WiFi para conexão ESP-ESP | Roteando sinal | Transmite a localização e indica a localização do ESP em movimento. |

## 

## 

## 3.3. Interações

Nessa versão inicial de interações do ESP(PRISMA) com o ESP Beacon destacam-se 3 formas de comunicação, onde a ação do usuário pode ou não ser necessária para melhor execução do sistema.

O primeiro caso, ao iniciar o processo de acionamento do ESP(Prisma), optamos por facilitar a experiência do Usuário e não submetê-lo ao processo de pressionar botões e esperar respostas do sistema, sendo assim, para ligar o ESP(Prisma)é necessário apenas conectá-lo a uma fonte de alimentação móvel, ou seja, uma bateria como o powerBank.

O segundo caso, ocorre automaticamente, que é a conexão entre os ESP's, ou seja, o ESP(Prisma) e o ESP Beacon, se conecta via WiFi e se comunicam, demonstrando a distância e o tempo entre eles. Foi observado que em alguns testes, o ESP(Prisma) pode apresentar falhas ao se conectar automaticamente, o que exige do usuário apenas uma ação, ao perceber que a luz do led vermelho está piscando e o buzzer tocando em intervalos de 1 segundo, o usuário saberá que algo não está correto. E a indicação da luz piscando e o buzzer tocando indica que o ESP(Prisma) não está conectado à rede. Para corrigir tal ato, é necessário resetar a placa do ESP(Prisma) e esperar a luz azul acender e permanecer acesa. Após resetar o ESP(Prisma) a luz permanecerá acessa e o sistema voltará a funcionar normalmente.

E, no terceiro caso, ocorre a transmissão dos dados coletados do ESP(Prisma) para a tela de administração do gestor do estacionamento. Os dados já estão sendo coletados e estão corretos, mas é preciso enviá-los via comunicação HTTP, assim, os dados coletados são enviados ao banco de dados local e posteriormente, são impressos no frontend do administrador, indicando alguns dados essenciais para a vida do cliente final, um exemplo do que é impresso na tela do administrador é o tempo que levará para o usuário receber seu veículo, juntamente com dados essenciais sobre o prisma, como o motorista que buscou o carro.

Esses três casos, engloba a solução das dores do cliente, que é o administrador do estacionamento, mas também consegue indiretamente, solucionar as dores do cliente final e do manobrista, pois com esta solução há o conhecimento do tempo final de entrega de veículo, melhor assertividade de tempo de trabalho e produtividade de cada colaborador e consequentemente, maior possibilidade de bonificação por empenho do colaborador, além de maior compreensão dos colaboradores que não atingem as metas.

Abaixo é possível visualizar as configurações, as ações do usuário e a resposta do sistema nos três casos de interações explicadas.

**Tabela de interação entre ESP(Prisma) e ESP Beacon**

| **#** | **configuração do ambiente** | **ação do usuário** | **resposta esperada do sistema** |
| --- | --- | --- | --- |
| **1** | ESP(Prisma) ligado | Basta o usuário conectar uma fonte de energia móvel de entrada tipo C do ESP. | A luz do led azul acenderá e o ESP já estará pronto para ser utilizado. |
| 2 | ESP (Prisma) buscando se conectar no WiFi. | Ao ligar o prisma, é necessário observar se o mesmo está conectado no WiFi para começar a medir a distância e consequentemente o tempo, caso a luz do led vermelho começar a piscar e o buzzer começar a tocar, significa que o mesmo não está conectado ou desconfigurou, sendo necessário o reset na placa. | Ao resetar o ESP(Prisma), a luz do led vermelho parará de piscar, juntamente com um silenciamento do buzzer, indicando que o ESP(Prisma) está conectado na rede WiFi dos ESP. |
| 3 | ESP (Prisma) conectado e enviando dados ao banco de dados e visualizados no computador administrador | O usuário necessita está na aba dos prismas ativos e os colaboradores ativos. | Visualização dos automóveis, juntamente com os dados disponíveis (nº do prisma, nome e ID do colaborador, tempo para entrega). |