

Universidade Estadual de Campinas  
IE309 - Turma X



## Projeto Final - Smart Parking

Aluno: Daniel Filipe Vieira RA 262720

Aluno: Fabio Bezerra de Souza RA 264526

Aluno: Guilherme Lopes da Silva RA 208916

Aluna: Paula Larissa Dias Lima RA 262738

Professor: Dr. Fabiano Fruett

Professor: MSc Pedro Chaves

Professor: MSc Raphael Montali

Junho  
2020

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Referencial Teórico</b>	<b>2</b>
2.1	Soluções IoT . . . . .	2
2.2	Linguagem de programação . . . . .	3
2.3	Redes de sensores sem fio . . . . .	3
2.4	Circuitos eletrônicos . . . . .	4
<b>3</b>	<b>Abordando a metodologia das três fases</b>	<b>5</b>
3.1	Fase I - Entendendo o negócio . . . . .	5
3.1.1	<b>O negócio</b> . . . . .	5
3.1.2	<b>As coisas</b> . . . . .	6
3.1.3	<b>O especialista</b> . . . . .	7
3.1.4	<b>As regras do negócio</b> . . . . .	7
3.2	Fase II - Definição de requisitos . . . . .	8
3.2.1	<b>Exibição</b> . . . . .	8
3.2.2	<b>Abstração</b> . . . . .	9
3.2.3	<b>Armazenamento</b> . . . . .	9
3.2.4	<b>Borda</b> . . . . .	10
3.2.5	<b>Conectividade</b> . . . . .	10
3.2.6	<b>Nós sensores e atuadores</b> . . . . .	10
3.3	Fase III - Implementação . . . . .	12
3.3.1	<b>Nós sensores e atuadores</b> . . . . .	12
3.3.2	<b>Microcontrolador</b> . . . . .	15
3.3.3	<b>Conectividade</b> . . . . .	18
3.3.4	<b>Borda</b> . . . . .	19
3.3.5	<b>Armazenamento</b> . . . . .	19
3.3.6	<b>Abstração</b> . . . . .	20
3.3.7	<b>Exibição</b> . . . . .	20
<b>4</b>	<b>Protótipo</b>	<b>21</b>
4.1	Arquitetura geral do sistema . . . . .	21
4.2	Sistema de monitoramento . . . . .	22
4.2.1	Componentes . . . . .	22
4.2.2	Desenvolvimento da PCB . . . . .	23
4.2.3	Firmware . . . . .	25
4.3	Gerência . . . . .	28
<b>5</b>	<b>Conclusões</b>	<b>31</b>

# 1 Introdução

A evolução dos sistemas eletrônicos permite que algumas atividades simples possam ser adaptadas para responder de maneira inteligente, proporcionando não somente comodidade ao usuário, mas também a economia em muitos casos. Atrelada a esta evolução, o conceito de *Internet Of Things* - IoT (ou Internet das Coisas), tornou-se popular e conforme [1] é uma realidade que tem um potencial de crescimento impressionante, apesar de não ser tão difundido quanto desejado, e que bilhões de objetos futuramente serão conectados trocando informações e interagindo com o ambiente de forma inteligente. Assim, a IoT vem sendo estudada não somente para aplicações residenciais ou industriais, mas também para cidades promovendo o que conhecemos por *Smart Cities* ou cidades inteligentes.

Com isso, no âmbito das *Smart Cities* [2] nos afirmam que estas representam um novo conceito em abordagem de desenvolvimento urbano, podendo assim permitir a melhora da mobilidade urbana. Pensando nisso, desenvolver estacionamentos inteligentes é uma opção a se considerar nos projetos de *Smart Cities*, de modo a evitar os congestionamentos de tráfego e também auxiliar na redução da emissão de gases na atmosfera. Além disso, vale ressaltar que um sistema inteligente para monitorar as vagas gera dados acerca da ocupação destas o que auxilia o proprietário do estacionamento na gestão do local, permitindo-o elaborar novas estratégias para o seu negócio. Pensando nisso, propomos o desenvolvimento de um protótipo de estacionamento inteligente monitorado através de uma rede de sensores sem fio. Este monitoramento foi realizado de tal modo que possibilitava indicar onde havia uma vaga livre e qual a vaga do estacionamento passava mais tempo ocupada. Além de contribuir para o melhor gerenciamento do estacionamento por parte do proprietário, por exemplo, o monitoramento das vagas pode auxiliar a prever os dias e horários de maior movimento oferecendo maior comodidade ao usuário e permitindo ao dono do estacionamento desenvolver novas estratégias para o seu estabelecimento.

Assim, além desta seção introdutória, este trabalho apresenta mais quatro seções sendo elas: **Seção 2 - Referencial teórico** que busca através da literatura embasamento técnico científico que auxilie na compreensão deste trabalho. Posteriormente, a **Seção 3 - Abordando a metodologia das três fases**, apresenta a metodologia de desenvolvimento do trabalho em questão. Em seguida a **Seção 4 - Resultados**, traz os resultados obtidos a partir deste projeto e por fim, na **Seção 5 - Conclusões**, são apresentadas as conclusões alcançadas através deste trabalho.

## 2 Referencial Teórico

Para o desenvolvimento deste projeto de pesquisa, fez-se necessário o estudo nas áreas de: soluções IoT, linguagem de programação, redes de sensores sem fio e dimensionamento de circuitos eletrônicos. No contexto das soluções IoT, utilizamos a metodologia das três fases que será detalhada posteriormente na seção 3: “**Abordando a metodologia das três fases**”.

### 2.1 Soluções IoT

Conforme [3] *Internet of things* (IoT) ou internet das coisas pode ter diversas definições conforme diferentes perspectivas. No entanto para definir IoT, adotaremos o conceito proposto em [4] que a define como sendo dispositivos físicos incorporados a dispositivos eletrônicos que permitem que estes dispositivos físicos se conectem à internet. A partir da compreensão sobre o que é uma solução IoT, faz-se necessário enfatizar o que vem a ser a “coisa” no conceito de IoT. De acordo com [3] uma coisa pode ser qualquer dispositivo presente em nosso cotidiano, desde que este dispositivo tenha uma capacidade computacional e de comunicação com a internet. Além disso, ainda em [3] o autor nos afirma que a comunicação da coisa com a internet não precisa ser direta, por exemplo a coisa (sendo qualquer dispositivo) pode estabelecer contato com outro dispositivo através de uma conexão *bluetooth* e o segundo dispositivo pode transmitir esses dados para a nuvem através da conexão com a internet. Assim, podemos concluir que as soluções IoT não são apenas soluções que apresentam um processo automatizado que pode se realizar sem a interferência humana, mas soluções inteligentes capazes de auxiliar em diversas situações através de previsões e sugestões ao usuário, elaboradas a partir do compartilhamento de dados coletados por dispositivos em um determinado ambiente permitindo maior comodidade, mobilidade e até mesmo redução de custos em alguns casos.

Como solução IoT, propomos neste trabalho um sistema de estacionamento inteligente. Consideramos relevante informar que [5] nos adverte que a maioria dos estacionamentos são considerados apenas sistemas de estacionamento informativos (IPS)<sup>1</sup> ao invés de serem sistemas de estacionamento inteligentes (SPS)<sup>2</sup> pois, apenas fornecem informações sobre a localização e o número de vagas disponíveis no estacionamento, não informando com exatidão o espaço disponível para estacionar. Partindo deste pressuposto, buscamos desenvolver nosso estacionamento de modo que sensores ultrassônicos fossem capazes de detectar quando as vagas estavam ocupadas ou não, além de informar com exatidão qual vaga estava disponível e calcular o percentual de ocupação das vagas conforme um determinado período, e disponibilizar estas informações ao proprietário do estacionamento.

---

<sup>1</sup>Do inglês *Informative Parking System*

<sup>2</sup>Do inglês *Smart Parking System*

## 2.2 Linguagem de programação

A linguagem de programação neste projeto é considerada elemento chave para o seu funcionamento pois, a esta possibilitou o funcionamento inteligente do protótipo. É através da linguagem programada que máquinas e computadores são capazes de executar tarefas, e assim são consideradas elementos inteligentes. Conforme [6] a ilusão de que eles realizam tarefas de forma inteligente é proporcionada através desse conjunto ordenado de instruções, que é denominado de algoritmo. Ainda conforme os autores, um algoritmo é definido como uma sequência finita, ordenada e não ambígua de passos para solucionar determinado problema ou realizar uma tarefa. Além dessa definição, os autores supracitados reforçam no capítulo 1 do seu livro “*Introdução à programação*”, que o conceito de algoritmo foi apresentado formalmente pela primeira vez por Alan Turing e Alonzo Church, em 1936 como sendo um conjunto não ambíguo e ordenado de passos executáveis que definem um processo finito. Assim, desenvolvemos uma linguagem programada (ou um algoritmo) capaz de executar a principal tarefa do nosso sistema: monitorar, através de sensores ultrassônicos, as vagas ocupadas e disponíveis em um estacionamento. Dentro os diversos tipos de linguagem de programação existentes, enfatizamos que para o desenvolvimento do nosso algoritmo fizemos o uso das linguagens C e Python para os algoritmos do módulo de monitoramento e sistema de gerência, respectivamente.

## 2.3 Redes de sensores sem fio

A rede de sensores sem fio (RSSF) ou em inglês, *Wireless sensor Network* (WSN), é uma rede composta por dispositivos, que são chamados de nós. Conforme apresentado em [7], os nós são usados para capturar informações e/ou controlar o ambiente ao seu redor. Uma rede de sensores sem fio é composta basicamente por 3 componentes: nós sensores, *gateway* e observador (usuário), conforme exibido na Figura 1:

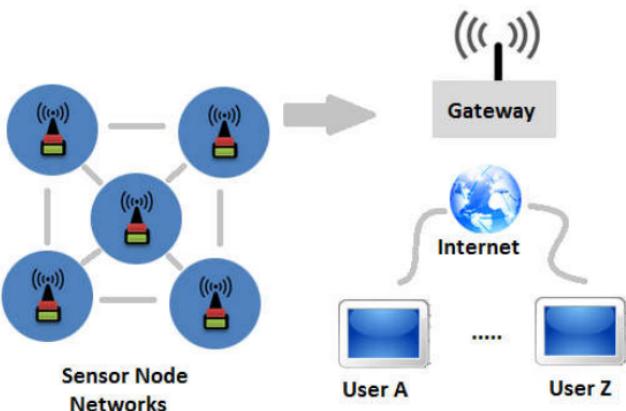


Figura 1: Arquitetura típica de uma RSSF. Adaptado de [7].

Os nós são compostos por dispositivos inteligentes com capacidade para sensoriamento, processamento e comunicação (geralmente sem fio). A comunicação com o *gateway* pode ser feita de diversas formas, onde as mais comuns são: ZigBee, Bluetooth, WIFI, etc. O *gateway* é responsável por enviar comunicação ao observador(usuário) através da internet.

Este trabalho consiste em uma aplicação típica de RSSF, onde os nós sensores estarão monitorando as vagas e o *gateway* será uma SBC<sup>3</sup> onde será feito o tratamento e envio dos dados para o próximo nível.

## 2.4 Circuitos eletrônicos

Uma das definições da palavra eletrônica pode ser encontrada em [8] como sendo: “Ciência que estuda os fenômenos em que se manifestam elétrons livres. Técnica baseada no emprego de dispositivos com utilização de elétrons em estado livre”. Assim, podemos concluir que os circuitos eletrônicos tratam da ligação entre dispositivos eletrônicos de modo a instituir um caminho fechado para os elétrons livres ordenando-os em corrente elétrica.

Neste trabalho, foi implementado um circuito eletrônico de modo a interligar os dispositivos, dentre os quais podemos destacar os microcontroladores e conforme é definido em [9] “Os microcontroladores estão presentes em quase tudo o que envolve a Eletrônica, diminuindo o tamanho, facilitando a manutenção e gerenciando tarefas internas de aparelhos eletroeletrônicos.” Ainda em [9] nos é apresentado que “Com o passar dos anos e com os avanços da tecnologia, os microcontroladores tornaram-se uma das melhores relações custo/benefício em se tratando de soluções que demandam processamento, baixo custo de hardware e pequena necessidade de espaço físico”. Sendo assim, um microcontrolador pode ser considerado um elemento não oneroso e essencial para a otimização do sistema eletrônico proposto, de modo a torná-lo inteligente.

---

<sup>3</sup>Do inglês *Single Board Computer*

### 3 Abordando a metodologia das três fases

Nesta seção serão abordadas as três fases da metodologia, a qual possibilitou o desenvolvimento deste projeto de maneira clara e organizada. Inicialmente descrevemos a metodologia das três fases, e atrelado à metodologia supracitada, mais precisamente nas fases 2- Definição dos requisitos e 3- Implementação, enfatizamos o modelo de referência utilizado para atender à demanda do projeto. O modelo adotado é denominado *Open Source Reference Model - OSRM* ou modelo de referência aberto. Com o modelo de referência, é possível exibir quais são as tecnologias envolvidas nos diversos níveis. O objetivo de dividir a solução em níveis é facilitar a visualização das etapas e possibilitar o detalhamento das necessidades e desafios de cada etapa. Isso possibilita criar uma base para análise das diferentes tecnologias [10]. A Figura 2 a seguir, detalha o modelo OSRM.

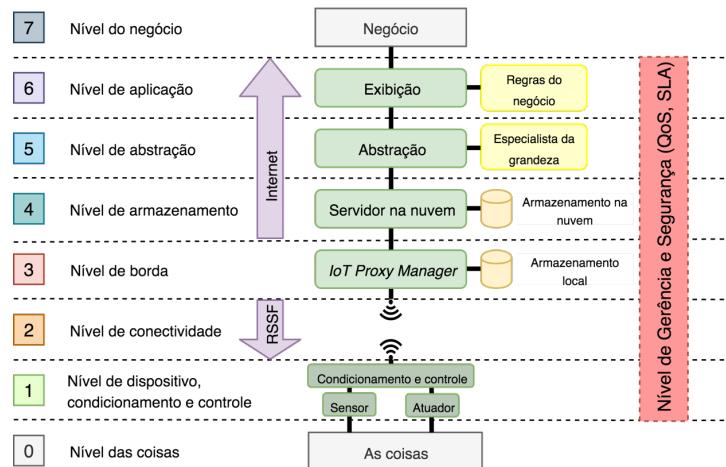


Figura 2: Modelo de referência OSRM. Adaptado de [11].

Na subseção 3.1 descrevemos a Fase I, onde são apresentados os tópicos a respeito do negócio, buscando compreender qual problema deveria ser resolvido e seus principais desafios<sup>4</sup>. Na subseção 3.2 apresentamos a Fase II, na qual é mapeada a definição de premissas e requisitos, estudando e discutindo sobre as tecnologias que poderiam atender as especificações do negócio em questão. E por fim, trouxemos na subseção 3.3 a Fase III, onde deve-se iniciar a implementação tomando como base a definição de requisitos realizada na Fase II.

#### 3.1 Fase I - Entendendo o negócio

##### 3.1.1 O negócio

Neste projeto, objetivou-se o desenvolvimento de um protótipo de estacionamento inteligente que atendesse as necessidades dos seus proprietários. Estes estacionamentos

<sup>4</sup>Nesta etapa, é importante compreender quais são as necessidades do cliente.

estão, a princípio, localizados em ambientes fechados (shoppings e mercados) e os proprietários poderiam obter informações sobre a contagem de vagas destes estacionamentos de maneira individual em função dos dias da semana. Além disso, seria interessante que a solução pudesse ajudar os clientes do próprio estacionamento, por meio de LEDs indicadores de vaga livre/ocupada, podendo esta ser uma opção a ser implementada conforme o desejo do proprietário. Pretendia-se obter estatísticas a respeito dessas contagens e chegar nas seguintes informações:

- **Qual é o dia de maior movimento** – Com essa informação, o proprietário poderá informar ao comércio quais os melhores dias/horários para se criar ofertas de seus produtos (quanto mais gente no local, maior a probabilidade de venda);
- **Qual a média do tempo de ocupação das vagas** – Com essa informação, é possível ter uma estimativa de ocupação das vagas, o que poderá ser uma informação útil ao motorista quando o estacionamento estiver cheio;
- **Quantos clientes vão ao estacionamento por dia** – Isso é importante para se obter uma estimativa de custos, por exemplo: se o estacionamento cobrar taxa de entrada, o controle da quantidade de carros estacionados por mês pode auxiliar no faturamento mensal do estabelecimento. Além disso, é possível verificar se a quantidade de vagas é suficiente para suprir a demanda de clientes, e até mesmo para controlar a escala e o horário de trabalho dos funcionários do estacionamento;
- **Quais as vagas preferidas dos clientes** – Algumas vagas podem ser mais solicitadas pelos clientes, em especial quando estão próximas das entradas dos locais. Conhecendo esta informação, o proprietário pode pensar em novas opções de *layout* e de vias para deslocamento rápido dentro do estacionamento conforme a demanda por estas vagas;
- **Quantas pessoas estacionam na vaga de deficientes por dia** – Pode-se ter uma ideia desse número de pessoas para saber se todas as vagas de deficientes estão sendo ocupadas ou não, saber se as pessoas que estão estacionando são, de fato, deficientes o que permitirá identificar a necessidade de incluir mais vagas para deficientes conforme a demanda, além de auxiliar em possíveis adequações do local a fim de melhorar a mobilidade destas pessoas conforme as suas necessidades.

### 3.1.2 As coisas

Sabe-se que o IoT permite a conexão entre as coisas, e o compartilhamento das informações obtidas através destas conexões. É importante destacar que [3] nos afirma que uma “coisa” pode ser qualquer objeto à nossa volta e que além disso, cada “coisa” deve

ter capacidade computacional e de comunicação com a internet. As “coisas” são entidades necessárias para alcançar os objetivos do negócio. Podem ser grandezas físicas, como umidade, luminosidade, ou dispositivos eletrônicos como relés e câmeras [11]. A Figura 3 representa uma “coisa” em uma solução IoT, onde há um dispositivo comum do cotidiano com uma certa capacidade computacional e comunicação com a internet.

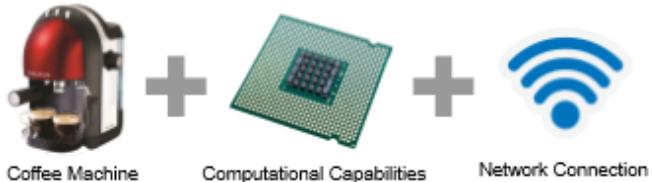


Figura 3: Coisas em IoT. Adaptado de [3].

As coisas que foram monitoradas neste projeto são as vagas do estacionamento. Cada uma delas foi monitorada por meio de sensores ultrassônicos do tipo HC-SR04.

### 3.1.3 O especialista

Os especialistas são os gerentes do estacionamento, pois estão em contato diário com os clientes e possuem conhecimento sobre as coisas que ocorrem nesses locais. Podemos incluir neste grupo também os próprios motoristas que mais utilizam o serviço do estacionamento pois, estes podem sugerir possíveis melhorias conforme as suas necessidades.

### 3.1.4 As regras do negócio

O negócio consistiu no monitoramento direto das vagas para contabilizar o número de veículos que estacionavam em cada vaga. Sendo assim, identificamos as seguintes regras (premissas e requisitos) para este negócio:

- A solução deveria ser de baixo custo, pois seria utilizado um módulo de monitoramento para cada uma das vagas do estacionamento. Obviamente, quanto menor o custo dos módulos, menor será o custo do projeto;
- Fez-se necessário o controle individual de cada vaga do estacionamento, por uma simples questão de organização e arquitetura do sistema que seria desenvolvido. Deveríamos desenvolver um sistema escalável, fácil de ser reproduzido. Isso também seria uma vantagem na hora de fazer a manutenção;
- O sistema a ser desenvolvido utilizaria uma solução Open-Source, de ponta a ponta, pois isso facilitaria a implementação de novas iniciativas;
- Exibição e Armazenamento de dados – Escolhemos sistemas altamente flexíveis, que fossem capazes de se adequar aos requisitos do projeto. Precisamos de servidores

com iniciativas de código aberto, que possibilitem fazer modificações mais profundas caso seja necessário.

- **Borda** – Pretendia-se obter um elemento de borda que trabalhasse com uma linguagem de programação usual e que apresentasse forte suporte na Web.

## 3.2 Fase II - Definição de requisitos

Nessa Seção será apresentada a definição de requisitos necessários para a elaboração da solução de IoT apresentada neste trabalho. Nessa etapa de requisitos, não serão listadas as tecnologias que possam suprir as necessidades dos clientes, uma vez que os detalhes das tecnologias serão mencionados na Fase III.

No geral, os principais requisitos para o desenvolvimento da solução de IoT são (*i*) o baixo custo de implementação e; (*ii*) escalabilidade em termos de atualizações futuras. Nesse sentido, as tecnologias escolhidas em cada uma das camadas do modelo de referência deverão suprir as condições (*i*) e (*ii*).

### 3.2.1 Exibição

A camada de exibição deveria ser composta por um sistema capaz de apresentar elevadas quantidades de informações ao cliente, uma vez que um estacionamento possui muitas vagas. Nesse sentido, ressalta-se que esse sistema deve ser escalável e organizado o suficiente para este fim. Além disso, as informações devem ser apresentadas de maneira elegante, de modo que se torne agradável aos olhos do cliente. É necessário que o sistema apresente os seguintes tópicos:

- **Indicador de vagas** – Uma variável deve indicar a disponibilidade de cada vaga, fornecendo o seu estado atual;
- **Movimento nos dias da semana** – Deseja-se conhecer o número de carros que ocupam cada vaga em função dos dias da semana. Dessa maneira, é possível ter uma ideia do tempo médio de ocupação de cada vaga, qual a média do tempo de ocupação das vagas, quais são as vagas preferidas dos clientes, qual o número estimado de motoristas que possuem algum tipo de deficiência física que frequentam o estacionamento, entre outros.
- **Controle de pessoas por causa da pandemia** – Devido ao cenário atual, tornou-se necessário a realização de um controle do número de pessoas que está acessando grandes estabelecimentos, como é o caso de shoppings. Com a contagem do número de carros que circula pelo estacionamento, é possível realizar um controle de acesso de pessoas aos estabelecimentos (por exemplo os shoppings).

### 3.2.2 Abstração

O sistema que será escolhido deve apresentar flexibilidade para que certas informações sejam abstraídas dos clientes (essas informações serão disponibilizadas apenas aos administradores do sistema, os donos do estacionamento não terão acesso a elas). Por exemplo, as informações de gerência da rede de sensores (estatísticas da potência de recepção) serão apresentadas aos administradores que entendem do assunto. Os requisitos nesta etapa são:

- **Presença do especialista** – O especialista é um elemento externo que determina a estratégia de transformação dos dados em informação fornecendo informações detalhadas sobre o problema a ser resolvido. E como a abstração se trata da capacidade de manipular os dados e deles extrair informações dos processos, faz-se necessária a presença do especialista [10]. Essas informações serão necessárias para que seja possível a divisão do sistema em duas instâncias, conforme é descrito no item a seguir.
- **Servidor com duas instâncias** – Serão necessárias duas instâncias. A primeira delas será disponível apenas para o cliente (dono do estacionamento), em que as informações da camada de exibição serão apresentadas. Já na segunda instância, um administrador terá acesso tanto às informações da camada de exibição, quanto aos dados gerados a partir das etapas de gerência de rede.

### 3.2.3 Armazenamento

Além de exibir os dados, é necessário que estes sejam armazenados. Para isso, conforme descrito em [10], são necessários dois elementos para realizar o armazenamento, sendo eles:

- **Servidor na nuvem** – No geral, este é contratado a fim de facilitar o seu gerenciamento de funcionamento. Além da facilidade de alterar a capacidade do mesmo de maneira simples, geralmente alterando o plano de contratação [10];
- **Capacidade de armazenamento** – A solução escolhida deve ter boa capacidade de armazenamento de dados. Nesta condição existe um impasse entre utilizar uma solução de armazenamento em nuvem ou armazenamento próprio. Caso opte pelo armazenamento em nuvem, o cliente dependerá de cobranças que podem ser realizadas conforme o uso, entretanto os dados podem estar disponíveis para o fornecedor da solução. Por outro lado, o cliente pode optar pelo armazenamento próprio por questões de segurança e criticidade dos dados envolvidos [10]. Ainda em [10] destaca-se que, ao adotar uma solução de armazenamento deste tipo existem questões como backup, capacidade de ampliação do espaço conforme necessário, consumo de energia, entre outras com as quais o cliente deverá se preocupar.

### 3.2.4 Borda

O elemento de borda ou *gateway* é o elemento responsável por fazer a conexão entre os dados coletados provenientes da RSSF para a internet. Faz-se necessário que o *Gateway* seja capaz de cumprir com os seguintes requisitos: (*i*) compatibilidade com a Estação Rádio Base (ERB) da tecnologia de conectividade sem fio; (*ii*) coleta dos dados que trafegam pela rede e posterior processamento dos mesmos e; (*iii*) suporte e compatibilidade com o protocolo TCP/IP, que é fundamental para a utilização de protocolos utilizados em aplicações de IoT (MQTT, HTTP, CoAP, etc).

### 3.2.5 Conectividade

No contexto da conectividade sem fio, a tecnologia a ser escolhida neste projeto deveria atingir distâncias razoáveis (dependendo do tamanho do estacionamento) em termos de transmissores e receptores, não sendo necessária a operação com elevadas taxas de dados (pacotes de dados transmitidos de 5 em 5 segundos já eram o suficiente para atender essa aplicação). Isso porque seriam enviadas pequenas quantidades de bytes no monitoramento das vagas. Nesse sentido, avaliou-se a possibilidade da implementação de diversos tipos de rede como por exemplo Wi-Fi, celular e redes de baixo consumo de energia, longo alcance e baixas taxas de processamento (Low Power Wide Area Network, *LPWAN*). As soluções de conectividade são classificadas em três categorias conforme [10], sendo elas:

- **Soluções de internet** – São protocolos proprietários que utilizam protocolos próprios na RSSF e disponibilizam os dados na internet via protocolo TCP/IP. Alguns exemplos são as tecnologias SigFox, LoraWan, LTE NoB (3GPP, 2018) e WI-Fi (2018);
- **Soluções de conectividade** – São protocolos proprietários que utilizam protocolos específicos. Alguns exemplos são: LoRa, ZigBee, Z-Wave (2018) e bluetooth (2018);
- **Soluções open source** – São soluções de código aberto que permitem o desenvolvimento de redes que atendam a um objetivo específico.

Após a realização de estudos e considerando os requisitos descritos acima, a equipe optou por utilizar um sistema open-source LPWAN para suprir a etapa de conectividade sem fio.

### 3.2.6 Nós sensores e atuadores

O dispositivo que realizou o monitoramento de cada vaga do estacionamento é composto pelas seguintes partes:

- **Microcontrolador** – Este componente deveria ser capaz de cumprir duas tarefas: (*ii*) processar as informações coletadas dos sensores, executando essa tarefa em

curtos intervalos temporais; (*ii*) realizar a contagem de vagas com base nos dados processados e enviar aos atuadores os comandos correspondentes e; (*iii*) enviar as informações processadas para um módulo de transmissão de dados sem fio. Além disso, sugeriu-se que o microcontrolador estivesse incorporado como uma solução open-source que possibilitasse suporte favorecido na internet, e que apresentasse um ambiente de desenvolvimento de firmware amigável de fácil interação, com escalabilidade e flexibilidade.

- **Sensor de monitoramento** – Com relação ao sensor para o monitoramento das vagas, não foi necessário que tivesse alta precisão, isto é, que apresentasse leituras muito próximas da realidade. No cenário deste projeto, foram feitas comparações entre duas distâncias muito diferentes: uma leitura com relação ao chão e a outra em relação à parte superior dos veículos. Nota-se que o sensor deveria comparar essas duas distâncias para tomar as devidas decisões. Portanto, conclui-se que não é um requisito de projeto trabalhar com sensores de alta precisão impactando, assim, no custo de implementação (sensores imprecisos tendem a ser de baixo custo). Além disso, destaca-se que existem estratégias de software para amenizar possíveis falhas que o sensor possa apresentar, como por exemplo a implementação de filtros e rotinas de detecção de falhas. Nesse sentido, apela-se para a implementação dessas estratégias a fim de tornar a solução mais confiável.

Por fim, ainda referente à escolha do sensor, destaca-se que o hardware deste projeto seja flexível ao uso de outros modelos de sensores, sendo assim adaptável. Neste ponto, o sistema não deve limitar-se ao uso de um único tipo de sensor.

- **Atuadores** – Os atuadores são componentes físicos capazes de transmitirem informações aos clientes do estacionamento<sup>5</sup>. Neste caso, é preciso que os clientes consigam identificar se a vaga está disponível ou não, considerando distâncias relativamente elevadas. Em outras palavras, um cliente precisará identificar, visualmente, se há uma vaga livre mesmo que esteja longe dela.

---

<sup>5</sup>Ressalta-se que o objetivo deste projeto é desenvolver uma solução de IoT para atender aos donos do estacionamento. Entretanto, a sinalização do estado das vagas para os clientes é uma estratégia interessante sob ponto de vista de logística interna.

### 3.3 Fase III - Implementação

Após a execução das etapas (*i*) Entendimento do negócio e (*ii*) Definição de requisitos, iniciam-se os procedimentos da Fase III objetivando a implementação da solução de IoT. Na Fase III, serão apresentadas as tecnologias que possam suprir as necessidades do projeto, levando em consideração as suas principais características. Posteriormente, após a definição das tecnologias e suas respectivas justificativas, será descrita a etapa de implementação.

#### 3.3.1 Nós sensores e atuadores

Conforme foi apresentado na subseção 3.1.2 “As coisas” dentro da seção 3 “Fase I - Entendendo o negócio”, o sensor escolhido para desenvolver o protótipo foi o ultrassônico HC-SR04. Este modelo de sensor ultrassônico funciona através do seguinte princípio: o sensor envia pulsos ultrassônicos e aguarda o retorno do sinal. Baseando-se no tempo de envio e retorno do sinal, o sensor calcula a distância entre o sensor e o objeto detectado. O sensor apresentado, tem alcance de leitura que varia de 2cm a 400cm ou de 0.02m a 4m, conforme [12]. Considerando que um carro popular tem altura média de 1.5m aproximadamente, este sensor atende satisfatoriamente ao principal objetivo deste projeto que resume-se em efetuar a contagem do número de veículos que ocupam as vagas do estacionamento a partir da detecção do veículo conforme a verificação periódica da altura do sensor em relação ao solo. A Figura 4 apresenta alguns modelos de carros com seus principais dados, dentre os quais está inclusa a altura dos veículos.

CONHEÇA OS PRINCIPAIS CONCORRENTES DO FIAT UNO MILLE			
<b>Effa M100</b> 	<b>Ford Ka 1.0</b> 	<b>Celta 1.0 2 portas</b> 	<b>Clio Campus 2 portas</b> 
<b>Combustível:</b> gasolina <b>Motor:</b> 970 cc e 47 cavalos <b>Comprimento:</b> 3.560 mm <b>Largura:</b> 1.600 mm <b>Altura:</b> 1.670 mm <b>Entre-eixos:</b> 2.335 mm <b>Peso:</b> 930 kg <b>Porta-malas:</b> 320 litros A partir de <b>R\$22.980</b>	<b>Combustível:</b> álcool / gasolina <b>Motor:</b> 1.000 cc e 70 cv <b>Comprimento:</b> 3.836 mm <b>Largura:</b> 1.812 mm <b>Altura:</b> 1.420 mm <b>Entre-eixos:</b> 2.452 mm <b>Peso:</b> 943 Kg <b>Porta-malas:</b> 263 litros A partir de <b>R\$ 26.190</b>	<b>Combustível:</b> álcool/gasolina <b>Motor:</b> 1.000 cc e 70 cv <b>Comprimento:</b> 3.788 mm <b>Largura:</b> 1.626 mm <b>Altura:</b> 1.408 mm <b>Entre-eixos:</b> 2.443 mm <b>Peso:</b> 845 Kg <b>Porta-malas:</b> 260 litros A partir de <b>R\$ 26.759</b>	<b>Combustível:</b> álcool/gasolina <b>Motor:</b> 1.000 cc e 76 cv <b>Comprimento:</b> 3.818 mm <b>Largura:</b> 1.639 mm <b>Altura:</b> 1.417 mm <b>Entre-eixos:</b> 2.443 mm <b>Peso:</b> 880 Kg <b>Porta-malas:</b> 255 litros A partir de <b>R\$ 26.890</b>

Figura 4: Dados característicos de alguns modelos de veículos. Adaptado de [13].

Conforme já é conhecido o princípio de funcionamento dos sensores ultrassônicos, o protótipo trabalha da seguinte maneira:

- Os sensores são posicionados a uma determinada altura em relação ao solo;
- Os sensores identificam a altura média de detecção do solo e enviam esta informação à uma unidade de processamento;

- A unidade de processamento deve se manter atualizada com informações das alturas medidas pelos sensores;
- Caso algum carro estacione na vaga, a altura média em relação ao solo irá diminuir o que indica que há um objeto detectado;
- Sendo detectada uma diferença de altura em relação ao solo, os dados coletados devem ser enviados à unidade de processamento;
- A unidade de processamento processa os dados e atualiza a unidade de armazenamento das informações;
- Conforme a atualização das informações processadas e armazenadas, LEDs indicadores serão acionados conforme a disponibilidade das vagas. Com isso, o sensor terá de verificar periodicamente a medida padrão de detecção do solo, e manter o status de cada vaga atualizado. A Figura 5 apresenta o sensor ultrassônico.



Figura 5: Sensor ultrassônico. Adaptado de [12].

A escolha do sensor ultrassônico é justificada através dos seguintes aspectos:

- **Custo benefício** - Neste projeto foi adotada a suposição que o custo do investimento exigido pelo cliente deve ser o mínimo possível. Conforme uma pesquisa de mercado foi identificada uma variação de preço entre R\$ 2.66 e R\$ 14.90 para o sensor ultrassônico apresentado<sup>6</sup>. A Tabela 1 apresenta alguns fornecedores do sensor e seus respectivos valores, vale ressaltar que os valores apresentados foram retirados de lojas online e não consideram o preço de frete.

Tabela 1: Valores do sensor HC-SR04 para diferentes fornecedores.

Fornecedor	Valor
AliExpress	R\$ 2.66
Curto circuito	R\$ 9.50
Eletrogate	R\$ 12.00
Master Walker	R\$ 13.90
Vida de silício	R\$ 14.90

<sup>6</sup>Valores coletados em maio de 2020

Além da Tabela 1, comparamos o sensor ultrassônico HC-SR04 com outros sensores ultrassônicos disponíveis no mercado conforme a Tabela 2:

Tabela 2: Comparando HC-SR04 com outros sensores ultrassônicos

Sensor	Característica	Aplicação
Sensor Sick	Alta precisão	Industrial
Sensor Telemecanique	Atende a distâncias de até 8m	Industrial
Sensor HC-SR04	Atende a distâncias de 0.02m a 4m	Protótipos e pequenos projetos

Assim, sensores a com alta precisão não são exigidos neste projeto tendo em vista que: não foi necessária precisão na medição pois o sistema compara duas medidas bastante distintas que são a altura do sensor em relação ao solo e a altura do sensor em relação à parte superior do veículo. Além disso, não foi exigido um sensor com alto range de detecção considerando que a altura padrão do estabelecimento em questão é mínima em relação ao solo. Também é importante frisar que, características de precisão e detecção aumentam o custo do dispositivo o que consequentemente aumenta o custo do projeto. Reitera-se que, conforme descrito na Fase 2, este projeto é flexível à implementação de outros modelos de sensores, o que não limita o sistema somente ao uso do HC-SR04.

- **Facilidade de manutenção conforme local de instalação** - No início deste documento, mais precisamente na subseção 3.1. “Fase I - Entendendo o negócio” página 6, é informado que este protótipo inicialmente atenderá a estacionamentos fechados livres da influência de agentes externos, o que permite manter os sensores em funcionamento dentro da sua vida útil, e além disso facilita a manutenção (como troca do dispositivo por exemplo), tendo em vista que o mantenedor não estará exposto à intempéries durante o ato da substituição do dispositivo caso seja necessário.
- **Alcance de leitura favorável ao sistema** - Conforme informações disponibilizadas pelo *datasheet* do sensor em [12], o mesmo tem um alcance de detecção que varia de 0,02m a 4m, o que atende à necessidade imposta pelo projeto proposto considerando as alturas médias de veículos automotivos atuais (Vide Figura 3: Dados característicos de alguns modelos de veículos, p.10).
- **Dimensões mínimas** - O sensor possui um tamanho pequeno, o que permite o desenvolvimento de um sistema pequeno que ocupe espaço mínimo.

- **Complexidade mínima de detecção** - Conforme foi citado em seções anteriores, inicialmente o protótipo foi desenvolvido para atender a estabelecimentos de pequeno porte e ambientes *indoor*. Poderia ter sido desenvolvido um sistema mais complexo de detecção de vagas através de câmeras, por exemplo, mas este tipo de sistema não apresenta viabilidade para ambientes fechados pois, conforme [14] faz-se necessário o emprego de técnicas de processamento de imagem, e o uso destas técnicas pode apresentar limitações como por exemplo: se a vaga estiver encoberta por um objeto não é possível identificá-la, a não ser que exista outra câmera captando a mesma vaga mas a partir de um ângulo onde esta não esteja encoberta. Além disso, o ambiente deveria ter uma boa luminosidade para o emprego das técnicas de processamento de imagens, sendo assim, o uso de câmeras de detecção torna-se mais viável em ambientes *outdoor* pois, normalmente, além da iluminação natural durante o dia, há a disponibilidade de iluminação pública durante a noite dependendo do ambiente. Enquanto isso, os sensores não necessitam de luz para detectar ou não a existência de veículos nas vagas pois estes funcionam pelo princípio de ultrassom, onde ondas sonoras são emitidas e ao encontrarem um obstáculo essas ondas são refletidas e retornam. Além disso, o sensor não precisa estar em contato direto com o objeto de detecção e não possui sua operação prejudicada pela transparência, poeira, sujeira ou vapores/gases presentes no ambiente. Desde que o objeto reflita as ondas sonoras, é possível usar um sensor ultrassônico independentemente de seu acabamento superficial ou cor [15].
- **Baixo consumo** - O sensor trabalha com uma corrente de 15 mA, o que implica em um baixo consumo energético.

### 3.3.2 Microcontrolador

Conforme foi apresentado na seção 3.2.4 “Nos sensores e atuadores”, foi escolhido o microcontrolador ATmega328P para gerenciar os nós sensores respeitando os requisitos levantados na fase 2. Assim, esse microcontrolador será responsável pela interpretação das informações coletadas do sensor ultrassônico, realizar a contagem das vagas, acionar os LEDs indicativos e por fim, enviar as informações processadas através de comunicação SPI<sup>7</sup> para módulo de comunicação sem fio usando o protocolo Radiuino.

A escolha deste microcontrolador é justificada através dos seguintes aspectos:

- **Facilidade de aquisição em mercado nacional** -O ATmega328P é um microcontrolador muito popular, por ser usado em diversas placas compatíveis com a plataforma Arduino. Com essa popularidade, as principais lojas de componentes eletrônicos nacionais, o mantêm em estoque, onde é possível comprá-lo facilmente.

---

<sup>7</sup>Serial Peripheral Interface

A Tabela 3 apresenta alguns valores retirados das principais lojas de componentes nacionais:

Tabela 3: Valores do ATmega328P em diferentes fornecedores nacionais (valores de junho de 2020)

Loja	Valor
Baú da Eletrônica	R\$37,33
Robocore	R\$21,50
Usinainfo	R\$27,41

Há também a possibilidade de fazer a importação deste microcontrolador diretamente do fabricante ou através de representantes nacionais, sendo essa uma ótima opção para produção em larga escala.

- **Disponibilidade de placas de desenvolvimento em mercado nacional -** Como este microcontrolador está presente nas mais populares placas Arduino, encontrar placas desenvolvimento para prototipagem rápida não é uma tarefa difícil, mesmo no Brasil. Há diversas opções, sendo as mais comuns exibidas na Figura 6, a seguir:

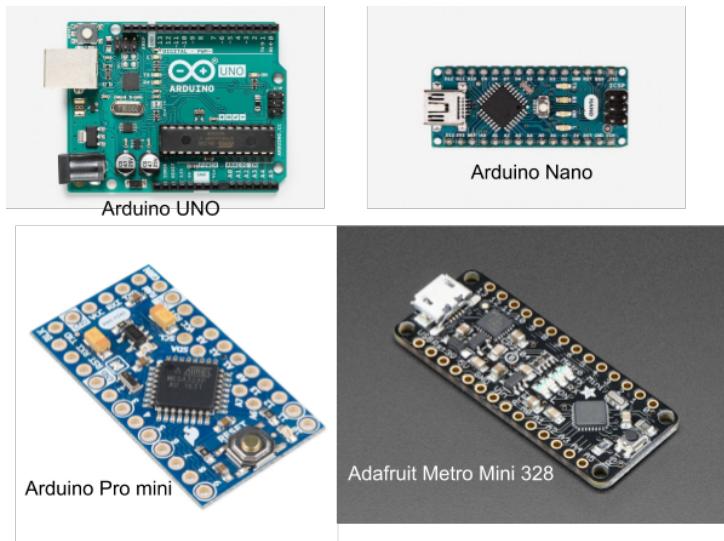


Figura 6: Placas de desenvolvimento com ATmega328P

Optamos por trabalhar com a placa **Arduino Pro Mini** por ser compacta e já trazer os recursos necessários para funcionamento do microcontrolador, além de ser fácil de soldar em uma placa.

- **Capacidade de processamento e armazenamento -** O ATmega328P é um microcontrolador de 8 bit da família AVR, uma excelente arquitetura RISC para aplicações em projetos inteligentes. Possui 32 KB de memória Flash, 2KB de SRAM

e 1KB de EEPROM. Ele pode operar em até 20 MIPS (Million Instructions Per Second). Além das capacidades de armazenamento e processamento, o ATmega328p traz diversos periféricos, como: conversor analógico digital, periféricos de comunicações (I2C, SPI, UART), timers, e saídas PWM. Ele pode operar em tensões de 1,8V a 5,5V e em temperaturas de -40°C a +85°C, conforme verificado em [16]

- **Quantidade de pinos suficiente para o projeto** - O ATmega328P possui até 32 pinos de I/O em 3 tipos de encapsulamentos, conforme exibido na Figura 7, a seguir:

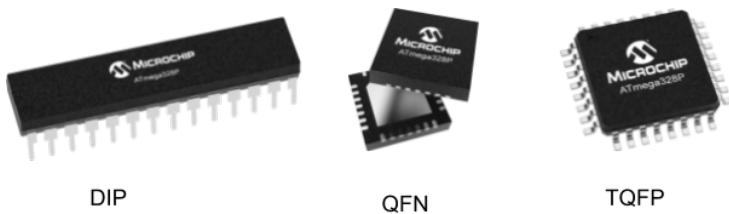


Figura 7: Encapsulamentos disponíveis para o ATmega328P

A quantidade de pinos atende os requisitos do projeto.

- **Periférico de comunicação SPI integrado** - Conforme já mencionado acima, o ATmega328P possui periférico SPI, podendo operar em modo mestre ou escravo. Esse periférico será usado para comunicação com o transceptor de RF.
- **Facilidade de programação usando framework Arduino** - O ATmega328P pode ser facilmente programado usando o *framework* Arduino. Boa parte do ecossistema Arduino foi desenvolvido com base no microcontroladores AVR. Além disso, o compilador AVR-GCC, usado pelo *framework* Arduino, é *open source* e pode ser usado em diferentes sistemas operacionais e ambientes de programação. Esses detalhes foram pontos positivos para *escolha do ATmega328P*, já que uma das premissas deste trabalho foi o uso de ferramentas *open source*.
- **Compatibilidade com bibliotecas Radiuino** - O protocolo Radiuino (que será apresentado mais a frente) foi desenvolvido com base no Arduino e com placas que possuem o ATmega328P. Apesar de ser *open source* e portável, o uso do ATmega328P facilita na implementação e uso de softwares já testados por anos.
- **Tensão de alimentação de 3,3V** - Como o módulo de rádio escolhido opera em 3,3V, há a necessidade do microcontrolador também opere em 3,3V para que não necessite de conversores de níveis na interface de comunicação. O uso de tais conversores, pode interferir na comunicação se não forem bem dimensionados, além de somarem mais um custo ao projeto. Como o ATmega328P opera em 3,3 V ele atende esse requisito.

### 3.3.3 Conectividade

Conforme foi descrito na subseção 3.2.5 “conectividade” a tecnologia escolhida para a rede de sensores deveria atingir distâncias razoáveis. Considerando as três categorias de classificação das soluções descritas em [10], optamos pelo uso de uma solução *open source* sendo o protocolo Radiuino.

O Radiuino basicamente, consiste na integração da plataforma Arduino com transceptores de Radiofrequência. Os dispositivos físicos (Hardwares) podem ser programados no Arduino IDE tomando como base as bibliotecas do Radiuino [17]. O hardware compatível com a plataforma Radiuino basicamente é composto por um controlador ATmega328p e um transceptor com capacidade de transmitir informações via radio frequência. O transceptor é o CC1101, disponível comercialmente por meio de módulos que operam em várias frequências (nas faixas de 433 MHz e 915 MHz).

Neste projeto optou-se pelo uso do módulo CC1101 operando em 433 MHz, tal como ilustrado na Figura 8. Trata-se de um módulo de comunicação flexível e com suporte de código aberto na Internet, que utiliza o processador AVR ATmega328 e o transceptor TI CC1101 RF. Mais especificações técnicas deste dispositivo podem ser obtidas em [18].

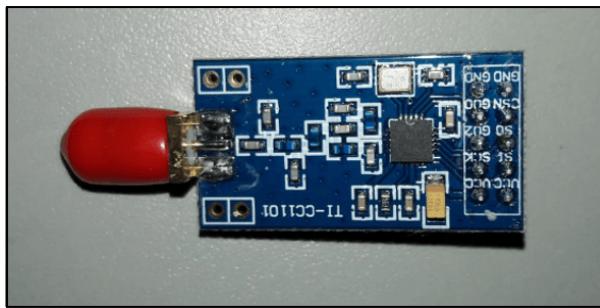


Figura 8: Módulo CC1101.

Com isso, a plataforma Radiuino permitiu criar uma rede de sensores sem fio (RSSF) com algumas vantagens como:

- **Plataforma open source** – O Radiuino tanto em hardware quanto em software trata-se de uma plataforma aberta, assim, qualquer usuário poderá adaptá-la conforme a sua necessidade;
- **Operação na faixa não licenciada** – Optando pelo uso do módulo CC1101, A transmissão dos dados do sistema ocorre pela banda não licenciada não sendo necessários gastos para esta finalidade, importando somente que os dispositivos utilizados sejam reconhecidos pelos órgãos regulamentadores.
- **Atende ao alcance mínimo desejado** – Na plataforma Radiuino o módulo transceptor CC1101 tem alcance de até 100m em ambientes *indoor* e de até 500m em

ambientes *outdoor*, o que atendeu perfeitamente a necessidade do projeto proposto, visto que à princípio são considerados ambientes *indoor* não muito extensos.

Apesar das vantagens citadas anteriormente, há uma desvantagem para o uso do protocolo RADIUINO em sua configuração padrão: a ausência de segurança deste. No entanto, o mesmo é flexível para este tipo de implementação, sendo assim, para trabalhos futuros, propõe-se implementar a segurança do protocolo<sup>8</sup>.

### 3.3.4 Borda

O *Gateway* foi padronizado em termos de iniciativas Open-Source para garantir suporte, escalabilidade, manutenção e flexibilização para atualizações futuras. Para tanto, foi utilizado um Raspberry Pi 3 (modelo B+), que atendeu perfeitamente as exigências destacadas na Fase 2 da metodologia. As principais especificações técnicas do Raspberry Pi 3 B+ são:

- Processador Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz;
- Clock de 1.4 GHz;
- Wifi 802.11 b/g/n/AC 2.4GHz e 5GHz;
- Memória RAM de 1GB;
- Bluetooth 4.2 BLE.

### 3.3.5 Armazenamento

Para armazenar as informações coletadas da rede de sensores, foi utilizado um servidor Zabbix instalado no Raspberry Pi 3 B+<sup>9</sup>. Atuando como sistema central do projeto, o Zabbix é basicamente composto por um servidor, um banco de dados MySQL e uma interface gráfica agradável aos usuários. Sendo assim, as informações foram armazenadas e exibidas no Zabbix, em especial para as informações relativas aos administradores (as informações de exibição serão abordadas mais a diante). Os principais módulos do Zabbix são:

- **Servidor Zabbix** – Trata-se do componente principal de todo o sistema, tanto é que, em muitas referências, o sistema Zabbix é tratado como “servidor Zabbix”.

---

<sup>8</sup>Existem muitas referências bibliográficas sobre segurança em redes de sensores sem fio. Em [19], por exemplo, o autor investiga a viabilidade de implementação de duas técnicas de criptografia no protocolo RADIUINO

<sup>9</sup>A equipe optou por trabalhar com servidores locais com o objetivo de facilitar a demonstração de funcionamento do protótipo. Mesmo assim, ressalta-se que é perfeitamente possível fazer a instalação do servidor Zabbix na nuvem, mas para isso, deveria ser contratado um serviço de dados (envolvendo custos).

Ele é responsável pelo processamento de dados, gerenciamento dos equipamentos monitorados, envio de alertas e exibição de relatórios;

- **Banco de dados** – É o local onde as informações são armazenadas. O Zabbix é compatível com as seguintes opções: MySQL/MariaDB, Oracle, PostgreSQL, SQLite e IBM DB2.
- **Interface Web** – Utilizada para a exibição das informações de forma gráfica e é por onde a maioria das configurações podem ser realizadas;
- **Zabbix Proxy** – É responsável pela coleta de dados de desempenho referente a um único servidor Zabbix, sendo capaz de coletar milhares de informações por segundo;
- **Agente Zabbix** – Trata-se de uma aplicação cliente do servidor. O agente é o Software capaz de coletar informações de um equipamento e enviá-las ao servidor.

### 3.3.6 Abstração

A camada de abstração foi feita utilizando o servidor Grafana. Trata-se de um servidor próprio para a exibição de informações de maneira elegante, uma vez que apresenta uma grande quantidade de recursos gráficos. Além disso, ressalta-se que existe a possibilidade da criação de várias instâncias para um mesmo projeto, que é um requisito fundamental para a aplicação em questão. Nesse caso, no servidor Grafana, foram criadas duas instâncias: de administração e do cliente. Essa divisão permitiu realizar a abstração de algumas informações que seriam exibidas diretamente ao cliente, tais como as informações de gerência de rede. Nesse caso, o cliente visualizou apenas as informações relativas às vagas do estacionamento, e as informações de rede foram abstraídas. Por outro lado, na instância de administração, as informações para o gerenciamento da rede foram exibidas. No caso deste projeto, tais informações se resumem às estatísticas da potência de recepção entre os terminais e da taxa de sucesso de pacotes.

### 3.3.7 Exibição

Na camada de exibição, as informações de interesse do cliente também foram exibidas no servidor Grafana. Dentre elas, é possível citar o monitoramento do estado de cada uma das vagas do estacionamento e o número de carros que ocuparam cada vaga a partir dos dias da semana. A partir dessas informações, outras informações podem ser extraídas, tais como o dia de maior movimento, número de clientes ao longo de um dia, qual é a vaga preferida dos clientes, número de motoristas com alguma deficiência física que utilizam o estacionamento e outras informações que podem ser importantes do ponto de vista de negócios do cliente.

## 4 Protótipo

Nesta seção serão abordados os detalhes a respeito do protótipo desenvolvido pela equipe, considerando os equipamentos e sistemas descritos na Fase 3 da metodologia. A princípio, o protótipo foi definido para funcionar com uma ERB que iria monitorar quatro vagas. Sendo assim, teria sido desenvolvida uma maquete com quatro vagas e um sensor para cada vaga. Três vagas seriam destinadas a veículos conduzidos por pessoas sem deficiência, e uma vaga seria destinada a pessoas com deficiência (PcD). Para as vagas normais, um LED verde seria utilizado para indicar que a vaga está disponível; na vaga de PcD, um LED azul seria usado para indicar sua disponibilidade. Nas duas situações, um LED vermelho seria usado para sinalizar que a vaga estaria ocupada. Entretanto, devido às condições atuais provocadas pela pandemia, a equipe desenvolveu um módulo, com todos os componentes necessários para monitorar apenas uma vaga. A equipe também desenvolveu uma placa de circuitos exclusiva a este projeto, que iria compor cada um dos módulos de monitoramento e a ERB, mas não foi possível a sua confecção por conta da pandemia. Mesmo assim, todo o desenvolvimento será apresentado nas próximas subseções.

### 4.1 Arquitetura geral do sistema

O protótipo desenvolvido foi baseado no modelo de referência contido na metodologia de 3 fases. Na Figura 9 é apresentada a arquitetura do sistema. São quatro vagas e para cada uma delas, um módulo de monitoramento teria sido desenvolvido<sup>10</sup>.

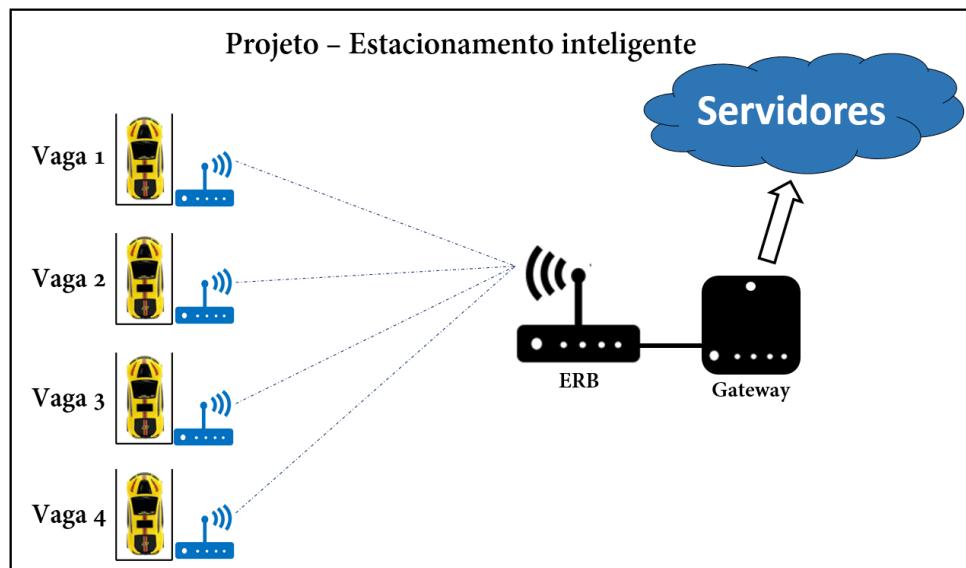


Figura 9: Arquitetura do sistema.

O protocolo de comunicação sem fio foi configurado para que os módulos de moni-

<sup>10</sup>Por conta da pandemia, apenas um módulo de monitoramento foi desenvolvido.

toramento recebessem uma solicitação da ERB a cada dois segundos. Os módulos de monitoramento foram responsáveis por : (i) monitorar o estado atual da vaga; (ii) realizar a contagem do número de carros que ocupou a vaga em função dos dias da semana e; (iii) realizar um processamento estatístico da intensidade do sinal recebido e taxa de sucesso de pacotes de dados. Para enviar as informações coletadas da rede de sensores, foi utilizado um *Gateway* integrado à ERB e as informações foram enviadas a um servidor Zabbix, por meio do suporte TCP/IP. O protocolo utilizado para esta comunicação foi o **Zabbix Sender**, que se trata de um *Socket TCP/IP*. E com o objetivo de realizar a abstração, foi implementada uma aplicação no servidor Grafana, permitindo a divisão do sistema em duas instâncias (uma para o cliente e outra para o administrador). Na sequência, serão apresentados os detalhes do desenvolvimento.

## 4.2 Sistema de monitoramento

Nesta subseção serão destacados as principais características do sistema de monitoramento da vaga, envolvendo os desenvolvimentos de hardware e software.

### 4.2.1 Componentes

Na Figura 10 é apresentado o módulo de monitoramento. Seus componentes são:

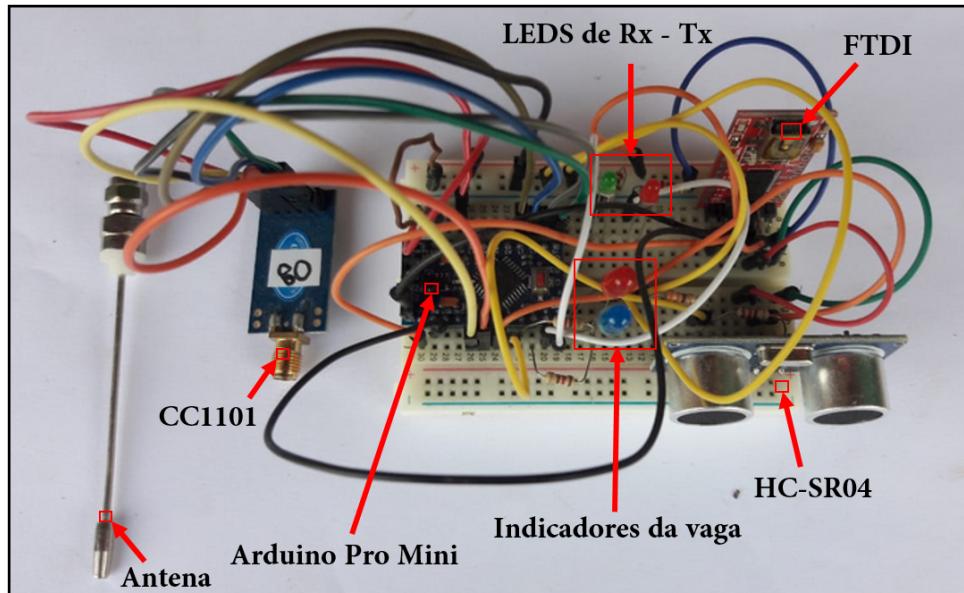


Figura 10: Módulo de monitoramento.

- Arduino Pro mini;
- Programador FDTI;
- Sensor HCSR-04;

- LEDS que indicam o estado da vaga;
- LEDs que indicam transmissão e recepção de dados da rede de sensores;
- Módulo de comunicação de RF CC1101;
- Antena de ganho 2,15 dBi.

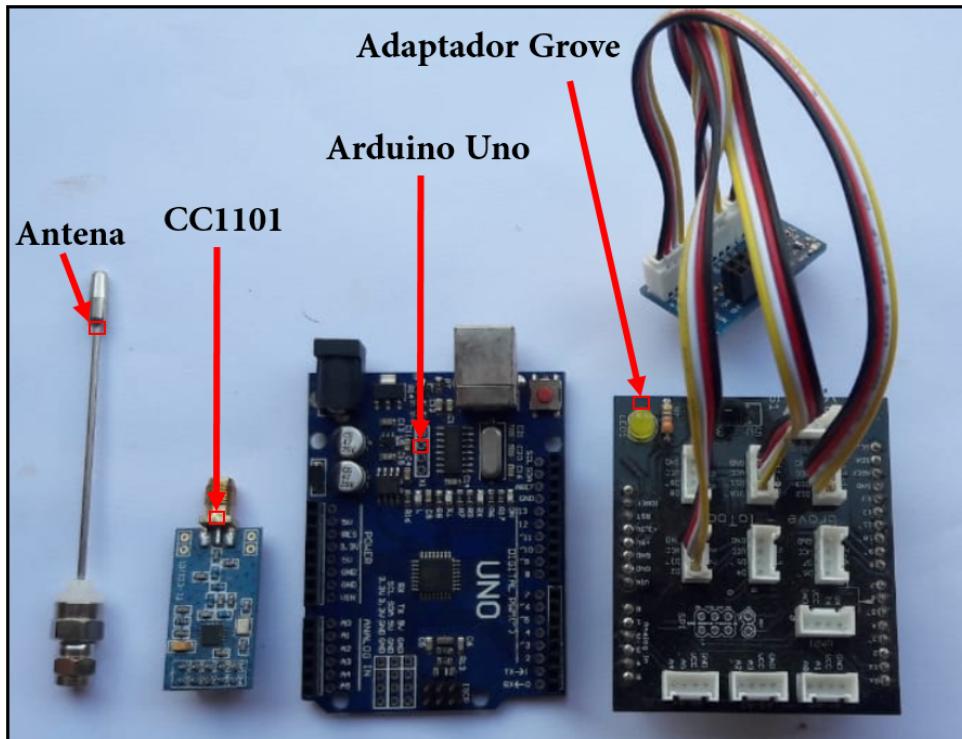


Figura 11: Estação Rádio Base.

Na Figura 11 são apresentados os componentes da ERB:

- Arduino Uno;
- Módulo de comunicação de RF CC1101;
- Antena de ganho 2,15 dBi;
- Adaptador Grove.

#### 4.2.2 Desenvolvimento da PCB

Para monitorar a vaga do estacionamento foi projetada uma placa de circuito impresso customizada utilizando o microcontrolador ATMega 328P contido no Arduino Pro Mini, sensor ultrassônico HC-SR04, transceptor de rádio CC1101 da *Texas Instruments*. Na figura 12 é apresentada a visualização 2D de como a placa irá ficar.

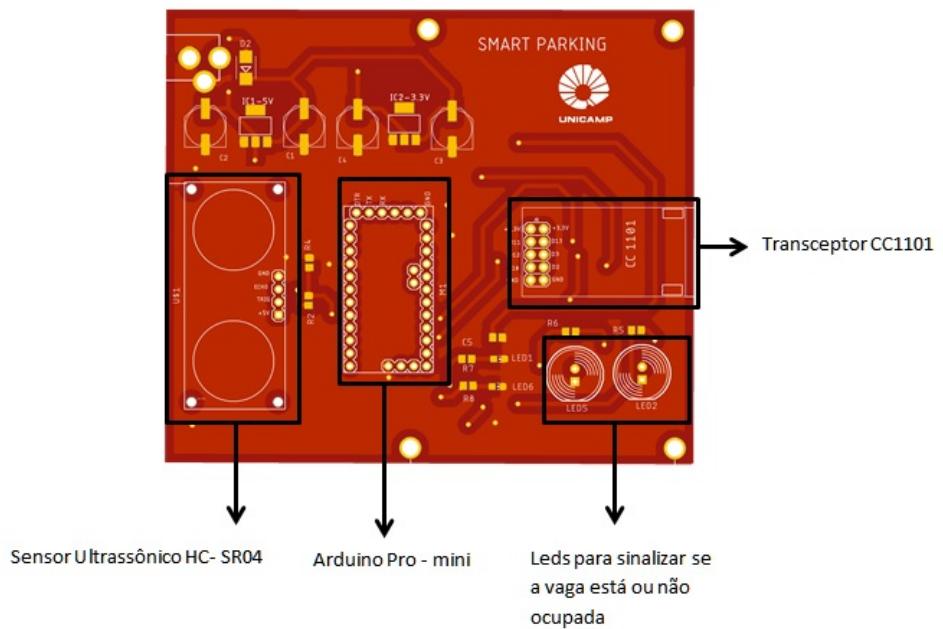


Figura 12: Módulo de monitoramento.

Os componentes principais dessa placa de circuito impresso são: o Arduino Pro - Mini responsável por realizar toda a lógica de funcionamento do projeto realizando a leitura do sensor ultrassônico para detectar se a vaga está ocupada ou não e assim acionar o LED azul ou verde para indicar que a vaga está livre ou acionar o LED vermelho para indicar que a vaga está ocupada e enviar o status da vaga utilizando o transceptor de rádio CC 1101 para o *Gateway*. A alimentação dessa placa pode ser feita por uma fonte de 9 V ou por uma bateria de 9 V, pois a placa possui dois reguladores de tensão: um de 3.3 V para alimentar o Arduino Pro mini e o transceptor CC 1101, e um regulador de 5 V para alimentar o sensor ultrassônico. O Arduino Pro Mini não é tolerante a entrada de 5 V, então não é permitido ligar o pino echo do sensor diretamente na porta digital do Pro Mini, com isso foi calculado um divisor resistivo para conectar o pino echo ao Arduino Pro Mini. O transceptor de rádio CC 1101 utiliza a alimentação de 3.3 V e utiliza a comunicação SPI (*Serial Peripheral Interface*). Além do layout da placa de circuito impresso foi feita a montagem do circuito do módulo de monitoramento utilizando o software Fritzing.

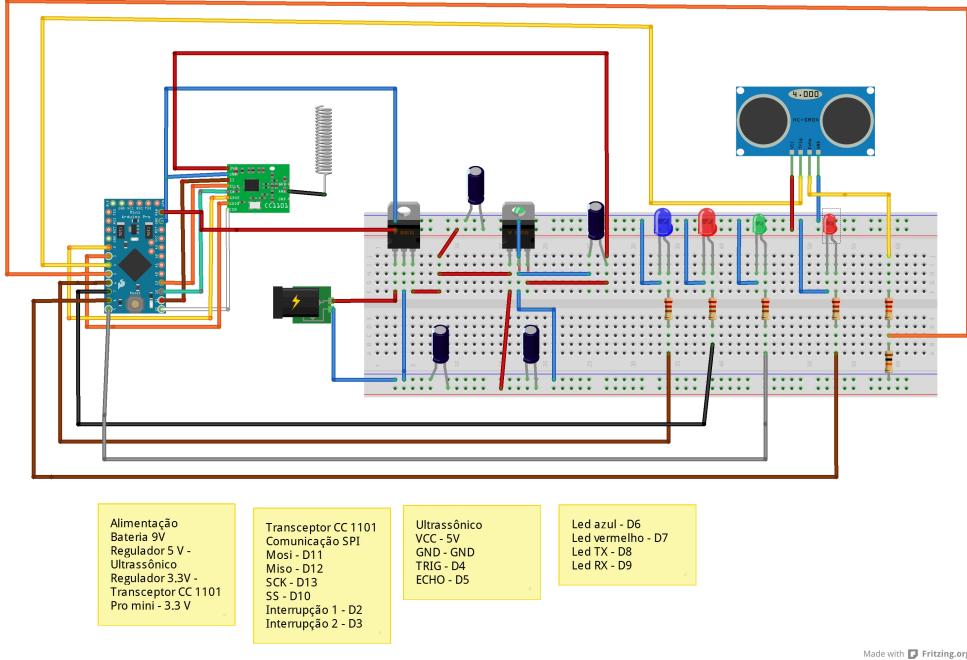


Figura 13: Montagem do circuito do módulo de monitoramento na protoboard.

#### 4.2.3 Firmware

O Firmware deste projeto foi composto por três partes. A primeira consistiu em desenvolver a aplicação para o monitoramento das vagas com base na lógica da contagem do número de carros. Além disso, recursos computacionais foram implementados para evitar, ao máximo, os erros de leitura do sensor.

Na Figura 14 é apresentado o fluxograma lógico. Inicialmente, visando obter a distância de referência em relação ao chão, foram feitas dez leituras para que uma distância média fosse calculada. Em seguida, se iniciaram as leituras para a verificação de ocupação das vagas e as distâncias obtidas eram comparadas com a distância de referência. Se a distância atual lida fosse menor do que a distância de referência, então a vaga estaria ocupada e isso era indicado por um LED vermelho; caso contrário, a vaga estaria livre (indicada por um LED verde ou azul, dependendo do tipo de vaga em questão).

Para a realização da contagem de vagas, foi feita uma comparação entre duas leituras de estado, tal como é apresentado no fluxograma. Nesse sentido, duas variáveis de controle foram utilizadas (para monitorar o estado atual e o estado anterior da vaga). Já a variável de armazenamento da contagem de vagas só era incrementada se o estado atual retornasse nível lógico 1 e o estado anterior resultasse em nível lógico 0. Isto é, no momento atual a vaga estaria ocupada e, anteriormente, estaria vazia. E ainda, ressalta-se que esse procedimento de comparação entre as variáveis de controle foi executado várias vezes (resultando em um tempo de resposta) para que o contador de vagas pudesse ser incrementado. Isso foi necessário para evitar possíveis contagens de maneira equivocada, como por exemplo uma pessoa passando na zona de atuação do sensor.

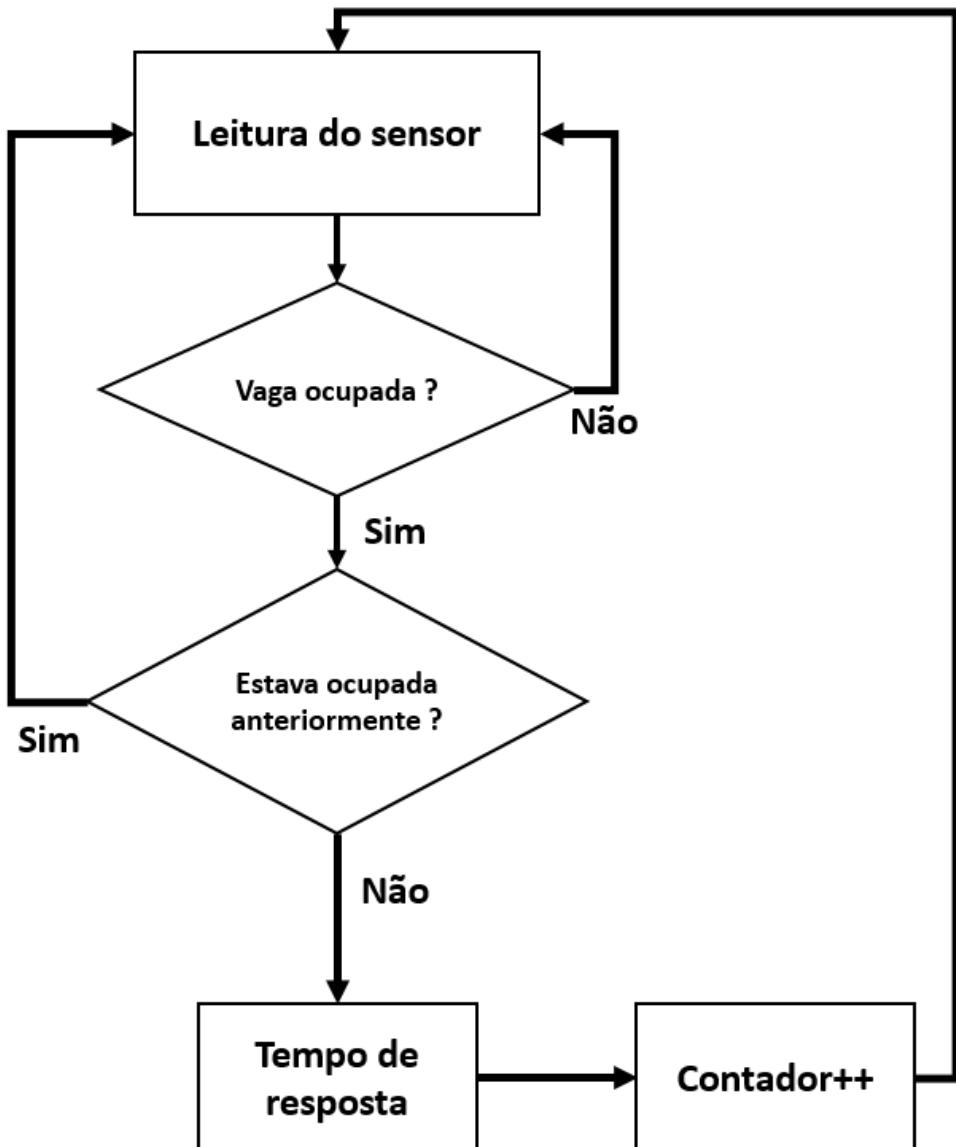


Figura 14: Fluxograma da contagem de vagas.

Na segunda parte, foi trabalhado o processamento estatístico e análises de intensidade de sinal e taxa de sucessos de pacotes (Packet Success Rate, PSR). Ressalta-se que, devido à complexidade do projeto e levando em consideração que um sistema real seja implementado com muitos terminais de rede, é possível que haja inviabilidade de que todo o processamento estatístico seja implementado no *Gateway*. Nesse sentido, optou-se pela estratégia de incorporar as análises estatísticas em cada um dos módulos de monitoramento. Nesse cenário, as estatísticas obtidas foram:

- **Potência média** – Uma média das últimas 10 medidas da potência foi obtida, caracterizando uma média móvel;
- **Desvio padrão** – é o desvio padrão da potência ao decorrer de 10 medições;
- **PSR** – É a taxa de sucesso de pacotes, obtida ao decorrer de um dia de medição.

A implementação do processamento estatístico no terminal de rede foi essencial para o funcionamento da estratégia utilizada, devido ao fato de que os terminais de rede sempre recebem os pacotes de dados que chegam da ERB (pelo fato de estarem na mesma frequência). Sendo assim, os processamentos são realizados localmente em cada terminal e, no momento correto, as informações já processadas são encaminhadas ao sistema de gerência. Na Figura 15 é apresentado o programa responsável pela realização das análises estatísticas da potência de recepção. Os valores de potência eram coletados mesmo que o pacote de dados não fosse para o devido terminal de rede.

```

for (int i = 0; i < 10; i++) {
    potLinear += 0.001 * pow(10.0, (vetorPotencia[i] / 10.0));
}

potLinear /= 10.0;
potMedAux = 10.0 * log10(potLinear / 0.001);
potMed = potMedAux;

for (int i = 0; i < 10; i++) {
    acumulador += pow((vetorPotencia[i] - potMed), 2.0);
}

acumulador /= 10.0;
acumulador = sqrt(acumulador);
desvPad = acumulador;

potLinear = potMedAux = acumulador = 0;

```

Figura 15: Código para as análises estatísticas.

E por fim, a terceira parte dizia respeito ao sistema de comunicação com a ERB, utilizando o CC1101 para a transmissão sem fio. Todas as informações foram inseridas no pacote de dados para que pudessem ser enviadas ao sistema de gerência. Na Figura 16 é apresentado o trecho de código em que as informações, descritas por meio de variáveis, são inseridas no pacote de dados. As variáveis são:

- **estadoAtual** – Indica o estado atual da vaga (livre ou ocupada);
- **contagemVagas** – Apresenta quantos carros ocuparam a vaga ao longo de um dia;
- **semana** – Possui a função de informar ao microcontrolador sobre o início do dia da semana;
- **counter** – Trata-se de um contador de pacotes, que foi utilizado para o cálculo da PSR.

```

inline void APP::receive(packet * pkt)
{
    //contaErros = (pkt->Data[0]*256 + pkt->Data[1]);
    pkt->Data[0] = (byte)(estadoAtual); //byte 16
    pkt->Data[1] = (byte) (contagemVagas / 256); //byte 17
    pkt->Data[2] = (byte) (contagemVagas % 256); //byte 18

    int potMed1 = abs(potMed * 100);
    pkt->Data[3] = (byte)((potMed1) / 256); // byte 19
    pkt->Data[4] = (byte)((potMed1) % 256); // byte 20

    int desvPad1 = desvPad * 100;
    pkt->Data[5] = (byte)((desvPad1) / 256); // byte 21
    pkt->Data[6] = (byte)((desvPad1) % 256); // byte 22

    semana = pkt->Data[7]; //byte 23
    if (semana == 1) {
        contagemVagas = 0;
        counter = 0;
        semana = 0;
    }

    pkt->Data[8] = (byte) (counter/16777216); //byte 24
    pkt->Data[9] = (byte) (((counter%16777216)/65536)); //byte 25
    pkt->Data[10] = (byte) (((((counter%16777216)%65536)/256)); //byte 26
    pkt->Data[11] = (byte) (((((counter%16777216)%65536)%256))); //byte 27

    // Envia o pacote para a camada de transporte
    Transp.send(pkt);
    return;
}

```

Figura 16: Informações inseridas no pacote de dados.

### 4.3 Gerência

Na etapa de gerência, utilizou-se o Raspberry Pi 3 (modelo B+) para a instalação dos servidores Zabbix e Grafana. O código desenvolvido em linguagem *Python* foi desenvolvido e integrado à base rádio para que os dados fossem coletados da rede de sensores. As informações foram enviadas para o banco de dados do servidor Zabbix. Nele, uma interface gráfica foi gerada para a visualização do estado da vaga.

Na Figura 17 é apresentada a interface gráfica de monitoramento do estado atual da vaga. Dessa maneira, os administradores do sistema poderão ter conhecimento a respeito da situação atual do estacionamento ao longo dos dias da semana. E ainda, o servidor Zabbix foi integrado ao Grafana, no qual a abstração foi realizada. No Grafana, foram criadas duas instâncias: a primeira delas foi exibida ao cliente e a segunda aos administradores do sistema. Na Figura 18 são exibidas as informações da aplicação. Além da informação sobre o estado atual da vaga, foi apresentada uma contagem de vagas ao longo dos dias da semana, para cada um dos módulos de monitoramento. Neste caso, ressalta-se que esses dados foram obtidos por meio de simulações, uma vez que apenas um módulo de monitoramento foi desenvolvido.

### Projeto Final – Estacionamento inteligente

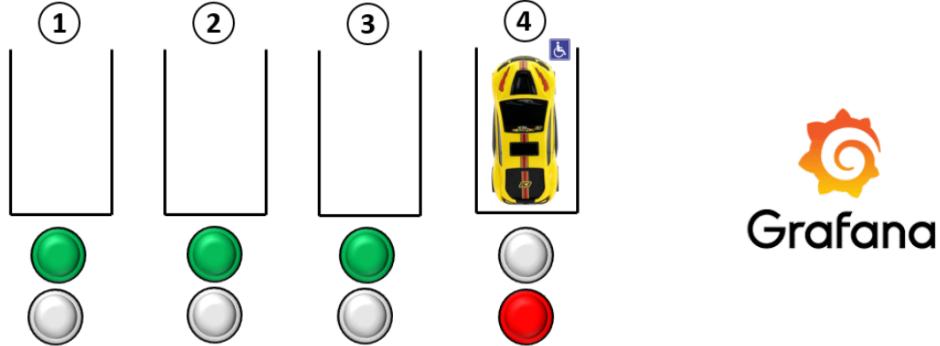


Figura 17: Monitoramento das vagas.

As simulações foram baseadas nas seguintes suposições: Considerando o cenário de um shopping, as vagas estarão disponíveis das 10 às 22h, totalizando 12 horas de funcionamento; supondo que, em média, um carro ocupa uma vaga durante um período de 1h30, então 8 carros ocupam a vaga no período de 12 horas. Além disso, também foi considerado que existem dias de maior movimento, como por exemplo nos fins de semana. E com base nessas suposições, simulações com valores aleatórios foram realizadas e os valores da Figura 18 foram obtidos.

Sensor 1 - DOMINGO <b>9</b>	Sensor 1 - SEGUNDA <b>4</b>	Sensor 1 - TERÇA <b>5</b>	Sensor 1 - QUARTA <b>7</b>	Sensor 1 - QUINTA <b>9</b>	Sensor 1 - SEXTA <b>7</b>	Sensor 1 - SÁBADO <b>9</b>	ESTADO ATUAL <b>LIVRE</b>
Sensor 2 - DOMINGO <b>7</b>	Sensor 2 - SEGUNDA <b>5</b>	Sensor 2 - TERÇA <b>7</b>	Sensor 2 - QUARTA <b>7</b>	Sensor 2 - QUINTA <b>8</b>	Sensor 2 - SEXTA <b>5</b>	Sensor 2 - SÁBADO <b>14</b>	ESTADO ATUAL <b>LIVRE</b>
Sensor 3 - DOMINGO <b>9</b>	Sensor 3 - SEGUNDA <b>7</b>	Sensor 3 - TERÇA <b>4</b>	Sensor 3 - QUARTA <b>9</b>	Sensor 3 - QUINTA <b>9</b>	Sensor 3 - SEXTA <b>8</b>	Sensor 3 - SÁBADO <b>15</b>	ESTADO ATUAL <b>LIVRE</b>
Sensor 4 - DOMINGO <b>13</b>	Sensor 4 - SEGUNDA <b>6</b>	Sensor 4 - TERÇA <b>9</b>	Sensor 4 - QUARTA <b>7</b>	Sensor 4 - QUINTA <b>4</b>	Sensor 4 - SEXTA <b>4</b>	Sensor 4 - SÁBADO <b>13</b>	ESTADO ATUAL <b>OCUPADA</b>

Figura 18: Instância de exibição ao cliente.

Na Figura 19 é apresentada a instância de gerência de dados, em que é possível obter a PSR do enlace de RF, a média e o desvio padrão da potência de recepção. Ressalta-se que essas informações são abstraídas do cliente, e estão disponíveis somente aos administradores e especialistas capazes de analisá-las.

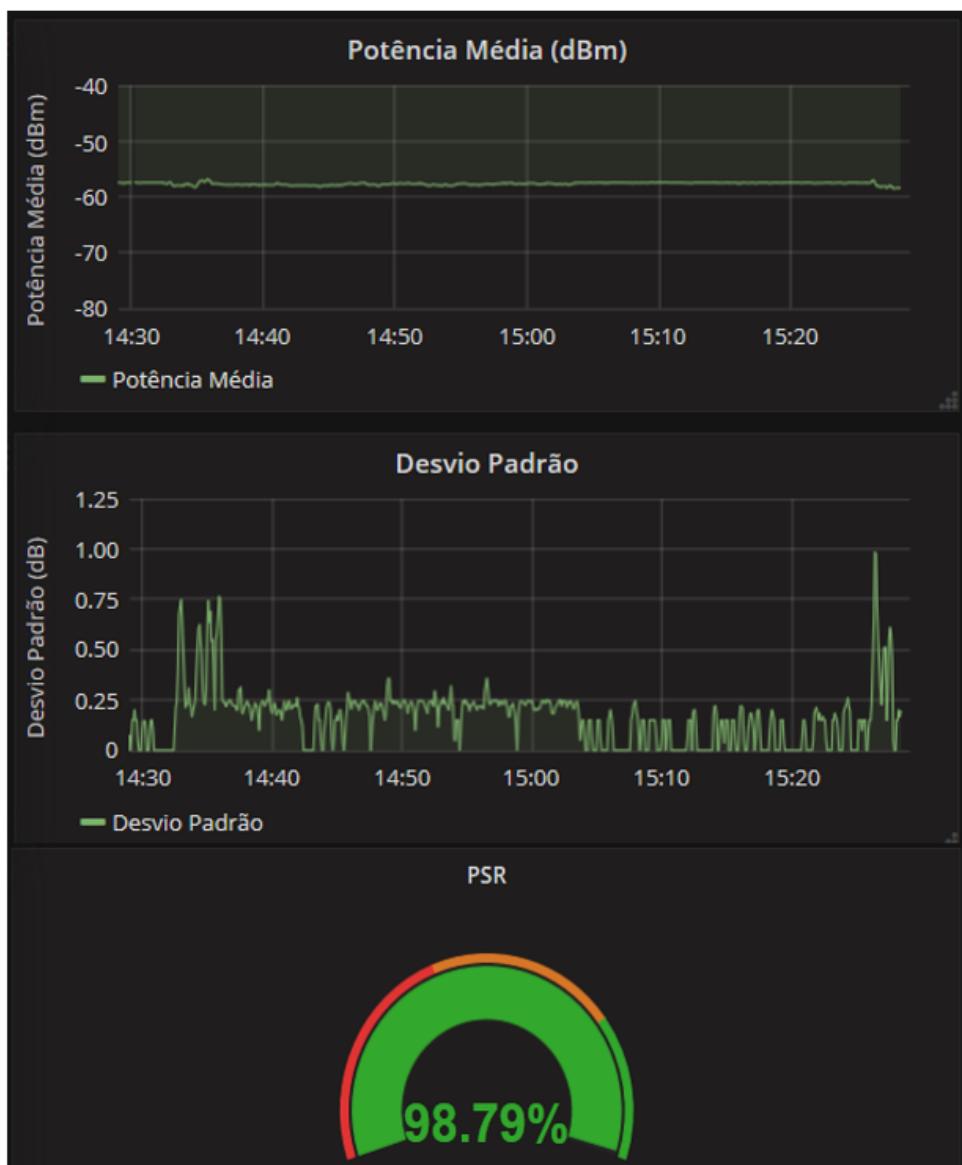


Figura 19: Instância de gerência.

## 5 Conclusões

Este trabalho apresentou como solução IoT um protótipo desenvolvido para atender a estacionamentos inteligentes. Inicialmente o protótipo propõe que sejam monitoradas quatro vagas de um estacionamento, sendo três vagas comuns, e uma vaga para pessoas com deficiência. Além disso, propõe-se que o mesmo atenda a estacionamentos pequenos e fechados, ou ambientes *indoor*.

O desenvolvimento do trabalho foi baseado na metodologia das três fases visando atender os seguintes aspectos: na fase 1 fez-se necessária a compreensão do negócio levando em conta a sua exigência, as regras do negócio bem como suas restrições e o especialista. Na fase 2, foi necessário realizar a definição dos requisitos em relação ao negócio em questão, para tanto seguimos o modelo de referência aberto OSRM para soluções IoT o qual divide a solução em níveis sendo eles: nível das coisas, nível dos dispositivos, nível de conectividade, nível de borda, nível de armazenamento, nível de abstração, nível de aplicação e nível do negócio. E por fim, na fase 3 foi proposta a implementação do negócio na qual são definidas as tecnologias disponíveis no mercado que atendem a implementação da solução. Na fase 3 também é utilizado o modelo OSRM, considerando que divisão em níveis facilita a visualização de cada etapa e identifica as necessidades em cada uma.

A solução proposta adere à seguinte restrição: o custo da solução deve ser mínimo, e a mesma, além de gerar informações úteis para os motoristas, também deve fornecer dados ao dono do estacionamento para que este possa desenvolver ações de gestão. Com isso, foi implementada uma ERB e módulo de monitoramento ambos a baixo custo, além de um *gateway* através do qual os dados seriam disponibilizados em nuvem. Além de ser uma solução a baixo custo, e ter capacidade de monitoramento, armazenamento, e transmissão de dados através de tecnologia sem fio, o protótipo em questão incorpora análises estatísticas dos dados nos módulos de monitoramento, bem como processamento estatístico e análises de intensidade de sinal e taxa de sucesso de pacotes (PSR). Por fim, foram realizados testes práticos a fim de verificar o funcionamento da solução. Reitera-se que devido à pandemia só foi possível realizar o experimento com apenas um sensor, ou seja, considerando apenas uma vaga, entretanto salientamos que em termos de firmware, o projeto foi implementado e atende às quatro vagas inicialmente propostas. Além disso, foi observado que o protótipo atende ao objetivo proposto, indicando quando uma vaga do estacionamento está ocupada, além de gerar e disponibilizar dados estatísticos através de uma interface com a finalidade de auxiliar o proprietário a melhorar a gestão do estabelecimento.

Como perspectivas futuras propõe-se a incorporação de técnicas de segurança ao protocolo Radiuino considerando a necessidade de criptografia para um projeto de grande porte. Além disso, considera-se a substituição dos sensores ultrassônicos por câmeras a fim de monitorar as vagas através de técnicas de processamento de imagens.

## ANEXO A - Orçamento das placas de circuito

Orçamento para fabricação de cinco e dez placas de circuito respectivamente. Para cinco placas, uma é destinada à ERB e as demais são destinadas aos módulos de monitoramento. Para dez placas, duas são destinadas às ERB e as demais são destinadas aos módulos de monitoramento.

<b>Cliente :</b> Email :danielvieira2006@gmail.com <b>Sr.(a) :</b> Daniel Vieira	<b>LUCH TEC - CIRCUITO IMPRESSO</b> Rua Antonio Tava - 1036 - Guarulhos . SP CEP : 07175-050 CNPJ: 23.597.757/0001-00 IE : 796.398.070.118 www.luchtec.com.br Tel : 2279-7096 Data :13/04/2019			
<b>ORÇAMENTO N°0109</b>				
<b>Gerbers 12.04.2020</b>				
<b>MODELO /EDIÇÃO:</b>				
MATERIAL :	FIBRA	ESPESSURA: 1,6MM		
LARGURA MM +- :	105	COMPRIMENTO MM+- : 90		
FACES :	2	CHAPADO : NÃO		
MASCARA :	2	COR : VERMELHA		
SIMOLOGIA :	2	COR : BRANCO		
HOT AIR	NÃO	ESTANHADA SIM		
<b>CUSTO ADICIONAIS</b>				
FOTOLITO	R\$ 130,00			
Teste Eletrico	R\$ -			
<b>Quantidade</b>	<b>Valor P/ UNIT</b>	<b>Total</b>	<b>Total C/ Adicionais</b>	<b>Prazo de entrega</b>
5	R\$ 77,00	R\$ 385,00	R\$ 515,00	11 Á 13 DIAS UTEIS
10	R\$ 42,00	R\$ 420,00	R\$ 550,00	11 Á 13 DIAS UTEIS
<b>Obs :</b> Teste eletrico e HAL Opcional Condição de pagamento:50% e 50% avista Validade do orçamento 15 dias Frete por Conta DESTINATARIO Fotolito cobrado separado De acordo com a produção de complexidade de PCI poderão ocorrer variação de +/- 10% nas quantidades solicitadas para produções.				

Figura 20: Orçamento para fabricação das placas de circuito

## Referências

- [1] FERREIRA L.C.B.C. et al. A pbl - based methodology for iot teaching. *IEEE Communications Magazine*, 57(11):20–26, november 2019.
- [2] TEKOUABOU S. C. K. et al. Improving parking availability prediction in smart cities with iot and ensemble-based model. *Journal of King Saud University-Computer and Information Sciences*, 2020.
- [3] C. PERERA. *Sensing as a service for internet of things: A roadmap*. Leanpub Publishers, 2017.
- [4] MAHENDRA B. M. et al. Iot based sensor enabled smart car parking for advanced driver assistance system. *2nd IEEE International Conference on Recent Trends in Electronics, Information e Communication Technology - RTEICT*, pages 2188–2193, 2017.
- [5] LOOKMUANG R. ; NAMBU T. K.; USANAVASIN S. Smart parking using iot technology. *5th International Conference on Business and Industrial research (ICBIR)*. IEEE, pages 1–6, 2018.
- [6] FORMIGA A.A. SOUSA B.J., JUNIOR J.J.L.D. *Introdução à programação*. Editora UFPB, 2014.
- [7] M. Kocakulak and I. Butun. An overview of wireless sensor networks towards internet of things. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–6, Jan 2017.
- [8] Dicionário Online Dicio. Significado de eletrônica., Maio 2020.  
<https://www.dicio.com.br/elettronica/>.
- [9] MARTINS N. A. *Sistemas microcontrolados. Uma abordagem com o Microcontrolador PIC 16F84*. Novatec Ltda, 2005.
- [10] DEO A.L.B. Proposta de um modelo open source de referência para iot - foco em pme, Junho 2018. <http://tede.bibliotecadigital.puc-campinas.edu.br:8080/jspui/handle/tede/1168>.
- [11] FRUET et al. Aula 1 - a internet das coisas fundamentos e oficina., Junho 2020.  
<https://classroom.google.com/w/NjE40Dg10Tk5MDda/t/all>.
- [12] Elec Freaks. Ultrassonic ranging module hc-sr04, Junho 2020.  
[https://img.filipeflop.com/files/download/Datasheet\\_HCSR04.pdf](https://img.filipeflop.com/files/download/Datasheet_HCSR04.pdf).

- [13] G1 Globo.com. G1 andou no novo fiat mille economy, Junho 2020.  
<http://g1.globo.com/Noticias/Carros/0,,MUL740235-9658,00-G+ANDOU+NO+NOVO+FIAT+MILLE+ECONOMY.html>.
- [14] LIMA D.S. Um sistema oportunista para detecção de vagas de estacionamento utilizando placas inteligentes acopladas em câmeras de segurança, Junho 2014.  
<http://www.repository.ufal.br/handle/riufal/1610>.
- [15] VIDAL V. Sensor ultrassônico hc-sr04 com arduino, Junho 2017.  
<https://blog.eletrogate.com/sensor-ultrassonico-hc-sr04-com-arduino/>.
- [16] Microchip. Atmega328p, Junho 2020.  
[http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf).
- [17] Radiuino. O que é, Junho 2020. <http://radiuino.cc/o-que-e/>.
- [18] Texas Instruments. Low-power sub-1 ghz rf transceiver, Junho 2017.  
[https://www.ti.com/lit/ds/symlink/cc1101.pdf?ts=1592856509108&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/cc1101.pdf?ts=1592856509108&ref_url=https%253A%252F%252Fwww.google.com%252F).
- [19] R. N. de Freitas. Impacto no desempenho de rssf com diferentes técnicas de cifragem. Master's thesis, Pontifícia Universidade Católica de Campinas, 2018.