



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**MATHEUS CAVALCANTI DOS SANTOS**

**PROTÓTIPO DE SMART PARKING PARA CIDADES INTELIGENTES E  
MOBILIDADE URBANA**

**Assis/SP**

**2019**



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**MATHEUS CAVALCANTI DOS SANTOS**

**PROTÓTIPO DE SMART PARKING PARA CIDADES INTELIGENTES E  
MOBILIDADE URBANA**

Projeto de pesquisa apresentado ao Curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a):** Matheus Cavalcanti dos Santos

**Orientador(a):** Prof. MSc. Guilherme de Cleve Farto

Assis/SP

2019

# PROTÓTIPO DE SMART PARKING PARA CIDADES INTELIGENTES E MOBILIDADE URBANA

MATHEUS CAVALCANTI DOS SANTOS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

**Orientador:** \_\_\_\_\_

Prof. MSc. Guilherme de Cleve Farto

**Examinador:** \_\_\_\_\_

Dr. Almir Rogério Camolesi

Assis/SP

2019

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente a Deus por essa oportunidade, sem Ele não estaríamos aqui para ter conquistas como essa.

Ao professor, Guilherme de Cleve Farto, pela ajuda e orientação constantes, principalmente pela confiança na minha capacidade, por sempre estar disponível a ajudar e tirar dúvidas, jamais perderei o conhecimento e conceitos que foram passados a mim.

Aos amigos, João Victor Viel, Renato Virto Moreira, Lucas Correia, Gabriel Alan Madureiro, Rodrigo Coradi, Mario Villani e Leonardo Brancalhão, que sempre me ajudaram e contribuíram de alguma maneira no meu conhecimento, sem vocês não seria capaz de chegar até aqui, a amizade de vocês é um grande presente, espero levar para toda a vida.

Aos familiares e minha namorada, por sempre me apoiarem durante toda a trajetória do curso, todos vocês fazem parte dessa conquista e ajudaram no meu crescimento pessoal, devo tudo o que sou a minha família.

## RESUMO

Em consequência do constante crescimento do número de carros nas cidades, as pessoas estão cada vez mais com dificuldades de se locomoverem, principalmente em grandes cidades, onde o número de carros é ainda maior. Com o aumento significativo deste número, atividades rotineiras de milhões de pessoas têm se tornado problemáticas, um exemplo é o momento em que os condutores de veículos precisam estacionar seus veículos para irem trabalhar, fazer compras, sair com a família, ir ao supermercado, entre outras tarefas do dia a dia.

A grande dificuldade em encontrar uma vaga para estacionar não é apenas por conta do elevado número de veículos, mas também por infraestrutura das próprias cidades, e estacionamentos públicos e privados. Devido a esse problema, a proposta desse trabalho é apresentar uma solução de baixo custo e simples utilização. Um protótipo de Smart Parking para a mobilidade urbana, ou seja, um projeto de IoT com finalidade de ajudar motoristas a encontrarem vagas disponíveis usando apenas o seu celular, sem ter necessidade de perder tempo procurando vagas de forma convencional.

**Palavras-chave:** Smart Parking, Mobilidade Urbana, Cidades Inteligentes, Internet das Coisas.

## **ABSTRACT**

As a result of the steady increase in the number of cars in cities, people are struggling to get around, especially in big cities, where the number of cars is even bigger. With the significant increase in this number, routine activities of millions of people have become problematic, an example is when drivers need to park their cars to go to work, shop, go out with the family, going to the supermarket, and other day-to-day tasks.

The great difficulty in finding a parking space is not only due to the high number of vehicles, but also by infrastructure of the cities themselves, such as public and private parking. Due to this problem, the goal of this project is to present a of low cost and simple use solution. A Smart Parking prototype for urban mobility, as an IoT project to help drivers find available places using only their mobile phone, without having to waste time looking for parking spots in a conventional way.

**Keywords:** Smart Parking, Urban Mobility, Smart Cities, Internet of Things.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1:</b> Gráfico de dispositivos por pessoas.....	14
<b>Figura 2:</b> Pirâmide da sabedoria .....	15
<b>Figura 3:</b> Arduino Uno.....	16
<b>Figura 4:</b> Componentes do Arduino Uno .....	17
<b>Figura 5:</b> Exemplo de Código Arduino.....	19
<b>Figura 6:</b> Sensor Ultrassônico HC-SR04.....	20
<b>Figura 7:</b> Raspberry Pi .....	21
<b>Figura 8:</b> Smart Parking .....	24
<b>Figura 9:</b> Sequência de sinais de controle UART.....	29
<b>Figura 10:</b> Detecção de Objeto, Sensor HC-SR04.....	30
<b>Figura 11:</b> Transmissão de Dados, Sensor HC-SR04.....	31
<b>Figura 12:</b> Conexão Arduino x Sensor.....	31
<b>Figura 13:</b> Código Comunicação Arduino.....	32
<b>Figura 14:</b> Código Comunicação Raspberry.....	33
<b>Figura 15:</b> Arquitetura Web Service.....	35
<b>Figura 16:</b> Front-end e Back-end.....	36
<b>Figura 17:</b> Aplicação Web RESTful.....	37
<b>Figura 18:</b> Arquitetura do projeto .....	40
<b>Figura 19:</b> API Python .....	41
<b>Figura 20:</b> Smart Parking Menu .....	42
<b>Figura 21:</b> Smart Parking vagas .....	43
<b>Figura 22:</b> Smart Parking consultar vagas.....	44

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>10</b>
1.2 OBJETIVOS.....	11
1.3 JUSTIFICATIVAS .....	12
1.4 MOTIVAÇÃO .....	12
1.5 PERSPECTIVAS DE CONTRIBUIÇÃO .....	13
1.6 METODOLOGIA DE PESQUISA .....	13
1.7 ESTRUTURA DO TRABALHO.....	14
<b>2. INTERNET DAS COISAS .....</b>	<b>15</b>
2.1 CONCEITO.....	15
2.2 IMPORTÂNCIA DO IOT .....	16
<b>3. ARDUINO E SENSORES .....</b>	<b>18</b>
3.1 INTRODUÇÃO.....	18
3.2 ARDUINO UNO .....	19
3.3 SENSORES.....	21
3.4 SENSOR ULTRASSÔNICO .....	22
3.5 SENSOR ULTRASSÔNICO HC-SR04.....	22
<b>4. RASPBERRY PI .....</b>	<b>24</b>
4.1 INTRODUÇÃO.....	24
4.2 ARQUITETURA.....	25
<b>5. PROPOSTA DO TRABALHO .....</b>	<b>26</b>
5.1 SMART PARKING.....	26
5.2. TECNOLOGIAS E RECURSOS ADOTADOS.....	27

5.3. PROTOCOLO UART .....	27
<b>6. DESENVOLVIMENTO DO TRABALHO .....</b>	<b>29</b>
6.1. COMUNICAÇÃO SENSOR HC-SR04 X ARDUINO.....	29
6.2. COMUNICAÇÃO ARDUINO X RASPBERRY PI COM PROTOCOLO UART .	31
6.3. COMUNICAÇÃO RASPBERRY X SPRING BOOT.....	33
6.4. APLICAÇÃO WEB SPRING BOOT & ANGULAR .....	35
6.4. PROJETO DESENVOLVIDO .....	37
<b>7. CONCLUSÕES .....</b>	<b>41</b>
<b>REFERÊNCIAS.....</b>	<b>43</b>

# 1. INTRODUÇÃO

Atualmente, pode-se observar que a humanidade tem evoluído a todo o instante, principalmente em áreas relacionadas a tecnologia, onde se gasta a maior quantidade de tempo e dinheiro para solucionar e apresentar novas técnicas e maneiras diferentes para ajudar as pessoas, e solucionar os problemas que afetam a sua qualidade de vida.

Apesar da tecnologia resolver problemas que as pessoas se deparam em suas vidas, ainda existem áreas onde os problemas não estão completamente resolvidos. Um bom exemplo que é o foco desse trabalho é em relação ao grande volume de automóveis nas pequenas e grandes cidades. Pesquisas feitas pelo Denatran disponibilizadas no site da AND – Associação Nacional dos Detrans, demonstram um aumento significativo do número de veículos no país, onde em 2009 circulavam aproximadamente 29 milhões de veículos e em 2017 foi registrado um pouco mais de 43 milhões de veículos no Brasil (Denatran, 2017). Além do problema de locomoção das pessoas pelo fato das grandes rodovias não suportarem tantos veículos rodando ao mesmo tempo, um dos maiores problemas que existe na maioria das cidades é a dificuldade de achar um lugar para deixar os seus carros estacionados por uma determinada quantidade de tempo.

Um tema que vem crescendo ao longo dos anos é o termo Internet of Things (IoT) traduzido em português Internet das Coisas. Segundo ao Amazon Web Services, IoT é um termo que foi criado por Kevin Ashton, um tecnólogo britânico que criou um sistema de sensores conectando o mundo físico à Internet. A IoT tem como principal conceito, criar esses sistemas fazendo uma conexão direta ao mundo real, como objetos, animados ou inanimados, geralmente esses sistemas ficam armazenados 100% em nuvem, ou seja, ficam sempre online. São infinitas possibilidades de desenvolver projetos que utilizam esse conceito, inclusive áreas, como comércio, indústria, transporte, agricultura, engenharia.

Em relação ao transporte, observa-se que é uma área em constante evolução tanto na tecnologia utilizada nos veículos como na busca de aprimoramento na locomoção geral desse grande número de automóveis. Ao realizar uma simples pesquisa pode-se obter várias informações e artigos relacionados aos estacionamentos inteligentes, e outras soluções para esse problema que atinge uma grande parte dos condutores. Atualmente pode se encontrar algumas teses e artigos com objetivos semelhantes a esse trabalho. Uma tese feita para a obtenção de um mestrado (PINHEIRO, 2017) propôs uma aplicação

móvel para a gestão inteligente de locais de estacionamento, onde o usuário tem a localização dos estacionamentos com vagas disponíveis. Foi utilizado sensores para fazer a monitoração das vagas e mostrando ao usuário as mesmas disponíveis. Atualmente existe uma empresa da Escócia a Smart Parking Ltd que possui uma proposta onde há utilização de sensores mais modernos para monitorar estacionamentos privados e públicos, onde nos privados existe uma central na entrada do estacionamento que um monitor exibe a quantidade de vagas disponíveis e quais estão livres, para ajudar o motorista a encontrar sua vaga mais rapidamente. Apesar de estar sendo implementado em vários países da Europa tem um custo mais elevado pela tecnologia utilizada nesses estacionamentos.

Um grande desafio dessa área é trazer as mesmas possibilidades de utilização com opções de dispositivos mais baratos e um melhor custo-benefício em um contexto geral. Tendo em vista essa dificuldade (FRANCO, 2014) sugeriu alternativas de soluções para Smart Parking com tecnologias sem fios com maior facilidade de acesso e custos. O autor propôs soluções utilizando sensores magnéticos, sensores baseados em ultrassons, processamento de imagens e contadores de entrada/saída. Em sua análise o autor pode concluir que os sensores magnéticos se adequam melhor em ambientes indoor e outdoor, além de apresentar melhores adaptações as condições adversas de clima e terem um menor custo.

O presente trabalho tem como objetivo desenvolver um sistema para estacionamentos com a utilização de sensores de proximidade, monitorar quais vagas estão ocupadas e quais estão livres. Com o mesmo sistema integrado com uma aplicação móvel disponível para o usuário onde por meio do aplicativo com GPS o usuário poderá localizar os estacionamentos que possuem vagas disponíveis.

## 1.2 OBJETIVOS

Esse trabalho apresenta o objetivo, desenvolver uma aplicação móvel integrada com um sistema local, encontrado no próprio estacionamento físico. Com o uso de sensores de proximidade o usuário pelo seu celular poderá encontrar o estacionamento com vagas disponíveis, por meio de uma busca pelo seu destino, no GPS o aplicativo informará vagas próximas ao seu destino definido pelo mesmo. No local do estacionamento onde o sistema ficará instalado será exibida todas as vagas existentes, as ocupadas e livres para que o funcionário possa monitorar todas as vagas com maior eficiência.

### 1.3 JUSTIFICATIVAS

Com o aumento significativo da população mundial, soluções e ideias que surgem a todo o momento são essenciais para a melhora na qualidade de vida da população. Estudos e pesquisas relacionadas ao IoT tem auxiliado a vida de muitas pessoas, pois os benefícios que ela traz são bastante relevantes. O estudo desse trabalho vem de encontro com esses princípios, a busca de soluções alternativas para problemas mais comuns do cotidiano das pessoas. Um exemplo do grande crescimento do IoT é a empresa Libelium, totalmente focada nesse tipo de tecnologia, possui mais de 10000 desenvolvedores pelo mundo, em empresas localizadas em 120 países diferentes, ela possui um Marketplace com grande variedade de soluções com IoT, inclusive Smart Parking.

Segundo (PINTO, 2017) os estacionamentos inteligentes podem ser um grande sucesso, pois reduz os problemas de congestionamento e poluição dentro das cidades. Porém o sucesso depende muito do comprometimento dos próprios usuários. A redução dos veículos em busca de vagas de estacionamento é positiva, não só para os motoristas com a economia de tempo de combustível, mas para um todo geral, onde o congestionamento seria reduzido, tornando a locomoção um tanto quanto mais tranquila e objetiva.

### 1.4 MOTIVAÇÃO

Uma das maiores motivações para a pesquisa desse trabalho, foi a dificuldade dos dias de hoje e o estresse que sempre há, durante a locomoção e a demora para se encontrar uma vaga para deixar o veículo, além do combustível gasto durante esses processos. É um problema que atinge vários motoristas em muitas cidades, principalmente nas grandes metrópoles.

A decisão por escolher o tema IoT, foi por ser uma área em grande desenvolvimento, afim de contribuir nas pesquisas relacionadas a essa área de estacionamentos inteligentes. Com intuito de utilizar as tecnologias atuais e apresentar uma solução alternativa com um custo um pouco mais reduzido para ser implantado.

A facilidade que o usuário terá em utilizar e se beneficiar com o aplicativo móvel, foi de grande importância para a concretização da ideia, tendo em vista principalmente a agilidade no uso e o tempo ganho para encontrar o estacionamento com vagas disponíveis.

## 1.5 PERSPECTIVAS DE CONTRIBUIÇÃO

Este trabalho poderá contribuir com futuros estudos envolvidos com IoT relacionados aos sensores de movimento, tendo como principal característica seu custo-benefício, pois o mesmo busca apresentar o que se tem de mais atual na área, sem deixar de ser acessível a qualquer profissional da área. Possuindo uma solução simples ao problema apresentado na introdução, de forma que seja uma porta a novas ideias relacionadas ao cotidiano das pessoas e seus desafios.

Traz como contribuição direta ao usuário da aplicação, uma melhora significativa à sua qualidade de vida, em vista do tempo perdido no trânsito, que tem afetado grande parte do dia de milhares de pessoas.

A tecnologia com a criação da Internet foi uma grande revolução, a Internet das Coisas tem se destacado mundialmente trazendo o mesmo impacto que a Internet teve, esse trabalho tem como proposta analisar essa abrangência do IoT e demonstrar a capacidade do impacto da tecnologia nas atividades mais comuns das pessoas, tornando possível desenvolver mais soluções com os mesmos objetivos.

## 1.6 METODOLOGIA DE PESQUISA

O presente trabalho tem inicialmente como metodologia estudos e análise de projetos retirados de artigos científicos, TCC, teses, revistas, livros, empresas específicas da área, com visão ampla do objeto a ser estudado, tendo como conceito parte da pesquisa descritiva onde, serão levantados informações e dados sobre o tema, visando uma análise no ambiente natural, para que exista uma maior concretização das informações identificadas.

Após o levantamento dos dados e análises, o trabalho terá como característica parte da pesquisa explicativa onde haverá explicação dos motivos, causas e problemas relacionados ao tema principal do trabalho. Por fim a metodologia se encerrará com o desenvolvimento prático de um software com documentação seguindo os padrões da engenharia de software integrado com sensores de proximidade estudados previamente, concluindo a parte prática com uma aplicação móvel onde estará conectada a internet e integrada com o sistema principal do trabalho.

## 1.7 ESTRUTURA DO TRABALHO

A estrutura deste trabalho será composta das seguintes partes:

- **Capítulo 1 – Introdução:** Neste capítulo é contextualizada a área de estudo e apresentará os objetivos, justificativas, motivação, perspectivas de contribuição e metodologia de pesquisa para o desenvolvimento deste trabalho.
- **Capítulo 2 – Internet das Coisas:** Neste capítulo, introduz-se sobre o que é e como funciona a Internet das Coisas (IoT) e a importância da mesma na evolução da tecnologia atual.
- **Capítulo 3 – Arduino e Sensores:** Neste capítulo, é contextualizada a placa Arduino, seu funcionamento e proposta de uso, juntamente com os sensores, analisando as possibilidades que se pode ter ao usar esses componentes.
- **Capítulo 4 – Raspberry Pi:** Neste capítulo, é apresentado o conceito da placa, analisando versões e apresentando sua arquitetura.
- **Capítulo 5 – Proposta de Trabalho:** Neste capítulo, apresenta-se o *Smart Parking* como solução ao tráfego da atualidade, bem como as tecnologias e recursos adotados.
- **Capítulo 6 – Conclusão Parcial:** Neste capítulo, apresentam-se as vantagens da implementação do Smart Parking, e o benefícios que as pessoas podem ter.
- **Referências**

## 2. INTERNET DAS COISAS

Esse capítulo tem como objetivo, apresentar conceitos e exemplificar com projetos reais de IoT, partindo para a área desse trabalho.

### 2.1 CONCEITO

O conceito de Internet das Coisas (IoT) não é novo, ele existe há vinte anos, quando a internet ficou mais popular, já se pensava em conectar equipamentos com outros através da internet e redes. Esse conceito está cada vez mais em alta, pois com a tecnologia atual, a facilidade que se tem de desenvolver e conectar coisas tem sido o fator chave pelo qual muitas empresas têm tomado direções para a utilização do IoT em diversas áreas.

Isso se deve também ao crescimento explosivo dos smartphones e tablets que rapidamente se popularizaram e atingiram mais de 12 bilhões de dispositivos no mundo, isso em 2010. Trazendo a estatística de quase 2 dispositivos por pessoa (CISCO IBSG, 2011). A mesma fonte prevê um aumento significativo desse número de dispositivos até 2020.

A imagem abaixo, demonstra uma previsão de quantos dispositivos estarão conectados até 2020, pelo número da população mundial.

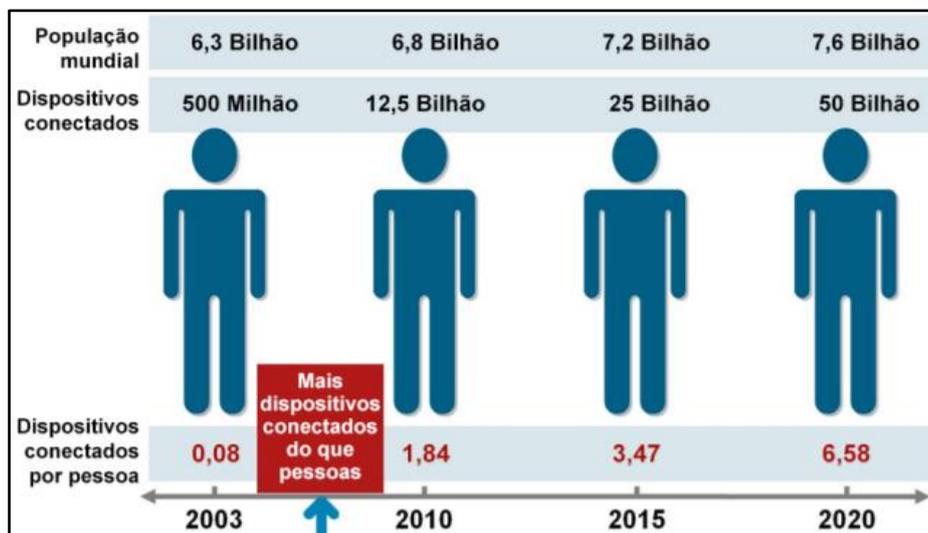


Figura 1: Gráfico dispositivos por pessoas

Fonte: Cisco IBSG, abril de 2011

Segundo (OLIVEIRA, 2017) a Internet das Coisas não é apenas ligar as “coisas” pela internet, mas também as tornar inteligentes, sendo processar e coletar informações ou

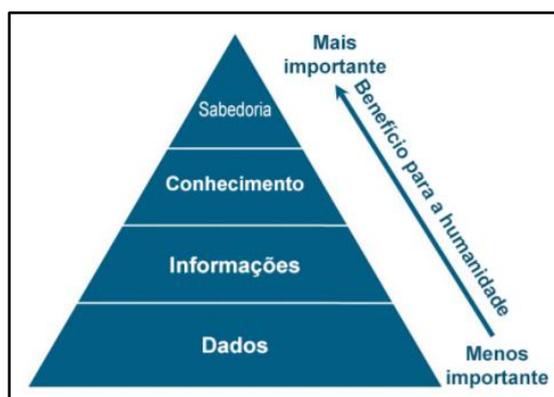
dados do ambiente em que estão conectadas. Afinal quando se tem implementado, é capaz de mudar toda a forma na qual nos relacionamos com essas coisas, com uso de sensores, câmeras, painéis e dispositivos relacionados.

## 2.2 IMPORTÂNCIA DO IOT

De acordo com (EVANS, 2011) o IoT é essencial para o progresso humano, pois com o crescimento em alta velocidade da população, as pessoas desejam ter vidas mais saudáveis, combinando esse desejo com a capacidade de evolução da Internet(IoT) para realizar as análises, coletas e distribuições dos dados em grande escala, a humanidade obterá todo o conhecimento necessário para não apenas viver mas prosperar nos próximos anos e séculos. Principalmente porque segundo o autor os seres humanos transformam os dados em sabedoria, e nós evoluímos porque nos comunicamos.

Essa atitude de compartilhar o conhecimento e as informações vem de tempos mais antigos, principalmente quando o ser humano descobriu o fogo, que foi possível avançar de forma gigantesca através da descoberta. Um exemplo mais recente é a descoberta da estrutura de hélice do DNA, as moléculas que carregam informações genéticas de uma geração para a outra. Foi extremamente importante para a medicina e a genética após os biólogos James Watson e Francis Crick em abril de 1953, terem publicado seu artigo.

Na figura 2, é representada em forma de pirâmide o conceito do que se tem de mais benéfico para a humanidade, do menos importante ao mais importante.



**Figura 2:** Pirâmide da sabedoria

Fonte: Cisco IBSG, abril de 2011

É importante destacar que é uma relação direta entre os dados e a sabedoria, sendo que quanto mais dados são gerados, mais conhecimento e sabedoria as pessoas adquirem. Com isso a IoT aumenta consideravelmente a entrada desses dados para o

processamento, juntando isso com o poder da Internet, as pessoas têm muito mais capacidade de se desenvolverem e evoluir tecnologicamente.

Apesar da IoT ser muito efetiva nos projetos e situações propostas, ainda com a tecnologia atual existem algumas barreiras e desafios que impedem a mesma de evoluir e se estabilizar nas áreas em que estão. Um exemplo disso é a energia dos sensores que são utilizados, pois para que a IoT atinja seu potencial máximo é necessário ter dispositivos e sensores autossustentáveis, para que tenha um menor custo financeiro e também um menor impacto ambiental.

### 3. ARDUINO E SENSORES

O presente capítulo tem como objetivo apresentar a placa Arduino, dissertando sobre suas características, funcionamento e fins específicos. Juntamente com a placa, será apresentado alguns tipos de sensores de proximidade, mais utilizados para fins de estudo e projetos acadêmicos.

#### 3.1. INTRODUÇÃO

A figura 3 demonstra um dos Arduinos mais utilizados, principalmente pelo baixo custo e fácil acessibilidade da maior parte dos jovens e adolescentes.



**Figura 3:** Arduino Uno

Fonte: <https://uploads.filipeflop.com/2014/09/01.png>

De acordo com (NOBLE, 2013) em seu livro, o Arduino surgiu em 2005 na Itália, através de um professor, chamado Massimo Banzi e um pesquisador, David Cuartielles. A ideia surgiu de desenvolver um microcontrolador que poderia ser utilizado pelos estudantes de arte e design em seus projetos, tendo em vista sempre o menor custo possível.

A placa depois de desenvolvida teve o nome de Arduino em referência a um bar local frequentado por alunos e professores.

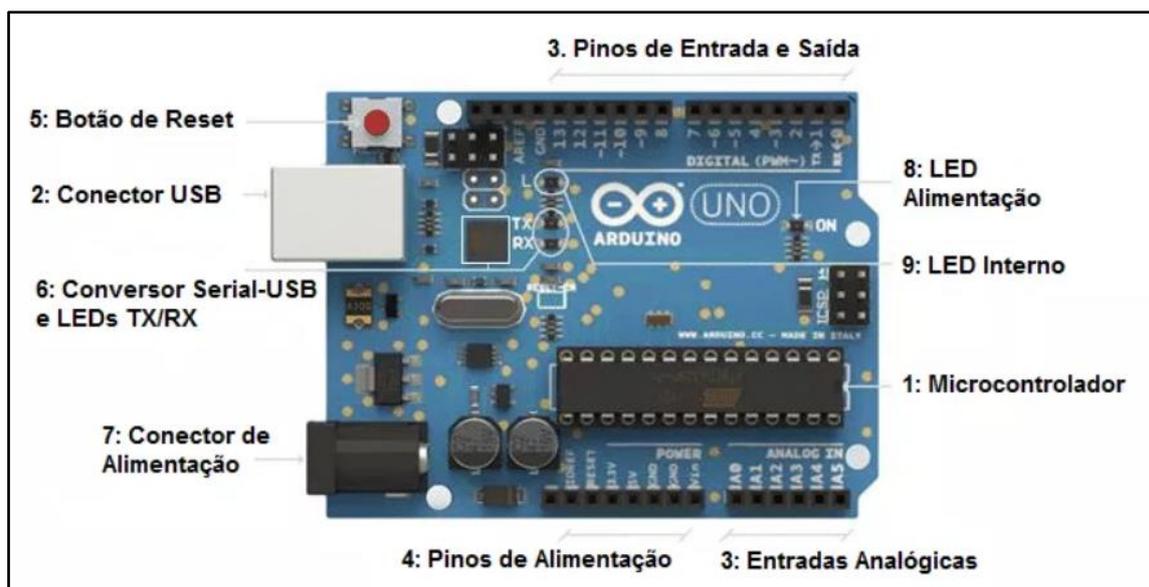
Teve sua popularidade rapidamente alcançada principalmente por ser simples de ser utilizado e ter um baixo custo, assim facilitando muitos estudantes a desenvolverem seus projetos.

Segundo (THOMSEN, 2014) o Arduino tem infinitas possibilidades de uso, pois é compatível com vários dispositivos, não necessariamente, tecnológicos como sensores e LEDs, mas tem uma variedade de opções imensa, sendo possível criar e elaborar projetos únicos, como criar um brinquedo novo, automatizar sua casa, carro, tudo depende de quem irá elaborar o projeto.

Para (MOTA, 2017) o Arduino tem um grande potencial para ser usado em robótica educacional, e no Brasil, existem várias instituições que utilizam a robótica para ensinar e instruir de forma mais efetiva a lógica de programação com o Arduino em projetos pequenos.

### 3.2. ARDUINO UNO

A figura 4 apresenta os principais componentes do Arduino Uno e algumas informações sobre cada um deles.



**Figura 4:** Componentes do Arduino Uno

Fonte: <https://i0.wp.com/porta.vidadesilicio.com.br/wp-content/uploads/2017/05/Arduino-Uno.png?w=712&ssl=1>

Existem vários tipos de modelos de Arduino, o modelo Arduino Uno é o mais popular, por ser bem simples e bastante eficiente. Seus principais componentes são: o Microcontrolador, Conector USB, Pinos de Entrada e Saída, Pinos de Alimentação, Botão de Reset, Conversor Serial-USB e LEDs TX/RX, Conector de Alimentação, LED de Alimentação e o LED Interno.

Sobre a placa, o microcontrolador utilizado é o ATmega328, contem 32kb de memória flash e 2kb de SRAM, de forma que, os programas são salvos na memória flash e as variáveis são armazenadas na memória SRAM, isso para que os dados da memória flash não se perca caso o Arduino seja desligado ou reiniciado, situação pelo qual a memória SRAM pode passar. Abaixo está um exemplo de código de um programa de Arduino que faz um LED piscar.

O próximo exemplo é uma imagem de um código do Arduino, que é utilizado como estudo inicial da plataforma, onde o Led que está conectado a placa acende com um determinado intervalo.

```
1 //Programa : Pisca Led Arduino
2 //Autor : FILIPEFLOP
3
4 void setup()
5 {
6     //Define a porta do led como saida
7     pinMode(13, OUTPUT);
8 }
9
10 void loop()
11 {
12     //Acende o led
13     digitalWrite(13, HIGH);
14
15     //Aguarda o intervalo especificado
16     delay(1000);
17
18     //Apaga o led
19     digitalWrite(13, LOW);
20
21     //Aguarda o intervalo especificado
22     delay(1000);
23 }
```

**Figura 5:** Exemplo de Código Arduino

Fonte: <https://www.filipeflop.com/blog/o-que-e-arduino/>

### 3.3 SENSORES

De acordo com (LOUREIRO, 2013), na última década houve um grande avanço tecnológico nas áreas dos sensores, com exemplos de sensores capazes de fazer monitoramento e controle de diferentes condições de clima de uma área rural, ou até mesmo do próprio oceano.

Na opinião do autor o maior avanço têm sido na área dos microprocessadores, com novos materiais sendo desenvolvidos, como os micro sistemas eletromecânicos (MEMS – *Micro Electro-Mecanical Systems*) e principalmente a comunicação sem fio tem estimulado o desenvolvimento desses sensores “inteligente”, pois a possibilidade de utilizar em vários ambientes, o torna muito mais versátil do que os sensores comuns com fios. A tendência dessa evolução constante, é visar abaixar os custos desses sensores e melhorar a capacidade, para obter resultados mais satisfatórios.

As redes de sensores têm possibilidades de alcançar diversas áreas como:

**Indústria:** Onde sensores podem ser incorporados em peças variadas de uma indústria, com principal objetivo de monitorar e realizar testes nos processos da mesma.

**Ambiente:** É capaz de ser utilizado em qualquer tipo de ambiente, tanto em ambientes internos como prédios, casas, quanto externos, com exemplos de oceanos, florestas, e até mesmo vulcões.

**Tráfego:** Usado em maior escala para monitorar tráfego de veículos em rodovias e áreas urbanas. Sendo a área no qual o presente trabalho se insere, se ramificando para os estacionamentos.

**Segurança:** Essencialmente utilizado em locais onde exigem mais segurança, como centros comerciais, bancos, entre outros.

**Medicina/Biologia:** Tem sido investido muito tempo e dinheiro para aprimorar todo o tipo de tecnologia que envolva a medicina, inclusive os sensores, que estão muito presentes nesse tipo de ambiente, como sensores que monitoram o funcionamento de órgãos como o coração.

**Militar:** Área na qual se destaca por ser a pioneira em avanços tecnológicos, onde se investe todo o esforço para aprimorar equipamentos e novas maneiras de utilizar a tecnologia a favor de um exército ou nação. Tem o exemplo de sensores que detectam o movimento de companhias não desejadas.

### 3.4 SENSOR ULTRASSÔNICO

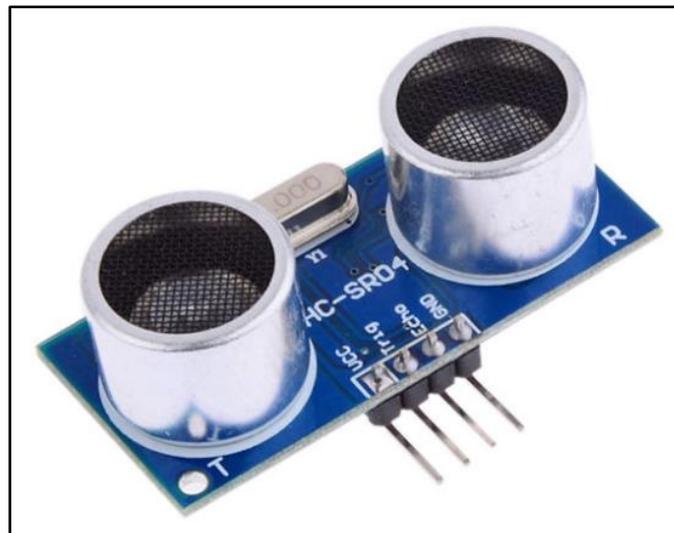
Segundo (Wendling, 2010) os sensores ultrassônicos são extremamente úteis para a detecção de objetos que estejam a uma certa distância, desde que os mesmos não sejam muito pequenos para serem possíveis de serem detectados.

O autor deixa como princípio de funcionamento desse sensor sendo: um oscilador que emite ondas ultrassônicas (em torno de 42kHz), que resultam num comprimento de onda que torna capaz de detectar os objetos.

As ondas refletidas pelo objeto são captadas pelo sensor, que recebe um sinal podendo ser processado as informações sobre o objeto no qual ocorreu a reflexão. Além disso o sensor também pode funcionar com um emissor e receptor.

### 3.5 SENSOR ULTRASSÔNICO HC-SR04

Na figura 6 é apresentado o sensor ultrassônico utilizado nesse projeto, onde o mesmo é capaz de medir a proximidade de qualquer objeto colocado em sua frente.



**Figura 6:** Sensor Ultrassônico HC-SR04

Fonte: <https://uploads.filipeflop.com/2017/07/9SS01-2.jpg>

Esse sensor é capaz de medir distâncias de 2cm a 4cm, possui uma boa precisão e sua principal vantagem é o baixo custo. Além de possuir um circuito pronto com emissor e receptor e 4 pinos (VCC, Trigger, ECHO, GND) para que possa medir as distâncias.

Para que a medição seja feita, é preciso alimentar o módulo e colocar o pino Trigger em nível alto por mais de 10us. Dessa forma o sensor emitirá uma onda sonora que, ao

encontrar um obstáculo, retornará em direção ao módulo. Durante o tempo de emissão e recebimento do sinal, o pino ECHO ficará em nível alto.

Algumas especificações do HC-SR04:

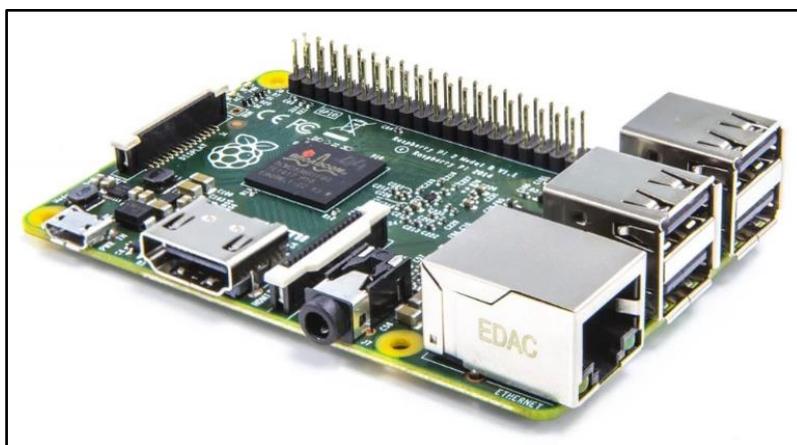
- Alimentação: 5V DC
- Corrente de Operação: 2mA
- Ângulo de efeito: 15°
- Alcance: 2cm ~ 4cm
- Precisão: 3mm

## 4. RASPBERRY PI

Neste capítulo será abordado os principais conceitos do Raspberry Pi, analisando sua arquitetura, seus componentes e apresentando sua origem.

### 4.1. INTRODUÇÃO

A figura 7 temos o Raspberry Pi 3, modelo atualmente mais avançado do mercado, é muito utilizado em projetos de eletrônica de várias instituições educacionais, principalmente por ser tão completo quanto um computador convencional, obviamente seu poder de processamento não equivale a um computador, mas é muito abrangente suas aplicações, por conter várias entradas e componentes.



**Figura 7:** Raspberry Pi

Fonte: <https://imagens.canaltech.com.br/99117.162501-Raspberry-Pi.jpg>

Segundo (DUARTE, 2013), a ideia inicial de criação do Raspberry Pi surgiu em 2006, na universidade de Cambridge, com o principal objetivo de ensinar as crianças e adolescentes a programação, de forma simples e direta. Os primeiros modelos do Raspberry Pi foram baseados no microcontrolador Atmel ATmega644.

Sua aceitação foi extremamente positiva, principalmente, por ser simples, versátil e de baixo custo, além de ter todas as informações relacionadas ao Raspberry em seu site oficial. O que despertou mais o interesse das pessoas para ter esse contato com o dispositivo, foi a questão de o código ser totalmente aberto, que faz multiplicar as possibilidades.

### 4.2 ARQUITETURA

Existem vários modelos de Raspberry Pi, que podem ser usados em vários modelos e projetos diferentes. Conforme (UPTON, 2017) Co criador do Raspberry Pi, desde o seu

lançamento com dois modelos, foram chegando mais modelos ao mercado, um aumento significativo. Atualmente a família Raspberry Pi consiste em cinco modelos principais: Raspberry Pi A+, modelo B+, Raspberry Pi 2, Raspberry Pi 3 e o modelo Raspberry Pi Zero. Com exceção do modelo Zero que possui um tamanho menor, para ter um baixo custo, os outros possuem a aparência e os componentes semelhantes aos dois iniciais da família.

Quase todas as placas Raspberry Pi há um semicondutor, comumente conhecido como circuito integrado ou chip. Esse tipo é chamado *system-on-chip* (SoC), que fornece ao Raspberry Pi seu processamento de propósito geral, renderização gráfica e recursos de entrada/saída. Dependendo do seu modelo, pode ser o Broadcom BCM2835 original, o mais rápido BCM2836 quad-core ou ainda mais poderoso BCM2837 de 64 bits.

Nos modelos A+, B+ e Zero, na parte de cima desse chip existe outro semicondutor, que é uma memória do Raspberry Pi para armazenar os dados temporários enquanto se executa um programa, já nas versões Raspberry Pi2 e 3, esse chip fica localizado na parte de baixo da placa. Esse tipo de memória é conhecido como memória RAM, ou seja, ela é volátil, que significa que o que estiver armazenado nela será perdido quando a placa for desligada.

Logo abaixo do SoC estão as saídas de vídeo da placa. Possui uma entrada HDMI (*High-Definition Multimedia Interface*), um conector que já está bastante em uso nos dias de hoje principalmente em monitores e TVs. É fornecida uma porta de vídeo composto, especialmente para as TVs mais antigas que não possuem soquete HDMI, a porta do *jack* AV de 3,5mm preto e prata fica à direita do soquete HDMI. Em questão da qualidade de imagem, é um pouco menor do que a HDMI, porém existe a possibilidade de usar mais componentes do apenas um padrão definido.

No Raspberry Pi Zero tem um layout um tanto diferente, pois não possui um soquete HDMI como nos outros modelos, no lugar dele, tem um *soquete mini-HDMI*, que por consequência exige um cabo específico ou um adaptador mini-HDMI.

Em uma visão geral sobre o Raspberry Pi, nota-se que, ele se encaixa na categoria de computadores de baixo custo, justamente por sua proposta simples e direta. Ser um facilitador de inclusão digital e iniciar os alunos de diversas idades a programação.

É um dispositivo bom em quesito de processamento em projetos de automação, além de proporcionar uma grande abrangência de opções profissionais e de experiências educacionais, tanto na área de programação, como de projetos de eletrônica, robótica e automação.

## 5. PROPOSTA DO TRABALHO

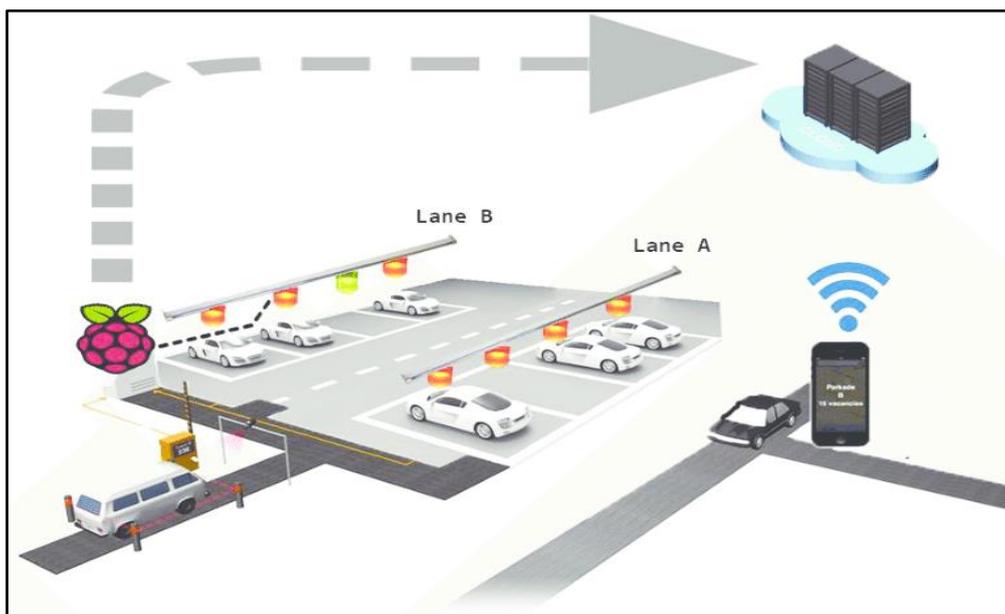
Esse trabalho teve como objetivo, desenvolver uma aplicação de Smart Parking, usando o conceito de IoT, incluindo o Arduino com sensor de distância para verificar as vagas disponíveis e o Raspberry foi utilizado para se comunicar com o Arduino, pois nele reside uma API em Python que se comunica com a aplicação do cliente na linguagem JAVA, essa aplicação feita em Java e Angular é responsável por disponibilizar os dados para o usuário, tudo isso afim de auxiliar os condutores de veículos nas grandes cidades, de encontrar vagas em estacionamentos.

O projeto consistiu em criar toda a infraestrutura de Arduino com os sensores, integrado com o Raspberry, onde a API do Python diretamente rodando no Raspberry, envia as informações, para a aplicação feita em Java e Angular, dessa forma, o usuário pode ter acesso às informações dos sensores em tempo real.

Concluindo a proposta do trabalho, a aplicação que o usuário tem acesso, fica disponível na Web, ou seja, sem a necessidade de baixar software para poder acessá-lo, sendo necessário apenas uma conexão com a internet.

### 5.1 SMART PARKING

No exemplo abaixo, é apresentado um pequeno conceito de um Smart Parking, totalmente conectado a sensores, com seus dados armazenados no servidor e disponíveis para o usuário final.



**Figura 8:** Smart Parking

Fonte: Abhirup Khanna, 2016

A principal proposta de um Smart Parking é oferecer facilidade, objetividade e agilidade na hora de encontrar uma vaga disponível. O intuito desse trabalho foi desenvolver uma solução que tenha um menor custo, mas que também cumpra as necessidades que o condutor possui ao se locomover nas cidades de hoje.

Inicialmente os sensores juntamente com o Arduino, monitoram todas as vagas do estacionamento, através da aplicação web, o usuário sabe qual vaga está livre e qual está ocupada, pelas cores verde e vermelho respectivamente.

Assim que um carro entra na vaga automaticamente o sensor de distância manda um sinal ao Arduino, fazendo assim que o LED inicial verde se torne vermelho indicando ao software que aquela vaga está ocupada, dessa forma ocorre para as demais vagas do estacionamento.

## 5.2 TECNOLOGIA E RECURSOS ADOTADOS

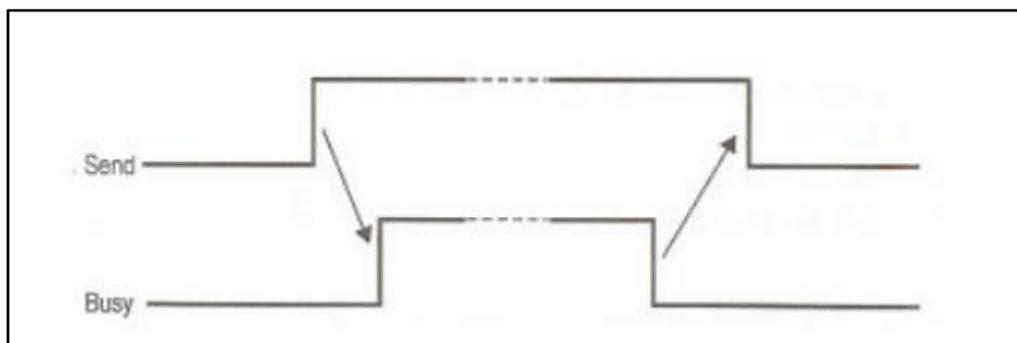
No primeiro momento foi implementado um software para o Arduino que se comunica com o sensor de distância (HC-SR04), tornando possível a inclusão dos LEDs para indicar as vagas livre e ocupadas conforme descrito anteriormente. Após finalizado a implementação do Arduino, será iniciado a implementação da API em JAVA que ficará no Raspberry Pi para receber os dados que vem diretamente do Arduino. A API será responsável por tratar toda a informação do software e criar a regra de funcionamento do mesmo.

Em um segundo momento foi feita a integração do Arduino com o Raspberry possibilitando assim a comunicação direta de ambos através do Protocolo UART (*Universal Asynchronous Receiver Transmitter*) de forma assíncrona como o protocolo já descreve, através disso foi feita a implementação do software utilizando o Java com Spring Boot e WebServices. Sendo esses WebServices consumidos pelo Angular, uma plataforma de aplicações web, baseado na linguagem TypeScript, que é liberado pela própria equipe do Google, empresa que tem todo o direito da plataforma.

## 5.3 PROTOCOLO UART (*UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER*)

Segundo (TALIANI, 2012), O conceito do protocolo UART é uma comunicação serial feita de forma assíncrona, é utilizado principalmente quando se deseja uma comunicação feita com barramento de baixo custo, porém há uma desvantagem, a velocidade dessa comunicação não é das mais rápidas, pois o hardware utilizado nesse tipo de comunicação é relativamente simples, utilizando apenas um fio para cada direção.

Quando o sistema deseja fazer a comunicação e o envio de informações, ele habilita o sinal *Send* indicando ao transmissor que deseja enviar os dados através da porta serial. Após isso, o controlador UART habilita o sinal *Busy* mostrando ao sistema que é possível enviar o byte desejado, podendo assim, a transmissão ser iniciada. Ao final da transmissão, o controlador UART desabilita o sinal *Busy* e o sistema desabilita o sinal *Send*, esse exemplo pode ser visto na figura 9.



**Figura 9:** Sequência de sinais de controle UART

Fonte: Redes De Comunicação Automotiva: CARACTERISTICAS, TECNOLOGIAS E APLICAÇÕES (Dias, 2010)

A transmissão de dados através do UART inicia-se com o start bit que marca o início de um frame com oito bits, dessa forma, o receptor é responsável por sincronizar para que seja capaz de receber essa sequência de dados vindos do transmissor.

A principal característica do protocolo UART é sua simplicidade na troca de informações, onde pode ser aplicado quando se deseja uma comunicação entre dois dispositivos de ponto a ponto.

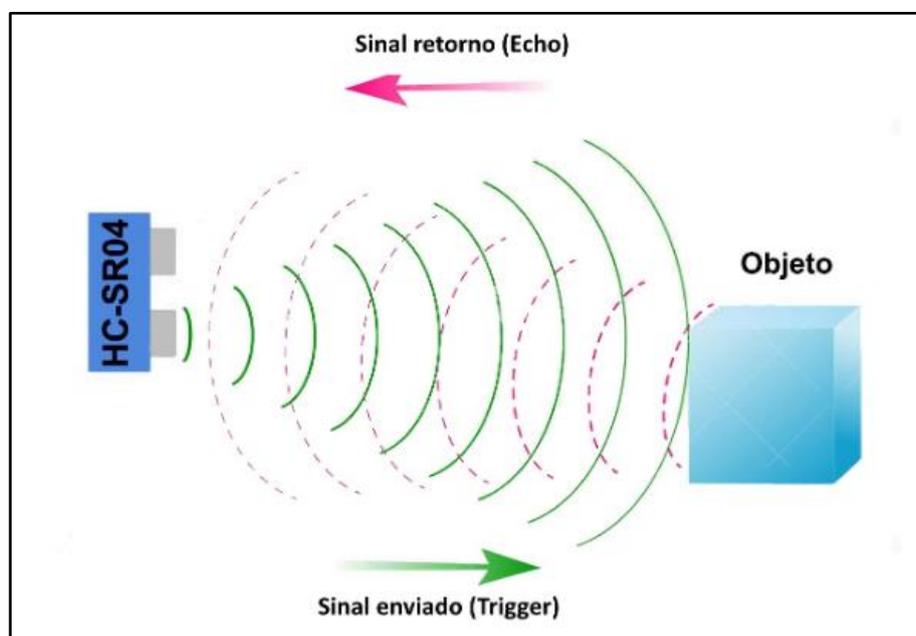
## 6. DESENVOLVIMENTO DO TRABALHO

Esse capítulo tem como objetivo apresentar toda metodologia adotada no desenvolvimento do trabalho, explorando os conceitos utilizados na forma prática.

### 6.1 COMUNICAÇÃO SENSOR HC-SR04 X ARDUINO

O sensor HC-SR04 é bastante comum nesses tipos de projetos, utilizando o Arduino, principalmente pelo seu baixo custo e facilidade de uso. No presente trabalho foi utilizado esse sensor de proximidade que é capaz de medir distâncias entre 2 cm e 4 metros, com uma ótima precisão de 3mm visando o tipo de projeto proposto, teve um ótimo aproveitamento.

Basicamente ele funciona na forma de envio de sinais ultrassônicos pelo sensor, que espera pelo retorno desse sinal, e com base no tempo entre o envio e retorno, calcula a distância entre o sensor e o objeto detectado na sua frente. A seguir temos uma imagem com um exemplo de como funciona essa detecção do objeto pelo sensor.



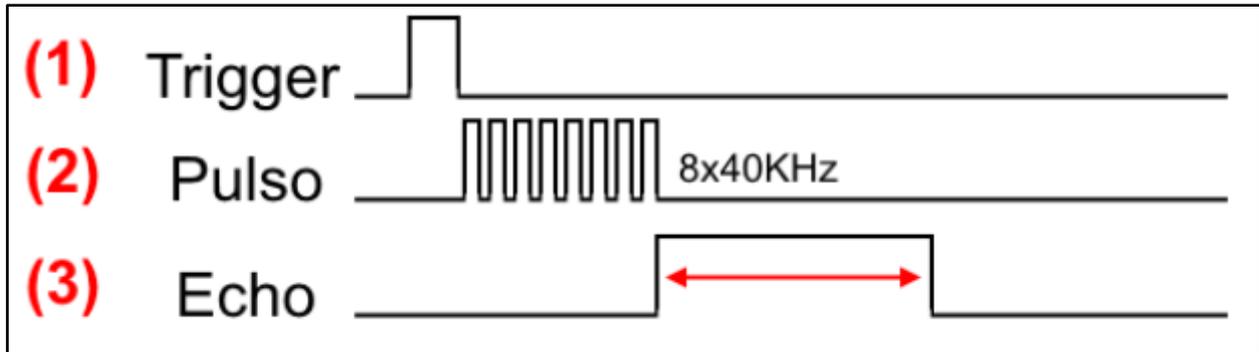
**Figura 10:** Detecção de Objeto, Sensor HC-SR04

Fonte: [https://uploads.filipeflop.com/2011/07/HC\\_SR04\\_Trigger\\_Echo.jpg](https://uploads.filipeflop.com/2011/07/HC_SR04_Trigger_Echo.jpg)

Primeiramente é enviado um pulso de 10 $\mu$ s, indicando o início da transmissão de dados. Depois disso, são enviados 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação

Distância = (Tempo Echo em nível alto \* velocidade do som) / 2. Para ligação do sensor ao microcontrolador, são utilizados 4 pinos: VCC, TRIGGER, ECHO e GND.

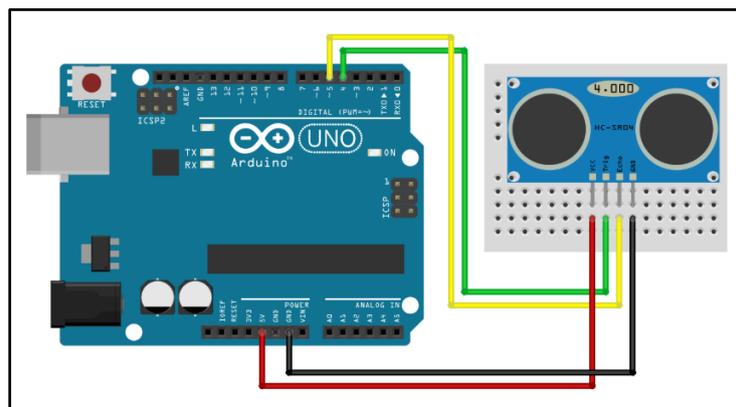
Na figura 11 podemos ver como é enviado e recebido esses pulsos, gerados na transmissão desses dados.



**Figura 11:** Transmissão de Dados, Sensor HC-SR04

Fonte: <https://uploads.filipeflop.com/2011/07/Diagrama-tempo-hc-sr04.png>

Para que a arquitetura do projeto fosse mais organizada e melhor apresentada, foi utilizado uma Protoboard como ferramenta de auxílio nas conexões entre os sensores e o Arduino. Dessa forma foi capaz de elaborar uma organização onde foi capaz de conectar mais de um sensor no mesmo Arduino, trazendo mais autonomia ao projeto. Na imagem abaixo é um exemplo de como pode ser feita essa conexão entre o Arduino e o sensor, com o auxílio da Protoboard.



**Figura 12:** Conexão Arduino x Sensor

Fonte: [https://uploads.filipeflop.com/2011/07/Arduino\\_HC\\_SR04\\_bb.png](https://uploads.filipeflop.com/2011/07/Arduino_HC_SR04_bb.png)

## 6.2 COMUNICAÇÃO ARDUINO X RASPBERRY PI COM PROTOCOLO UART

Como foi visto acima na Seção 5.3, o protocolo UART faz uma comunicação assíncrona com as portas seriais dos dispositivos. A arquitetura desse trabalho é baseada nesse protocolo, e através dele foi possível desenvolver a comunicação serial entre a placa Arduino Uno e o Raspberry Pi 3. No Arduino existe duas portas responsáveis por fornecer essa funcionalidade, são elas, a porta RX que é o receptor e a porta TX, o transmissor. Para que ocorra essa comunicação de forma correta o *Baud Rate* (Taxa de Transmissão) de ambos dispositivos devem ser os mesmos, nesse projeto foi utilizado a taxa de transmissão no valor de 9600 bit/s. No Raspberry a conexão é feita através dos pinos 10 (RX) e o pino 8 (TX) localizados na GPIO.

Para que seja possível a comunicação entre os dois dispositivos, é necessário fazer algumas configurações e o código capaz de enviar os dados através da porta serial. A seguir podemos ver na imagem um exemplo de um código feito na ferramenta do Arduino, que é capaz de ligar um Led conforme a resposta recebida do Raspberry.

```
1 // Programa: Comunicacao Arduino com Raspberry Pi
2 // Autor: FILIPEFLOP
3
4 const int pino_led = 13;
5 char buf;
6 int estado = 1;
7
8 void setup()
9 {
10  pinMode(pino_led, OUTPUT);
11  Serial.begin(115200);
12 }
13
14 void loop()
15 {
16  while (Serial.available() > 0)
17  {
18    buf = Serial.read();
19    // Caso seja recebido o caracter L, acende o led
20    if (buf == 'L')
21    {
22      // Liga ou desliga o led da porta 13
23      digitalWrite(pino_led, estado);
24      // Envia a resposta para o Raspberry
25      Serial.print("Recebido! - Estado Led: ");
26      if (estado == 1)
27      {
28        Serial.println("Ligado");
29      }
30      else
31        Serial.println("Desligado");
32    }
33    // Inverte o estado do led
34    estado = !estado;
35  }
36 }
```

**Figura 13:** Código Comunicação Arduino

Fonte: <https://www.filipeflop.com/blog/comunicacao-serial-arduino-com-raspberry-pi/>

No Raspberry é um pouco diferente, de início temos que alterar um arquivo localizado no diretório /etc/ do Raspian (Sistema Operacional baseado no Debian, feito especialmente para executar no Raspberry) chamado inittab. Na alteração desabilitamos o Sistema de mostrar a tela de login diretamente na serial. Após a alteração deve-se salvar o arquivo editado e reiniciar o Raspberry.

Na figura 14 podemos ver um exemplo de código feito em Python que é responsável por fazer a comunicação com o Arduino, nesse exemplo toda vez que o usuário aperta o botão o Raspberry envia um sinal para o Arduino, que por sua vez é responsável por ligar o Led.

```
1  #Programa : Comunicacao Raspberry e Arduino
2  #Autor : FILIPEFLOP
3
4  import RPi.GPIO as GPIO
5  import time
6  import serial
7
8  #Configura a serial e a velocidade de transmissao
9  ser = serial.Serial("/dev/ttyAMA0", 115200)
10
11  GPIO.setmode(GPIO.BOARD)
12
13  #Define o pino do botao como entrada
14  GPIO.setup(18, GPIO.IN)
15
16  #Mensagem inicial
17  print ("Pressione o botao...")
18
19  while(1):
20      #Verifica se o botao foi pressionado
21      if GPIO.input(18) == True:
22          #Envia o caracter L pela serial
23          ser.write("L")
24          print("Enviado - L")
25          #Aguarda resposta
26          resposta = ser.readline()
27          #Mostra na tela a resposta enviada
28          #pelo Arduino
29          print resposta
30          #Aguarda 0,5 segundos e reinicia o processo
31          time.sleep(0.5)
```

**Figura 14:** Código Comunicação Raspberry

Fonte: <https://www.filipeflop.com/blog/comunicacao-serial-arduino-com-raspberry-pi/>

Para esse trabalho foi utilizado esse mesmo conceito de comunicação, com algumas diferenças nos códigos vistos nos exemplos acima, porém na parte do Raspberry foram utilizadas algumas bibliotecas específicas para que fosse possível utilizar todos os recursos do Arduino e sensores, juntamente com a aplicação web que foi proposta.

A biblioteca utilizada no Python que foi essencial para a comunicação do Raspberry com a aplicação foi a Requests. Através dela é possível realizar requisições HTTP para qualquer

API (application programming interface) Rest disponível, com isso foi possível mandar os dados que estavam sendo gerados pelo sensor, passar pelo Arduino, ser enviado para o Raspberry, e no fim, ele realizar uma requisição para a aplicação em Spring Boot. Esses dados recebidos na aplicação foram possíveis ser armazenados no banco de dados, no qual foi utilizado o banco de dados MySQL, que é muito conhecido pela grande parte dos desenvolvedores.

### 6.3 COMUNICAÇÃO RASPBERRY X SPRING BOOT (JAVA)

Para que fosse possível realizar essa comunicação, seria necessário utilizar o conceito que está bastante em alta no mundo da tecnologia hoje em dia, os Web Services. Através dele é possível realizar vários tipos de aplicações e gerar APIs totalmente independentes uma das outras, o que torna os projetos de desenvolvimento de softwares, muito mais ágeis, seguros e escaláveis.

De acordo com (GOMES, 2014), os Web Services surgiram como uma evolução dos modelos de computação distribuída, que foram muito utilizados na década de 90, entre eles estão o RMI, CORBA e DCOM. Quando a Internet começou a evoluir e o mercado corporativo entrou nesse mundo, a tecnologia que hoje está cada vez mais comum, os web services surgiu, pois também houve a necessidade de integrar aplicações na web, não somente redes locais, como era na época. O primeiro padrão de desenvolvimento dessa tecnologia a surgir foi o SOAP (Simple Object Access Protocol). Basicamente essa tecnologia pode ser definida como softwares integrados e interagindo entre si, enviando e recebendo informações, sem se preocupar com o hardware em uso, sistema operacional ou linguagem de programação. A única premissa é que no caso do SOAP essa troca de dados deve ser feita no formato XML (Extensible Markup Language).

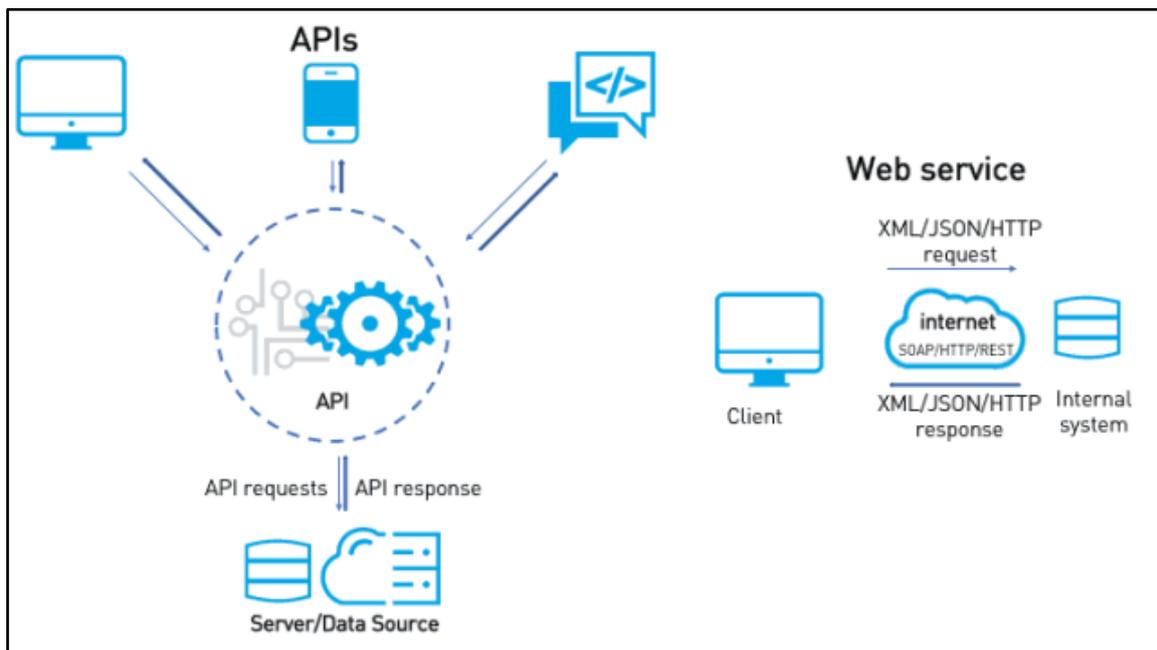
Além do SOAP, surgiu um padrão que tem sido adotado por muitos desenvolvedores e grandes empresas, o tão conhecido padrão REST (Representational State Transfer). Essa arquitetura provê serviços que podem ser consumidos por outras aplicações ou até mesmo o cliente. O padrão REST é mais indicado para a integração de sistemas através da internet, por fornecer serviços escaláveis, de fácil modificação e manutenção, e principalmente por ter boa performance, o que torna esse padrão muito bem aceito e estável na internet.

Esse projeto está baseado no padrão REST e na sua implementação, o Raspberry fornece serviços no qual a aplicação Java com o framework Spring Boot, pode facilmente consumi-la, tornando possível, o acesso aos dados do Raspberry. A criação desse *web service* na parte do Raspberry foi feita através da biblioteca disponível no Python chamada Requests,

com ela é possível implementar serviços para serem consumidos utilizando a maioria das funcionalidades do protocolo HTTP.

O código em Python foi implementado no próprio Raspberry, assim o mesmo precisa estar conectado na internet para que possa realizar as requisições HTTP diretamente para aplicação web.

Na figura 15 podemos observar como é a arquitetura do *web service* e das APIs que o compõem.



**Figura 15:** Arquitetura Web Service

Fonte: <https://medium.com/codiumclub/web-services-guide-to-become-full-stack-developer-9224bed7b4cb>

O mundo da tecnologia hoje está cercado de várias APIs, para inúmeros tipos de situações e aplicações diferentes, os exemplos mais básicos que temos, é API de endereço, no qual um software qualquer pode fazer uma requisição para determinada API de CEP, e a mesma retorna na resposta da requisição, todos os dados detalhados do respectivo CEP passado para ela. Outro exemplo é a API do Google Maps, que inclusive foi utilizada nesse projeto, onde o próprio Google fornece as informações de praticamente o mundo inteiro, basta ter uma *key* que se equivale a uma senha ou login, a partir daí seu software ou aplicativo mobile, pode ter acesso a qualquer informação que desejar.

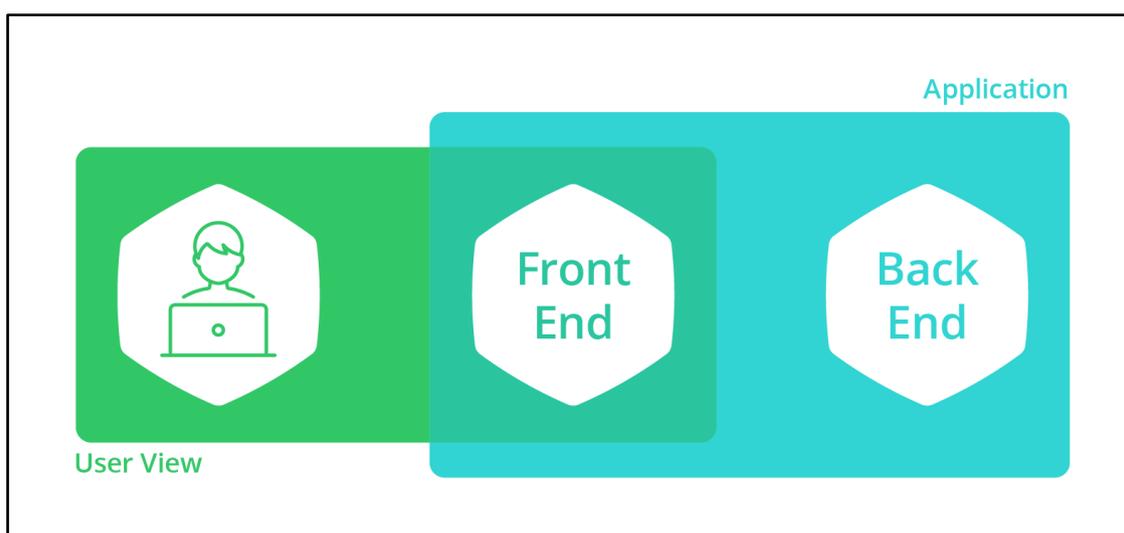
## 6.4 APLICAÇÃO WEB SPRING BOOT & ANGULAR

A aplicação web foi desenvolvida utilizando a linguagem Java, juntamente com um dos *frameworks* mais utilizados na atualidade, o Spring. Por ser muito completo, o Spring fornece várias opções de projetos a serem criados para web, grandes empresas utilizam esse framework que cada vez mais está provando sua capacidade de ser seguro, escalável e ter uma boa capacidade de processar dados.

O Java é responsável por fornecer os serviços para que o *framework* Angular possa consumi-lo, toda a manipulação dos dados geralmente é feita no lado do servidor, onde a aplicação Java se encontra, isso forma o conceito de *Back-end* ou *Server-side*, onde o usuário final não possui acesso e nem sabe como realmente funciona, essa geração dos dados.

O que o usuário tem acesso fica disponível na página web, ou aplicativo que ele mesmo baixa, esses padrões de desenvolvimento, estão sendo muito utilizados, principalmente pela segurança que propõe, pois o usuário consegue alterar apenas o que está na página visível para ele, todos os dados que ele deseja alterar devem ser permitidas no lado do servidor da aplicação (*back-end*), esse lado do cliente é bastante caracterizado por linguagens famosas, como HTML (*Hypertext Markup Language*), JavaScript e CSS (*Cascading Style Sheets*) e se chama de *Front-end*.

No exemplo abaixo podemos observar como é a arquitetura da aplicação baseada em *web services* e APIs REST.



**Figura 16:** Front-end e Back-end

Fonte: <https://blog.hackerrank.com/wp-content/uploads/2018/10/backend-development-diagram.png>

Nesse projeto foi utilizado a linguagem TypeScript com o *framework* Angular na parte do *front-end*, com ele foi possível consumir os serviços que o *back-end* disponibiliza, na arquitetura REST essa troca de dados é constante.

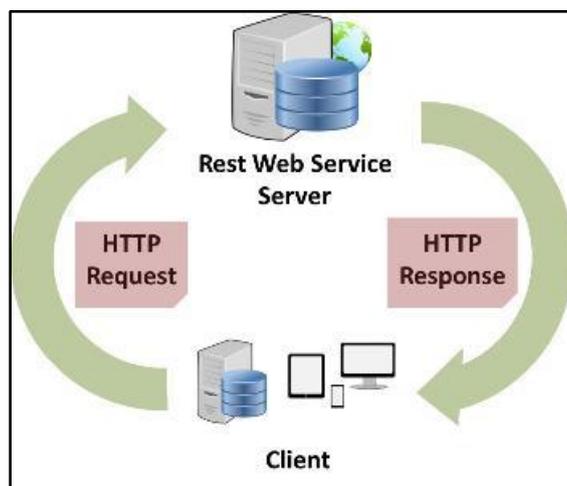
Segundo (LECHETA, 2015), o REST utiliza o protocolo HTTP para criar serviços que retornam dados nos formatos XML ou JSON (*JavaScript Object Notation*) e tem o objetivo de fornecer uma alternativa para o SOAP e o WSDL. Existe o termo chamado RESTful que de acordo com o autor foi criado para indicar que determinado sistema ou serviço segue os princípios do REST, por exemplo os serviços mais conhecidos como, GET, POST, PUT e DELETE, então se um *web service* seguir estes princípios básicos, poderemos dizer, que este é um *web service* RESTful. Porém o REST tem mais algumas características:

Cada requisição ao serviço deve retornar de forma independente e não deve conter estado, para que isso ocorra, utiliza-se o protocolo HTTP, no qual já está estabelecido na maior parte da internet.

Utiliza uma URI (Uniform Resource Identifier), de forma geral é uma sintaxe para identificar recursos. No REST, cada recurso deve conter uma URI para que seja possível consultar informações sobre ele.

E por fim, no REST existe os tipos de conteúdo (*mime-type*) para solicitar conteúdo e retornar conteúdo. Isso possibilita que o cliente do serviço, ou seja, a aplicação ou até mesmo o usuário final, especifique se deseja que o conteúdo seja retornado em XML ou JSON, por exemplo.

Na figura 16, podemos ver o funcionamento desse ciclo de requisições e respostas, na arquitetura REST.



**Figura 17:** Aplicação Web RESTful

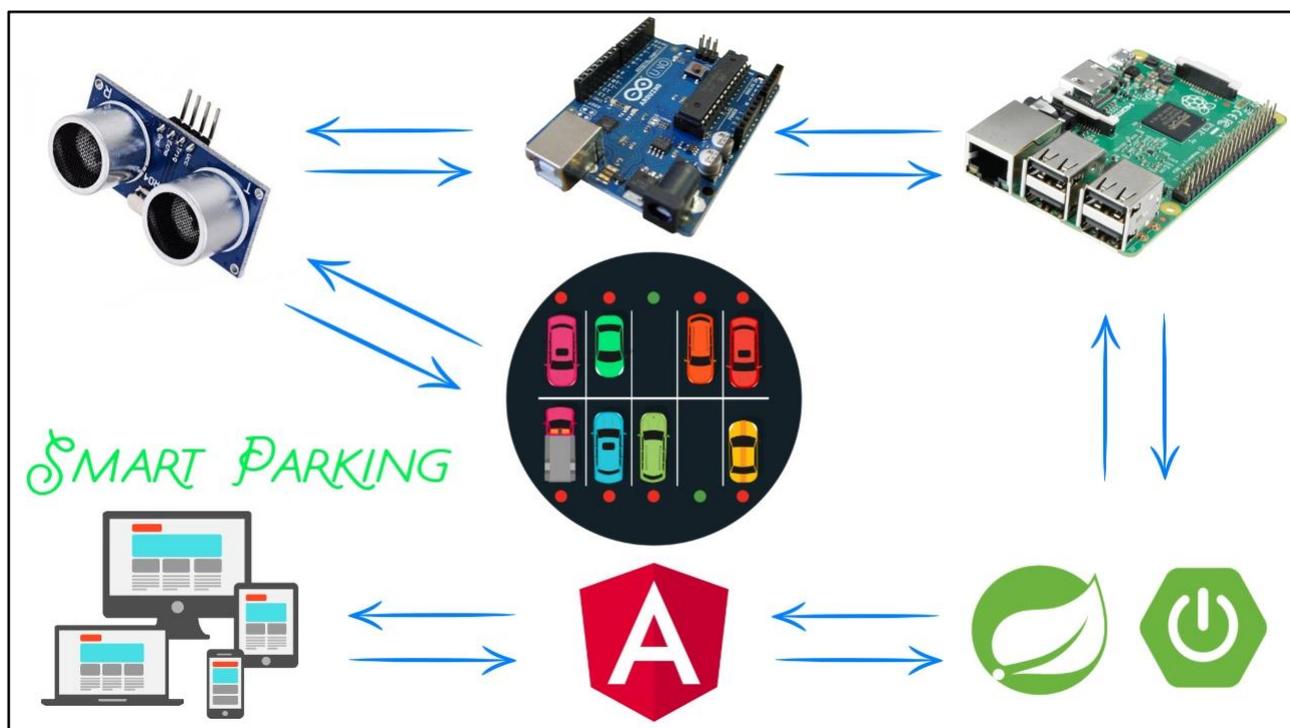
Fonte: <https://www.stsoftware.com.au/site/ST/blog/article/rest-define/>

A aplicação para o usuário final, foi desenvolvida utilizando essas tecnologias apresentadas, de forma que o cliente tem acesso direto de seu navegador, independente da plataforma que ele estiver usando, computador ou smartphone, apenas uma conexão com a internet é necessária para ter acesso aos recursos da aplicação.

#### 6.4 PROJETO DESENVOLVIDO

Essa seção tem como objetivo, apresentar o projeto que foi desenvolvido juntamente com o presente trabalho, utilizando as tecnologias analisadas anteriormente.

Após a comunicação dos sensores ultrassônicos e o Arduino o próximo passo, foi criar o meio que faria a integração do Arduino com o Raspberry. Para isso utilizei o protocolo UART. A seguir podemos ver uma figura que demonstra toda a arquitetura do projeto desenvolvido.



**Figura 18:** Arquitetura do projeto

Fonte: Matheus Cavalcanti, 2019

A partir da comunicação feita entre o Raspberry, foi desenvolvido uma API em Python para fazer requisições com os dados recebidos do Arduino e sensores. Assim tornou possível a aplicação Spring Boot ter acesso a esses dados por meio da arquitetura REST. O Python utiliza a biblioteca Requests para a implementação dos *web services*.

Na figura 18 é um exemplo da API feita em Python para realizar essas requisições para o Spring.

```
url = 'http://10.0.3.191:8080/integracao/enviar'
headers = {
    'Content-Type': 'application/json',
}

body = {
    'msg': respostaArd
}

msg = requests.post(url=url, headers=headers, json=body)

time.sleep(0.1)
```

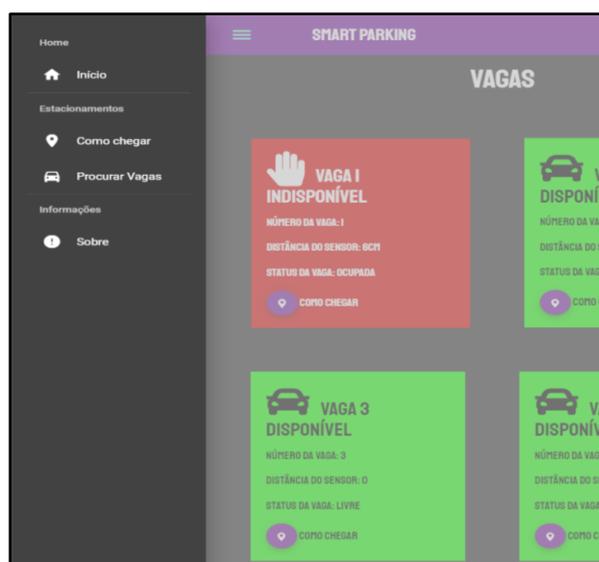
**Figura 19:** API Python

Fonte: Matheus Cavalcanti, 2019

Essa API é responsável por fazer uma requisição POST para o Spring Boot com os dados recebidos do Arduino, a partir dela foi possível salvar essas informações no banco de dados diretamente do Spring utilizando o Spring Java Persistence API (JPA), para isso foi utilizado o banco MySQL.

Após as informações salvas no banco de dados, as mesmas ficam disponíveis como API para serem consumidas pelo Angular, fornecendo as páginas de navegação para o usuário final, podendo ser acessadas diretamente do navegador, tanto pelo smartphone, computador ou qualquer dispositivo que tenha conexão com a internet e um *browser* para executá-lo.

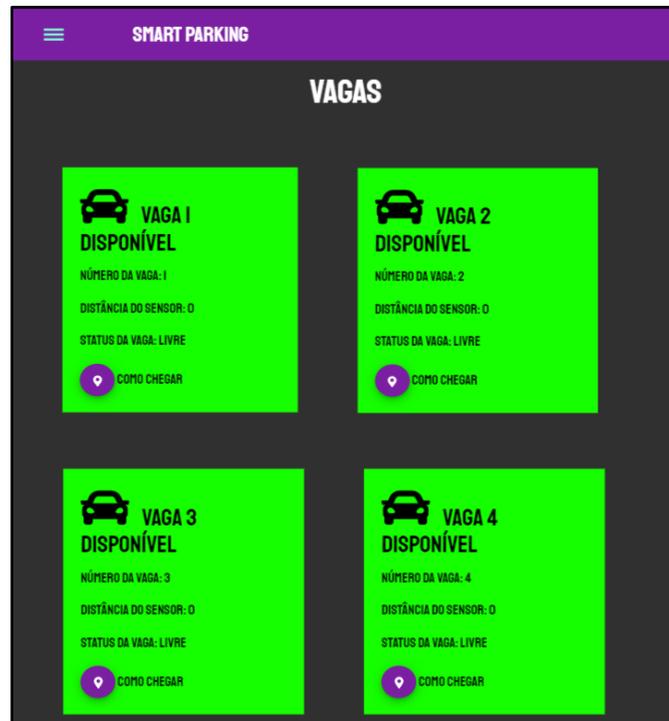
Na figura 19 é uma captura feita da aplicação desenvolvida no Angular, disponível ao usuário.



**Figura 20:** Smart Parking Menu

Fonte: Matheus Cavalcanti, 2019

No Menu ficam disponíveis as opções que o usuário possui para navegar na aplicação. Na próxima imagem vemos o funcionamento da página de vagas, onde o usuário consulta as vagas que o estacionamento possui.



**Figura 21:** Smart Parking vagas

Fonte: Matheus Cavalcanti, 2019

A partir da página de consulta de vagas, o usuário pode verificar quais vagas estão ocupadas e quais estão livres, tudo isso feito de forma dinamicamente, através do Angular é possível identificar o status de cada vaga, graças a API que a mesma consome diretamente de seu *back-end* no Spring Boot.

Na figura abaixo podemos observar essa consulta com mais detalhes, sendo as vagas ocupadas na cor vermelho e as livres na cor verde.



**Figura 22:** Smart Parking consultar vagas

Fonte: Matheus Cavalcanti, 2019

Por último, o usuário tem à sua disposição um GPS diretamente na aplicação para ele ter a localização exata de onde fica o estacionamento, no qual ele mesmo já identificou a vaga que pretende utilizar.

## 7. CONCLUSÕES

Tendo em vista o que foi abordado no presente trabalho, o IoT tem se mostrado capaz para resolver problemas de mais diversas áreas e ambientes, como em mercados, no trânsito, parques, hospitais, shoppings e nas próprias casas. Essa curiosidade e sucesso demonstrados ao IoT, se deve principalmente na sua utilidade bastante intuitiva, além de ser muito efetiva nas situações propostas.

Para projetos pequenos o Arduino sempre foi uma ótima opção, pois entrega uma grande variedade de possibilidades, não deixando de ter um baixo custo. A integração que ele pode oferecer com diversos componentes é grande, como nesse trabalho que será abordado, um pequeno projeto que utilizará os sensores de distância, no modelo ultrassônico HC-SR04, um modelo simples, mas que entrega uma boa qualidade no seu uso.

Juntamente com o Arduino, será integrado um Raspberry Pi, para que a arquitetura e os dados gerados pelos sensores e os LEDs sejam disponibilizadas para uma aplicação de Backend, desenvolvida em Java, que ficará armazenada no próprio Raspberry. Por fim, uma aplicação Web será desenvolvida para o cliente final, através do seu smartphone encontrará o estacionamento que possua vagas disponíveis, pois essa informação será gerada pelos sensores de distância e depois interpretada pela aplicação, disponibilizando de forma clara e objetiva ao cliente, a quantidade de vagas disponíveis, para que ele não perca tempo na busca da sua vaga para estacionar.

O protótipo se mostrou ser simples no seu uso, porém pode ser de grande contribuição para futuros projetos, envolvendo os estacionamentos inteligentes, mobilidade urbana e áreas afins.

O projeto mostrou também, a capacidade de integração das principais tecnologias presente no mercado, com os conceitos e práticas do IoT. De como os *web services* REST podem ser úteis nessa integração, principalmente por trazer independência as tecnologias, padronizando o tratamento e troca dos dados, tornando o projeto capaz de ser alterado a qualquer momento e evoluído futuramente.

Como por exemplo, há a possibilidade de inserir gráficos e dados analíticos sobre determinado estacionamento ou até mesmo uma vaga, com informações sobre a frequência que a vaga é utilizada, o tempo que permanece ocupada e disponível. Outro exemplo é desenvolver um sistema de locação de vagas e pagamentos diretamente do mesmo projeto,

sendo possível o usuário escolher alguma vaga e reservá-la por algum período e trazendo a opção de pagar diretamente da aplicação, utilizando seu cadastro e informações sobre ele

## REFERÊNCIAS

AMAZON, Olawale Oladehin, Brett Francis. **Core Tenets of IoT**. Amazon Web Services. Disponível em: < <http://d0.awsstatic.com/whitepapers/core-tenets-of-iot1.pdf> >. Acesso em: 08 out. 2018.

DENATRAN, Associação Nacional dos Detrans. **Brasil já tem 1 carro a cada 4 habitantes**. Brasília-DF, Brasil. Disponível em: < <http://www.and.org.br/brasil-ja-tem-1-carro-a-cada-4-habitantes-diz-denatran> >. Acesso em: 13 out. 2018.

EVANS, Dave. **A Internet das Coisas Como a próxima evolução da Internet está mudando tudo**. Cisco IBSG. Disponível em: < [https://www.cisco.com/c/dam/global/pt\\_br/assets/executives/pdf/internet\\_of\\_things\\_iot\\_ibsg\\_0411final.pdf](https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf) >. Acesso em: 28 jan. 2019.

FRANCO, Pedro Felipe dos Santos. **Soluções para Smart Parking com base em tecnologias sem fios**. Universidade de Aveiro, Portugal. Disponível em: < <https://ria.ua.pt/bitstream/10773/13812/1/solu%2b%2b%20para%20smart%20parking%20com%20base%20em%20tecnologias%20sem%20fios.pdf> >. Acesso em: 14 out. 2018.

GOMES, Daniel. **Web Services SOAP em Java - 2ª Edição: Guia prático para o desenvolvimento de web services em Java**. São Paulo, SP. Novatec Editora Ltda, out. 2014.

LECHETA, Ricardo. **Web services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google**. São Paulo, SP. Novatec Editora Ltda, set. 2015.

LIBELIUM. **The IoT Marketplace by Libelium shows at IoT Solutions World Congress more than 100 solutions to improve sustainability and competitiveness**. Disponível em: < <http://www.libelium.com/the-iot-marketplace-by-libelium-shows-at-iot-solutions-world-congress-more-than-100-solutions-to-improve-sustainability-and-competitiveness> >. Acesso em: 22 out. 2018.

LOUREIRO, Antônio. **Redes de Sensores Sem Fio**. Belo Horizonte-MG, Brasil. Disponível em: < [https://s3.amazonaws.com/academia.edu.documents/31991228/179\\_Loureiro\\_Nogueira\\_Ruiz\\_Min\\_i\\_Nakamura\\_Figueiredo.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1552325766&Signature=PnMHZ1PNHDX3UsfowjvMCYnJU%3D&response-content-](https://s3.amazonaws.com/academia.edu.documents/31991228/179_Loureiro_Nogueira_Ruiz_Min_i_Nakamura_Figueiredo.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1552325766&Signature=PnMHZ1PNHDX3UsfowjvMCYnJU%3D&response-content-)

disposition=inline%3B%20filename%3DRedes\_de\_Sensores\_Sem\_Fio.pdf >. Acesso em: 10 fev. 2019.

MOTA, Allan. **O que é Arduino e como funciona**. Brasil. Disponível em: < <https://portal.vidadesilicio.com.br/o-que-e-arduino-e-como-funciona> >. Acesso em: 08 fev. 2019.

NOBLE, Joshua. **Arduino em Ação**. São Paulo, SP. Novatec Editora Ltda, ago. 2013.

OLIVEIRA, Sérgio. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. São Paulo, SP. Novatec Editora Ltda, jun. 2017.

PINHEIRO, Leonardo Fernandes. **Smart Parking: Uma aplicação móvel para Gestão Inteligente de Locais de Estacionamento**. Universidade de Aveiro, Portugal. Disponível em: < <https://ria.ua.pt/bitstream/10773/23460/1/tese.pdf> >. Acesso em: 07 out. 2018.

PINTO, Marcelo da Rocha. **Internet das coisas, cidades inteligentes e mobilidade urbana: um estudo de caso sobre os Smart Parkings em vias públicas e os impactos na qualidade de vida da população**. Niterói-RJ, Brasil. Disponível em: < [https://app.uff.br/riuff/bitstream/1/5671/1/TCC\\_MARCELO\\_DA\\_ROCHA\\_PINTO.pdf](https://app.uff.br/riuff/bitstream/1/5671/1/TCC_MARCELO_DA_ROCHA_PINTO.pdf) >. Acesso em: 25 out. 2018.

SMART PARKING. **Smart Parking Limited**. Smart Parking Ltd, Scotland. Disponível em: < <https://www.smartparking.com/about-us> >. Acesso em: 09 out. 2018.

KIANPINSHEH, Amin. **Smart Parking System (SPS) Architecture Using Ultrasonic Detector**. Universiti Sains Malaysia, Malaysia. Disponível em: < <https://pdfs.semanticscholar.org/d985/e50fc73cd28477af544d52b1e4c9f11f09d5.pdf> >. Acesso em: 15 jan. 2019.

SANTOS, Bruno. **Internet das Coisas: da Teoria à Prática**. Belo Horizonte-MG, Brasil. Disponível em: < <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf> >. Acesso em: 17 jan. 2019.

DIAS, M. M D. **Redes De Comunicação Automotiva: CARACTERÍSTICAS, TECNOLOGIAS E APLICAÇÕES**. São Paulo, SP. Érica, jun. 2010. 220p.

TALIANI, Helio. **Estudo dos Protocolos de Comunicação das Arquiteturas Eletroeletrônicas Automotivas, com foco nas suas características e respectivas aplicações, visando o direcionamento para o uso adequado e customizado em cada categoria de veículo.** São Caetano do Sul-SP, Brasil. Disponível em: < <https://maua.br/files/monografias/estudo-dos-protocolos-de-comunicacao-das-arquiteturas-eletroeletronicas.pdf> >. Acesso em: 19 jul. 2019.

THOMSEN, Adilson. **O que é Arduino.** São Paulo-SP, Brasil. Disponível em: < <https://www.filipeflop.com/blog/o-que-e-arduino> >. Acesso em: 29 jan. 2019.

WENDLING, Marcelo. **Sensores.** Guaratinguetá-SP, Brasil. Disponível em: < <http://www.joinville.udesc.br/portal/professores/saimon/materiais/Sensores.pdf> >. Acesso em: 18 fev. 2019.