

Douglas Navarro, Renan Yuri Lino

**Redes de Sensores de Presença**

São Paulo

2017



Douglas Navarro, Renan Yuri Lino

## **Redes de Sensores de Presença**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia.

Área de Concentração:  
Engenharia de Sistemas Eletrônicos

Orientador: Fernando Josepetti Fonseca

São Paulo

2017



Douglas Navarro, Renan Yuri Lino

## **Redes de Sensores de Presença**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia.

Área de Concentração:  
Engenharia de Sistemas Eletrônicos

Orientador: Fernando Josepetti Fonseca

São Paulo

2017

### Catálogo-na-publicação

Lino, Renan Yuri

Redes de Sensores de Presença / R. Y. Lino, D. Navarro -- São Paulo, 2017.

75 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Internet das Coisas 2.Sensor de Presença 3.Edifícios Inteligentes  
4.Computação em Nuvem 5.ESP8266 I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.  
III.Navarro, Douglas

Douglas Navarro, Renan Yuri Lino

## **Redes de Sensores de Presença**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia.

Trabalho aprovado. São Paulo, 08 de dezembro de 2017:

---

**Fernando Josepetti Fonseca**  
Orientador

---

**Marcelo Knörich Zuffo**  
Convidado

São Paulo  
2017





*"Computers are useless. They can only give you answers."  
Pablo Picasso*



# Resumo

Atualmente não há qualquer inteligência sobre como o prédio da Engenharia Elétrica da Escola Politécnica da USP é ocupado e utilizado pelas pessoas. Deste modo, qualquer proposta de intervenção ou melhoria no edifício é feita sem dados que ajudem a justificá-la, criando uma barreira para o processo de contínua melhoria e atualização. O mesmo problema pode ser encontrado em qualquer edifício público ou particular com intenso fluxo de pessoas, gerando uma demanda por dados sobre a movimentação dos indivíduos dentro dos espaços e subespaços. Além das demandas relacionadas à segurança e melhoria na gestão dos edifícios, a existência de dados confiáveis sobre a ocupação do ambiente também é de interesse econômico para diversas instituições capazes de monetizar a posse da propriedade com a exibição de peças publicitárias em locais de interesse. O objetivo deste trabalho é desenvolver uma rede de sensores de presença que colete dados sobre quais espaços são utilizados, por quantas pessoas, em quais momentos do dia, *etc*, sem, entretanto, identificar positivamente os usuários para respeitar a privacidade dos ocupantes. Este trabalho também propõe a geração de relatórios sobre os dados adquiridos de forma a entregar para a gestão do prédio *insights* sobre como o mesmo é utilizado.

**Palavras-chave:** Internet das Coisas, Sensor de Presença, Edifícios Inteligentes, Computação em Nuvem, ESP8266.



# Abstract

Currently there is no intelligence whatsoever regarding the way that the Electrical Engineering of Polytechnic School of the University of São Paulo building is occupied. Thus, any proposal of intervention or improvement on the building is made without data that help justify it, creating a barrier for the continuous process of update and improvement of this building. The same problem can be found in any public or private building with an intense flow of people, generating a demand for data on the movement of individuals within spaces and subspaces. In addition to the demands related to security and better management of buildings, the existence of reliable data on the occupation of the environment is also of economic interest to several institutions able to monetize the ownership of the property with the display of advertising pieces in places of interest. The goal of this project is to develop a network of presence sensors that collects data on how the spaces are occupied, by how many people, at which times of the day, *etc*, without, however, positively identifying users to respect the privacy of occupants. This project also includes the generation of reports on the acquired data in order to deliver to the building management team insights on how the building is utilized.

**Keywords:** Internet of Things, Presence Detection, Smart Buildings, Cloud Computing, ESP8266



# Lista de ilustrações

Figura 1 – Árvore de necessidades . . . . .	20
Figura 2 – Matriz de pesos para os grupos de requisitos . . . . .	20
Figura 3 – Matriz de pesos para o primeiro grupo de requisitos . . . . .	20
Figura 4 – Matriz de pesos para o segundo grupo de requisitos . . . . .	21
Figura 5 – Matriz de pesos para o terceiro grupo de requisitos . . . . .	21
Figura 6 – Requisitos de marketing . . . . .	22
Figura 7 – Matriz de compromissos . . . . .	25
Figura 8 – Matriz de compromissos de engenharia . . . . .	26
Figura 9 – Decomposição funcional de nível 0 . . . . .	27
Figura 10 – Decomposição funcional de nível 1 . . . . .	27
Figura 11 – A plataforma de desenvolvimento nodeMCU . . . . .	31
Figura 12 – Sensor infravermelho reflexivo . . . . .	32
Figura 13 – Sensor ultrassônico . . . . .	32
Figura 14 – Matriz de decisão das plataformas de desenvolvimento de embarcados. . . . .	37
Figura 15 – Pontuações acumuladas na matriz de decisão . . . . .	37
Figura 16 – Sonares desobstruídos . . . . .	40
Figura 17 – Sonar 2 detecta um obstáculo . . . . .	40
Figura 18 – Sonar 1 detecta um obstáculo . . . . .	41
Figura 19 – Fluxograma do algoritmo de detecção de passagem . . . . .	42
Figura 20 – Fluxograma do algoritmo de transmissão das ocorrências . . . . .	45
Figura 21 – Diagrama de blocos do SoC ESP8266-12E . . . . .	46
Figura 22 – Fotografia do módulo ESP8266-12E . . . . .	47
Figura 23 – Modelo de dados da aplicação no servidor web . . . . .	49
Figura 24 – Diagrama da arquitetura MVC . . . . .	52
Figura 25 – Camadas da aplicação e tecnologias empregadas no desenvolvimento . . . . .	53
Figura 26 – Diagrama da infraestrutura da aplicação . . . . .	54
Figura 27 – Layout da placa de circuito impresso do sistema embarcado . . . . .	56
Figura 28 – Fotografia de dois protótipos confeccionados . . . . .	57
Figura 29 – Interface navegável da API no endpoint para o modelo Movement . . . . .	58
Figura 30 – Interface navegável da API no endpoint para o modelo Movement . . . . .	59
Figura 31 – Widget de contagem de pessoas no dashboard . . . . .	59
Figura 32 – Widget de gráfico do histórico de pessoas no dashboard . . . . .	60
Figura 33 – Widget de gráfico de fluxo de pessoas no dashboard . . . . .	60
Figura 34 – Widget de estado dos sensores no dashboard . . . . .	61





# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Identificação dos Problemas e Necessidades do Cliente</b>	<b>17</b>
<b>1.2</b>	<b>Declaração das Necessidades do Cliente</b>	<b>18</b>
<b>1.3</b>	<b>Declaração dos Objetivos do Projeto</b>	<b>19</b>
1.3.1	Árvore de Necessidades	20
<b>1.4</b>	<b>Requisitos de Engenharia</b>	<b>22</b>
1.4.1	Identificação dos Requisitos de Engenharia	22
1.4.2	Matriz de Compromissos	25
1.4.3	Matriz de Compromissos de Engenharia	25
<b>1.5</b>	<b>Descrição funcional</b>	<b>27</b>
1.5.1	Detalhamento da solução e conceito	28
<b>2</b>	<b>ESTADO DA ARTE</b>	<b>29</b>
<b>2.1</b>	<b>Visão Geral</b>	<b>29</b>
<b>2.2</b>	<b>Revisão da literatura</b>	<b>29</b>
<b>2.3</b>	<b>Tecnologias Relevantes</b>	<b>30</b>
<b>2.4</b>	<b>Discussão sobre o sensor</b>	<b>31</b>
<b>2.5</b>	<b>Discussão sobre o microcontrolador</b>	<b>32</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>35</b>
<b>3.1</b>	<b>Sistema embarcado</b>	<b>35</b>
<b>3.2</b>	<b>Servidor</b>	<b>38</b>
<b>4</b>	<b>IMPLEMENTAÇÃO</b>	<b>39</b>
<b>4.1</b>	<b>Sistema embarcado</b>	<b>39</b>
4.1.1	Detecção de presença e de passagem	39
4.1.2	Registro de horário das ocorrências	43
4.1.2.1	Relógio de Tempo Real	43
4.1.3	Transmissão das ocorrências de passagem	43
4.1.4	Microcontrolador	46
4.1.5	Sensor Ultrassônico	48
<b>4.2</b>	<b>Servidor</b>	<b>48</b>
4.2.1	Arquitetura do software no servidor	48
4.2.2	Modelos de dados	49
4.2.3	Segurança e controle de acesso	50
4.2.4	Interfaces para usuários	50

4.2.5	Pilha de software . . . . .	51
4.2.6	Infraestrutura do servidor . . . . .	52
<b>5</b>	<b>RESULTADOS FINAIS . . . . .</b>	<b>55</b>
<b>5.1</b>	<b>Facilidade de Instalação . . . . .</b>	<b>55</b>
<b>5.2</b>	<b>Interfaces de Usuário Intuitiva . . . . .</b>	<b>57</b>
5.2.1	API . . . . .	57
5.2.2	Painel de Administração . . . . .	58
5.2.3	Dashboard . . . . .	58
<b>5.3</b>	<b>Confiabilidade dos dados . . . . .</b>	<b>60</b>
<b>5.4</b>	<b>Privacidade do usuário . . . . .</b>	<b>61</b>
<b>6</b>	<b>ANÁLISE . . . . .</b>	<b>63</b>
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>65</b>
<b>8</b>	<b>TRABALHOS FUTUROS . . . . .</b>	<b>67</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>69</b>
	<b>ANEXO A – DATASHEET DO SENSOR HC-SR04 . . . . .</b>	<b>72</b>
	<b>ANEXO B – DATASHEET DA DURACELL CR2032 . . . . .</b>	<b>74</b>
	<b>APÊNDICE A – ESQUEMÁTICO DO SISTEMA EMBARCADO . . . . .</b>	<b>75</b>

# 1 Introdução

O problema da ocupação de espaços fechados é de interesse geral para diversos segmentos da indústria e do comércio, viabilizando informações capazes de otimizar a alocação de recursos e gerar novas fontes de receita para ambientes com grande circulação de indivíduos.

Dentro deste contexto, destacamos a necessidade de coletar dados sobre o fluxo em tempo real e de maneira distribuída e, ao mesmo tempo, gerar relatórios completos e de fácil interpretação para todos os interessados e envolvidos na gestão do espaço.

## 1.1 Identificação dos Problemas e Necessidades do Cliente

O estudo da ocupação de espaços fechados é interessante para qualquer gestor que deseja otimizar o uso e a circulação das pessoas dentro de um ambiente selecionado. Compreender a dinâmica da ocupação dos espaços permite traçar planos de ações capazes de alocar recursos de forma mais inteligente, reduzindo custos com energia elétrica, condicionamento térmico e aumento a qualidade do serviço prestado com a alocação correta de pessoal para vigilância, limpeza e outras atividades associadas.

No caso do cliente que será atendido por este projeto a atividade é a ocupação e utilização de espaços fechados, como o prédio da Engenharia Elétrica da Poli e seus diversos sub-espacos. Para este cliente em específico, a otimização da alocação de recursos justifica-se pela redução de gastos imposta pela crise orçamentária da universidade, que restringiu o número de funcionários disponíveis para as atividades de suporte e gestão do uso do edifício, além de problemas recorrentes com a segurança pessoal e patrimonial nas suas dependências.

Todas as pessoas envolvidas com a administração do prédio sofrem com a falta de dados sobre, entre outros aspectos:

- O número de pessoas que acessa o prédio diariamente;
- A variação do número de usuários do prédio de acordo com horários e épocas do ano;
- Como as pessoas utilizam os sub-espacos que existem dentro do prédio.

Nota-se que há gargalos na circulação de pessoas em determinadas áreas, baixa circulação de pessoas em outras, e, de forma geral, uma má utilização do espaço do prédio.

Um produto capaz de atender as necessidades do cliente deve, obrigatoriamente:

- Respeitar a privacidade dos usuários do prédio, evitando a identificação pessoal positiva;
- Mapear o uso e o trânsito dos indivíduos dentro do prédio em função do tempo;
- Ser de fácil utilização para gestores e encarregados dos serviços, entregando as informações de forma clara e concisa através de gráficos e relatórios;
- Permitir a análise histórica dos dados para obtenção de relatórios sobre a utilização das áreas nos períodos de interesse;
- Permitir uma associação geográfica sobre o uso do espaço.

## 1.2 Declaração das Necessidades do Cliente

Conhecer a dinâmica do fluxo de indivíduos dentro de um espaço de interesse é um requisito chave para a otimização dos processos que envolvem a gestão de um edifício onde a circulação de pessoas varia de forma significativa em determinado período de tempo. Essas informações tornam-se ainda mais valiosas quando é necessário alocar recursos de vigilância, limpeza e consumíveis de forma inteligente e dinâmica, buscando reduzir custos e aprimorar o serviço oferecido. Para tanto, o gestor e seus colaboradores precisam de dados confiáveis, acessíveis e atualizados sobre a ocupação do espaço em função do tempo, permitindo a análise da utilização dos subespaços de interesse em uma faixa de tempo selecionada que permitirá, posteriormente, a definição das estratégias de gestão a serem adotadas. Nesse contexto, um sistema capaz de adquirir informações quantitativas sobre a circulação de pessoas de forma automática tornaria possível a aquisição de dados importantes e, consequentemente, produziria insights sobre o que pode ser feito para melhorar o uso do espaço, a alocação dos recursos e o dimensionamento do impacto de ações tomadas. Além de automático, o sistema precisa respeitar as legislações sobre privacidade em ambientes públicos e privados e disponibilizar as informações de forma clara e acessível, diretamente associada ao espaço geográfico delimitado.

## 1.3 Declaração dos Objetivos do Projeto

Desenvolver um sistema de aquisição, registro e exibição do fluxo e acesso de pessoas em pontos estratégicos do espaço de interesse, capaz de monitorar e armazenar informações sobre a circulação das pessoas por longos períodos de tempo, exibindo-os em uma interface amigável e estatisticamente relevante. O projeto utilizará sensores distribuídos e conectados através de um contexto de Internet das Coisas.

Primeiramente, um estudo sobre o sensor mais adequado para a aplicação será realizado levando em conta as necessidades declaradas pelo cliente, envolvendo a prototipagem e o teste de cada uma das alternativas disponíveis. Em seguida, a estrutura da comunicação será definida, analisando os protocolos e esquemas mais adequados para a aplicação. O dispositivo responsável pela coleta dos dados será então integrado com um serviço de armazenamento em nuvem para testes preliminares do protótipo, antes do desenvolvimento de uma plataforma própria para gerenciamento e visualização dos dados. Também será projetada uma placa de circuito impresso para fins de teste e prototipagem avançada.

Após a elaboração da prova de conceito a aplicação web própria será desenvolvida, objetivando o atendimento das necessidades específicas declaradas pelo cliente. Paralelamente, o protótipo das unidades sensoras será transformado em uma placa de circuito impressa definitiva, adicionando robustez e permitindo a fabricação em escala do produto.

### 1.3.1 Árvore de Necessidades

Figura 1 – Árvore de necessidades

Fonte: Autores.

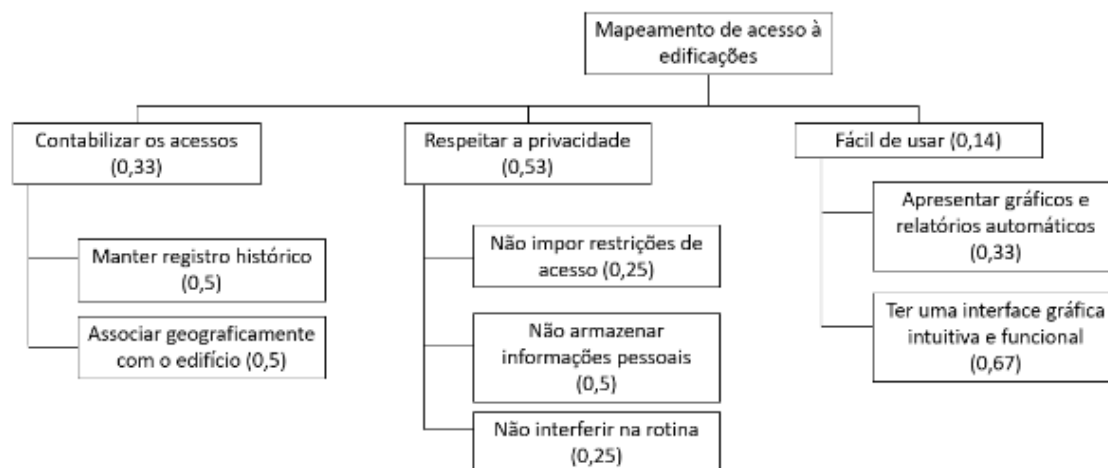


Figura 2 – Matriz de pesos para os grupos de requisitos

Fonte: Autores.

	Contabilizar acessos	Respeitar privacidade	Fácil de usar	Peso Geométrico	Score Normalizado
Contabilizar acessos	1	0,5	3	1,145	0,333
Respeitar privacidade	2	1	3	1,817	0,528
Fácil de usar	0,333	0,333	1	0,481	0,140

Figura 3 – Matriz de pesos para o primeiro grupo de requisitos

Fonte: Autores.

	Manter histórico	Associação geográfica	Peso Geométrico	Score Normalizado
Manter histórico	1	1	1	0,5
Associação geográfica	1	1	1	0,5

Figura 4 – Matriz de pesos para o segundo grupo de requisitos

Fonte: Autores.

	Sem restrição acesso	Sem info. pessoais	Sem interferir na rotina	Peso Geométrico	Score Normalizado
Sem restrição acesso	1	0,5	1	0,794	0,250
Sem info. pessoais	2	1	2	1,587	0,500
Sem interferir na rotina	1	0,5	1	0,794	0,250

Figura 5 – Matriz de pesos para o terceiro grupo de requisitos

Fonte: Autores.

	Gráficos e relatórios	Interface gráfica	Peso Geométrico	Score Normalizado
Gráficos e relatórios	1	0,50	0,707	0,333
Interface gráfica	2	1	1,414	0,667

## 1.4 Requisitos de Engenharia

Uma vez que os requisitos estipulados pelo cliente foram declarados nas seções anteriores, efetuou-se o levantamento dos requisitos de engenharia necessários para atender as necessidades dos usuários do projeto.

### 1.4.1 Identificação dos Requisitos de Engenharia

Os requisitos de engenharia foram estipulados atentando às propriedades de serem abstratos, verificáveis, não ambíguos e rastreáveis.

Figura 6 – Requisitos de marketing

Fonte: Autores.

Identificador	Requisito de Marketing
1	Facilidade de instalação
2	Interface de usuário intuitiva
3	Confiabilidade dos dados
4	Privacidade do usuário



Req. Marketing	Requisito de Engenharia	Justificativa	Classificação
3	Detectar a entrada e a saída de indivíduos da área delimitada com, pelo menos, 95% de precisão	Para que os dados obtidos sejam relevantes o sistema precisa apresentar um desempenho elevado na contabilização do fluxo de pessoas	Desempenho
3	Reter os dados coletados localmente por, pelo menos, 1 dia	Mesmo que o sistema desenvolvido viabilize a coleta de dados para a nuvem, cada entidade de sensoriamento deve ser capaz de reter os dados obtidos localmente caso haja algum problema que impossibilite a transmissão	Desempenho
3	Ter capacidade de transmitir os dados para a rede	Visando a mínima interferência humana, os sensores devem reportar e transmitir as informações coletadas para uma entidade de gerenciamento central através de uma rede de comunicação	Conectividade
1	Os sensores devem operar com as tensões de 127V e 220V	A instalação do dispositivo será realizada em ambientes com tomadas comuns disponíveis para alimentação	Disponibilidade
2	A interface com o usuário deve ser acessível e disponibilizar todas as informações possíveis sobre o edifício monitorado	Os dados coletados devem estar disponíveis para análise por parte dos interessados em uma interface de fácil acesso, que centralize as informações existentes para obtenção de insights sobre a utilização do espaço	Funcionalidade
2	Pré-processar as informações para exibição de relatórios e gráficos	As informações coletadas pelos sensores e centralizadas devem ser resumidas de forma visual para facilitar a interpretação e a usabilidade do sistema	Funcionalidade

1	Utilizar um sensor não-invasivo, sem bloqueio físico de acesso	O objetivo do sistema é conhecer a dinâmica do fluxo de pessoas no edifício, sem impor controle de acesso às áreas monitoradas. Portanto, espera-se que as entidades de coleta de dados não interfiram na circulação dentro dos ambientes	Desempenho
4	Evitar a identificação positiva de indivíduos	A preocupação com a privacidade e o respeito à legislação vigente exigem que o sistema seja incapaz de identificar a registrar informações pessoais. Além disso, o objetivo da solução é apenas contabilizar o fluxo de acesso, tornando desnecessária a associação pessoal.	Legal

### 1.4.2 Matriz de Compromissos

A matriz de compromissos mostra como o atendimento de um requisito de marketing prejudica ou facilita o atendimento de um requisito de engenharia. Espaços vazios na matriz indicam a inexistência de relação relevante entre os requisitos. Os quadrados onde há setas para cima indicam relação de ajuda entre os requisitos e quadrados onde há setas para baixo indicam que o atendimento de um requisito prejudica o outro, inevitavelmente. Por exemplo, aumentar o tempo de operação autônoma prejudica a facilidade de instalação, pois são necessárias baterias maiores ou conexão à rede elétrica.

Figura 7 – Matriz de compromissos

Fonte: Autores.

		Precisão	Retenção dos Dados Localmente	Transmissão dos dados pela rede	Tempo de Operação Autônoma	Disponibilidade de acesso dos Dados	Pré-Processamento de Dados	Invasividade do Sensor	Identificação das Pessoas
		+	+	+	+	+	+	-	-
Facilidade de instalação	+				⇓			⇑	
Interface de usuário intuitiva	+			⇑		⇑	⇑		
Confiabilidade dos dados	+	⇑	⇑	⇑	⇑		↑	⇑	
Privacidade do usuário	+							↑	⇑
Interferência na passagem	-	⇓						⇑	↑

### 1.4.3 Matriz de Compromissos de Engenharia

A matriz de compromissos de engenharia mostra como o atendimento de um requisito de engenharia prejudica ou facilita o atendimento dos outros requisitos de engenharia. Para este projeto foram mapeadas poucas relações.

A primeira foi entre retenção local dos dados e transmissão dos dados pela rede. A retenção dos dados localmente ajuda a transmissão dos dados pela rede pois esta

rede pode falhar ocasionalmente e, tendo os dados armazenados localmente por algum tempo, é possível tentar retransmiti-los.

A disponibilidade de acesso dos dados prejudica a retenção local dos dados pois, idealmente, nenhum dado seria armazenado localmente - eles seriam imediatamente acessíveis através da internet. Entretanto, esta situação não ocorre em uma implementação. A transmissão dos dados pela rede é importante para a disponibilidade de acesso aos dados, já que um requisito depende diretamente do outro. Finalmente, a invasividade do sensor é prejudicada pela precisão do sensor, já que quanto mais invasivo é o sensor menor é a complexidade necessária para garantir uma boa precisão nas detecções. A tradicional catraca giratória é um bom exemplo disto. No outro extremo, câmeras podem detectar pessoas passando mas a tecnologia de visão computacional é bastante mais avançada.

Figura 8 – Matriz de compromissos de engenharia

Fonte: Autores.

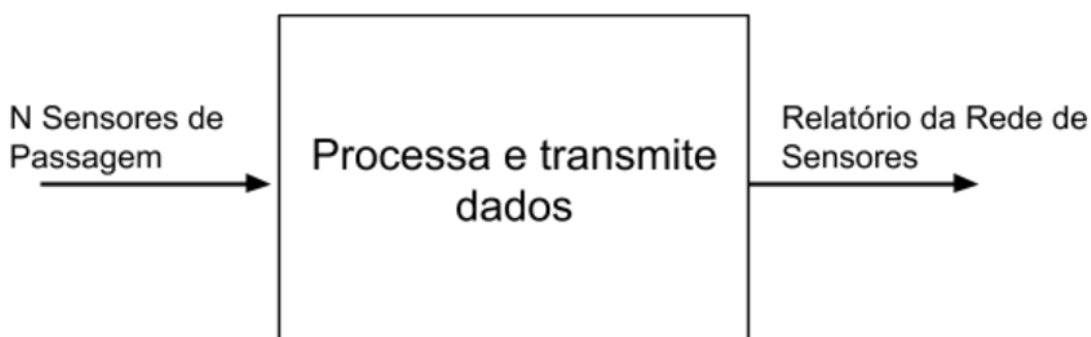
		Precisão	Retenção dos Dados Localmente	Transmissão dos dados pela rede	Tempo de Operação Autônoma	Disponibilidade de acesso dos Dados	Pré-Processamento de Dados	Invasividade do Sensor	Identificação das Pessoas
		+	+	+	+	+	+	-	-
Precisão	+							↓	
Retenção dos Dados Localmente	+			↑		↓			
Transmissão dos dados pela rede	+					↑↑			
Tempo de Operação Autônoma	+								
Disponibilidade de acesso dos Dados	+								
Pré-Processamento de Dados	+								
Invasividade do Sensor	-								
Identificação das Pessoas	-								

## 1.5 Descrição funcional

A descrição funcional do projeto através de diagramas de bloco mostra como o sistema deverá gerar e processar a informação para geração de valor do produto. Em nível zero, utilizamos um único bloco para mostrar como o sistema opera da forma mais abstrata possível: os sensores de passagem processam e transmitem dados para o sistema de geração de relatórios.

Figura 9 – Decomposição funcional de nível 0

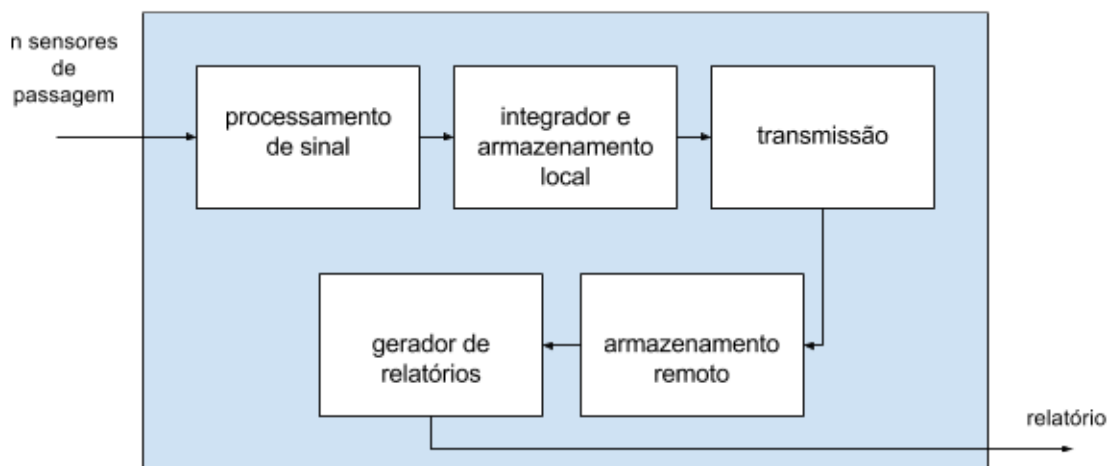
Fonte: Autores.



Uma descrição de nível um mostra etapas intermediárias e importantes para o funcionamento apropriado do produto, sendo possível discernir quais etapas serão realizadas pelos sistemas embarcados (processamento de sinal, armazenamento local e transmissão) e quais serão realizadas pelo sistema externo à rede: geração de relatórios e armazenamento remoto.

Figura 10 – Decomposição funcional de nível 1

Fonte: Autores.



### 1.5.1 Detalhamento da solução e conceito

Os sensores de passagem são sensores ultrassônicos de distância. Cada dispositivo incorpora um sensor, que através de constantes medidas de distância, conseguem inferir se houve ou não a passagem de uma pessoa na direção em que aponta. Por exemplo, um sensor ultrassônico posicionado no teto de um corredor e que aponta para o chão, irá constantemente medir uma distância de, digamos, 3 metros, que é a distância entre o teto e o chão. Entretanto, quando uma pessoa passa, esta medida deve mudar para ( 3- altura da pessoa ) metros. Este sensor não emite luz, som, tampouco exige contato físico com o usuário, sendo ideal para o requisito de invasividade.

Cada sinal do sensor é processado localmente pelo dispositivo que adquire este sinal de modo a aumentar a confiabilidade da medida de distância. O sensor ultrassônico é sujeito a realizar leituras discrepantes por causa de fenômenos acústicos, como reverberação da onda sonora. Este processamento pode ser considerado bastante simples e consiste apenas no cálculo da média de um número de leituras. Este processamento transforma uma série de leituras consecutivas mas com algumas medidas defeituosas e, fazendo o cálculo de uma média, retorna uma leitura bem mais confiável. Este bloco funcional está diretamente relacionado ao requisito confiabilidade.

O sinal devidamente tratado contém a informação sobre o fluxo efetivo de pessoas no local de instalação e, portanto, deve ser armazenado localmente para atender aos requisitos funcionais de engenharia apresentados anteriormente. Dessa forma, otimizamos os recursos de rede evitando transmissões frequentes e desnecessárias, além de aumentar a robustez das unidades sensoras.

A transmissão dos dados locais para a nuvem se dá em intervalos pré-programados, quando os valores armazenados no integrador são formatados e empacotados de acordo com as definições do protocolo de comunicação utilizado para envio de informações ao servidor remoto.

Os dados recebidos pelo servidor devem ser armazenados de forma persistente por longos períodos de tempo, para permitir a geração de análises e relatórios temporais de acordo com os requisitos estabelecidos pelo cliente.

O módulo gerador de relatórios é responsável por acessar a informação persistente armazenada nos servidores e construir gráficos, tabelas e análises sobre o uso do edifício de acordo com as informações demandadas pelo usuário do sistema.

## 2 Estado da arte

Nesta seção será apresentado um panorama das soluções atuais para o problema abordado. São também citados artigos que analisam o problema com uma visão de internet das coisas e edifícios inteligentes. Também será apresentada uma análise preliminar de possíveis soluções.

### 2.1 Visão Geral

O controle do número de pessoas em um determinado espaço é fundamental para determinadas aplicações e já existe em contextos onde sua existência é um fator crítico como, por exemplo, os restaurantes universitários, sistemas de transporte público e outros espaços com fluxo inconstante de pessoas e sujeitos a lotação máxima.

Na maioria das aplicações atuais, a aquisição e o registro do número de indivíduos é feita de forma primitiva utilizando catracas, que contabilizam o número de usuários que atravessam um determinado ponto (geralmente um gargalo). O contador da catraca deve ser registrado constantemente e de forma manual, tornando o processo de coleta e análise dos dados altamente ineficiente. Além disso, as catracas impõe uma barreira física ao usuário, que acaba por interferir na sua interação com o ambiente em situações em que não se deseja restringir o acesso, mas simplesmente realizar uma contagem.

Algumas aplicações mais modernas podem ser encontradas, por exemplo, nos acessos internos de integração das estações na Linha 4 - Amarela do Metrô de São Paulo. Pequenos totens são instalados nas áreas de controle de fluxo e, utilizando sensores de barreira, as unidades são capazes de contabilizar o fluxo de usuários entrando e saindo do sistema de transporte sem imposição de um bloqueio físico.

O projeto em questão procura fazer esta aquisição e registro em tempo real através da internet e sem a necessidade de uma catraca ou obstáculo físico. Isso permite que a informação possa ser distribuída para uma infinidade de usuários em tempo real, além de ser facilmente reproduzido para que realize medidas em diversos pontos do edifício.

### 2.2 Revisão da literatura

Com o crescente interesse no desenvolvimento de edifícios inteligentes e o esforço de pesquisa concentrado em soluções que aumentem a eficiência energética

de residências e espaços comerciais, o interesse pelo mapeamento do fluxo de pessoas em espaços fechados tem sido abordado frequentemente em artigos científicos.

Os sensores e abordagens utilizadas são bastante diversos. É abordada, por exemplo, a necessidade de utilizar diversos tipos de sensores para a obtenção de leituras confiáveis sobre o número de ocupantes de um determinado espaço.(LUPPE; SHABANI, 2017),

Uma abordagem bastante completa da arquitetura fog-cloud é discutida por Dutta e Roy, onde os dispositivos se reportam a uma entidade central para permitir o controle geral do prédio e uma visualização completa do estado do edifício.(DUTTA; ROY, 2017)

Minoli et. al. discute as principais tecnologias que permitem uma implementação real de Internet das Coisas em contextos de edifícios inteligentes, discutindo os principais desafios de segurança impostos.(MINOLI; SOHRABY; OCCHIOGROSSO, 2017)

Algumas discussões encontradas na literatura são estritamente teóricas, fornecendo modelos matemáticos para a modelagem da ocupação de edifícios (LIU; ZHANG; GENG, 2017; HUA et al., 2016; JIN et al., 2017; REYNA et al., 2017).

Nas implementações reais com utilização de sensores, a aplicação do Arduino e do nodeMCU como entidades computacionais é encontrada como solução de baixo custo e rápida visualização de resultados (LUPPE; SHABANI, 2017).

## 2.3 Tecnologias Relevantes

A tecnologia para realizar tal projeto está crescendo bastante nos últimos anos tanto do ponto de vista de hardware quanto de software.

No hardware temos SoC's como o ESP8266, da chinesa Espressif, que tem baixo custo e plena capacidade de comunicação WiFi. O ESP8266 pode ser encontrado em diversas versões, inclusive integrado à plataformas de desenvolvimento como o nodeMCU. Para aplicações mais robustas em requisitos computacionais, como soluções envolvendo processamento de imagens, existem plataformas baseadas em Linux embarcado disponíveis como o Raspberry Pi e o Intel Edison, ambas com boas soluções de conectividade.

No campo dos sensores os tradicionais sensores de barreira infravermelha são uma opção de baixo custo e já amplamente utilizadas na aplicação em questão. Existem outras alternativas a serem consideradas, como câmeras e sensores de temperatura, com suas vantagens e desvantagens, que serão cuidadosamente analisadas durante o desenvolvimento da solução.



Figura 11 – A plataforma de desenvolvimento nodeMCU



No software, especificamente na camada de comunicação, temos protocolos leves pensados justamente para dispositivos limitados como o ESP8266, tal como é o protocolo MQTT (Message Queue Telemetry Transport) ou o CoAP (Constrained Application Protocol). Os dois protocolos foram desenvolvidos para aplicações M2M (Machine-To-Machine), em contextos de aplicação de dispositivos computacionalmente limitados conectados a redes limitadas em termos de banda e conectividade.

## 2.4 Discussão sobre o sensor

Inicialmente desejamos que o sensor detecte simplesmente a passagem de pessoas, não necessariamente em qual direção. Considerada esta exigência, temos as seguintes opções:

Sensor IR: Um par LED infra-vermelho/fotodiodo é alinhado e o LED emite constantemente luz infravermelha. Quando o fotodiodo conduz, sabemos que houve reflexão da luz e, portanto, a passagem de uma pessoa. Há sensores que são vendidos especificamente para este propósito, como o representado na figura a seguir, que utilizando lentes permite até ajustar a distância de detecção. Um sensor típico é apresentado na Figura 12.

- Distância: 3-30cm
- Consumo: 100mA
- Preço: R\$ 24,81 no Mercado Livre
- Tensão: 10V a 30V (saída em lógica 5V)

Sensor Ultrassom: Este sensor opera com princípio semelhante ao sensor IR, mas emitindo um sinal sonoro de alta frequência no lugar do sinal luminoso. Uma

Figura 12 – Sensor infravermelho reflexivo



breve pesquisa permitiu estimar diversas vantagens no uso deste sensor como baixo consumo de corrente, medida de maior alcance além de custo menor. Um sensor do tipo HC-SR04 é apresentado na Figura 13.

- Modelo: HC-SR04
- Distância: 2-450 cm
- Corrente em Stand by: Menos de 2mA
- Preço: R\$ 6,89 no Mercado Livre

Figura 13 – Sensor ultrassônico



## 2.5 Discussão sobre o microcontrolador

Dentre as diversas opções de sistemas microcontrolados ou até relativamente poderosos computadores embarcados como o Raspberry Pi, decidimos que o sistema que mais se adequa às especificações deste projeto é o NodeMCU, apresentado na Figura 11. O NodeMCU é um kit de desenvolvimento que facilita a prototipagem de projetos baseados no ESP8266-12.

O ESP8266-12 é um microcontrolador com Wi-Fi integrado que pode ser programado tanto em C quanto em Lua, tendo seu firmware constantemente atualizado<sup>1</sup>.

Considerando o grande potencial desta plataforma, é possível notar que é a mais versátil e ao mesmo tempo acessível, sendo encontrada à venda online de vendedores brasileiros por preços que vão de R\$25,00 a R\$30,00

---

<sup>1</sup> para mais detalhes visite o repositório <https://github.com/nodemcu/nodemcu-firmware> no GitHub



## 3 Materiais e Métodos

Nesta seção serão apresentadas as ideias que motivaram a escolha das ferramentas utilizadas no contexto do desenvolvimento deste projeto, bem como os procedimentos utilizados no mesmo desenvolvimento. O sistema finalizado apresenta componentes de *hardware* e *software* e, essa última, em contextos bastantes distintos: no de um sistema embarcado e no de uma aplicação *web*.

Sendo assim, as variáveis e questionamentos que surgiram em cada etapa do desenvolvimento foram analisadas sempre com atenção às limitações do contexto e das ferramentas utilizadas para o mesmo, produzindo resultados de natureza bastante distinta para atender os objetivos de engenharia e de marketing.

### 3.1 Sistema embarcado

Adotando um processo de implementação incremental, não foram considerados plataformas de desenvolvimento orientadas a produção em grande escala e com alto desempenho, como kits de desenvolvimento para FPGAs <sup>1</sup> ou para processadores ARM. A desvantagem de se abordar a implementação orientada a produção em grande escala é que se sacrifica muito esforço para uma etapa que não será atingida.

Tal esforço surge do fato de que nessas implementações o desenvolvimento se inicia a partir de camadas de abstração muito simples, algo vantajoso para obter alto desempenho do produto final, mas que exige tempo incompatível com o estipulado por este trabalho. Os requisitos de marketing e de engenharia não exigem indicadores de desempenho elevados, especialmente relativos a tempo de processamento das informações, sendo desnecessário abordar a implementação de forma orientada a desempenho. Um sistema de segurança como, por exemplo, um sistema de *air-bags* de um veículo, sofreria tais exigências.

As plataformas de desenvolvimento compatíveis com a abordagem adotada mapeadas, considerando todos os modelos de cada família, foram:

- Arduino
- Raspberry Pi
- NodeMCU

---

<sup>1</sup> Field Programmable Gate Array é um chip que contém milhões de portas lógicas que podem ser programadas para implementar qualquer tipo de lógica. É bastante utilizado para sistemas fabricados em grande escala.

Para decidir qual plataforma de desenvolvimento seria mais adequada, foram considerados os fatores:

- Rapidez e facilidade de implementação

Apesar de todas as plataformas consideradas serem voltadas para prototipagem rápida e não para produção industrial, neste quesito a grande disponibilidade de bibliotecas de código e outros recursos para o Arduino e NodeMCU, que dispõe da mesma forma de tais recursos, fortalece estes dois candidatos.

Os Raspberry Pi, por serem uma plataforma mais próxima de um computador de propósito geral, facilitam o desenvolvimento de aplicações de alto nível de abstração. Como a arquitetura do sistema desenvolvido neste trabalho já assume que os serviços web (de alto nível de abstração) serão executados na nuvem, este candidato perde o seu brilho.

- Compatibilidade entre poder de processamento disponível e necessário

O Arduino em suas versões mais simples não dispõe de poder de processamento abundante, com um microcontrolador da família AVR-8, de 8 bits, do fabricante *Atmel* mas ainda assim suficiente para o projeto em questão.

O NodeMCU tem o seu núcleo de processamento integrado a interface *WiFi* com um processador de 32 bits, mais potente que o AVR de 8 bits mas ainda assim adaptado a sistemas embarcados que buscam baixo consumo energético.

Os modelos de Raspberry Pi disponíveis trazem um processador bastante potente para a aplicação deste trabalho, se aproximando de um computador de propósito geral. O Raspberry Pi é capaz de processar vídeo e realizar outras tarefas semelhantes a de um computador doméstico, sendo uma má escolha de acordo com este critério.

- Facilidade de montagem de protótipo funcional e instalável

A família Arduino oferece diversas opções de placas para realizar a prototipagem, dispondo, entre seus diversos representantes, versões mais compactas e fáceis de instalar em um protótipo funcional. Além disso, a vasta integração de módulos de *hardware* permite alterar os sensores que serão utilizados com facilidade. É um bom candidato segundo este critério. É importante também mencionar que os *chips* relativamente simples da família AVR-8 facilitam projeto de uma placa de circuito impresso de face única para um protótipo instalável.

O módulo ESP8266 apresenta dimensões reduzidas e conexões um pouco mais complicadas do que os microcontroladores AVR-8. Diferente dos microcontroladores AVR-8 que apresentam encapsulamentos *Dual In Line* facilmente soldáveis

manualmente, seus pinos exigem uma montagem *surface mount* que, apesar de dificultar a produção manual de protótipos, não impossibilita tal produção.

Os representantes da linha Raspberry Pi exigiriam que todo o kit de desenvolvimento utilizado seja mantido em um protótipo de campo, já que o processador exige técnicas industriais de soldagem, bem como a placa de circuito impresso também fica complexa demais para produção viável de apenas algumas unidades.

- Reprodutibilidade

Este quesito avalia a facilidade de se reproduzir o hardware. É um quesito a ser considerado já que o projeto parte do princípio de que deve haver diversos sensores para compor a rede.

De forma análoga ao quesito 'facilidade de montagem de protótipo funcional e instalável', as plataformas NodeMCU e Arduino levam vantagem.

Entretanto, o peso deste quesito é menor considerando que este trabalho não foca na etapa de produção em grande escala dos sensores.

Nas figuras 14 e 15 é possível observar a matriz de decisão do kit de desenvolvimento com os pesos escolhidos entre 1 e 5.

Figura 14 – Matriz de decisão das plataformas de desenvolvimento de embarcados.

Fonte: Autores.

Critérios de Decisão	Peso	Raspberry Pi		Arduino		Node MCU	
		Nota	Ponderação	Nota	Ponderação	Nota	Ponderação
Rapidez e facilidade de implementação	3	3	9	5	15	4	12
Compatibilidade entre poder de processamento disponível e necessário	4	1	4	3	12	5	20
Facilidade de montagem de protótipo funcional e instalável	4	1	4	4	16	4	16
Reprodutibilidade	2	1	2	4	8	4	8

Figura 15 – Pontuações acumuladas na matriz de decisão

Fonte: Autores.

Ponderação Final	
Candidatos	Pontuação
Raspberry Pi	19
Arduino	51
NodeMCU	56

A plataforma NodeMCU será a escolhida.

## 3.2 Servidor

A aplicação web para centralização dos dados e geração dos relatórios pode ser desenvolvida em servidores próprios ou em serviços na nuvem.

A economia com a terceirização da infraestrutura e a possibilidade de escalar a performance da aplicação com poucos cliques é um diferencial importante em relação à opção de rodar um servidor próprio. Além disso, a configuração na nuvem permite o teste do produto com nenhum investimento em hardware próprio, diminuindo drasticamente o investimento inicial requerido pelo projeto.

Entretanto, vale destacar o maior controle e segurança sobre os dados exercido quando a aplicação é executada em equipamento próprio e centralizado.

Optou-se pelo uso da infraestrutura em nuvem, tendo em vista a agilidade no desenvolvimento obtida com esta escolha.



## 4 Implementação

Neste capítulo serão discutidos todos os algoritmos e estruturas, tanto do sistema embarcado quanto do servidor, que foram necessários para o funcionamento desejado do sistema. Também serão discutidas decisões de projeto feitas para viabilizar tais estruturas e algoritmos.

### 4.1 Sistema embarcado

Definir o que é um sistema embarcado não é tarefa fácil, mas pode-se dizer que é qualquer sistema computacional que não é um *desktop*, ou outro computador de propósito geral (GIVARDIS, 2002). Neste projeto, o propósito do sistema é bastante específico: detectar e reportar quantas pessoas passaram em uma ou outra direção. Portanto, pode ser considerado um sistema embarcado.

#### 4.1.1 Detecção de presença e de passagem

Dentre os sensores levantados após pesquisa, independentemente da natureza de seu funcionamento, no contexto deste projeto tal sensor será utilizado como um produtor de feixe de luz ou de som que, tendo tal feixe interrompido ou habilitado devido a presença de um obstáculo, possibilita que o sistema infira a presença de tal obstáculo.

Para a detecção de presença de um obstáculo, basta a existência de um único feixe. Para que o sistema seja capaz inferir se houve movimento, basta analisar o padrão com o qual este feixe é interrompido ao longo do tempo.

Entretanto, para inferir a direção em que houve o movimento de uma pessoa, é necessário que haja dois ou mais sensores em ação. Posiciona-se os sensores a uma certa distância e produzindo os feixes paralelamente. Isso permite que o sistema detecte se houve o acionamento de um sensor relativamente ao outro no tempo. Dependendo de qual ou quais sensores foram acionados primeiro, é possível concluir a direção do movimento do obstáculo.

Na figuras 16, 17 e 18 está representado tal esquema de detecção de passagem de movimento utilizando sensores ultrassônicos. Para os sensores ultrassônicos, note que a ativação do sensor ocorre com o retorno da onda sonora devido à reflexão no obstáculo, que, na ausência de tal obstáculo, demora demais para retornar e é ignorada pelo sistema. Na figura 16, há ausência de obstáculos. Depois, há detecção de presença no sensor ultrassônico 2, como mostrado na figura 17. Finalmente, a detecção de presença no sensor ultrassônico 1 em uma janela curta de tempo após a

primeira detecção permite concluir que houve movimento da direita para a esquerda da figura, como representado na figura 18.

Figura 16 – Sonares desobstruídos

Fonte: Autores.

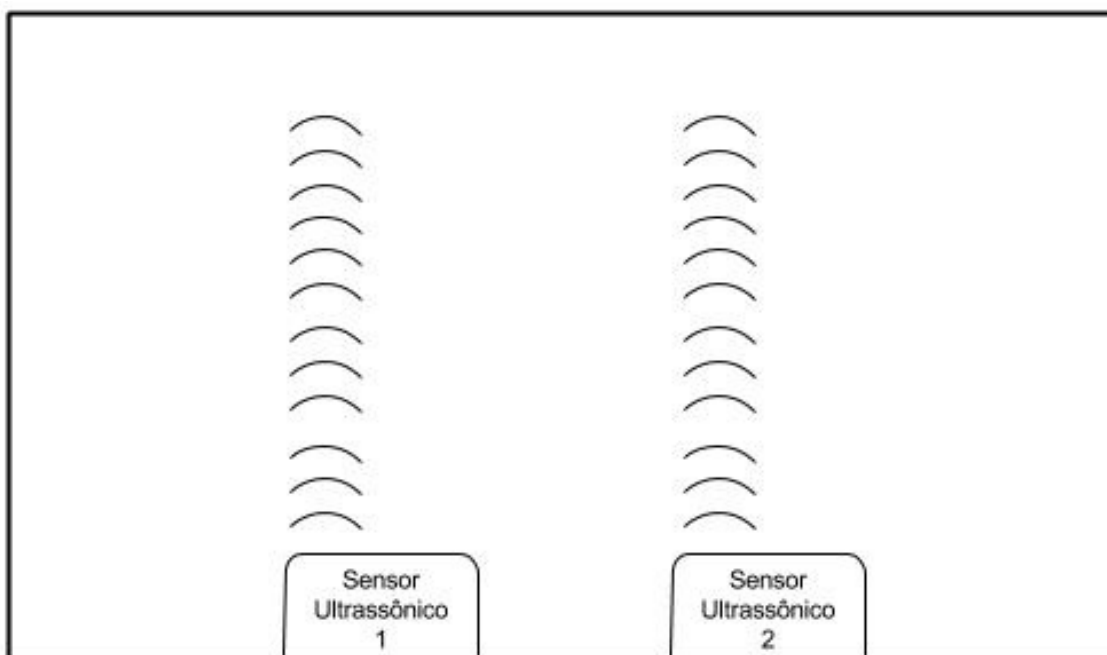


Figura 17 – Sonar 2 detecta um obstáculo

Fonte: Autores.

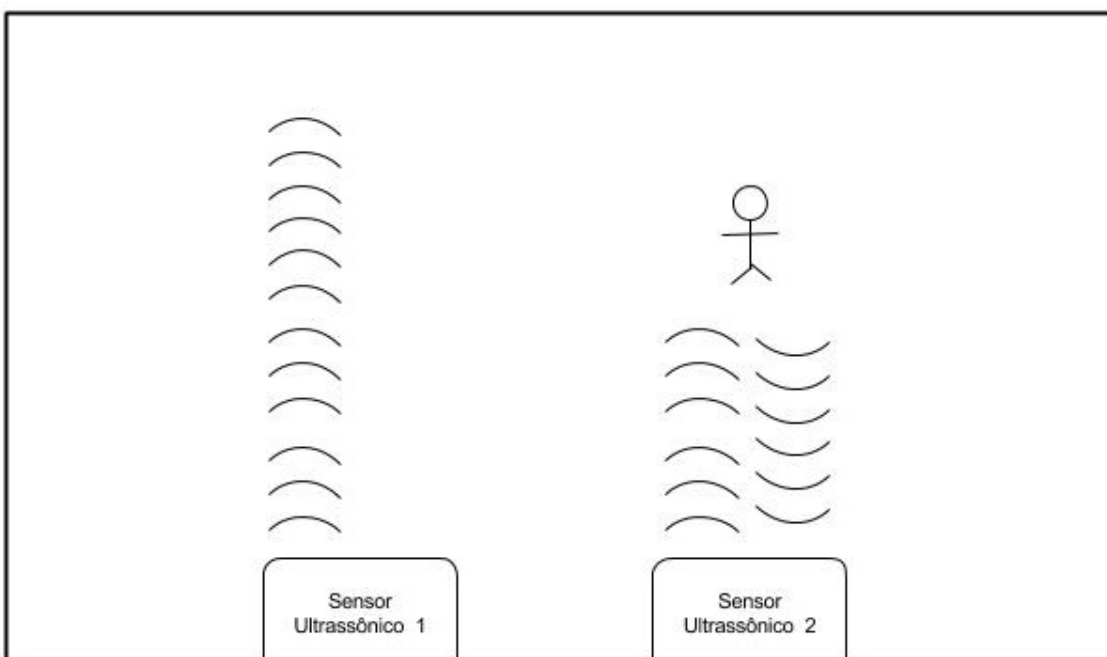
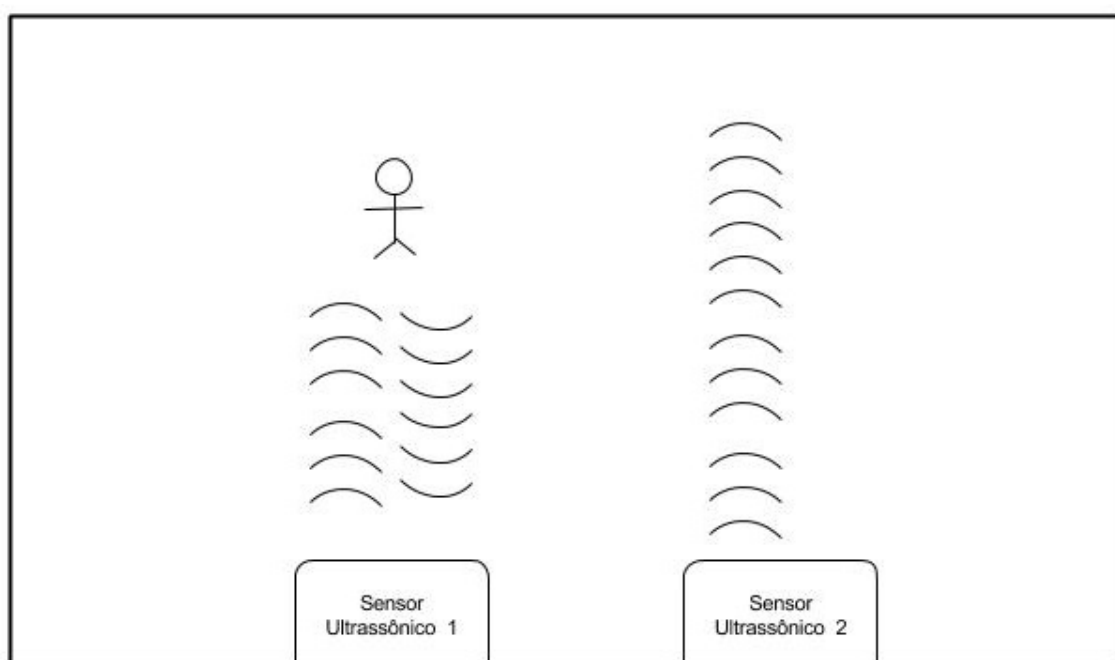


Figura 18 – Sonar 1 detecta um obstáculo

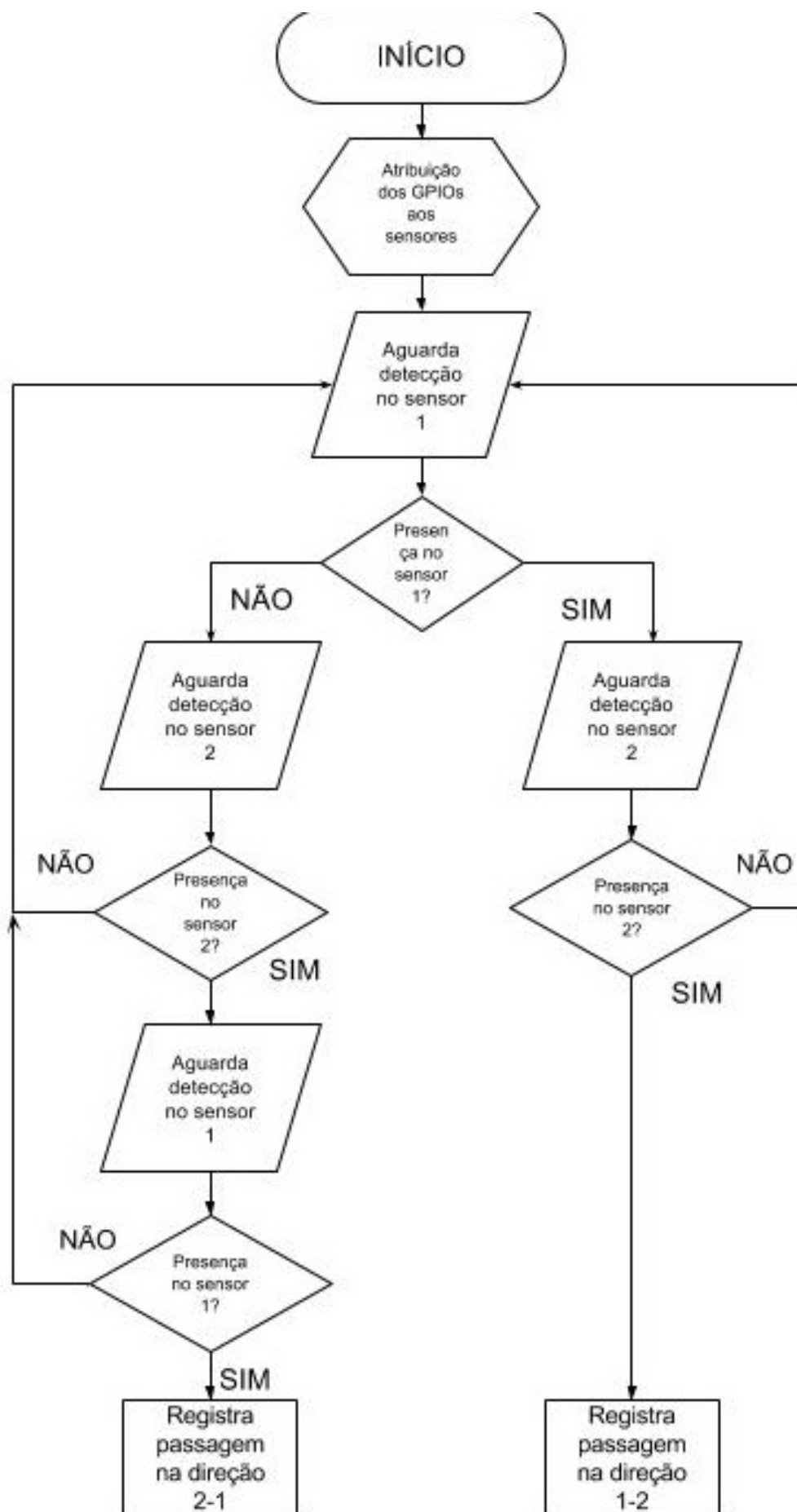
Fonte: Autores.



A lógica que realiza tal detecção de forma contínua pelo sistema pode ser descrita através do fluxograma representado na figura 19.

Figura 19 – Fluxograma do algoritmo de detecção de passagem

Fonte: Autores.



### 4.1.2 Registro de horário das ocorrências

Para que a visualização das ocorrências de entradas ou saídas faça sentido, é fundamental que haja uma data atrelada às mesmas. Esta data pode ser produzida através de um relógio no lado do servidor, registrando quando as mensagens são recebidas, ou ela pode ser produzida no lado do sensor, registrando quando as mensagens de fato são produzidas. Neste projeto ambos horários serão considerados, sendo importante o horário do lado do sensor para garantir a qualidade da informação e do lado do servidor para monitorar o atraso do sistema.

#### 4.1.2.1 Relógio de Tempo Real

Conforme discutido anteriormente, é fundamental para o sistema ter localmente uma referência de tempo. Tal referência será adquirida através de um módulo RTC, composto principalmente de um circuito integrado digital alimentado por bateria capaz de contar tempo de forma precisa por períodos de anos.

Os principais representantes de tais circuitos integrados são o DS1307 e o DS3231. A diferença entre os dois chips é que o DS3231 é, de acordo com os *datasheets* de ambos, mais preciso graças a um mecanismo de compensação contra a variação do oscilador interno devido às temperaturas externas. Para o DS1307 não é especificada uma acurácia, mas para o DS3231 a acurácia é de  $\pm 2$  partes por milhão o que resulta em atrasos da ordem de minutos por ano, desprezíveis para a aplicação discutida neste trabalho.

Uma grande vantagem de se utilizar tais circuitos é a contagem de dias do mês, meses e anos de forma correta até o ano 2100, o que é uma exigência do projeto para garantir a qualidade da informação. Além disso, estes relógios são imunes a quedas de energia e não precisam ser reconfigurados, já que contam com uma bateria externa. O consumo padrão é de 500nA e uma bateria CR2032 geralmente tem uma capacidade de 240mAh, permitindo um período padrão de descarga de aproximadamente 54 anos.

Como o circuito deve contar com alimentação constantemente, é desprezível uma análise da capacidade da bateria do módulo RTC, pois a mesma nunca irá descarregar. O fabricante Duracell garante um prazo de validade de 10 anos para estas baterias, então este é o período máximo recomendado para que a mesma bateria seja utilizada no sistema.

### 4.1.3 Transmissão das ocorrências de passagem

A transmissão das ocorrências não deve ser frequente demais, pois isso pode sobrecarregar o servidor e, de acordo com a arquitetura proposta, sequer é possível

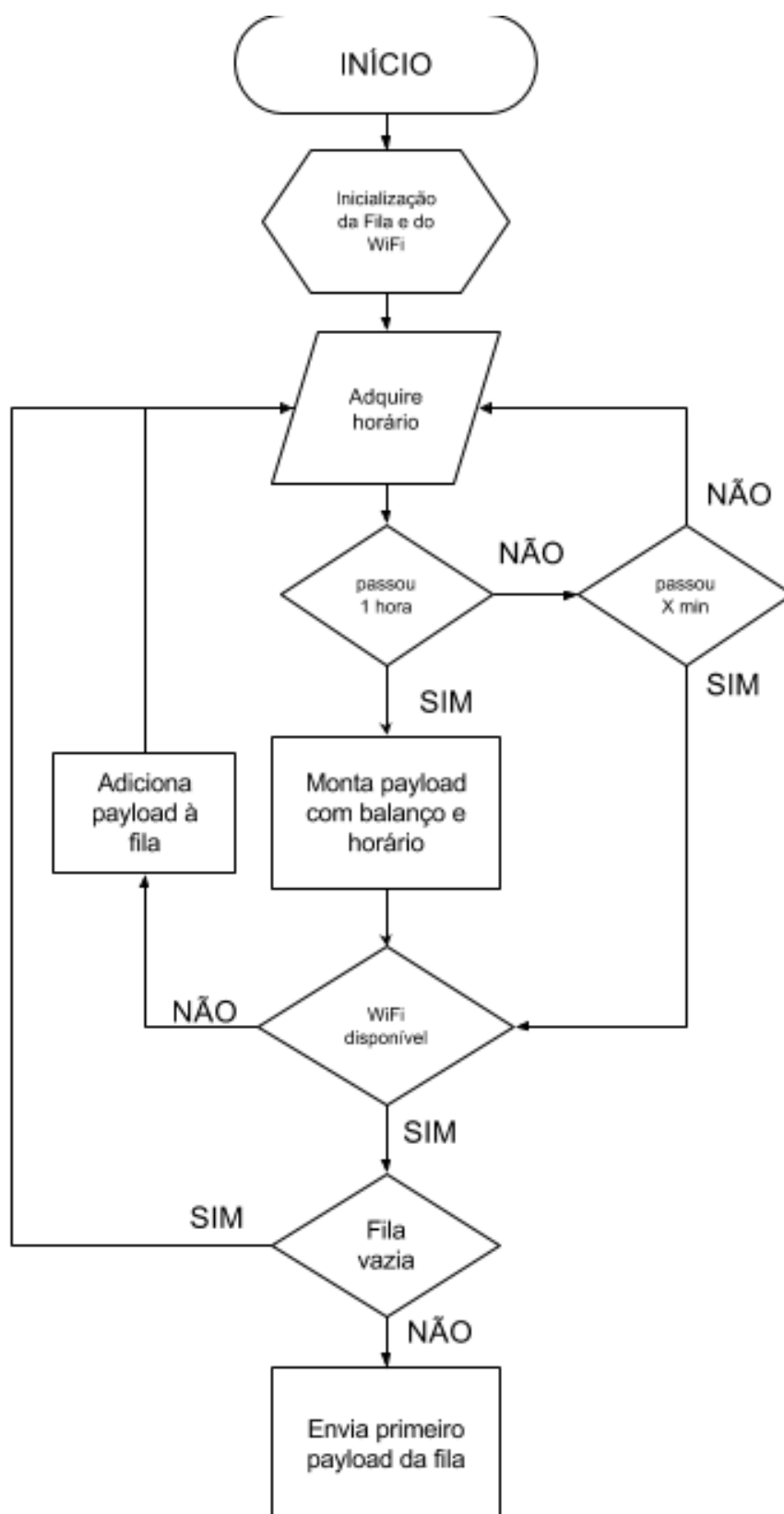
observar ocorrências com resolução menor que a de horas. Então, o mais adequado é registrar as ocorrências de hora em hora, na forma de um balanço.

Após uma hora de coleta de ocorrências, deve ser composta uma mensagem HTTP POST que será enviada ao servidor como um modelo do tipo Movement. De acordo com este modelo, o campo value deve conter a quantidade integrada de detecções, ou seja, o balanço das passagens. Uma passagem na direção de entrada do ambiente incrementa em uma unidade o balanço, enquanto uma passagem na direção de saída do ambiente decrementa de uma unidade o mesmo balanço. Neste mesmo modelo deve ser registrada a hora e data da produção de tal balanço. Este campo é importante para que seja mantida a fidelidade entre o momento em que os dados foram produzidos e o momento em que foram registrados. Dependendo das condições da rede, tais momentos podem diferir bastante e comprometer a informação.

Então, produzidos e armazenados os dados, podemos notar que deve haver um mecanismo de armazenamento local dos dados enquanto eles, por algum problema com a transmissão, não puderem ser enviados. Uma estrutura eficiente para armazenar e organizar tais dados é a fila. A lógica que deve ser executada pelo sistema pode ser representada pelo fluxograma da Figura 20:

Figura 20 – Fluxograma do algoritmo de transmissão das ocorrências

Fonte: Autores.



Note que a checagem de X em X minutos tem a função de descarregar a fila,

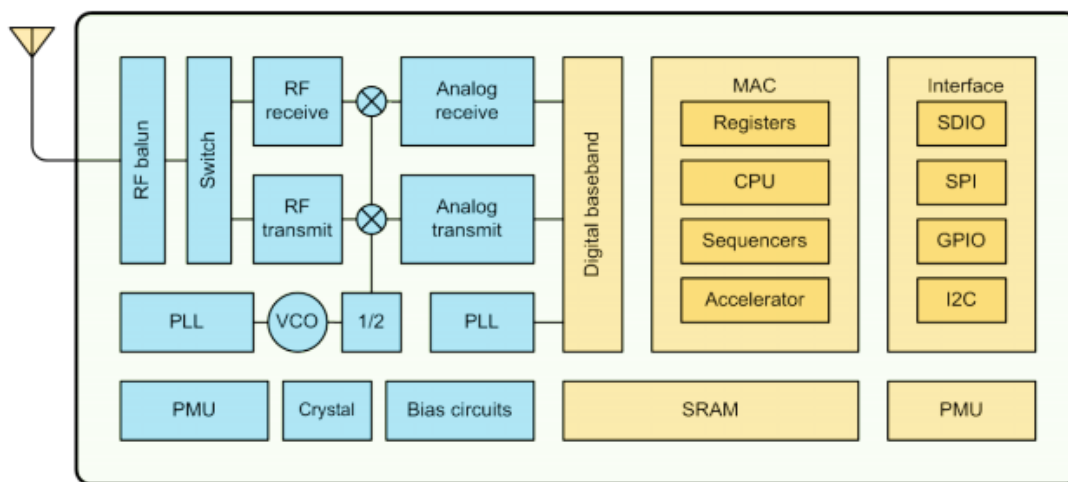
permitindo que o sistema compense um período de tempo sem conectividade. Para  $X = 15$  minutos, o sistema pode ficar 4 horas consecutivas sem conectividade. Para  $X = 5$  minutos, o sistema pode ficar 12 horas sem conectividade. É possível ajustar o tempo de verificação  $X$  de forma que ele sempre seja capaz de descongestionar o sistema fazendo  $X = 60 \text{ minutos} / \text{número de itens acumulados na fila}$ , desde que este número seja sempre maior ou igual a um.

#### 4.1.4 Microcontrolador

O sistema escolhido para a implementação foi o ESP8266-12E. O ESP8266-12E é um System On a Chip que conta com um microcontrolador Tensilica L106 de 32 bits de baixo consumo energético e com WiFi e antena integrados.

Figura 21 – Diagrama de blocos do SoC ESP8266-12E

Fonte: (ESPRESSIF, 2017)



Os módulos ESP são produzidos pela empresa chinesa Espressif e a mesma fornece um Software Development Kit na linguagem C, o que permite desenvolver aplicações de baixo nível de abstração para o módulo. Tais aplicações de baixo nível de abstração são de alto desempenho, permitindo o uso de sistemas operacionais de tempo real (RTOS), personalização de operações de radiofrequência, configuração da frequência de operação do chip e diversas operações capazes de modificar o funcionamento do sistema em um nível muito próximo do hardware. Nesta abordagem de desenvolvimento de software há pouquíssimas bibliotecas disponíveis e todo o desenvolvimento da aplicação deve ser executado a partir de uma base fundamental de comandos e funções do módulo. (ESPRESSIF, 2017)



Figura 22 – Fotografia do módulo ESP8266-12E

Fonte: (ESPRESSIF, 2017)



Entretanto, há também outros kits de desenvolvimento de software, baseados no primeiro que já foi mencionado, e que foram desenvolvidos pela comunidade de usuários dos módulos e outras empresas e focam em facilitar o desenvolvimento das aplicações com mais camadas de abstração. Tais kits buscam diminuir o tempo de desenvolvimento e implementação do projeto em detrimento do desempenho final do sistema.

Entre estas opções de kits temos o NodeMCU, que utiliza a linguagem Lua e dispõe de um acervo médio de bibliotecas; temos também o MicroPython (MICROPYTHON, 2017), que permite o desenvolvimento em linguagem Python mas atualmente com documentação pobre e poucas bibliotecas; outra opção é o Mongoose OS, (MONGOOSE-OS, 2017) que permite o desenvolvimento em javascript para software de prototipagem e C e C++ para software de produção (produto final); finalmente, o ambiente Arduino também é uma opção, onde foi criada uma interface entre o código fundamental em C, o compilador C++ utilizado pelo Arduino e o gravador de memória utilizado pelo Arduino para transferir os programas compilados para o módulo.

Recentemente foi criado também o ambiente de desenvolvimento *PlatformIO*

(PLATFORMIO, 2017) para apoiar o desenvolvimento de sistemas embarcados e que permite, neste caso, profissionalizar o ambiente Arduino utilizando a mesma base de código, mas trazendo vantagens e facilidades no processo de debug, de escrita de código da aplicação, e de produção de aplicações para diversas famílias de hardware no mesmo ambiente.

Para a finalidade, prazo, e quantidade de unidades a serem produzidas neste projeto, a opção mais adequada é o ambiente Arduino, que permite performance e confiabilidade suficientes. Esta opção também permite explorar e testar diversas possibilidades de sensores, algoritmos e verificar determinados indicadores de desempenho com agilidade.(ARDUINO, 2017)

#### 4.1.5 Sensor Ultrassonico

O sensor ultrassônico adotado foi o HC-SR04. É um sensor de baixo custo e capaz de detectar distâncias de 400 máximo centímetros e no mínimo 4 centímetros, oferecendo assim uma escala confortável para a aplicação. Este sensor é operado em grande parte pelo sistema, pois o sistema é responsável por produzir um pulso de 10 microssegundos no pino *Trigger* do sensor, que irá produzir a onda sonora. Então o sistema deve esperar uma borda de subida no pino *Echo* e verificar quanto tempo passou entre os dois eventos. Seja este tempo  $T$ . A distância total de ida e volta do sinal sonoro equivale a  $T$  multiplicado pela velocidade do som. Como queremos apenas metade disso, dividimos o resultado por dois.

Há, entretanto, cuidados que devem ser tomados ao se utilizar este sensor. Segundo o *datasheet* encontrado em anexo, não se deve produzir pulsos frequentes demais no pino *Trigger*, pois pode ocorrer sobreposição de ondas sonoras ecoantes. A recomendação é a de emitir um pulso a cada 60 milissegundos.

### 4.2 Servidor

A implementação da aplicação web exige um estudo detalhado para garantir que todos os requisitos de marketing e engenharia sejam atendidos.

Discutimos a seguir as principais decisões tomadas em relação à arquitetura e organização dos dados na aplicação do servidor.

#### 4.2.1 Arquitetura do software no servidor

A operação da rede de sensores distribuídos exige a presença de uma entidade central para armazenar o grande volume de dados produzido pelas unidades sensoras. Além disso, os requisitos de marketing exigem que os dados coletados sejam de

fácil interpretação para o usuário, permitindo a geração de relatórios e insights para contribuir com o gerenciamento do espaço monitorado.

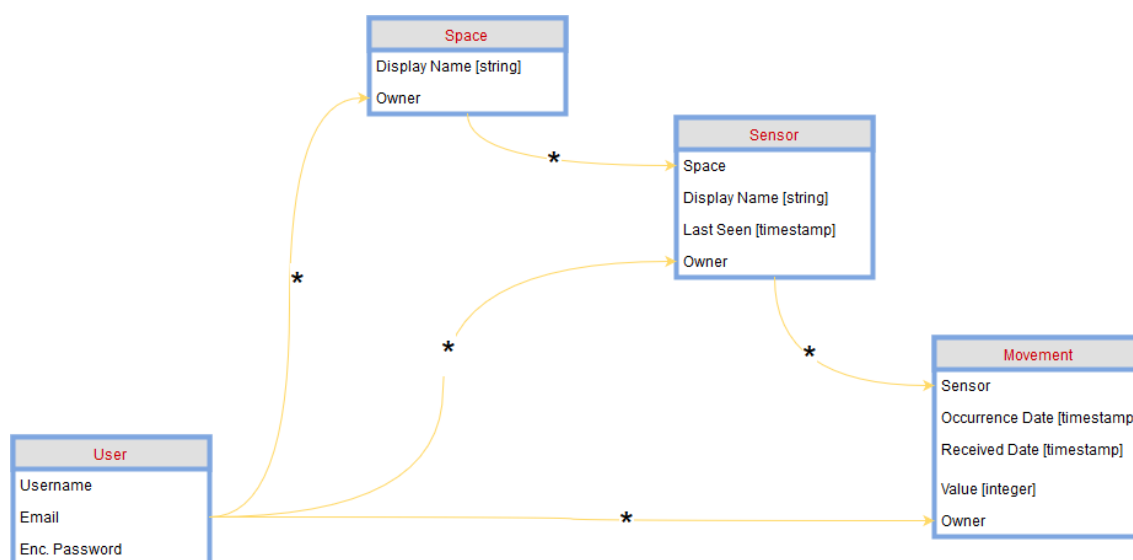
Em complemento ao projeto das unidades de detecção de entrada e saída, foi desenvolvido um serviço em nuvem para coleta e apresentação dos dados. A aplicação foi desenvolvida para funcionar de modo responsivo em navegadores web, suportando desktops e smartphones.

## 4.2.2 Modelos de dados

Para representar as estruturas reais dentro do servidor de forma clara e lógica para os operadores, desenvolvemos classes para abstração de cada uma das entidades que compõe a rede de sensores distribuídos. A Figura 23 representa a relação entre cada uma dessas entidades.

Figura 23 – Modelo de dados da aplicação no servidor web

Fonte: Autores.



A classe User abstrai os usuários do serviço, provendo controle de permissões e login nas interfaces do aplicativo web. Esta classe aparece relacionada em todas as outras através do campo “Owner”, para permitir o controle preciso de permissões de modificação e adição de entidades-filhas.

A classe Space abstrai os ambientes monitorados, podendo representar salas, vãos, corredores, banheiros ou qualquer outra partição com fluxo de pessoas capaz de receber sensores de monitoramento. Fisicamente, um espaço de monitoramento é limitado em seus acessos por unidades de sensores e essa relação se reflete dentro da arquitetura de software: um Space associa 0 ou mais Sensors.

A abstração das placas de sensoriamento fica por conta da classe Sensor, responsável por representar as entidades físicas de detecção de fluxo dentro do

servidor. Cada Sensor está associado a um único Space (note que, entretanto, um Space pode associar diversos Sensors).

Do ponto de vista do dispositivo físico e seu firmware não existe qualquer vínculo com o espaço monitorado, permitindo que os sensores sejam redistribuídos livremente. Dessa forma, uma unidade de monitoramento de fluxo precisa conhecer apenas sua própria identidade, e o servidor se encarrega de associá-lo ao espaço configurado.

A unidade de informação enviada por um sensor é abstraída pela classe Movement, que representa a detecção de um ou mais movimentos de entrada ou saída de um ambiente. Quando um sensor detecta a passagem de uma pessoa, ele registra essa informação internamente e informa ao servidor através da estrutura Movement no momento adequado. Um sensor pode informar imediatamente cada movimentação de forma individual ou acumular dados para envio em intervalos pré-definidos de tempo. Obviamente, um Sensor associa 0 ou mais Movements.

### 4.2.3 Segurança e controle de acesso

Para garantir o desenvolvimento de um sistema capaz de atender múltiplos usuários dentro da mesma infraestrutura física com segurança e privacidade de dados, todas as entidades de abstração estão associadas ao modelo User. Dessa forma, é possível fazer uma checagem rápida e eficiente das permissões de leitura, modificação, exclusão e criação de entidades-filhas que respeite o contexto de cada usuário.

Por exemplo, o registro de um Movement só é possível quando o Sensor associado pertence ao usuário autenticado que solicitou sua criação. A listagem de entidades também só é permitida ao dono, garantindo que múltiplos usuários de organizações distintas possam compartilhar a mesma infraestrutura de servidor.

### 4.2.4 Interfaces para usuários

A aplicação desenvolvida tem, pelo menos, dois tipos de clientes bastante distintos: os operadores (que incluem administradores de contas e usuários padrão) e as unidades sensoras.

Enquanto o primeiro grupo é composto por humanos que majoritariamente consomem dados armazenados no servidor, o segundo grupo é composto por dispositivos que operam alimentando a base de dados com as informações colhidas do ambiente.

Para atender à necessidade específica de cada um desses grupos a aplicação web deverá contar com interfaces capazes de receber e fornecer informações no formato adequado.

A solução para os operadores com funções administrativas deve incluir uma

interface gráfica amigável que permita a criação e manipulação das entidades, bem como suas relações.

Para os usuários padrão, responsáveis pelo monitoramento dos espaços e uso dos gráficos gerados, existe a demanda por uma interface intuitiva e direta e, ao mesmo tempo, segura em termos de alteração acidental ou não autorizada das configurações feitas pelos administradores de conta.

Finalmente, os sensores necessitam de uma interface simples para enviar as informações colhidas.

De forma a atender aos requisitos dos três grupos delineados, dividimos a aplicação em três interfaces distintas: o dashboard, para os usuários padrão; o painel de administração, para os administradores de conta; e a API, para interação com os sensores.

#### 4.2.5 Pilha de software

O desenvolvimento de aplicações web modernas se dá, em sua grande maioria, através de frameworks que entregam diversas facilidades para o programador.

O mercado conta hoje com inúmeros frameworks baseados nas linguagens de programação mais populares. Por afinidade com a linguagem Python, escolhemos o framework Django. Além disso, o Django conta com a facilidade de adicionar e gerenciar módulos através do gerenciador de pacotes nativo do Python.

O padrão de desenvolvimento utilizado no Django é conhecido como MVC (Model-View-Controller). Esse padrão, prepresentado na Figura 24 (WIKIPEDIA, 2017) permite que os modelos de dados desenhados anteriormente sejam mapeados diretamente para classes da aplicação, facilitando a sua implementação. Este padrão tem sido adotado em projetos profissionais por permitir o desenvolvimento contínuo com incremento da complexidade da aplicação sem perder a organização do código.

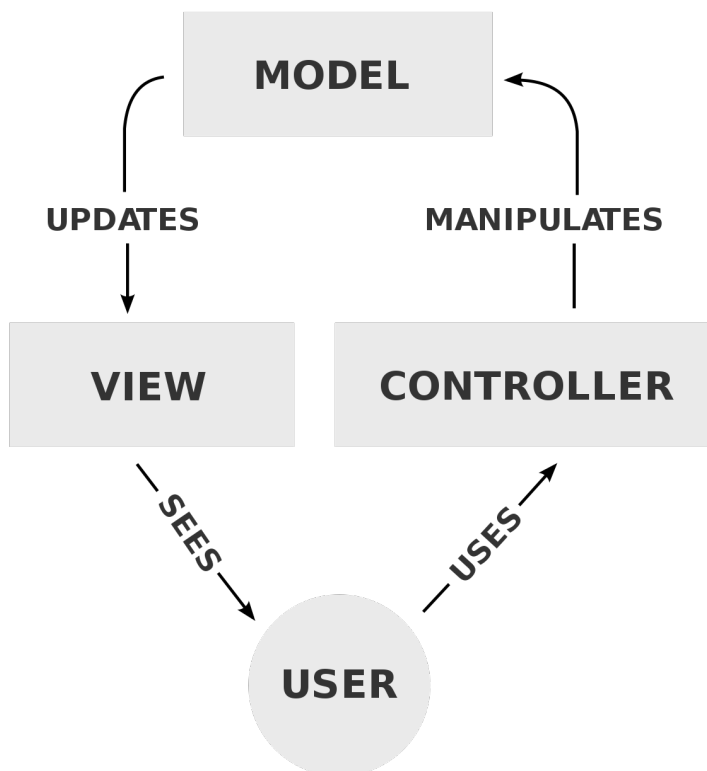
Outro componente importante da aplicação é o banco de dados. Em uma aplicação escalável espera-se que o armazenamento de dados seja confiável para manter a integridade das informações. Existem hoje diversas soluções para bancos de dados relacionais no mercado, com características próprias de desempenho, escalabilidade e suporte.

Optou-se pela utilização do PostgreSQL, um software open source com compatibilidade total à linguagem SQL.

Uma vez definidas as camadas básicas da aplicação, foram definidas as ferramentas utilizadas na construção das interfaces definidas no capítulo anterior. Essa definição levou em conta a capacidade de adição de módulos prontos ao Django e está

Figura 24 – Diagrama da arquitetura MVC

Fonte: (WIKIPEDIA, 2017)



resumida na Figura 25.

Para a API e o painel de administração, o Django conta com módulos open source distribuídos pela comunidade, capazes de entregar interfaces gráficas prontas e diversas ferramentas de controle de acesso e manipulação das estruturas de dados.

Apenas o *dashboard* precisou de uma implementação completa, baseada na tradicional pilha web: HTML, CSS e JavaScript.

#### 4.2.6 Infraestrutura do servidor

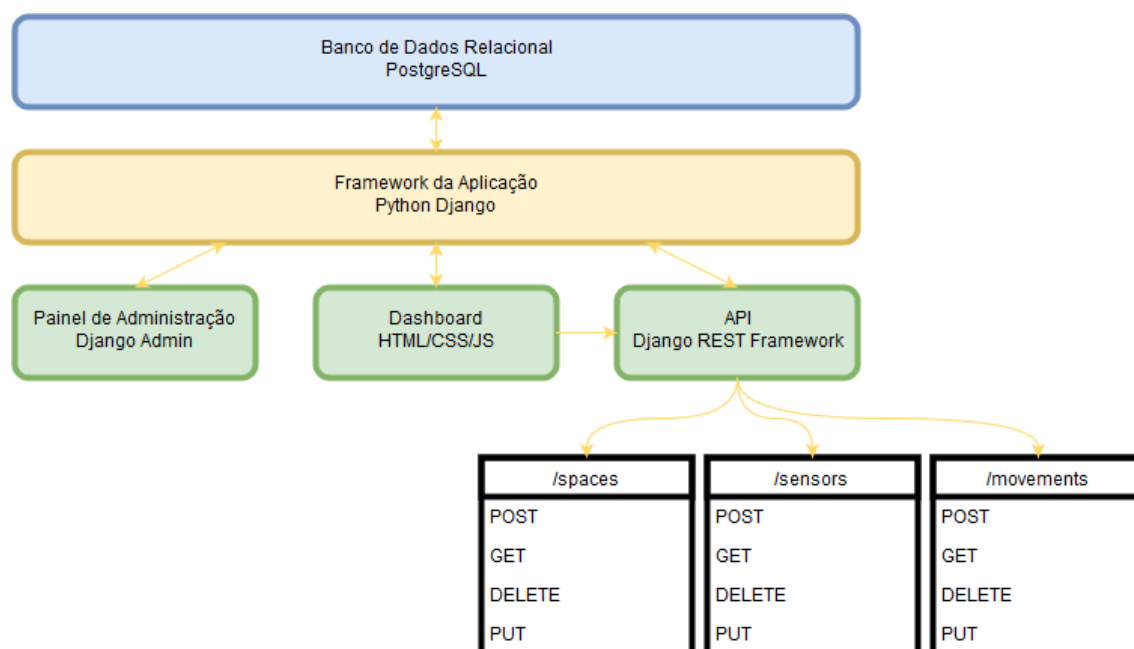
A escolha da infraestrutura física para o servidor é tão importante quanto a arquitetura de software. Espera-se que a infraestrutura para esta aplicação seja escalável e ao mesmo tempo econômica para a fase de testes.

Muitas soluções surgiram recentemente no mercado, permitindo a prototipagem rápida de aplicações web com baixo ou nenhum custo inicial. Dentre elas, destaca-se o Heroku, provedor de infraestrutura em nuvem para aplicativos web. O diferencial do Heroku está na configuração automática e na integração com serviços de versionamento, como o GitHub.

Para aplicações pequenas, como o teste realizado nesse projeto, o Heroku

Figura 25 – Camadas da aplicação e tecnologias empregadas no desenvolvimento

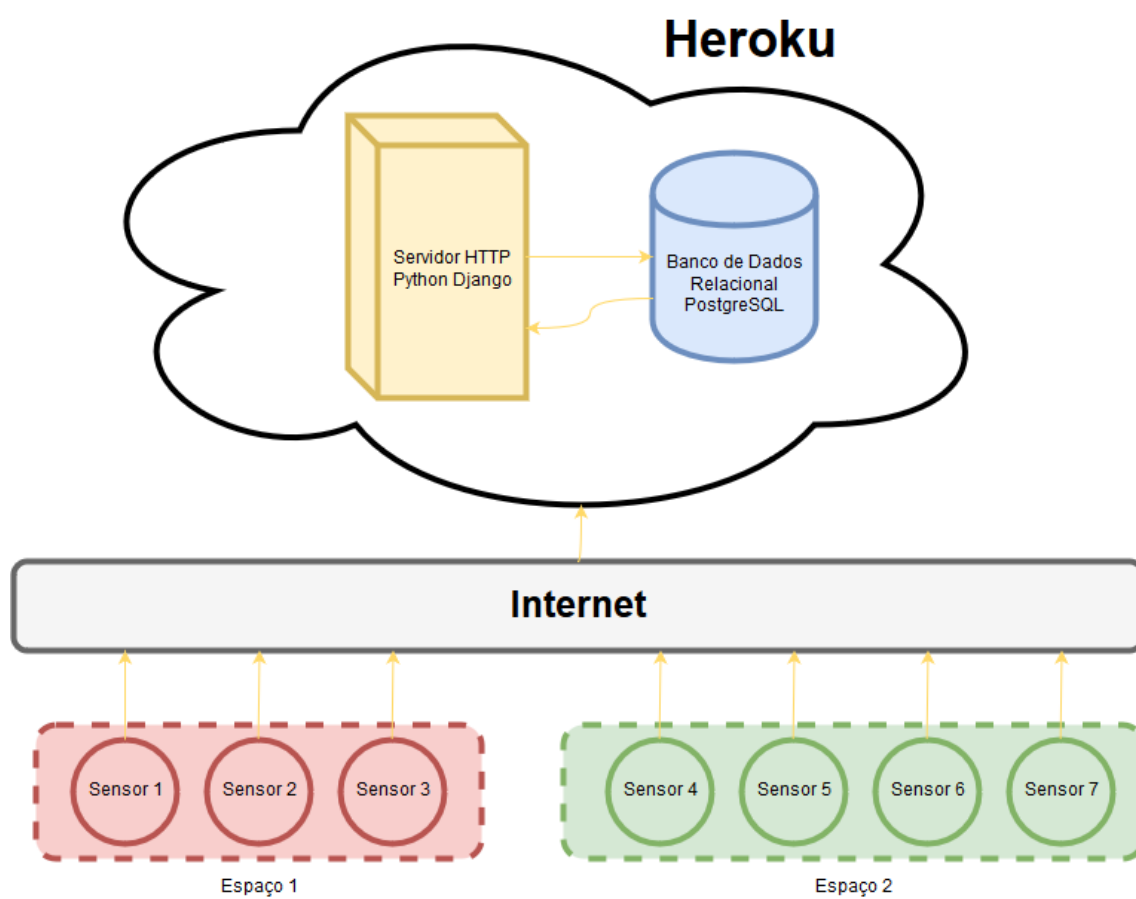
Fonte: Autores



fornece servidores, banco de dados e domínios de forma gratuita e permite escalar o armazenamento e a capacidade de processamento com apenas um clique. A Figura 26 representa a estrutura alocada para a aplicação.

Figura 26 – Diagrama da infraestrutura da aplicação

Fonte: Autores





## 5 Resultados Finais

Nesta seção serão apresentados os resultados do desenvolvimento e testes do projeto, evidenciando quais requisitos de marketing foram atendidos até o presente momento e explicando como o produto consegue atender tais exigências através da engenharia do produto.

Os requisitos de marketing a serem abordados são:

- Facilidade de instalação
- Interface de usuário intuitiva
- Confiabilidade dos dados
- Privacidade do usuário

### 5.1 Facilidade de Instalação

Apesar de o *hardware* concebido até o presente momento não ser otimizado para produção industrial e ter sido confeccionado manualmente, ele apresenta dimensões aceitáveis para instalação de teste, com uma placa de circuito impresso de 6 cm de largura por 20 cm de comprimento, conforme as figuras 27 e 28.

Ainda considerando a fase de experimentação e testes em que o projeto se encontra, foi utilizada como principal fonte de energia para cada sistema um conjunto de baterias de íon-lítio de 5 Volts, utilizado popularmente para recarregar *smartphones*. Tal solução é bastante conveniente para o protótipo, e capaz de manter o sistema em operação por aproximadamente 50 horas considerando a capacidade de 2600 mAh da bateria utilizada e um consumo médio de 50 mA do sistema.

A utilização de um circuito de alimentação utilizando a rede elétrica implicaria um sacrifício desnecessário para a etapa de desenvolvimento em que o projeto se encontra. Ainda assim, é possível utilizar adaptadores AC-DC bastante acessíveis para substituir a bateria e alimentar o circuito através da rede elétrica, caso tal cenário seja almejado.

Figura 27 – Layout da placa de circuito impresso do sistema embarcado

Fonte: Autores

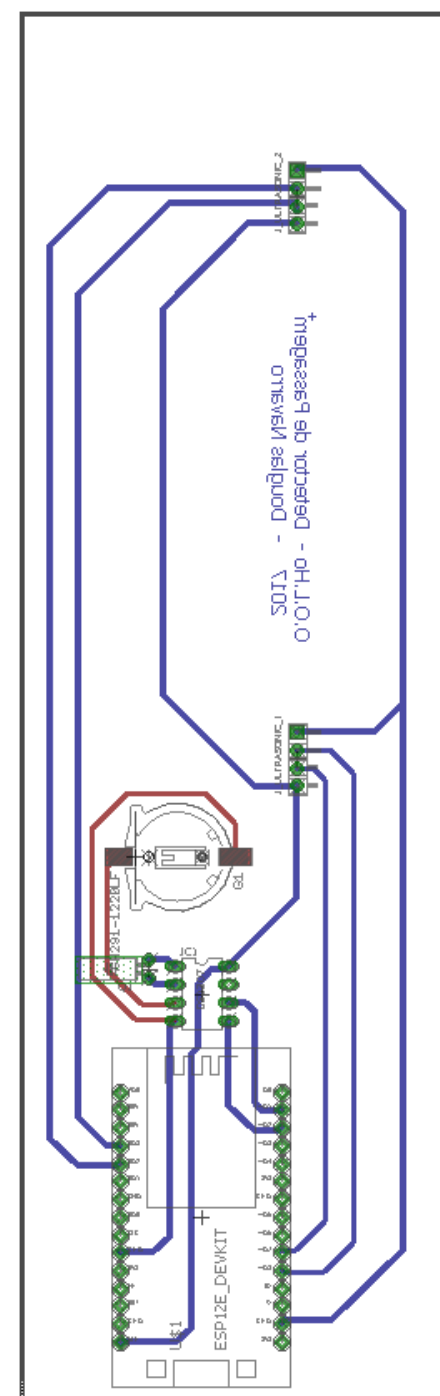
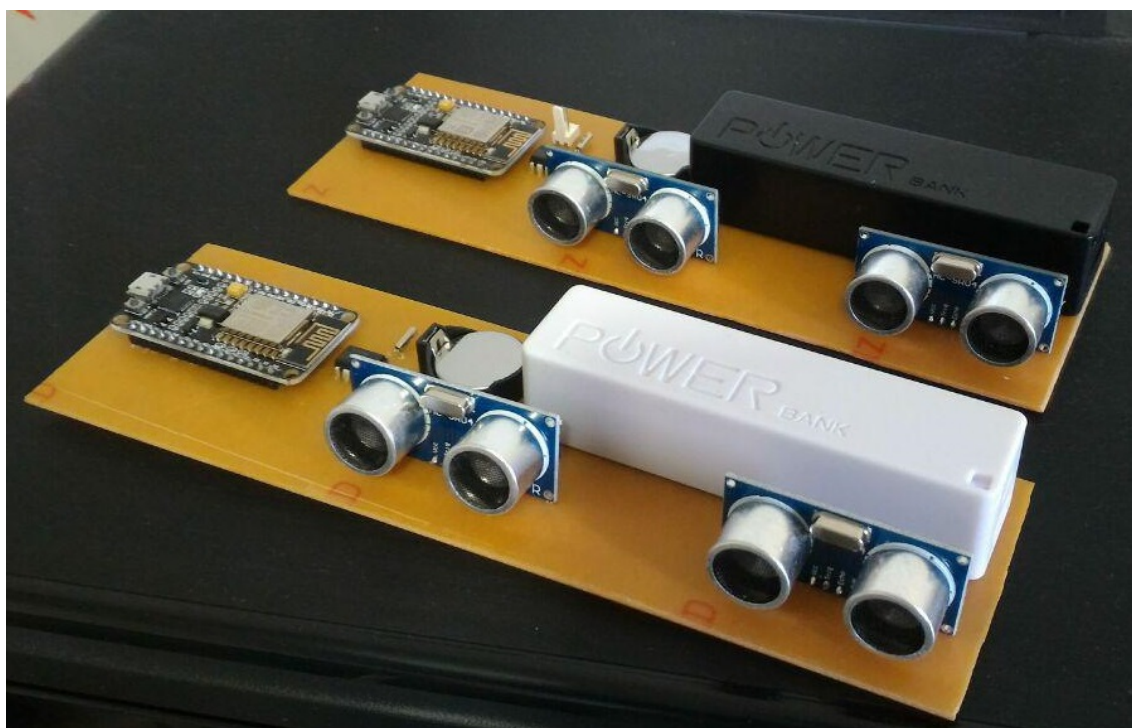


Figura 28 – Fotografia de dois protótipos confeccionados

Fonte: Autores



## 5.2 Interfaces de Usuário Intuitiva

A implementação da aplicação web foram satisfatórios, entregando todas as interfaces previamente definidas nas seções anteriores.

### 5.2.1 API

A API da aplicação foi implementada através do Django REST Framework, um módulo poderoso capaz de entregar interfaces RESTful com pouco esforço por parte do programador a partir dos modelos de dados já definidos.

A API conta com serializadores, que convertem a estrutura de dados trabalhada em um objeto do tipo JSON. A função do serializador é gerar uma representação reconstituível do objeto modelado, permitindo a criação, leitura e modificação por parte dos clientes da API.

O acesso aos serializadores é feito através dos controladores da aplicação, responsáveis pela lógica de negócios de cada um dos modelos de dados. Os controladores da aplicação também contém as restrições de acesso e checagem de permissões de usuário.

Em nosso projeto, optamos por expor através da API três modelos de dados: Space, Sensor e Movement. Seguindo os padrões de desenvolvimento RESTful, os três

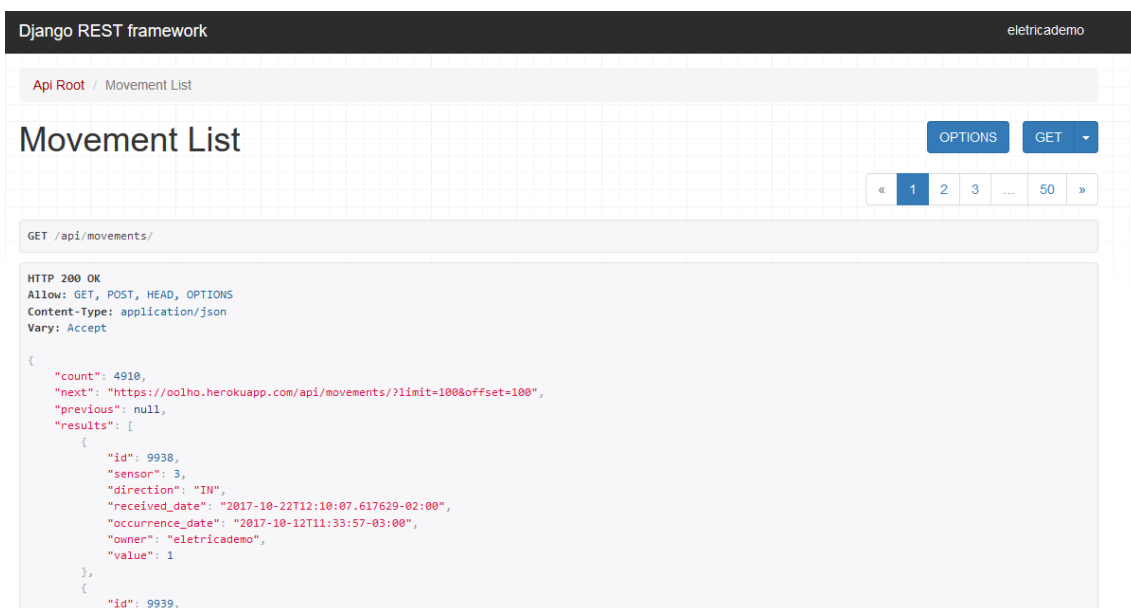
endpoints permitem a criação, modificação, remoção e leitura (POST, PUT, DELETE e GET) dos objetos.

O objetivo principal da API é oferecer uma interface adequada para as unidades sensoras, e, para isso, a exposição do modelo Movement seria suficiente. Entretanto, optamos pela exposição adicional dos modelos Space e Sensor para facilitar o desenvolvimento de aplicações nativas em outras plataformas (Android, iOS e aplicações desktop).

Além disso, o Django REST Framework provê interfaces gráficas opcionais para a API, permitindo a comutação entre uma visão navegável por browser convencionais e a versão reduzida processável por clientes limitados como a nossa unidade sensora como representado na Figura 29.

Figura 29 – Interface navegável da API no endpoint para o modelo Movement

Fonte: Autores



## 5.2.2 Painel de Administração

O painel de administração foi criado para permitir o controle administrativo das contas, usuários e associação de relação entre os objetos.

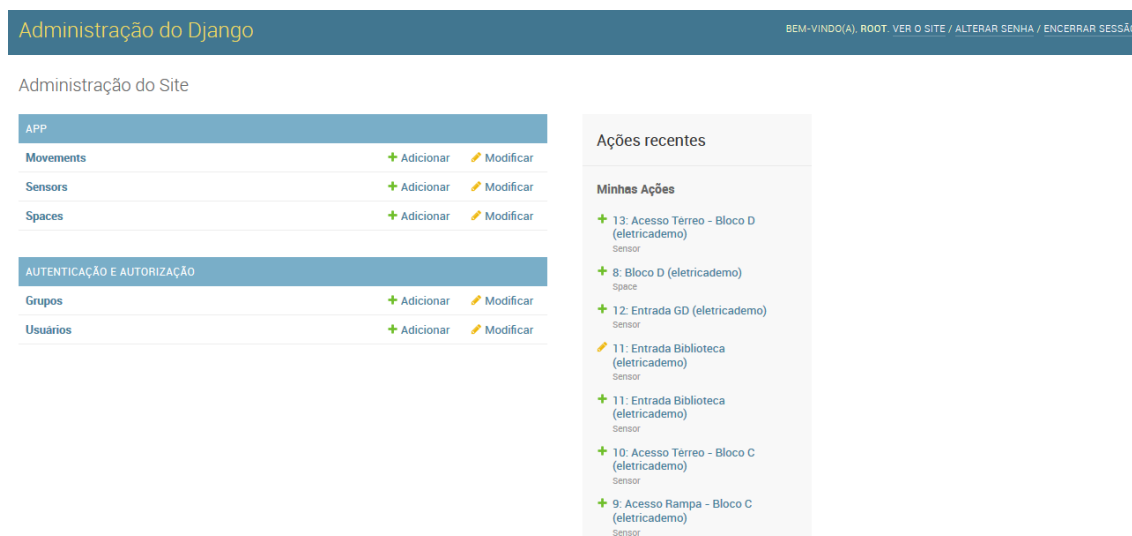
O Django já fornece um módulo padrão com a instalação para controle administrativo dos modelos com interface gráfica representada na Figura 30.

## 5.2.3 Dashboard

O dashboard foi desenvolvido para permitir a visualização dos dados coletados pelos sensores, de forma intuitiva para o operador final.

Figura 30 – Interface navegável da API no endpoint para o modelo Movement

Fonte: Autores



A visão principal do dashboard é baseada nos espaços, e dentro da visão de cada espaço o dashboard fornece widgets com informações importantes para o operador do sistema.

O primeiro widget, representado na Figura 31, apresentado é o resumo de pessoas dentro do espaço, permitindo uma visão rápida da situação atual.

Atendendo ao requisito de fornecer relatórios e insights baseados no histórico de fluxo ao espaço, o dashboard conta com gráficos agregados e filtráveis através de controles. Esses gráficos, representados nas Figuras 32 e 33 são obtidos através de um endpoint adicional na API, permitindo a geração dos gráficos em outras aplicações além do aplicativo web.

Figura 31 – Widget de contagem de pessoas no dashboard

Fonte: Autores

Bloco A

Agora,

**259**

pessoas no espaço

Por fim, para fins de diagnóstico da rede, o dashboard fornece o status dos sensores associados ao espaço exibido, mostrando a data da última interação como na Figura 34.

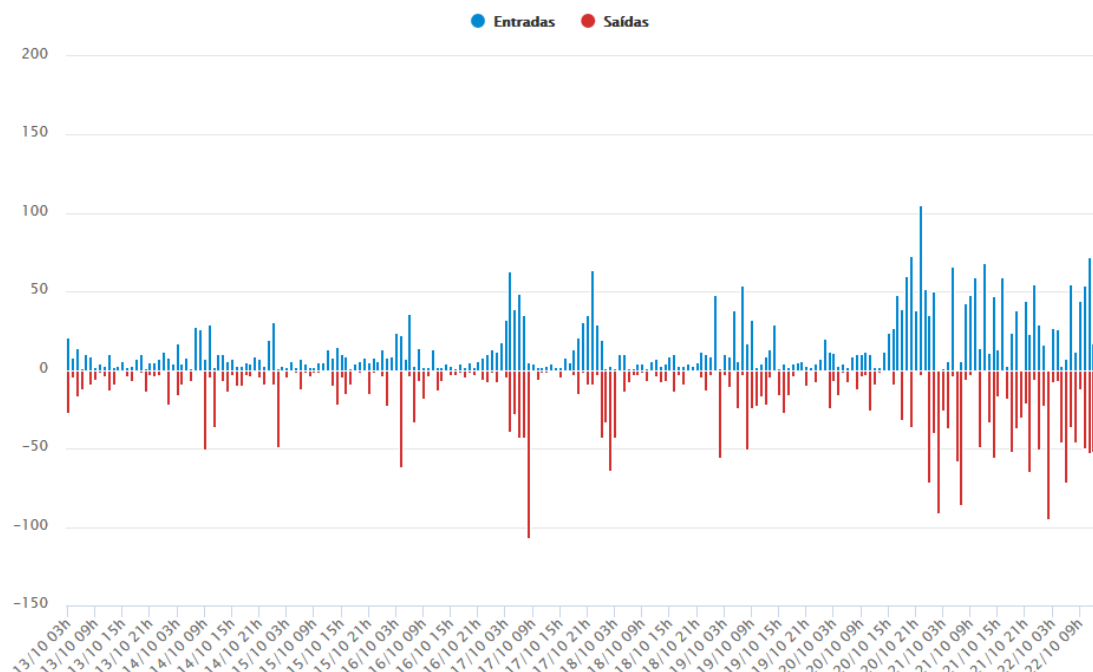
Figura 32 – Widget de gráfico do histórico de pessoas no dashboard

Fonte: Autores



Figura 33 – Widget de gráfico de fluxo de pessoas no dashboard

Fonte: Autores



### 5.3 Confiabilidade dos dados

A confiabilidade dos dados é composta por dois fatores: a acurácia das detecções de passagem e a acurácia do registro de horário. A acurácia do registro de horário é garantida pelo desvio de até  $\pm 20\text{ppm}$  da frequência de oscilação do cristal externo ao relógio DS1307, permitindo atrasos de aproximadamente 10 minutos por ano, a uma temperatura típica de ambientes internos. (MAXIM-INTEGRATED, 2005)

Figura 34 – Widget de estado dos sensores no dashboard

Fonte: Autores

### Sensores

ID	Nome	Visto Por Último	Espaço
4	Acesso Térreo - Bloco A	4 de Novembro de 2017 às 10:33	Bloco A
5	Acesso Rampa - Bloco A	16 de Outubro de 2017 às 00:54	Bloco A

A acurácia das detecções de passagem varia bastante dependendo da distância que há entre o par de sensores ultrassônicos e a frequência de disparos de sinais sonoros pelos mesmos, bem como o ambiente em que o sistema é instalado e a janela de tempo de detecção considerada. Em testes realizados com anteparos simples (como uma régua escolar) que se deslocam a menos de 10 centímetros do par de sensores a uma velocidade compatível com a janela de tempo considerada, a acurácia do sistema pode ser de até 100% considerando até uma centena de passagens.

Entretanto, utilizando pessoas de fato cruzando a linha de visada dos sensores os resultados de acurácia diminuem para até 50%, dependendo das configurações de janela de detecção de tempo e distância entre os sensores ultrassônicos. Esta acurácia pode, contudo, ser elevada realizando ajustes ao código que é executado pelo sistema, bem como ajustando a posição dos sensores ultrassônicos de um mesmo sistema.

## 5.4 Privacidade do usuário

O sistema é incapaz de produzir identificação positiva de pessoas, dada a simplicidade do funcionamento dos sensores. Não havendo captura de imagens, registro biométrico ou leitura eletrônica de dispositivos de identificação pessoal, não é possível identificar as pessoas que cruzam a visada dos sensores.





## 6 Análise

A validação obtida com o protótipo foi satisfatória para demonstrar que o projeto, na forma que foi concebido, é tecnicamente viável.

A grande variação de acurácia na detecção de passagens verificada nos testes revela que muitos há fenômenos desconhecidos e relacionados à utilização do par de sensores ultrassônicos. Foi possível notar que a distância entre os sensores é crucial para que a onda ultrassônica produzida por um não seja capturada pelo outro. Se eles forem posicionados muito próximos, certamente um anteparo irá produzir uma reflexão da onda sonora que pode ser capturada pelo sensor ultrassônico que não emitiu a onda sonora. Isso leva a falsos positivos e passagens sendo detectadas no sentido errado.

Entretanto, é importante que o sistema não tenha uma dimensão muito elevada, sendo adotada a distância de 8 centímetros verificada experimentalmente como sendo um bom compromisso entre acurácia e dimensão do protótipo.

Com relação à aplicação web, obtivemos uma interface amigável que gera dados relevantes para o usuário. A aplicação foi desenhada para se adaptar à diferentes formatos de tela, possibilitando o controle e visualização de dados em desktops, notebooks, tablets e celulares. O dashboard permite um controle intuitivo do período visualizado para avaliação de séries históricas, mas permite também uma visualização instantânea com os dados atuais.

A interface de API permite expandir a aplicação para programas nativos em smartphones e outras plataformas, além de abstrair o acesso para qualquer versão do sensor.

A aplicação em nuvem é escalável e conta com facilidades previstas pelo serviço de infraestrutura, como balanceamento de carga para pico de acessos.

Apesar da implementação atual ser tecnicamente funcional, restam validações de mercado e de usabilidade com o usuário final para avaliar o cumprimento dos requisitos de marketing estabelecidos no início do projeto.

Dentre as hipóteses de usabilidade a serem avaliadas, a relevância dos widgets já implementados é um ponto crítico na experiência do usuário. Existe a necessidade de conduzir testes e entrevistas para investigar a demanda pela implementação de outros widgets no dashboard.



## 7 Conclusão

O trabalho exercitou o desenvolvimento do produto desde a determinação das necessidades do cliente até fase de prototipagem funcional, englobando conhecimentos de eletrônica, programação, desenvolvimento de sistemas embarcados e engenharia em geral.

O produto concebido é capaz de coletar informações sobre o fluxo de pessoas dentro de espaços delimitados sem interferência física na movimentação dos indivíduos e sem identificação positiva, preservando a privacidade dos usuários do ambiente.

A acurácia do sistema embarcado não é a ideal e depende bastante de calibração do código que é executado no sistema bem como as condições em que a instalação é feita. Entretanto, o sistema opera de forma robusta, sem necessidade de interferência do cliente após configuração inicial.

O gerador de relatórios apresenta de maneira visual e intuitiva os dados coletados, permitindo identificar os horários e os ambientes de maior movimento através de um filtro da série histórica armazenada.

Em suma, os requisitos de marketing foram em grande parte atingidos, não havendo necessidade de se retornar às fases mais fundamentais de desenvolvimento, mas apenas de incrementar, aprimorar e escalar a solução proposta a partir de iterações de *feedback* do cliente.

É importante ressaltar o potencial e a escalabilidade do projeto que, estando bastante inserido em um contexto de Internet das Coisas, pode produzir resultados poderosos a partir de unidades simples de tecnologia constantemente conectadas. Um sistema embarcado com alto nível de aprimoramento pode ser produzido em grandes quantidades para tornar acessível o custo do produto. Por outro lado, um serviço web moldado com funcionalidades adaptadas para cada cliente pode permitir que o sistema seja utilizado em qualquer tipo de local: universidades, hospitais, lojas, *etc.*

Os de grandes volumes de dados, cuja produção pode ser viabilizada por este projeto, pode ser tratada estatisticamente para criar informação de qualidade e que pode ser de valor para qualquer entidade: governo, empresa e até cidadão. Finalmente, o trabalho desenvolvido pode ser considerado um recurso de grande valor para a criação espaços mais inteligentes em diversas escalas.



## 8 Trabalhos futuros

O sistema embarcado pode ser levado para um estágio de desenvolvimento que considere de forma mais rigorosa a produção dos sinais dos sensores ultrassônicos. Uma solução melhor é a de haver um microcontrolador ou circuito eletrônico mais simples e dedicado apenas a produzir de forma eficiente sinais de interrupção para o ESP8266 quando houvesse detecção de presença em cada sensor. O mecanismo de amostragem constante, ou *polling*, fica bastante sujeito a falhas e a abordagem de se assumir uma janela de tempo entre as detecções exige calibragem do sistema.

As dimensões físicas também podem ser bastante reduzidas, considerando que os protótipos foram confeccionados utilizando procedimentos manuais e bastante rudimentares. Um processo de fabricação industrial permitiria uma solução embarcada mais compacta.

Uma ideia interessante para a plataforma de visualização dos dados é a produção de mapas de calor da ocupação do espaço. O cliente poderia enviar para o sistema uma imagem da planta do espaço e associar à imagem os diversos sensores do sistema. Este, então, produziria diversas versões desta imagem para o cliente, em tempo real, de quais espaços estão sendo mais intensamente utilizados através da mudança de cor de determinados espaços na planta.

É importante ressaltar que a solução como um todo carece de validações de mercado e de experiência do usuário, através de testes e entrevistas extensivas com clientes para validação da solução apresentada e formulação de um modelo de negócios.



# Referências

ARDUINO, R. oficial do projeto esp8266. Repositorio oficial do projeto esp8266-arduino. In: . [s.n.], 2017. Disponível em: <<https://github.com/esp8266/Arduino>>. Acesso em: 25 sept. 2017.

DUTTA, J.; ROY, S. Iot-fog-cloud based architecture for smart city: Prototype of a smart building. In: *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*. [S.l.: s.n.], 2017. p. 237–242.

ESPRESSIF. Endereço Eletrônico da Espressif. In: . [S.l.: s.n.], 2017. [Online; acessado em 06 de Outubro de 2017].

GIVARDIS, F. V. e T. In: *Embedded System Design: A Unified Hardware/Software Introduction*. [S.l.: s.n.], 2002. p. 1.

HUA, Q. et al. Occupancy detection in smart buildings using support vector regression method. In: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. [S.l.: s.n.], 2016. v. 02, p. 77–80.

JIN, M. et al. Occupancy detection via environmental sensing. *IEEE Transactions on Automation Science and Engineering*, PP, n. 99, p. 1–13, 2017. ISSN 1545-5955.

LIU, Z.; ZHANG, J.; GENG, L. An intelligent building occupancy detection system based on sparse auto-encoder. In: *2017 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. [S.l.: s.n.], 2017. p. 17–22.

LUPPE, C.; SHABANI, A. Towards reliable intelligent occupancy detection for smart building applications. In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. [S.l.: s.n.], 2017. p. 1–4.

MAXIM-INTEGRATED. Timekeeping accuracy, automatic and affordable. In: . [s.n.], 2005. Disponível em: <<https://www.maximintegrated.com/en/app-notes/index.mvp/id/3566>>. Acesso em: 12 oct. 2017.

MICROPYTHON. Homepage do micropython.org. In: . [s.n.], 2017. Disponível em: <<https://micropython.org>>. Acesso em: 25 sept. 2017.

MINOLI, D.; SOHRABY, K.; OCCHIOGROSSO, B. Iot considerations, requirements, and architectures for smart buildings x2014;energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, v. 4, n. 1, p. 269–283, Feb 2017. ISSN 2327-4662.

MONGOOSE-OS. Homepage do mongoose-os. In: . [s.n.], 2017. Disponível em: <<https://mongoose-os.com>>. Acesso em: 25 sept. 2017.

PLATFORMIO. Homepage do platformio. In: . [s.n.], 2017. Disponível em: <<http://platformio.org>>. Acesso em: 25 sept. 2017.

REYNA, E. A. de et al. Enhanced indoor localization through crowd sensing. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2017. p. 2487–2491.

WIKIPEDIA. *Model–view–controller* — *Wikipedia, The Free Encyclopedia*. 2017. [Online; accessed 10-December-2017]. Disponível em: <<https://en.wikipedia.org/w/index.php?title=Model%E2%80%93view%E2%80%93controller&oldid=813672337>>.





# ANEXO A – Datasheet do sensor HC-SR04



Tech Support: [services@elecfreaks.com](mailto:services@elecfreaks.com)

## Ultrasonic Ranging Module HC - SR04

### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

### Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

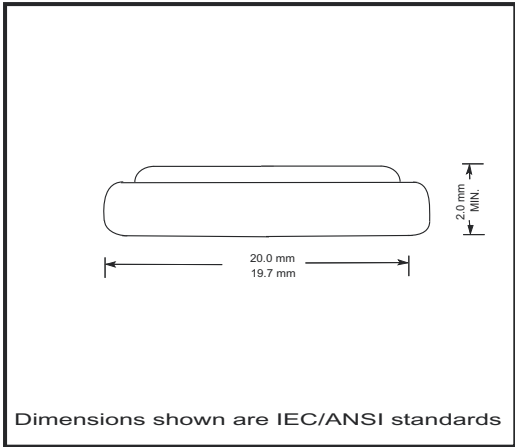


# ANEXO B – Datasheet da Duracell CR2032

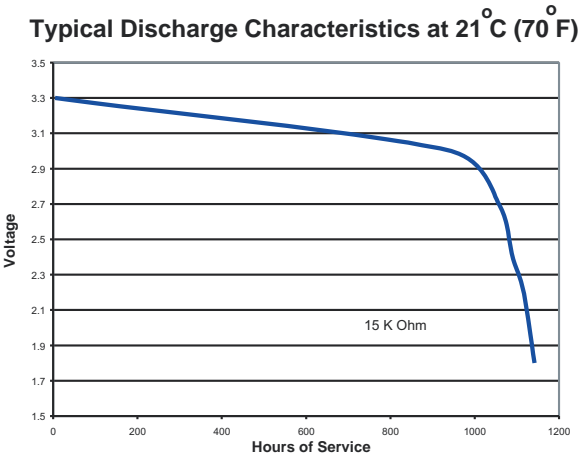
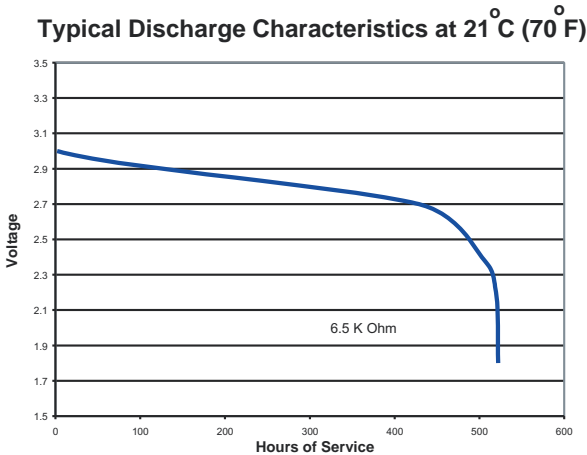
**DURACELL®**  
Lithium-Manganese Dioxide Battery

**2032**  
Size: CR2032

Zn/MnO<sub>2</sub>



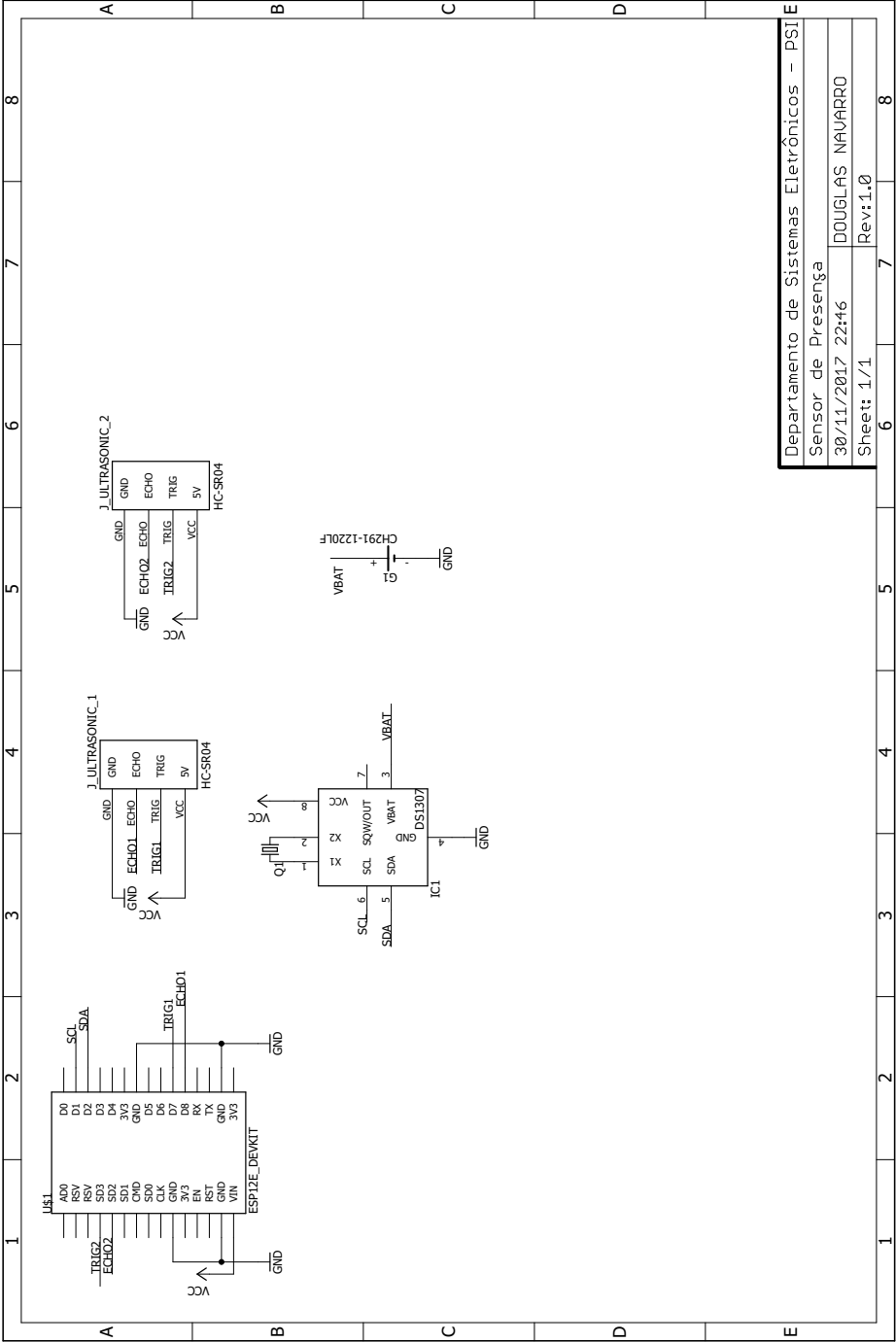
Nominal Voltage:	3 V
Average Weight:	3.14 gm (0.11 oz.)
Volume:	0.63 cm <sup>3</sup> (0.04 in. <sup>3</sup> )
Terminals:	Flat
Operating Temperature Range:	-20°C to 54°C (-4°F to 130°F)
ANSI:	5004LC
IEC:	CR 2032



**DURACELL®**  
BATTERIES  
Berkshire Corporate Park  
Bethel, CT 06801 U.S.A.  
Telephone: Toll-free 1-800-544-5454  
Internet: [www.duracell.com](http://www.duracell.com)

\* Delivered capacity is dependent on the applied load, operating temperature and cut-off voltage. Please refer to the charts and discharge data shown for examples of the energy / service life that the battery will provide.

# APÊNDICE A – Esquemático do sistema embarcado



Departamento de Sistemas Eletrônicos – PSI	E
Sensor de Presença	
30/11/2017 22:46	
DOUGLAS NAVARRO	
Sheet: 1/1	
Rev:1.0	
6	7
8	