

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
CURSO DE ESPECIALIZAÇÃO EM INTERNET DAS COISAS

LADISLAU ESTEVÃO KROKOSZ

**LOCALIZAÇÃO UTILIZANDO INFORMAÇÕES DE  
ROTEAMENTO EM REDES DE SENSORES SEM FIO**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2018

LADISLAU ESTEVÃO KROKOSZ

**LOCALIZAÇÃO UTILIZANDO INFORMAÇÕES DE  
ROTEAMENTO EM REDES DE SENSORES SEM FIO**

Monografia de Especialização, apresentada ao Curso de Especialização em Internet das Coisas, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Ohara Kerusauskas  
Rayel

CURITIBA  
2018



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Câmpus Curitiba

Diretoria de Pesquisa e Pós-Graduação  
Departamento Acadêmico de Eletrônica  
Curso de Especialização em Internet das Coisas



---

## **TERMO DE APROVAÇÃO**

### **LOCALIZAÇÃO UTILIZANDO INFORMAÇÕES DE ROTEAMENTO EM REDES DE SENSORES SEM FIO**

por

**LADISLAU ESTEVÃO KROKOSZ**

Esta monografia foi apresentada em 30 de Novembro de 2018 como requisito parcial para a obtenção do título de Especialista em Internet das Coisas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Ohara Kerusauskas Rayel  
Orientador

---

Prof. M.Sc. Danillo Leal Belmonte  
Membro titular

---

Prof. M.Sc. Omero Francisco Bertol  
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico esse trabalho à minha esposa Mara, que sempre me incentiva a dar continuidade aos estudos e me apoia nos desafios do conhecimento.

## AGRADECIMENTOS

Agradeço à minha mãe e aos meus irmãos, que fizeram valer os anos de estudo e os esforços despendidos na minha formação.

Aos professores do curso, os quais proporcionaram uma ampla visão das novas tecnologias introduzidas no mercado.

Ao meu orientador, Prof. Dr. Ohara Kerusauskas Rayel, pela paciência e pelo tempo dedicado às explicações e ajustes necessários para a conclusão do trabalho.

À CELEPAR, empresa na qual trabalho com muito orgulho, que tanto me apoiou na realização desse projeto.

Fico também bastante agradecido a todas as pessoas que de alguma forma me apoiaram ao longo dos meses dedicados a este curso de especialização.

A tecnologia está mudando rapidamente a maneira como interagimos com o mundo à nossa volta (MAGRANI, 2018).

## RESUMO

KROKOSZ, Ladislau Estevão. **Localização utilizando informações de roteamento em redes de sensores sem fio**. 2018. 45 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Esse trabalho aborda o uso de sensores para o rastreamento de pessoas ou objetos inseridos em um determinado ambiente. No âmbito da Internet das Coisas (IoT, do inglês *Internet of Things*), através do uso de um aplicativo instalado no aparelho celular, por exemplo, um usuário pode saber a localização daquilo que deseja monitorar, proporcionando maior controle e segurança para empresas, instituições, e pessoas. Neste trabalho, a localização aproximada é obtida através de informações de roteamento em redes de sensores sem fio que utilizam protocolos de Internet das Coisas.

**Palavras-chave:** Internet das coisas. Sensores. MQTT. 6LoWPAN. Monitoramento.

## ABSTRACT

KROKOSZ, Ladislau Estevão. **Localization through wireless sensor networks routing information.** 2018. 45 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

This work addresses the use of sensors in order to track people or objects that are present in a given environment. Within the internet of things, through an application installed on a mobile device, an user can determine the location of something that he wants to monitor, providing greater the control and security to companies, institutions, and people. In this work, the approximate location is obtained through routing information from wireless sensor networks that utilize Internet of Things protocols.

**Keywords:** Internet of things. Sensors. MQTT. 6LoWPAN. Monitoring.



## LISTA DE FIGURAS

Figura 1	–	Planta fictícia de uma escola	14
Figura 2	–	<i>Gartner Hype Cycle 2018</i>	17
Figura 3	–	Topologia básica de uma rede 6LoWPAN	20
Figura 4	–	Interconexão de redes através do uso do <i>Border Router</i> 6LBR	21
Figura 5	–	Tela do 6LBR para visualização de sensores	21
Figura 6	–	Tela de visualização da árvore de nós do 6LBR	22
Figura 7	–	<i>Broker</i>	23
Figura 8	–	Microcontrolador CC2650	24
Figura 9	–	Diagrama dos microcontroladores CC26xx	25
Figura 10	–	CC2650 <i>LaunchPad</i>	26
Figura 11	–	CC2650 <i>SensorTag</i>	26
Figura 12	–	<i>Raspberry Pi</i>	27
Figura 13	–	Quantidade de módulos dos principais repositórios	28
Figura 14	–	Diagrama do Rastreador das Coisas	29
Figura 15	–	Diagrama Entidade-Relacionamento	30
Figura 16	–	<i>Trigger</i> para inserção de sensores	31
Figura 17	–	Diagrama de coleta e registro de informações	32
Figura 18	–	Diagrama de requisições e respostas via REST API	32
Figura 19	–	Resposta do serviço de listagem de sensores no formato JSON	33
Figura 20	–	Utilização do sistema operacional <i>Android</i> no Brasil em 2012	34
Figura 21	–	Utilização do sistema operacional <i>Android</i> no Brasil em 2018	34
Figura 22	–	Diagrama das telas do aplicativo	35
Figura 23	–	Tela inicial do aplicativo	36
Figura 24	–	Tela do aplicativo para cadastro do servidor	37
Figura 25	–	Tela do aplicativo para visualização de locais	38
Figura 26	–	Tela do aplicativo para edição de local	38
Figura 27	–	Tela do aplicativo para visualização de sensores móveis	39
Figura 28	–	Tela do aplicativo para edição de sensor móvel	40
Figura 29	–	Tela do aplicativo para visualização de relatório	41
Figura 30	–	Tela do aplicativo para monitoração de local	42

## LISTA DE QUADROS

Quadro 1 – Cenários para Internet das Coisas .....	16
Quadro 2 – URIs para acesso aos <i>web services</i> .....	33

## LISTA DE SIGLAS

6LBR	Roteador de borda 6LoWPAN, do inglês <i>6LoWPAN Border Router</i>
6LoWPAN	IPv6 sobre redes de área pessoal com baixo consumo, do inglês <i>IPv6 over Low power Wireless Personal Area Networks</i>
API	Interface de Programação de Aplicativos, do inglês <i>Application Programming Interface</i>
CELEPAR	Companhia de Tecnologia da Informação e Comunicação do Paraná
CoAP	Protocolo para Aplicações Restritas, do inglês <i>Constrained Application Protocol</i>
CSS	Folhas de Estilo Cascadeadas, do inglês <i>Cascading Style Sheets</i>
DER	Diagrama Entidade-Relacionamento
DODAG	Grafo Acíclico Direcionado Orientado ao Destino, do inglês <i>Destination Oriented Directed Acyclic Graph</i>
GPS	Sistema de Posicionamento Global, do inglês <i>Global Positioning System</i>
HTML	Linguagem de Marcação de Hipertexto, do inglês <i>Hypertext Markup Language</i>
HTTP	Protocolo de Transferência de Hipertexto, do inglês <i>Hypertext Transfer Protocol</i>
iOS	Sistema Operacional para iPhone, do inglês <i>iPhone Operating System</i> ou <i>iPhone OS</i>
IoT	Internet das Coisas, do inglês <i>Internet of Things</i>
IP	Protocolo de Internet, do inglês <i>Internet Protocol</i>
IPv4	Protocolo de Internet versão 4, do inglês <i>Internet Protocol version 4</i>
IPv6	Protocolo de Internet versão 6, do inglês <i>Internet Protocol version 6</i>
JSON	Notação de Objeto de JavaScript, do inglês <i>JavaScript Object Notation</i>
MAC	Controle de Acesso de Mídia, do inglês <i>Media Access Control</i>
MCU	Unidade Microcontroladora, do inglês <i>Microcontroller Unit</i>
MQTT	Transporte de Telemetria através de enfileiramento de Mensagens, do inglês <i>Message Queuing Telemetry Transport</i>
MQTT-SN	Transporte de Telemetria através de enfileiramento de Mensagens para Redes de Sensores, do inglês <i>Message Queuing Telemetry Transport for Sensor Networks</i>

NPM	Gerenciador de Pacotes do Node, do inglês <i>Node Package Manager</i>
OSI	Interconexão de Sistemas Abertos, do inglês <i>Open System Interconnection</i>
PCB	Placa de Circuito Impresso, do inglês <i>Printed Circuit Board</i>
REST	Transferência de Estado Representacional, do inglês <i>Representational State Transfer</i>
RFID	Identificação por radiofrequência, do inglês <i>Radio-Frequency Identification</i>
RPL	Protocolo de Roteamento para redes de baixo consumo com perdas, do inglês <i>Routing Protocol for Low-Power and Lossy Networks</i>
RSSI	Indicador de Intensidade de Sinal Recebido, do inglês <i>Received Signal Strength Indication</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Linguagem de Consulta Estruturada, do inglês <i>Structured Query Language</i>
TCP	Protocolo de controle de Transmissão, do inglês <i>Transmission Control Protocol</i>
UDP	Protocolo de Datagramas do Usuário, do inglês <i>User Datagram Protocol</i>
URI	Identificador Uniforme de Recurso, do inglês <i>Uniform Resource Identifier</i>
USB	Barramento Serial Universal, do inglês <i>Universal Serial Bus</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	PROBLEMA	13
1.2	OBJETIVOS	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	14
1.3	JUSTIFICATIVA	15
1.4	ESTRUTURA DO TRABALHO	15
<b>2</b>	<b>INTERNET DAS COISAS</b>	<b>16</b>
2.1	SURGIMENTO E UTILIZAÇÃO	16
2.2	EVOLUÇÃO	16
<b>3</b>	<b>TECNOLOGIAS PARA IOT</b>	<b>18</b>
3.1	PROTOCOLOS	18
3.1.1	MQTT-SN	18
3.1.2	6LoWPAN	19
3.2	BORDER ROUTER	20
3.3	BROKER	22
3.4	CONTIKI	22
3.5	CC2650	23
3.6	CC2650 LAUNCHPAD	24
3.7	CC2650 SENSORTAG	24
3.8	RASPBERRY PI	25
3.9	MARIADB	26
3.10	NODE.JS	27
<b>4</b>	<b>RASTREADOR DAS COISAS</b>	<b>29</b>
4.1	SENSORES	29
4.2	SERVIDOR	30
4.2.1	Base de Dados	30
4.2.2	Serviços	31
4.3	APLICATIVO ANDROID	33
4.3.1	Cadastrar Servidor	36
4.3.2	Visualizar Locais	36
4.3.3	Editar Descrição de um Local	37
4.3.4	Visualizar Sensores	38
4.3.5	Editar Descrição e Designar Local para um Sensor Móvel	39
4.3.6	Verificar Locais Acessados por um Sensor Móvel	40
4.3.7	Monitorar um Local	40
4.4	RESULTADO	41
<b>5</b>	<b>COMENTÁRIOS FINAIS</b>	<b>43</b>
	REFERÊNCIAS	44

## 1 INTRODUÇÃO

A inserção da Internet das Coisas (IoT, do inglês *Internet of Things*) no cotidiano das pessoas está ocorrendo gradualmente. Novas ideias e novos produtos estão surgindo, propiciando soluções para diversas áreas da tecnologia.

Esse trabalho propõe uma solução para o rastreamento de objetos ou de pessoas inseridas em um determinado ambiente, como uma empresa ou escola, onde não exista a possibilidade do uso de sistema de posicionamento global (GPS, do inglês *Global Positioning System*), ou onde a aplicação de métodos de triangulação de sinais seja demasiadamente complexa ou inviável por qualquer outro motivo. O rastreamento proposto tem como base o monitoramento dos níveis dos sinais medidos na comunicação entre sensores móveis e fixos.

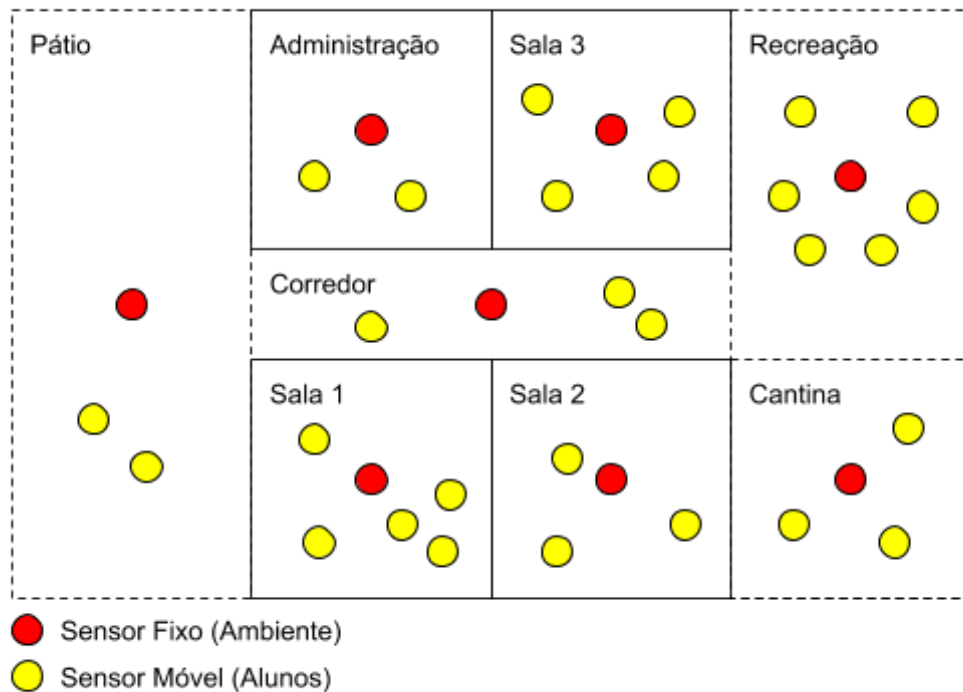
### 1.1 PROBLEMA

São frequentes nos noticiários, fugas de crianças de creches e escolas. Isso pode ocorrer por pequenos descuidos dos responsáveis pelas crianças nestes estabelecimentos, já que pode não ser possível monitorar seus alunos o tempo todo. As crianças por sua vez não têm ciência dos riscos que correm ao saírem dos ambientes das instituições, e o fazem geralmente com o objetivo de voltarem às suas residências.

Para evitar situações como essa, levando-se em consideração o ambiente de uma escola, imaginou-se a construção de uma rede de sensores fixos e móveis. Em cada sala ou ambiente a ser monitorado fica instalado um sensor, que é chamado de fixo. Cada aluno carrega um sensor, chamado de móvel, que fica inserido em um colar ou bracelete. Esses sensores carregados pelos alunos conectam-se com o sensor do ambiente mais próximo, e enviam a informação de proximidade para um servidor. A Figura 1 mostra uma planta fictícia de uma escola e a distribuição de sensores no ambiente.

A partir das informações de proximidades entre os sensores móveis e fixos registradas, com o uso de um aplicativo desenvolvido para aparelhos celulares, os professores e responsáveis pelos alunos podem monitorar em qual ambiente está cada criança, e ficam sabendo em tempo real se alguma criança está fora do ambiente para o qual ela foi destinada.

Figura 1 – Planta fictícia de uma escola



Fonte: Autoria própria.

## 1.2 OBJETIVOS

O desenvolvimento desse projeto ocorre em um momento de crescimento das tecnologias empregadas no contexto de internet das coisas. Portanto, ele deve fazer uso dos mais recentes recursos tecnológicos disponíveis no mercado.

### 1.2.1 Objetivo Geral

Desenvolver um protótipo de um sistema para a monitoração do posicionamento de pessoas dentro do ambiente de uma instituição, utilizando-se dos conceitos e tecnologias empregados em internet das coisas.

### 1.2.2 Objetivos Específicos

- Desenvolver códigos para sensores móveis e fixos que os permitam enviar informações de proximidades entre os mesmos;
- Modelar uma base de dados para registro dos dados;

- Desenvolver um código que, instalado em um servidor, efetue coleta de informações de sensores e registro das mesmas na base de dados, e que as torne disponíveis através de *webservices*;
- Preparar o código do servidor para prover serviços de manutenção de cadastro dos sensores;
- Desenvolver um aplicativo para aparelhos celulares que utilize os serviços do servidor para acessar as informações geradas.

### 1.3 JUSTIFICATIVA

É oportuno o desenvolvimento desse trabalho em um momento em que:

- A internet das coisas está apresentando um grande crescimento;
- A procura por novos dispositivos que estejam conectados na internet ou em uma simples rede de dados, e que facilitem o cotidiano das pessoas com recursos inovadores, é cada vez maior.

### 1.4 ESTRUTURA DO TRABALHO

A apresentação deste trabalho segue a seguinte estrutura:

**Capítulo 1 - Introdução:** apresenta o tema, o problema, os objetivos, a justificativa e essa estrutura do trabalho.

**Capítulo 2 - Internet das Coisas:** apresenta uma introdução à internet das coisas.

**Capítulo 3 - Estudo das Tecnologias:** descreve as tecnologias voltadas à internet das coisas e aquelas adotadas nesse projeto.

**Capítulo 4 - Rastreador das Coisas:** mostra detalhadamente o processo de desenvolvimento do protótipo.

**Capítulo 5 - Conclusão:** apresenta as considerações finais do trabalho.



## 2 INTERNET DAS COISAS

A internet das coisas conecta dispositivos diversos à rede mundial de computadores, dando-lhes capacidades de disponibilizar informações geradas através da leitura de sensores, e de realizar funções diversas através de comandos remotos. Isso possibilita o aprimoramento do uso desses dispositivos, tornando-os cada vez mais eficientes.

### 2.1 SURGIMENTO E UTILIZAÇÃO

Filho (2016) afirma que a internet das coisas se desenvolveu a partir da confluência da internet com os sistemas de Identificação por Radiofrequência (RFID, do inglês *Radio-Frequency Identification*) e outras redes sem fio de automação. São inúmeros os benefícios alcançados com o uso dessa tecnologia. O Quadro 1 exibe uma gama de possíveis cenários para a implantação de IoT.

**Quadro 1 – Cenários para uso da Internet das Coisas**

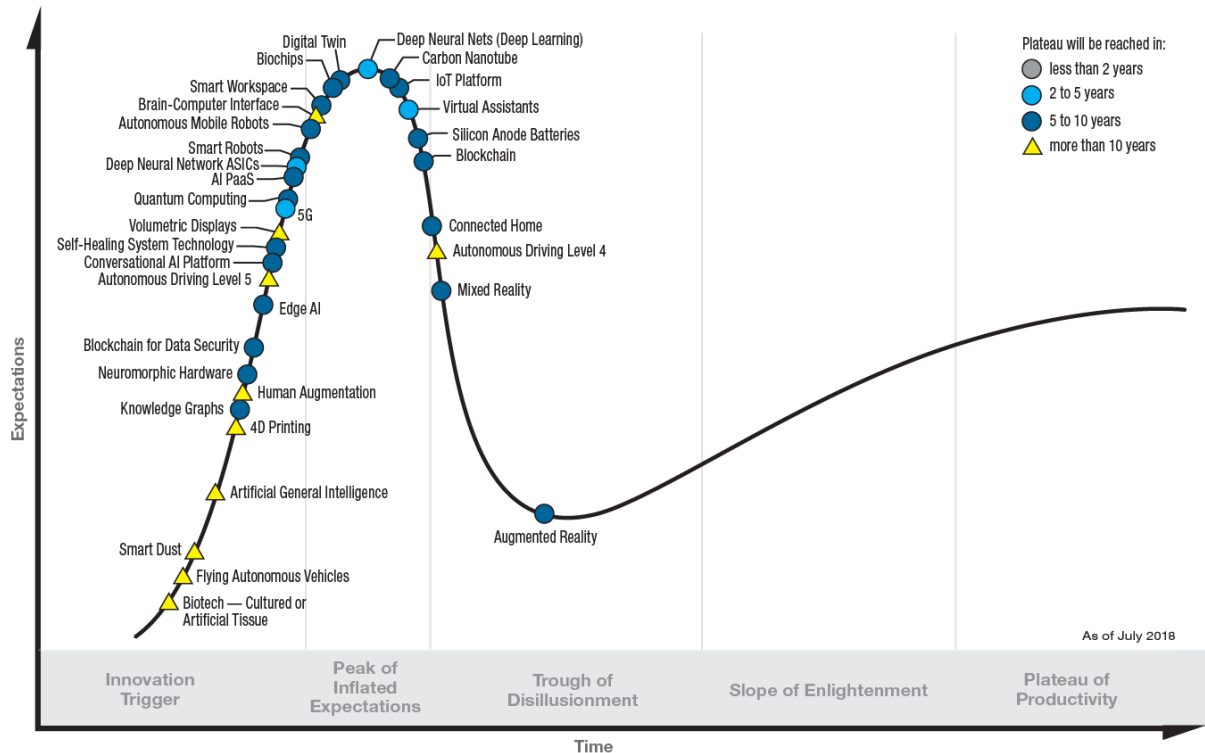
Cenário	Exemplo de Aplicação
Cidade	Serviços para a população
Indústria	Controle da produção e segurança
Saúde	Monitoramento da saúde de pessoas
Residência	Comodidade e economia de energia
Comércio	Automação e segurança patrimonial
Escritório	Controle de pessoal e qualidade do trabalho
Logística	Agilidade e qualificação de entregas
Trânsito	Integração entre veículos e sinalização
Rural	Controle da produção agropecuária
Energia	Geração e distribuição de energia otimizadas

**Fonte: Autoria própria.**

### 2.2 EVOLUÇÃO

A Figura 2 indica o posicionamento da internet das coisas (*IoT Platform*) no gráfico *Hype Cycle for Emerging Technologies*, construído pela empresa de pesquisa e consultoria Gartner para o ano de 2018. IoT aparece dentre as tecnologias emergentes que estão na fase de *Peak of Inflated Expectations*, o que demonstra grande expectativa para o desenvolvimento e crescimento dessa tecnologia.

Figura 2 – Gartner Hype Cycle 2018



Fonte: Gartner (2018).

A indústria encara a internet das coisas com otimismo e a aponta como importante elemento econômico nas próximas décadas. A estimativa de impacto econômico global vinculado ao cenário de IoT corresponde a mais de 11 trilhões de dólares em 2025 (MAGRANI, 2018). São previstos cerca de 100 bilhões de dispositivos inteligentes conectados até lá.

No Brasil, o crescimento da internet das coisas mostra-se bastante promissor. O Ministério de Ciência, Tecnologia, Inovação e Comunicações lançou, em outubro de 2017, o Plano Nacional de Internet das Coisas. O principal objetivo é inserir o país na revolução econômica e tecnológica da conectividade de dados (PULICE, 2017).

### 3 TECNOLOGIAS PARA IOT

Esse capítulo descreve os principais conceitos de internet das coisas e apresenta todo o *hardware* e *software* utilizado no desenvolvimento desse trabalho.

#### 3.1 PROTOCOLOS

Protocolos são os conjuntos de regras necessárias para a padronização da comunicação entre dispositivos capazes de enviar e receber informações. Para que ocorra a troca de mensagens entre seus componentes, a internet das coisas requer protocolos de rede específicos. Sendo assim, dois equipamentos não podem simplesmente enviar um fluxo de bits um para o outro e aguardarem ser interpretados. Os dispositivos precisam concordar com o protocolo.

O HTTP (Protocolo de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol*) é o protocolo de comunicação da camada de aplicação do modelo OSI (Interconexão de Sistemas Abertos, do inglês *Open System Interconnection*) que é utilizado na internet para a transferência de dados.

Para evitar a degradação do desempenho da rede devido a sobrecarga do protocolo HTTP, que opera sobre o protocolo TCP (Protocolo de Controle de Transmissão, do inglês *Transmission Control Protocol*) na camada de Transporte, em internet das coisas utilizam-se protocolos como o CoAP (Protocolo para Aplicações Restritas, do inglês *Constrained Application Protocol*) e o MQTT (Transporte de Telemetria através de enfileiramento de Mensagens, do inglês *Message Queuing Telemetry Transport*), mais indicados para pequenos dispositivos conectados à rede.

##### 3.1.1 MQTT-SN

O MQTT tem características que o destacam dentre os protocolos definidos para o uso em internet das coisas, como facilidade de implementação, qualidade de serviço e segurança.

Esse protocolo foi desenvolvido inicialmente pela IBM no final dos anos 90. Sua aplicação original era vincular sensores em *pipelines* de petróleo a satélites. Como seu nome sugere, ele é um protocolo de mensagem com suporte para a comunicação assíncrona entre as partes. Um protocolo de sistema de mensagens assíncrono desacopla o emissor

e o receptor da mensagem tanto no espaço quanto no tempo e, portanto, é escalável em ambientes de rede que não são confiáveis (YUAN, 2017).

O protocolo MQTT utiliza o *publish/subscriber* (publicador/subscritor) como padrão para troca de mensagens, onde os nós da rede que precisam receber determinados tipos de informações inscrevem-se em um *broker*, que é o intermediador no processo de comunicação da rede MQTT e gerencia as publicações e as subscrições. Os nós que têm informações utilizam o *broker* para publicá-las.

Como o protocolo MQTT foi projetado para usar TCP, o que não é indicado para redes de sensores sem fio e de baixo consumo de energia, foi desenvolvido o MQTT-SN (Transporte de Telemetria através de enfileiramento de Mensagens para Redes de Sensores, do inglês *Message Queuing Telemetry Transport for Sensor Networks*), que faz uso do Protocolo de Datagramas do Usuário (UDP, do inglês *User Datagram Protocol*), operando de forma semelhante ao protocolo original.

Algumas das vantagens do MQTT em relação aos outros protocolos utilizados em internet das coisas são:

- Risco de sobrecarga na rede minimizado, o que reduz a possibilidade de falhas;
- Protocolo de domínio público, o que o torna flexível e possibilita sua configuração de acordo com as necessidades do projeto;
- Desenvolvimento de códigos simplificados, que propicia seu funcionamento em dispositivos com restrições de processamento ou memória.

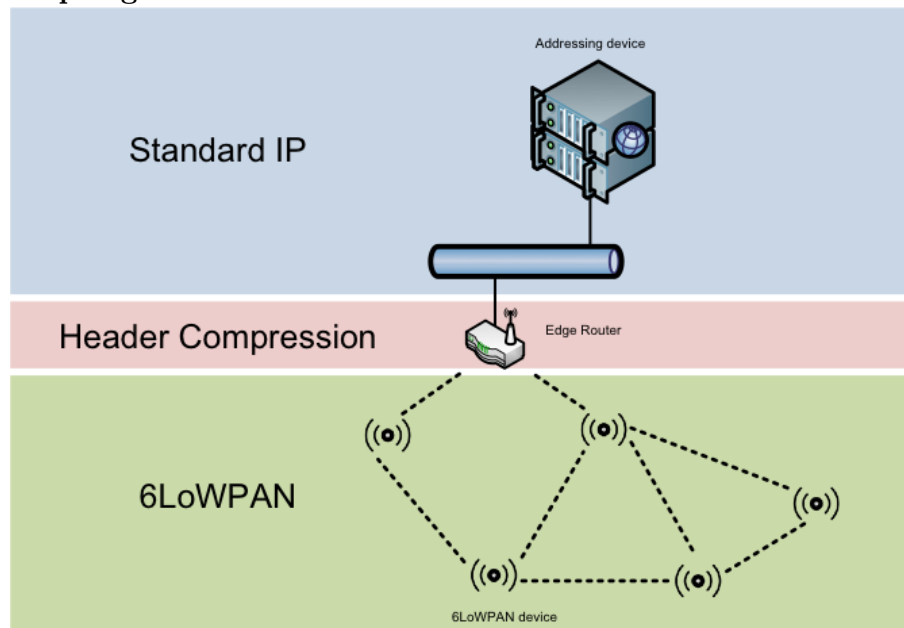
Levando-se em consideração essas vantagens, que possibilitam a conexão de múltiplos dispositivos e sensores à rede em um mesmo momento, o protocolo MQTT-SN foi escolhido para o desenvolvimento do protótipo construído para esse trabalho.

### 3.1.2 6LoWPAN

Com o protocolo IPv6 (Protocolo de Internet versão 6, do inglês *Internet Protocol version 6*) é possível atribuir um endereço IP (Protocolo de Internet, do inglês *Internet Protocol*) único para cada equipamento existente no mundo, já que são  $3,40 \times 10^{38}$  endereços disponíveis, muito acima dos 4.294.967.296 disponíveis com IPv4 (Protocolo de Internet versão 4, do inglês *Internet Protocol version 4*).

O 6LoWPAN (IPv6 sobre redes de área pessoal com baixo consumo, do inglês IPv6 over Low power Wireless Personal Area Networks) é um protocolo que adapta o IPv6 padrão para dispositivos projetados para ter baixo consumo de energia e menor poder de processamento. A Figura 3 exibe uma rede padrão, IPv4 ou IPv6, ligada à uma rede 6LoWPAN, o que faz-se através de um roteador de borda, cuja definição apresenta-se na Seção 3.2.

**Figura 3 – Topologia básica de uma rede 6LoWPAN**



Fonte: Embedded (2012).

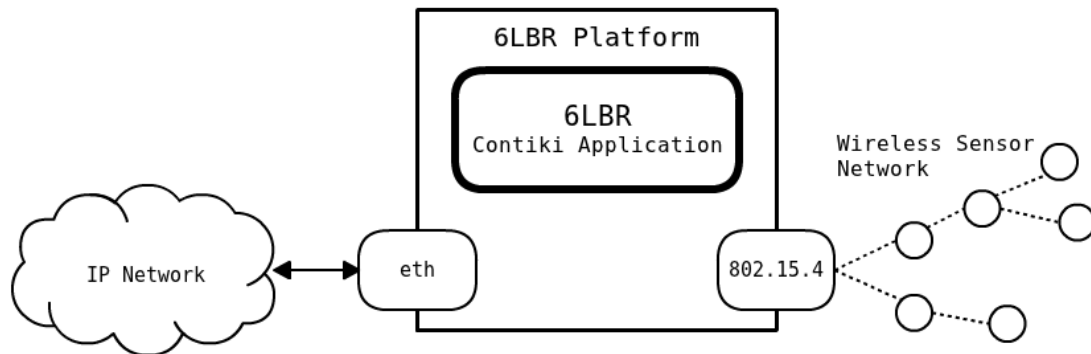
O protocolo 6LoWPAN define encapsulamento, *headers* e mecanismos de compressão. O endereço IPv6 (128 *bits*) é dividido em prefixo de rede (64 *bits*) e endereço da interface (64 *bits*). Um dos possíveis mecanismos de compressão de cabeçalho do 6LoWPAN que pode-se utilizar omite os 64 *bits* mais significativos, que são os do prefixo, porque eles são fixados para uma determinada rede 6LoWPAN. Os outros *bits*, destinados à interface, já garantem um espaço bastante grande para endereçamento, que chega a um número de até  $1,84467441 \times 10^{19}$  endereços. Existe opção de compressão de endereço de interface, sendo mais comum a compressão de 16 *bits*. Dessa forma, os 128 *bits* originais do endereço IPv6 podem ser comprimidos para 16 com o uso do protocolo 6LoWPAN.

### 3.2 BORDER ROUTER

Em uma rede 6LoWPAN, para que os sensores tenham acesso à rede IP externa e possam enviar e receber mensagens de outros dispositivos, é necessário que eles se

conectem a um roteador de borda (*border router* em inglês), que tem a função de adaptar as mensagens da rede 6LoWPAN para a rede IP externa e vice-versa. O esquema apresentado na Figura 4 demonstra essa interconexão de redes.

**Figura 4 – Esquema de conexão entre redes com o uso do 6LBR**



**Fonte: Cetic (2012).**

O 6LBR (Roteador de borda 6LoWPAN, do inglês *6LoWPAN Border Router*) é um roteador de borda de código aberto desenvolvido pela CETIC. Ao ser iniciado no servidor ele disponibiliza um portal com ferramentas úteis para a gestão dos componentes da rede 6LoWPAN, sendo possível acessá-lo através de um navegador de *internet* padrão. A Figura 5 exibe a tela que lista sensores conectados à rede e detectados pelo 6LBR em um determinado momento.

**Figura 5 – Tela do 6LBR para visualização de sensores**

**6LBR**  
6Lowpan Border Router

System **Sensors** Status Configuration Statistics Administration

Sensors Node tree PRR Parent switch Hop count Traffic Export

**Sensors**

**Sensors list**

Node	Type	Web	Coap	Parent	Up PRR	Down PRR	Last seen	Status
fd00::212:4b00:6e2:9f89	TI	web	coap	fe80::212:4b00:768:d35b	100.0%		4	OK
fd00::212:4b00:6e2:3b0b	TI	web	coap	fe80::212:4b00:768:d35b	100.0%		5	OK

**Actions**

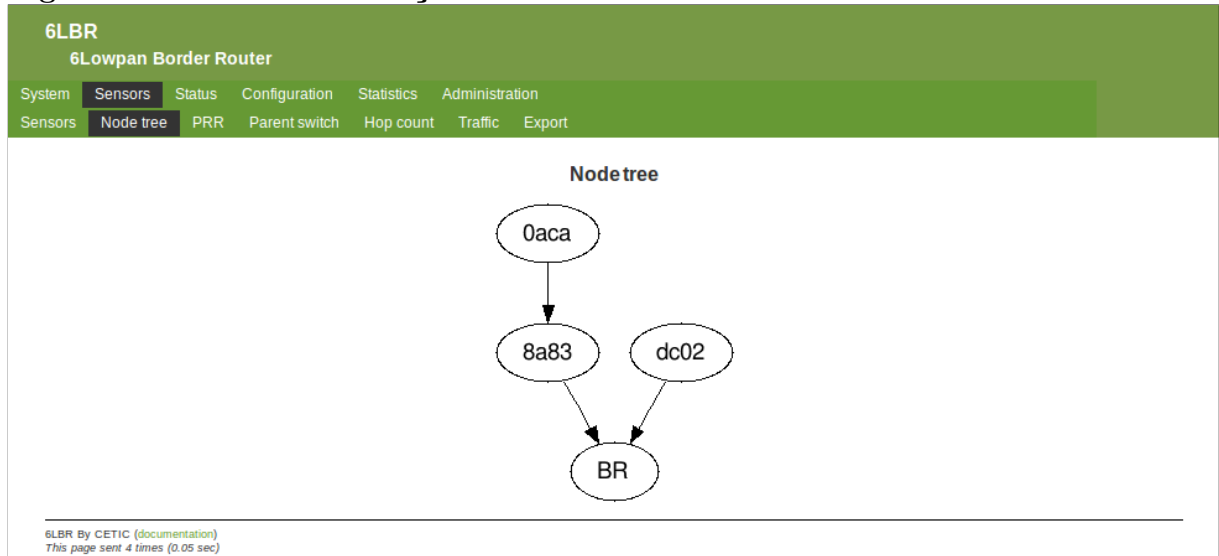
6LBR By CETIC (documentation)  
This page sent 10 times (1.23 sec)

**Fonte: Autoria própria.**

A Figura 6 apresenta a tela que mostra graficamente os sensores, representados

pelos números de identificação 0aca, 8a83 e dc02, conectados ao roteador de borda, representado pela abreviação BR.

**Figura 6 – Tela de visualização de *Node Tree* do 6LBR**



**Fonte: Autoria própria.**

### 3.3 BROKER

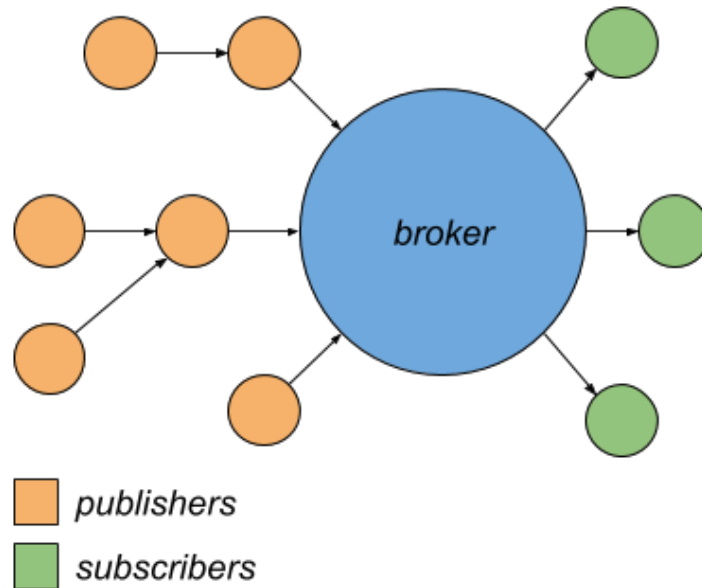
O *Broker* é o elemento da rede MQTT responsável por gerenciar as mensagens do protocolo. Através do padrão *publish/subscriber*, ele recebe as informações dos nós, como é o caso dos sensores, que publicam as mensagens em um tópico, e as envia para os nós inscritos nesse mesmo tópico, que podem ser aplicativos, servidores ou até mesmo outros sensores.

A Figura 7 mostra o esquema básico da ligação entre o *broker*, os nós que publicam mensagens em um determinado tópico e os nós que estão inscritos nesse tópico. Portanto, dispositivos e sensores devem se conectar ao *broker* para enviar ou receber mensagens.

### 3.4 CONTIKI

O *Contiki* é um sistema operacional de código aberto concebido há mais de uma década para ser implementado em *hardwares* de sensores com limitadas capacidades de processamento e memória, reduzida largura de banda de comunicação, e que apresentam baixo custo e consumo de energia reduzido.

Figura 7 – Exemplo das ligações entre o *broker*, *subscribers* e *publishers*



Fonte: Autoria própria.

Este sistema operacional possui suporte para diversos microcontroladores, incluindo o CC2650 da *Texas Instruments*, apresentado na Seção 3.5. As aplicações desenvolvidas para o *Contiki* são escritas na linguagem de programação “C”.

O projeto do *Contiki* atende aos protocolos IPv4 e IPv6, e também àqueles desenvolvidos baixo consumo de energia, como o 6LoWPAN, o CoAP e o protocolo de roteamento para redes de baixo consumo com perdas (RPL, do inglês *Routing Protocol for Low-Power and Lossy Networks*).

### 3.5 CC2650

Esse microcontrolador ou Unidade Microcontroladora (MCU, do inglês *Microcontroller Unit*) produzido pela *Texas Instruments* contém um processador ARM Cortex-M3 de 32 bits, que pode operar com *clock* de 48 MHz, e possui 128 kB de memória programável. O CC2650, que é exibido na Figura 8, é compatível com soluções que utilizam o protocolo 6LoWPAN.

O microcontrolador CC2650 é ideal para se coletar informações a partir da leitura de diversos tipos de sensores, sejam eles analógicos ou digitais. Ele faz isso de forma autônoma, enquanto o restante do sistema está no modo “suspenso” ou “dormindo”. Por esse motivo, o microcontrolador é utilizado em uma grande variedade de produtos que



**Figura 8 – Microcontrolador CC2650**



**Fonte: Texas (2016a).**

precisam apresentar reduzido consumo de energia. A Figura 9 exhibe o diagrama em blocos dos microcontroladores da série CC26xx.

### 3.6 CC2650 LAUNCHPAD

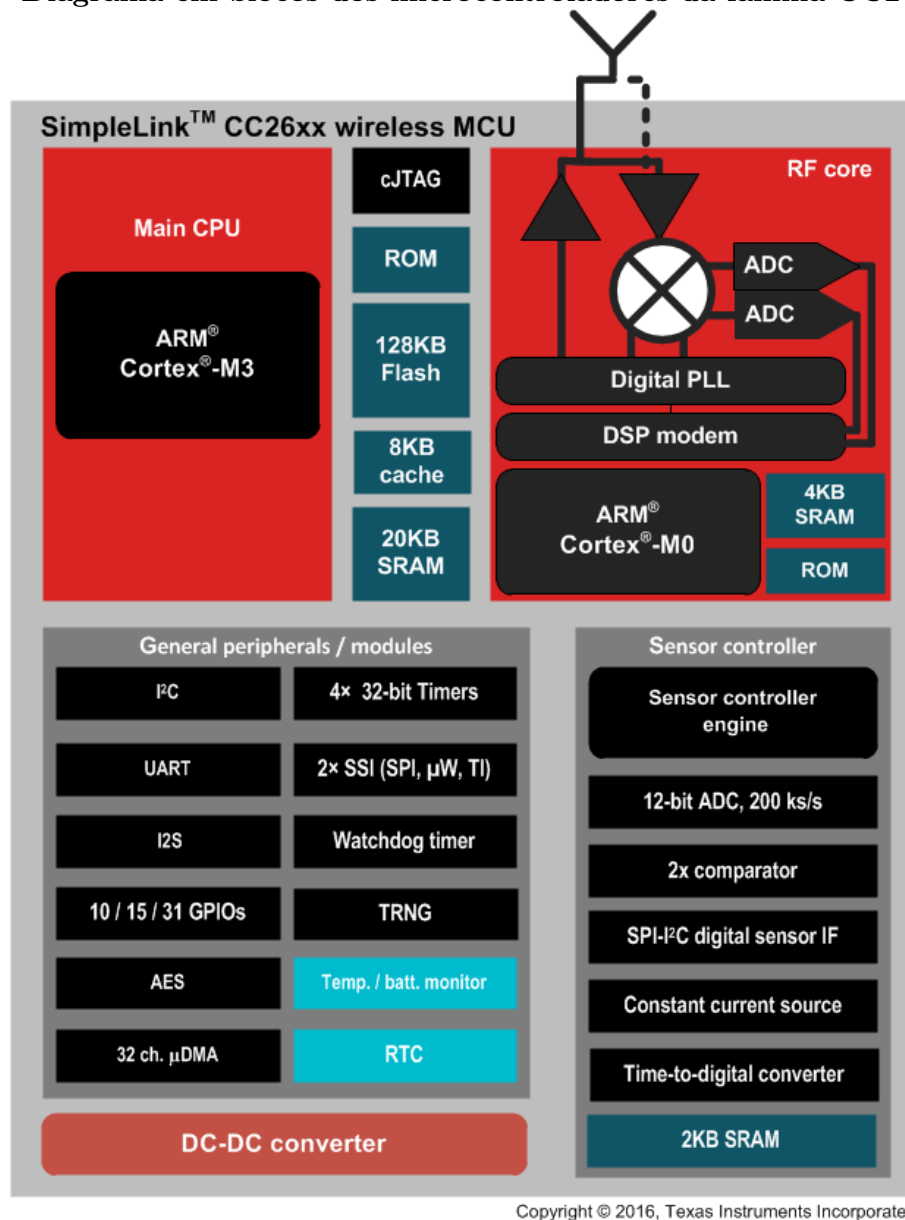
A Figura 10 apresenta o *kit* de desenvolvimento *SimpleLink™ CC2650 Wireless MCU LaunchPad™*, fabricado pela *Texas Instruments*. Ele facilita os testes dos códigos desenvolvidos para a MCU CC2650, e permite o acesso a toda a pinagem de entrada e saída do microcontrolador. A depuração desses códigos é efetuada através da porta USB (Barramento Serial Universal, do inglês *Universal Serial Bus*) do dispositivo.

### 3.7 CC2650 SENSORTAG

A Figura 11 mostra o *SimpleLink™ Bluetooth low energy/Multi-standard SensorTag*, outro *kit* que também é fabricado pela *Texas Instruments*. Ele se difere do CC2650 *LaunchPad* porque já possui diversos sensores em sua placa mãe, que podem medir temperatura, luminosidade, ruído, dentre outros. A placa é instalada em um estojo plástico, possui uma capa de proteção de borracha e pode funcionar independente de alimentação externa, pois contém suporte para uma pequena bateria.

O *SensorTag* tem um tamanho reduzido, sendo pouco menor que a metade do tamanho do *LaunchPad*, e possui menos componentes eletrônicos. Para a inclusão de facilidades no mesmo, é necessária a instalação de módulos de expansão como o *SimpleLink SensorTag Debugger DevPack*, que permite a conexão do conjunto a um computador através de porta USB para a escrita dos códigos desenvolvidos para o CC2650.

Figura 9 – Diagrama em blocos dos microcontroladores da família CC26xx



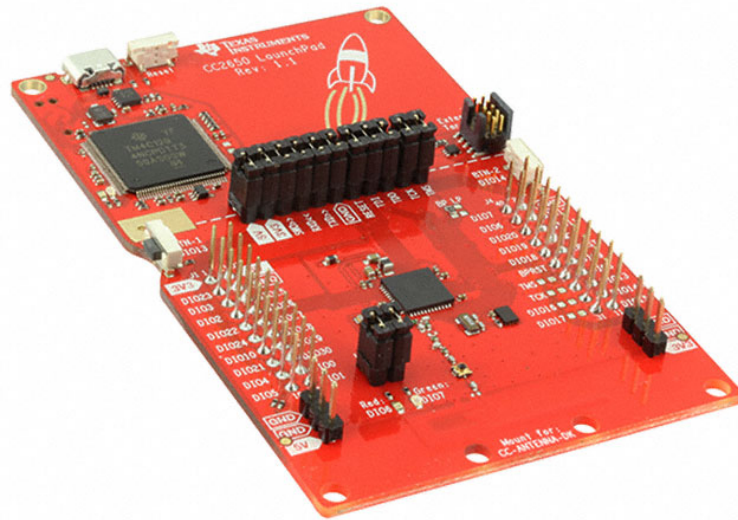
Fonte: Texas (2016a).

### 3.8 RASPBERRY PI

O *Raspberry Pi* é um pequeno computador que começou a ser produzido no Reino Unido pela *Raspberry Pi Foundation* no ano de 2012. Ele tem como objetivo a promoção do ensino na ciência da computação. Atualmente existem à venda pelo menos sete modelos, com diferentes capacidades de processamento e variada conectividade.

São diversos os sistemas operacionais que podem ser instalados nesse computador. O modelo utilizado como servidor no projeto é igual ao exibido na Figura 12, e usa o

Figura 10 – *Kit de desenvolvimento CC2650 LaunchPad*



Fonte: Texas (2016c).

Figura 11 – *Kit CC2650 SensorTag*



Fonte: Texas (2016b).

*Raspbian* como sistema operacional, que é uma distribuição *Linux* preparada para esse equipamento.

### 3.9 MARIADB

*MariaDB* é um Sistema de Gerenciamento de Banco de Dados (SGBD) que utiliza como interface a Linguagem de Consulta Estruturada (SQL, do inglês *Structured Query Language*). Ele foi desenvolvido como um *fork* do *MySQL*, um projeto bastante popular

**Figura 12 – Raspberry Pi 3 Model B**



**Fonte: Raspberry (2016).**

que teve os direitos adquiridos pela *Sun Microsystems*, a qual posteriormente foi vendida para a *Oracle*.

Para Cardoso (2015), a segurança é um foco especial para os desenvolvedores do *MariaDB* e o projeto mantém os seus próprios protocolos de segurança. Diversas falhas nesse tópico percebidas no *MySQL* foram encontradas pela equipe do *MariaDB* e posteriormente relatadas. A colaboração com o *MySQL* é estreita, o que garante que questões de segurança sejam sanadas imediatamente quando descobertas.

A equipe responsável pelo projeto *MariaDB* mantém a compatibilidade e a correspondência com APIs (Interface de Programação de Aplicativos, do inglês *Application Programming Interface*) e comandos do *MySQL*, e procura manter esse SGBD como software livre.

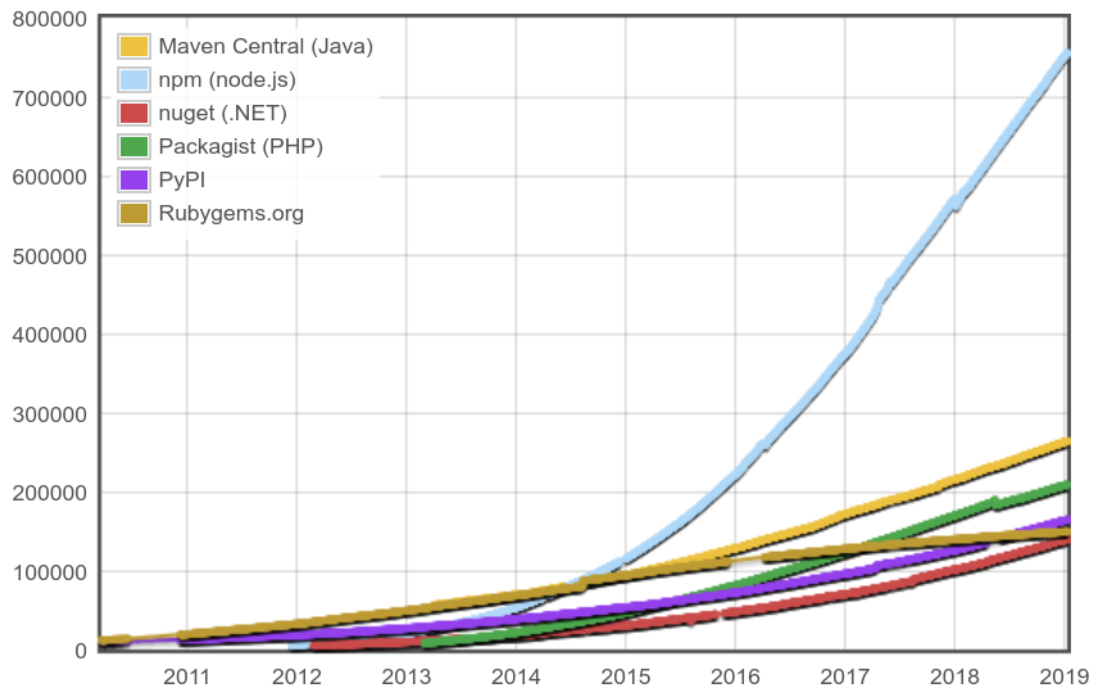
### 3.10 NODE.JS

*Node.js* é uma plataforma de código aberto criada para desenvolvimento de aplicações baseadas em rede do lado do servidor. Ele é composto pela união do interpretador *V8 JavaScript Engine* do *Google* e da biblioteca “*libuv*”, além de um conjunto de módulos. Sua linguagem de programação é o *JavaScript*, que é uma linguagem de *scripting* interpretada e atualmente é utilizada em muitas aplicações, sejam elas do lado do cliente ou do lado do servidor.

Dentre os diversos módulos desenvolvidos para *Node.js*, que podem ser obtidos

através do Gerenciador de Pacotes do *Node* (NPM, do inglês *Node Package Manager*), existem muitos criados para internet das coisas. A Figura 13 mostra o crescimento do uso dessa plataforma nos últimos anos, através de um comparativo da quantidade de módulos disponíveis nos repositórios de algumas das principais linguagens de programação.

**Figura 13 – Quantidade de módulos disponíveis nos principais repositórios**



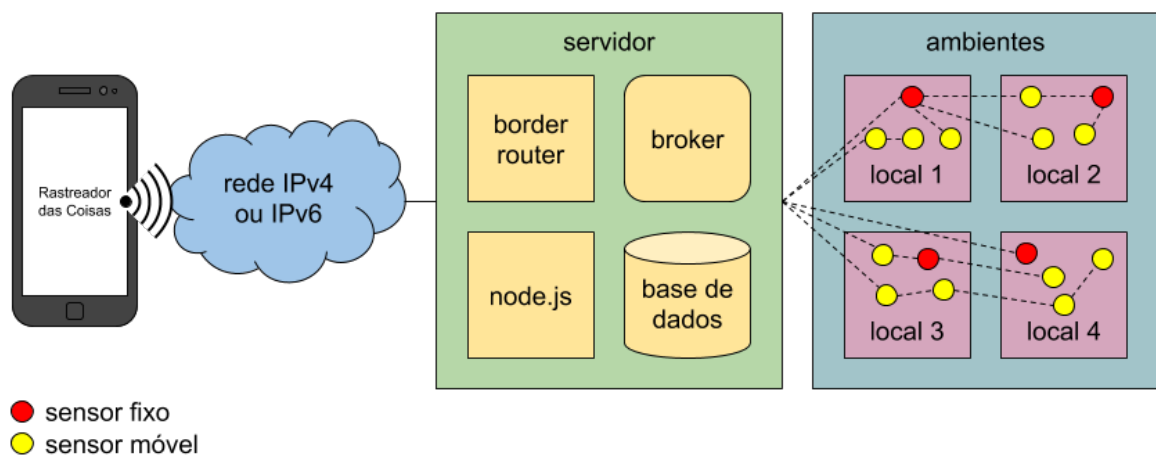
Fonte: DeBill (2019).

## 4 RASTREADOR DAS COISAS

Desenvolveu-se um projeto simples e de baixo custo. Ele utiliza dispositivos que, embora importados, são de fácil acesso para pesquisadores e estudantes. Para detalhes dos mesmos, pode-se consultar o Capítulo 3.

A Figura 14 é um diagrama que mostra os componentes do projeto. Os nós “pais” são os sensores fixos e os nós “filhos” são os sensores móveis.

**Figura 14 – Diagrama do Rastreador das Coisas**



**Fonte: Autoria própria.**

Para o desenvolvimento do aplicativo para celular e dos serviços disponíveis no servidor, utilizou-se *softwares* livres ou de código aberto.

### 4.1 SENSORES

O protótipo desenvolvido para este trabalho faz uso dos dispositivos *LaunchPad* e *Sensortag* nas funções de sensores ou nós. Embora o MCU CC2650 tenha diversas entradas e saídas para conexão com sensores ou servomecanismos, para o sistema de rastreamento construído, é somente necessária a utilização de sua capacidade de comunicação entre os nós para a determinação da localização dos alunos no ambiente.

Para a diferenciação dos sensores na rede, utilizou-se o endereço MAC (Controle de Acesso de Mídia, do inglês *Media Access Control*) de cada dispositivo como seu identificador, o que impede problemas de códigos de identificação duplicados ou conflitos nas informações registradas de nós fixos ou móveis.

## 4.2 SERVIDOR

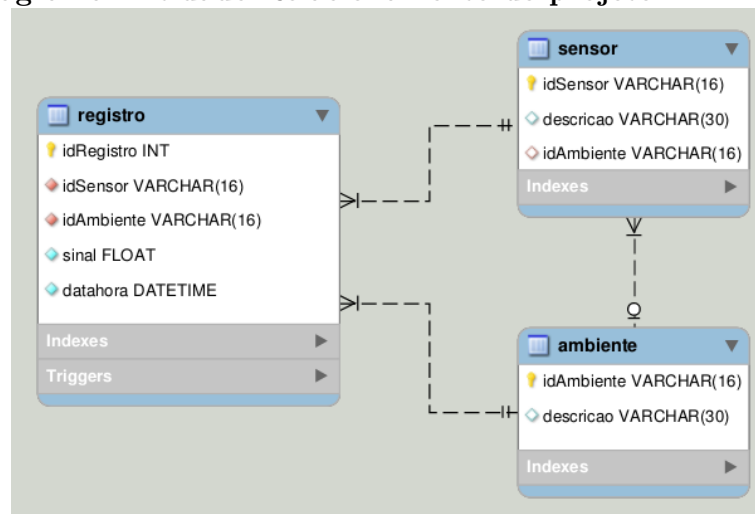
Instalou-se o servidor em um *Raspberry Pi 3 Model B*, com sistema operacional *Raspbian*. Ele concentra importantes elementos do esquema da internet das coisas, como pode-se verificar na Figura 14: o *border router* e o *broker*. Além disso, tem a função de armazenar os dados recebidos dos sensores em uma base de dados.

O equipamento conecta-se a um roteador compatível com IPv4 e IPv6 através de sua porta *ethernet*. A conexão com a rede 6LoWPAN ocorre através de um *Launchpad* preparado com o código *slip radio* disponível nos exemplos do pacote do *Contiki*.

### 4.2.1 Base de Dados

No servidor instalou-se o banco de dados *MariaDB*. A base de dados implementada para o projeto tem três tabelas, que registram as informações enviadas pelos sensores, e guardam as descrições dos mesmos e seus relacionamentos.

**Figura 15 – Diagrama Entidade-Relacionamento do projeto**



**Fonte: Autoria própria.**

As tabelas exibidas no Diagrama Entidade-Relacionamento (DER) da Figura 15 tem as seguintes funções:

- registro: armazenar os identificadores dos sensores móveis e fixos recebidos pelo servidor, e registrar data e hora de cada ocorrência;
- sensor: armazenar os identificadores dos sensores móveis, suas descrições e os identificadores dos ambientes relacionados;

- ambiente: armazenar os identificadores dos sensores fixos e suas descrições.

Para automatizar a inclusão de sensores nas tabelas sensor e ambiente, foi criado um *trigger*. Seu código pode ser visualizado na Figura 16. Para cada inserção na tabela registro, é verificada a existência dos sensores em suas respectivas tabelas. Então eles são inseridos, caso ainda não existam.

**Figura 16 – Script SQL para criação do *trigger***

```
DELIMITER $$
CREATE TRIGGER `registro_BEFORE_INSERT` BEFORE INSERT ON `registro` FOR EACH ROW
BEGIN
  IF ( SELECT COUNT(*) FROM sensor WHERE idSensor = NEW.idSensor ) < 1 THEN
    INSERT INTO sensor (idSensor) VALUES (NEW.idSensor);
  END IF;
  IF ( SELECT COUNT(*) FROM ambiente WHERE idAmbiente = NEW.idAmbiente ) < 1 THEN
    INSERT INTO ambiente (idAmbiente) VALUES (NEW.idAmbiente);
  END IF;
END$$
DELIMITER ;
```

Fonte: Autoria própria.

#### 4.2.2 Serviços

Desenvolveu-se um código na linguagem de programação *JavaScript*, que opera sobre a plataforma *Node.js* e faz uso de alguns módulos obtidos através do NPM:

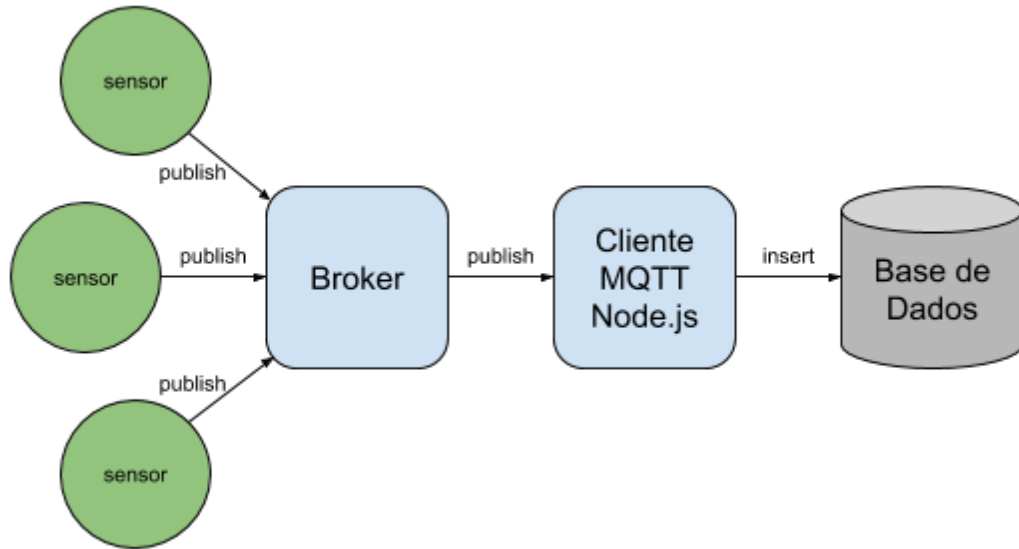
- express: *web framework*;
- mariadb: *client* da base de dados;
- mqtt: biblioteca para o protocolo MQTT.

Esse *script* funciona como um gerenciador de dados do sistema e tem como funções:

- Coletar, através da inscrição no tópico do protocolo MQTT, mensagens publicadas pelos sensores, e inserir os dados contidos nas mesmas na tabela de registro da base de dados;
- Fazer a comunicação entre o aplicativo Android e a base de dados através de *web services*, tornando possível a visualização das informações e alterações que se façam necessárias.



Figura 17 – Coleta e inserção das informações na base de dados



Fonte: Autoria própria.

A Figura 17 mostra o caminho percorrido pelas informações entre os sensores e a base de dados no processo de coleta das mesmas.

Os *web services* que propiciam a integração entre o aplicativo e a base de dados utilizam a arquitetura de Transferência de Estado Representacional (REST, do inglês *Representational State Transfer*). Esse fluxo de informações é demonstrado através da Figura 18.

Figura 18 – Trocas de requisições e respostas entre o aplicativo e a base de dados



Fonte: Autoria própria.

Criou-se um URI (Identificador Uniforme de Recurso, do inglês *Uniform Resource Identifier*) para cada requisição HTTP. O Quadro 2 exibe uma listagem dos URIs gerados. Como resposta, o serviço devolve ao aplicativo as informações no formato JSON (Notação de Objeto de *JavaScript*, do inglês *JavaScript Object Notation*), como o exemplo exibido

na Figura 19.

**Quadro 2 – URIs para acesso aos *web services***

<b>Função</b>	<b>URI (Sintaxe)</b>
Listar Sensores	http://<domínio ou endereço IP>/listar/sensores
Listar Ambientes	http://<domínio ou endereço IP>/listar/ambientes
Listar Sensores de um Ambiente	http://<domínio ou endereço IP>/listar/sensoresdoambiente?ambiente=<ambiente>
Exibir Detalhes de um Sensor	http://<domínio ou endereço IP>/detalhar/sensor?sensor=<sensor>
Ajustar Sensor	http://<domínio ou endereço IP>/ajustar/sensor?sensor=<sensor>&descricao=<descrição>&ambiente=<ambiente>
Ajustar Ambiente	http://<domínio ou endereço IP>/ajustar/ambiente?ambiente=<ambiente>&descricao=<descrição>
Exibir Últimas Localizações de um Sensor	http://<domínio ou endereço IP>/rastrear/sensor?sensor=<sensor>
Exibir Últimas Localizações de um Sensor em um Período	http://<domínio ou endereço IP>/rastrear/sensor?sensor=<sensor>&dhi=<dataInicial>&dhf=<dataFinal>

**Fonte: Autoria própria.**

**Figura 19 – Exemplo de resposta do serviço “Listar Sensores”, no formato JSON**

```
{
  "02124B0012050ACA": {
    "descricao": "João",
    "proximidade": "Sala 1"
  },
  "02124B0006E29F89": {
    "descricao": "Maria",
    "proximidade": "Cantina"
  }
}
```

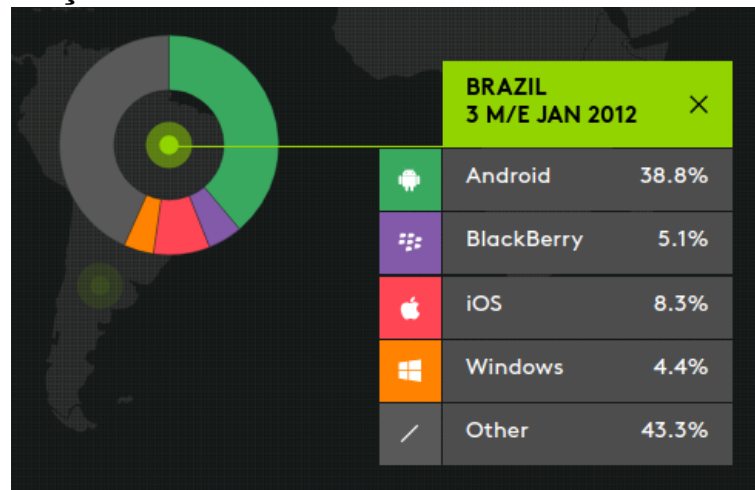
**Fonte: Autoria própria.**

### 4.3 APLICATIVO ANDROID

Devido à grande utilização dos dispositivos móveis, a forma mais eficiente para rastrear cada um dos nós, de forma segura, ilimitada e individual, foi a criação de um aplicativo que pode ser facilmente instalado nesses dispositivos e configurado para conectar ao servidor, receber os dados em tempo real e consultar o histórico, além de permitir renomear os sensores fixos e móveis.

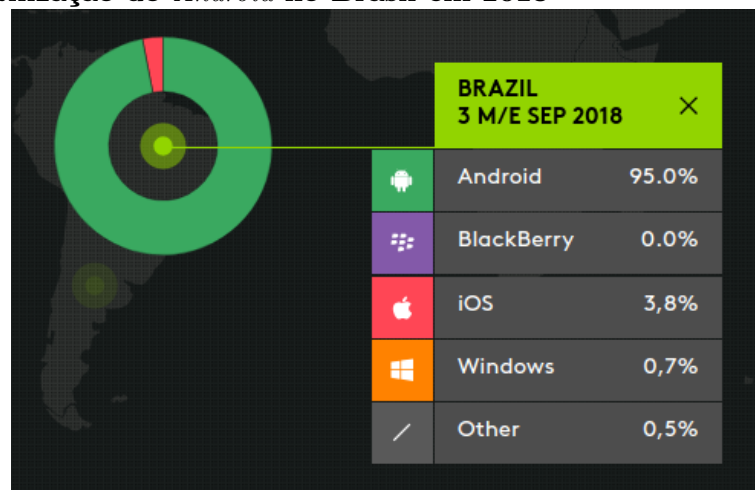
Para o protótipo do Rastreador das Coisas, optou-se por desenvolver o aplicativo para o sistema operacional *Android* por estar ele consolidado no mercado, com a imensa preferência dos brasileiros. Comparando-se as tabelas e os gráficos das Figuras 20 e 21, pode-se perceber o crescimento do *Android* no Brasil entre os anos de 2012 e 2018.

**Figura 20 – Utilização do *Android* no Brasil em 2012**



Fonte: Kantar (2019).

**Figura 21 – Utilização do *Android* no Brasil em 2018**

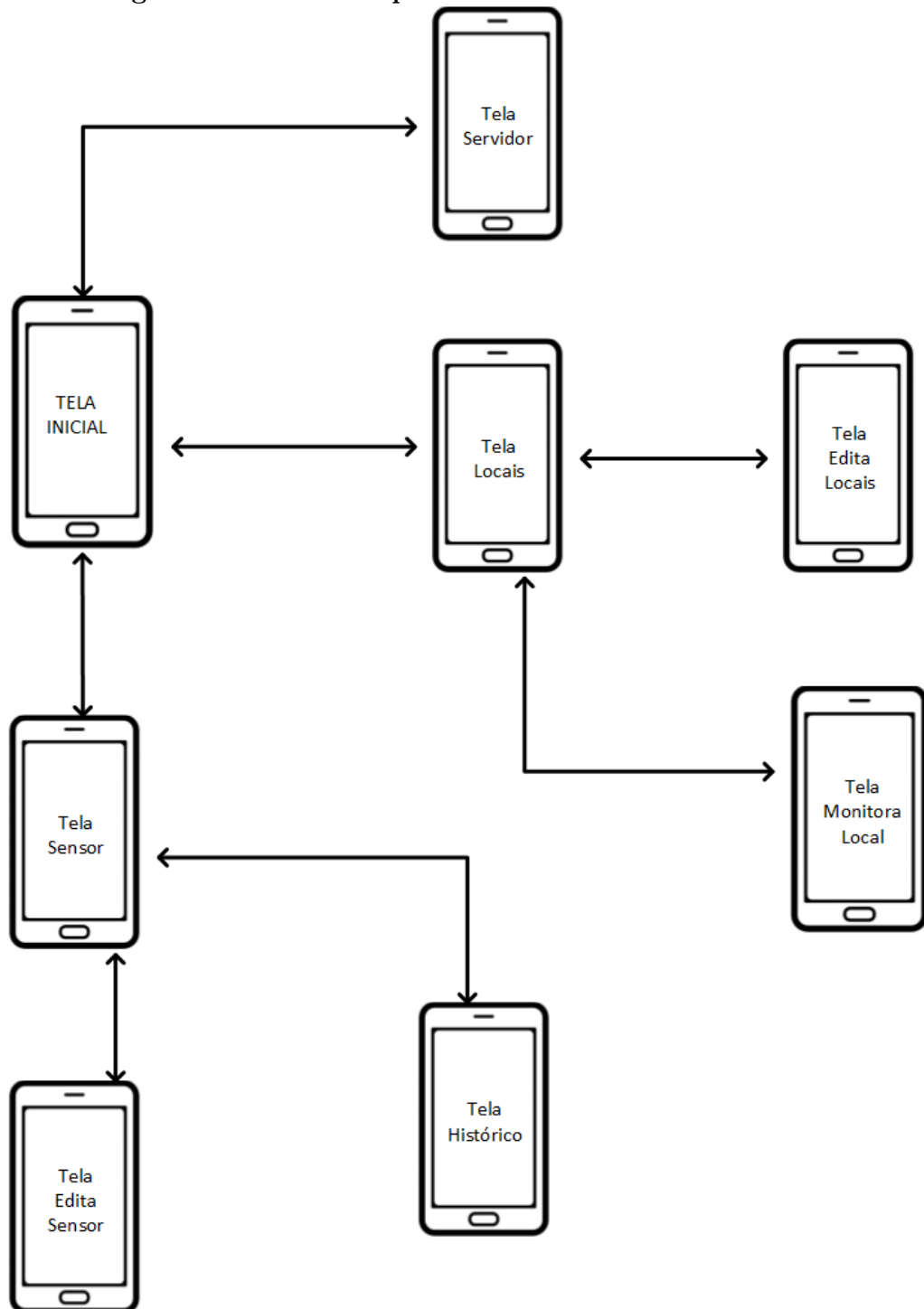


Fonte: Kantar (2019).

Através da utilização do *framework PhoneGap*, desenvolveu-se um aplicativo que opera sobre o sistema operacional *Android*. Esse *framework*, é utilizado para desenvolvimento de aplicativos móveis em código aberto, onde o mesmo código fonte, pode gerar aplicativos para *Android*, *Windows Phone* e *iOS* (Sistema Operacional para iPhone, do inglês *iPhone Operating System* ou *iPhone OS*). Para esse projeto optou-se por desenvolver usando *HTML* (Linguagem de Marcação de Hipertexto, do inglês *Hypertext*

Markup Language), JavaScript, Web SQL Database e Folhas de Estilo Cascadeadas (CSS, do inglês *Cascading Style Sheets*). O PhoneGap também é compatível com Java, C++ e Objective-C. A Figura 22 exibe o diagrama de telas do aplicativo, no qual as setas indicam as sequencias de acesso às funcionalidades.

Figura 22 – Diagrama de telas do aplicativo



Fonte: Autoria própria.

A tela inicial do aplicativo, que é exibida na Figura 23, mostra o menu geral do aplicativo, que aparece também nas outras telas e dá acesso a todas as funcionalidades.

**Figura 23 – Tela inicial do aplicativo**



**Fonte: Autoria própria.**

O aplicativo acessa informações contidas na base de dados através dos *web services* mencionados na Subseção 4.2.2, efetuando leituras, cadastros e alterações. O uso de serviços centralizados em um servidor proporciona o desenvolvimento de códigos menores e evita a inclusão de bibliotecas para acesso direto aos dados.

#### 4.3.1 Cadastrar Servidor

Após instalado no dispositivo *Android*, é necessário configurar o aplicativo para a conexão com o servidor. A Figura 24 mostra a tela desenvolvida para esse fim. O acesso está disponível através da opção “Servidor” no menu.

Os dados cadastrados são salvos localmente no dispositivo, através do *Web Sql*, que guarda as informações de nome, endereço IP e porta do servidor.

#### 4.3.2 Visualizar Locais

Quando o usuário acessa a opção “Locais” do menu, o aplicativo se conecta ao servidor e traz uma lista com todos os ambientes inseridos na base de dados. A tela pode

**Figura 24 – Tela para cadastro do servidor**

Locais Sensor Servidor

Rastreador  
das Coisas

Login information

Nome: Rastreador

IP: rastreador.servebeer.c

Porta: 8888

salvar

**Fonte: Autoria própria.**

ser observada na Figura 25. Os locais apresentados representam os sensores fixados nos ambientes da instituição. Enquanto não for cadastrada uma descrição para o sensor, o servidor enviará o próprio código de sensor fixo na listagem.

Se o usuário tocar em um ícone de edição, que é representado por um lápis, será exibida a tela de edição de descrição do local. Essa opção é apresentada na Subseção 4.3.3. No caso dele selecionar o sensor tocando em sua descrição, será exibida a tela de monitoração daquele local. A Subseção 4.3.7 descreve essa função do aplicativo.

#### 4.3.3 Editar Descrição de um Local

Através da tela de edição do local, que é exibida na Figura 26, o usuário pode alterar a descrição do sensor afixado em um determinado ambiente. Caso não tenha ocorrido algum cadastro ainda para esse dispositivo, o *web service* enviará o próprio código do mesmo como descrição.

O acesso a essa facilidade é feito através da tela de visualização de locais, descrita na Subseção 4.3.2.

Figura 25 – Tela de visualização de locais



Fonte: Autoria própria.

Figura 26 – Tela de edição de local



Fonte: Autoria própria.

#### 4.3.4 Visualizar Sensores

O acesso à opção “Sensor” do menu faz com que o aplicativo traga do servidor a listagem de sensores móveis ou de alunos, como pode-se ver na Figura 27. Essa lista

apresenta também o local onde o sensor móvel está no momento.

**Figura 27 – Tela de visualização de sensores**



**Fonte: Autoria própria.**

Com um toque na descrição de um sensor, é possível editar seus dados, o que é detalhado na Subseção 4.3.5. Também na tela de visualização de sensores pode-se acessar, através de um toque no ícone da coluna “histórico” da listagem, a tela que exhibe o relatório no qual verifica-se em quais ambientes o sensor móvel esteve. Detalhes sobre essa tela são exibidos na Subseção 4.3.6.

#### 4.3.5 Editar Descrição e Designar Local para um Sensor Móvel

A Figura 28 exhibe a tela de edição da descrição de sensor móvel e da designação do ambiente base para ele. O acesso a essa função do aplicativo pode ser feito através da visualização de sensores apresentada na Subseção 4.3.4.

O campo localidade é o ambiente em que o sensor móvel deve ter como base. Essa designação permite que a monitoração de local da Subseção 4.3.7 exhiba informações dos sensores móveis que estão dentro ou fora do ambiente predeterminado. Para editá-lo, deve-se utilizar o botão de seleção que permite fazer a alteração de localidade.



**Figura 28 – Tela de edição de sensor**

**Fonte: Autoria própria.**

#### 4.3.6 Verificar Locais Acessados por um Sensor Móvel

É possível visualizar os ambientes onde o aluno esteve através da tela exibida na Figura 29, que mostra também a data e horário em que ele foi detectado pela primeira vez no local. O acesso a essa função se dá através da tela de visualização de sensores, apresentada na Subseção 4.3.4.

#### 4.3.7 Monitorar um Local

Através da tela de monitoração de local, que é mostrada na Figura 30, o usuário pode verificar o status de cada sensor móvel associado ao local.

Nessa tela, que pode ser acessada através da visualização de locais como mostrado na Subseção 4.3.2, o professor, por exemplo, pode visualizar se os alunos da sala estão dentro ou fora do recinto. Na coluna “status”, alunos assinalados com ícones em forma de “V” com contorno na cor verde estão presentes no ambiente ao qual foram designados, e os que estão associados a ícones representados por um sinal de exclamação com contorno na cor vermelha, estão ausentes.

Figura 29 – Tela de visualização dos locais onde o aluno esteve



Sensor	Local	Data/Hora
Paulo	Pátio	2018-10-06T10:50:55.000Z
Paulo	Corredor	2018-10-06T10:51:59.000Z
Paulo	Sala 2	2018-10-06T10:52:25.000Z
Paulo	Corredor	2018-10-06T10:53:34.000Z
Paulo	Sala 1	2018-10-06T10:54:25.000Z

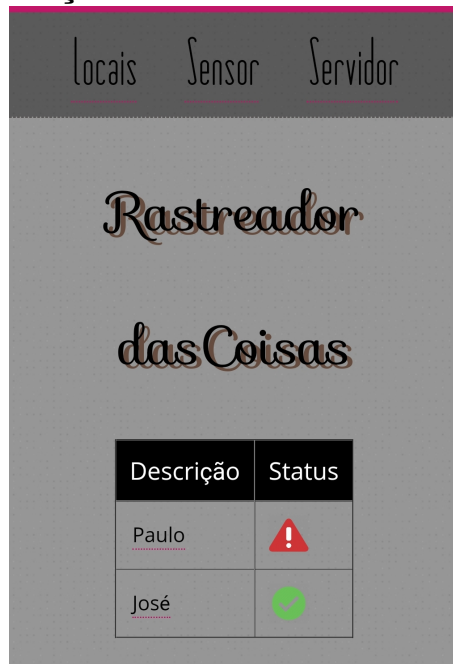
Fonte: Autoria própria.

#### 4.4 RESULTADO

Em um experimento inicial utilizou-se um exemplo de *multicast* contido no pacote do sistema operacional *Contiki*. O objetivo era fazer com que os sensores móveis enviassem as identificações de todos os sensores fixos próximos juntamente com os valores do Indicador de Intensidade de Sinal Recebido (RSSI, do inglês *Received Signal Strength Indication*) obtidos de cada um. Posteriormente o programa gerenciador de dados do servidor faria a seleção dos sensores fixos mais próximos aos móveis. Porém, na unificação dos códigos de *multicast* e de *mqtt-sn-tools* do *Contiki*, constatou-se que a memória *flash* do CC2650 não era suficiente para trabalhar com a mesclagem dos códigos desses exemplos, o que tornou essa solução inviável.

Nos experimentos seguintes utilizou-se o protocolo RPL. Os nós filhos, que são os sensores móveis, entram na rede através do Grafo Acíclico Direcionado Orientado ao Destino (DODAG, do inglês *Destination Oriented Directed Acyclic Graph*) do RPL,

**Figura 30 – Tela de monitoração de local**



**Fonte: Autoria própria.**

informam (através do *publish*) quem são seus nós pais, que são os sensores fixos mais próximos, e na sequência saem da rede. Isso se repete infinitamente a cada período de tempo que pode chegar a um minuto. Esse processo torna possível um nó móvel se deslocar entre os ambientes informando em qual nó fixo ele se conectou em um determinado momento, permitindo assim o sistema de rastreamento.

Após o desenvolvimento e testes do aplicativo *Android* e dos serviços, e com todos os componentes do projeto ativados, constatou-se que o protótipo é funcional. A movimentação dos sensores móveis entre os sensores fixos gera as informações de proximidade desejadas, as quais podem ser acompanhadas através das telas do aplicativo, como a tela de monitoração de local apresentada na Figura 30. O cadastramento de nomes dos sensores e dos locais pode ser efetuada de maneira fácil, pois cada sensor que é ativado, assim que envia seu código de identificação para o servidor, logo é exibido no aplicativo e pode ser configurado com a correta descrição.

Como resultado obteve-se um sistema de rastreamento que permite, através do uso de um aplicativo para dispositivos baseados no sistema operacional *Android*, monitorar local ou remotamente, o posicionamento das crianças dentro do ambiente de uma creche ou de uma escola utilizando-se sensores através do contexto da internet das coisas.

## 5 COMENTÁRIOS FINAIS

Ter um pequeno “computador” nas mãos, como é o caso de um *smartphone* ou de um *tablet*, até há pouco tempo era um sonho, mas atualmente é uma realidade, e todas as coisas que se conectam a ele o tornam indispensável. A evolução da internet das coisas está ocorrendo em um momento certo, pois as pessoas estão cada dia mais conectadas com as redes em volta e com a *internet*. IoT é um conceito muito importante para o controle do homem sobre a máquina, facilidade almejada há algum tempo como pode-se verificar na literatura futurista e em diversos filmes de ficção científica.

Com base nesse documento, podem-se elaborar trabalhos futuros, como um portal *web* com funcionalidades equivalentes às do aplicativo *Android* desenvolvido, e até mesmo aplicativos para outras plataformas. Também podem-se criar novos monitoramentos dos sensores em tempo real, assim como relatórios históricos diversos, incluindo-se nesse contexto algumas análises estatísticas e gráficas. A visão de um mapa dos ambientes é uma opção que pode agregar valor ao sistema. Outra possibilidade é o desenvolvimento do projeto de um circuito específico para o MCU CC2650 e uma PCB (Placa de Circuito Impresso, do inglês *Printed Circuit Board*) com tamanho reduzido, visando a aplicação prática de todo o sistema de rastreamento.

Os estudos e experimentos realizados no desenvolvimento deste trabalho apresentaram resultados satisfatórios, os quais podem ser aproveitados para a construção de diversas soluções dentro de empresas, escolas, hospitais e ambientes públicos, por exemplo, que tenham necessidade de monitoramento da localização de objetos, como malotes, móveis, equipamentos, de pessoas, como estudantes, crianças, médicos, e funcionários, e até mesmo de animais em geral.

O protótipo apresentado atende aos objetivos propostos para o presente trabalho. Para essa realização foram utilizados conhecimentos adquiridos no curso de especialização, como desenvolvimento em diferentes linguagens de programação, construção e manutenção de bases de dados, protocolos de redes de dados, redes de sensores sem fio e protocolos de baixo consumo de energia, como MQTT e CoAP, sistema operacional *Linux*, e toda a gama de conceitos aplicada na tecnologia de internet das coisas.

## REFERÊNCIAS

- CARDOSO, A. **O que é o MariaDB?** 2015. Disponível em: <<https://www.bravulink.com.br/o-que-e-o-mariadb>>. Acesso em: 29 jan. 2019.
- CETIC. **6LBR.** 2012. Disponível em: <<https://www.cetic.be/6LBR>>. Acesso em: 27 out. 2018.
- DEBILL, E. **Module Counts.** 2019. Disponível em: <<http://www.modulecounts.com>>. Acesso em: 10 jan. 2019.
- EMBEDDED. **How to set up a 6LoWPAN network.** 2012. Disponível em: <<https://www.embedded.com/electronics-blogs/embedded-cloud-talkers/4236873/How-to-setup-a-6LoWPAN-network>>. Acesso em: 29 nov. 2018.
- FILHO, M. F. **Internet das Coisas.** Palhoça: Unisul Virtual, 2016.
- GARTNER. **5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018.** 2018. Disponível em: <<https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018>>. Acesso em: 24 nov. 2018.
- KANTAR. **World Panel.** 2019. Disponível em: <<https://www.kantarworldpanel.com/global/smartphone-os-market-share>>. Acesso em: 15 jan. 2019.
- MAGRANI, E. **A Internet das Coisas.** Rio de Janeiro: FGV Editora, 2018.
- PULICE, C. **O Brasil está pronto para um plano de Internet das Coisas?** 2017. Disponível em: <<https://exame.abril.com.br/tecnologia/o-brasil-esta-pronto-para-um-plano-de-internet-das-coisas>>. Acesso em: 22 out. 2018.
- RASPBERRY. **Raspberry Pi 3 Model B.** 2016. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b>>. Acesso em: 24 out. 2018.
- TEXAS. **CC2650 SimpleLink multi-standard 2.4 GHz ultra-low power wireless MCU.** 2016. Disponível em: <<http://www.ti.com/product/CC2650>>. Acesso em: 31 out. 2018.
- TEXAS. **SimpleLink™ Bluetooth low energy/Multi-standard SensorTag.** 2016. Disponível em: <<http://www.ti.com/tool/CC2650STK>>. Acesso em: 22 out. 2018.
- TEXAS. **SimpleLink™ CC2650 wireless MCU LaunchPad™ Development Kit.** 2016. Disponível em: <<http://www.ti.com/tool/LAUNCHXL-CC2650>>. Acesso em: 22 out. 2018.

YUAN, M. **Conhecendo o MQTT**. 2017. Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Acesso em: 28 out. 2018.