UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUÍS HENRIQUE PUHL DE SOUZA

HABILITANDO UM PRÉDIO A LOCALIZAR CONTEXTUALMENTE DISPOSITIVOS UTILIZANDO REDES SEM FIO

BAURU
2016

LUÍS HENRIQUE PUHL DE SOUZA

HABILITANDO UM PRÉDIO A LOCALIZAR CONTEXTUALMENTE DISPOSITIVOS UTILIZANDO REDES SEM FIO

Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação apresentado ao Departamento de Computação da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho" – UNESP, Câmpus de Bauru.
Orientador: Prof. Dr. Eduardo Martins Morgado

BAURU
2016

Luís Henrique Puhl de Souza
Habilitando um Prédio a Localizar Contextualmente Dispositivos utilizando Redes Sem Fio/ Luís Henrique Puhl de Souza. – Bauru, 2016-
41 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Eduardo Martins Morgado

Monografia (Trabalho de Conclusão de Curso) –
Universidade Estadual Paulista “Júlio de Mesquita Filho”
Faculdade de Ciências - Campus Bauru
Departamento de Computação , 2016.

1. Localização. 2. Raspberry Pi. 3. Internet das Coisas. 4. Contexto I. Prof. Dr. Eduardo Martins Morgado. II. Universidade Estadual Paulista "Júlio de Mesquita Filho". III. Faculdade de Ciências. IV. Habilitando um Prédio a Localizar Contextualmente Dispositivos utilizando Redes Sem Fio

LUÍS HENRIQUE PUHL DE SOUZA

HABILITANDO UM PRÉDIO A LOCALIZAR CONTEXTUALMENTE DISPOSITIVOS UTILIZANDO REDES SEM FIO

Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação apresentado ao Departamento de Computação da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho" – UNESP, Câmpus de Bauru.

Aprovado em 13/02/2017.

BANCA EXAMINADORA

Prof. Dr. Eduardo Martins Morgado
Orientador

Profa. Dra. Simone das G. D. Prado

Profa. Dra. Roberta Spolon

RESUMO

ABSTRACT

SUMÁRIO

1 INTRODUÇÃO

Nos recentes anos de 2014 a 2016, a Internet das Coisas (IoT - *Internet of Things*) vem tomando o foco das atenções de empresas e entusiastas de Tecnologia da Informação (DZONE, 2015) e, como é esperado que uma quantia total de 6,4 bilhões de dispositivos conectados exista até o final de 2016 (GARTNER, 2015) e entre 26 bilhões (GARTNER, 2014) e 50 bilhões até 2020 com até 250 novas coisas conectando-se por segundo (Cisco Blog, 2013), as empresas líderes do segmento já incluem IoT como uma de suas áreas de atuação (IBM, 2016; ARM, 2016; MICROSOFT, 2016; INTEL, 2016; ORACLE, 2016; GOOGLE, 2016; AMAZON, 2016a).

Todo este movimento no mercado é justificado pelo baixo custo dos pequenos dispositivos computacionais (FUNDATION, 2015; ESP8266.NET, 2016) e grandes serviços na nuvem (KAUFMANN; DOLAN, 2015; AMAZON, 2016b). Este baixo custo possibilita a computação ubíqua descrita por Weiser (1999) que nesta obra é entendida como *"computação onipresente diluída no dia-a-dia"*. Também nesta obra, esta onipresença diluída no plano de fundo é a base e a consequência para o conceito e área de IoT, sendo esta a realizadora da computação ubíqua.

Uma vez contextualizado o mercado e a oportunidade de implementação da computação ubíqua, percebe-se a necessidade de dar aos elementos cotidianos (coisas) a capacidade info-computacional, tornando-os sensores e atuadores conectados, unicamente identificáveis e acessíveis através da rede mundial de computadores (LEMOS, 2013; KRANENBURG, 2012). Para tanto, este trabalho propõe a construção de um sensor que, através da rede, identifica e localiza contextualmente os elementos cotidianos.

1.1 Problema

Tamanha quantidade de dispositivos conectados pouco acrescenta na vida diária se humanos ou coisas não puderem simplesmente se encontrar. Tanto em ambiente real quanto virtual é essencial o contato e conhecimento entre as partes envolvidas para que uma interação complexa seja executada. Portanto, para que uma aplicação IoT funcione corretamente, o conhecimento do contexto em que todos os interessados, sejam coisas ou pessoas, estão inseridos é indispensável. Para a maioria das aplicações, a informação contextual de maior relevância é a localização física.

Em situações em que a localização contextual é essencial para o bom funcionamento de uma aplicação IoT, destaca-se a necessidade da coleta desta informação através de sensores ativos sempre que a aplicação requisite a ciência deste contexto em suas tomadas de decisão. E, também, para que outros (sistemas, pessoas e coisas) saibam a localização de qualquer dispositivo ao qual têm interesse de interagir, distribuindo efetivamente essa informação coletada sobre o contexto com todos os que se encontram envolvidos no mesmo contexto.

Um exemplo desta necessidade de localização de dispositivos dentro de um prédio seria um profissional saber onde está o dispositivo em seu local de trabalho, seja ele um vendedor e seu *tablet* para demostrar um produto fora de estoque em uma loja ou um médico e seu equipamento portátil.

1.1.1 Sobre Sistemas de Posicionamento

Sistemas de posicionamento (PS - *Positioning System*) são geralmente constituídos de um Ponto Origem Global escolhido (*O*) e um conjunto não vazio de Pontos de Referência (RP - *Reference Point*) cuja localização global em relação ao *O* é conhecida com uma certa precisão quando o sistema é construído - precisão interna. Então, para o usuário, um sistema de posicionamento oferece como resultado uma precisão de visualização menor que a sua precisão interna. Um PS tem interesse em determinar a posição de um ponto móvel (MU - *Mobile User*). Essa localização é feita encontrando um conjunto de distâncias associadas a cada um dos RPs em um sub-conjunto com dimensão variável de acordo com o método utilizado. Feito isso, é possível utilizar modelos matemáticos para, a partir das distâncias, encontrar uma posição do MU em relação aos RPs e uma nova transformação é aplicada para encontrar a posição relativa ao *O*.

Uma das maneiras de classificar PSs é entre as classes de Auto Posicionamento e Posicionamento Remoto. Os de Auto Posicionamento contém no MU todo aparato necessário para medir a distância dos RPs e calcular a posição em relação a *O*. Já os classificados como de Posicionamento Remoto tem o mínimo necessário na MU e todo o trabalho de cálculo de distância e posição global é feito nos RPs ou em uma unidade coordenadora

destes.

Para PSs eletrônicos baseados em radio-frequência (RF - *Radio Frequency*), geralmente, utilizam-se dois componentes básicos, Transmissores e Receptores, os quais assume-se que ao menos um destes está no RP e ao menos um outro no MU. Para calcular a distância entre MU e RP, utiliza-se as propriedades da comunicação por RF como tempo de chegada (TOA - *Time Of Arrival*), diferencial de tempo de chegada (TDOA - *Time Difference Of Arrival*) e ângulo de chegada de sinal (AOA - *Angle Of Arrival*).

Para maior precisão, é comum a utilização de múltiplas RPs geralmente com o número mínimo igual ao número de dimensões espaciais que deseja-se calcular. Nota que para sistemas distribuídos a sincronização de relógios é um problema intrínseco, então é fundamental que o tempo seja contado como dimensão.

Os sistemas classificados como “Sistema de Navegação Global por Satélite” (GNSS - *Global Navigation Satellite System*), como o tradicional estadunidense Sistema de Posicionamento Global (GPS - *Global Positioning System*), utilizam a técnica em que o dispositivo móvel contém o receptor e os transmissores são fixos em satélites na órbita terrestre (DJUKNIC; RICHTON, 2001). Devido a posição e número de satélites, o GPS e seus correlatos estão sempre presentes do ponto de vista de um observador da superfície terrestre, sendo para este tipo de usuário um sistema ubíquo.

Entretanto, a força do sinal GNSS não é suficiente para penetrar a maioria dos prédios, uma vez que estes dependem de visão direta (LOS - *Line-Of-Sight*) entre os satélites e o receptor. A reflexão do sinal muitas vezes permite a leitura em ambientes fechados, porém o cálculo da posição não será confiável (CHEN; KOTZ, 2000). Portanto, apesar da ubiquidade dos GNSSs em ambientes abertos, são necessárias soluções diferentes para obter um Sistema de Posicionamento para Ambientes Fechados (IPS - *Indoor Positioning System*), sendo a ubiquidade deste essencial para conquistar o mesmo nível de confiança trazido pelos GNSSs.

Para implementar este IPS, propõem-se o uso de tecnologias já implantadas em dispositivos móveis e essenciais para o funcionamento dos mesmos, especialmente as de camadas de comunicação, que são ubíquas no ambiente dos dispositivos móveis, como *Wi-Fi* (padrão *IEEE 802.11*) e *Bluetooth* (padrão *Bluetooth SIG*), para que os objetos conectados os quais tem-se interesse de encontrar a localização contextual não necessitem de modificações.

Outros protocolos de comunicação sem fio ubíquos existem (em especial, o celulares em todas as gerações 2G, 3G, 4G), porém não oferecem a mesma flexibilidade por trabalharem em uma faixa de radio-frequência licenciada e por questões de propriedade da rede que serão abordadas na seção de Localização Contextual desta mesma obra.

De forma semelhante, existem protocolos mais flexíveis (nas faixas não licenciadas

como *NFC*, infra-vermelho, *ZigBee* ou *SIGFOX*), porém estes não estão presentes na maiora dos aparelhos utilizados, tanto globalmente quanto localmente, removendo a característica da forma de comunicação ubíquoa que é foco deste trabalho.

Devido às restrições anteriores, justifica-se o foco nas tecnologias de comunicação *Wi-Fi* e *Bluetooth*, porém trabalhar com as duas tecnologias simultaneamente é um problema complexo por si só, então, a escolha de um ou outro deve ser feita, apesar de a nível global serem de equivalente importância para esta obra (ambas tem mesma importância e presença no mercado atual, permitem flexibilidade por possuírem protocolos conhecidos por todos em frequências livres de licenciamento, dentro da área de cobertura que são de nosso interesse e o usuário final já ser o proprietário da rede local criada). Esta escolha toma um único parâmetro como decisivo que é a observação do ambiente de teste do protótipo onde pouco existe o uso de *Bluetooth* que reflete o costume local de mantê-lo desligado em comparação com *Wi-Fi* que está sempre ligado em todos dispositivos, conectando estes diretamente à Internet. Portanto *Wi-Fi* é a tecnologia de maior interesse por pequena margem.

1.2 Motivação

A proposta deste trabalho é criar um ambiente consciente, onde o contexto locativo oriundo do posicionamento remoto de cada dispositivo móvel é administrado e divulgado pelo prédio conectado ao invés da auto-localização do aparelho, pois:

a) Uma vez encontrada a localização, é mais fácil propagar esta informação do ambiente para o aparelho em comparação ao autoposicionamento, pois a negociação entre o ambiente e o aparelho é nula quando o primeiro contém a informação- o ambiente sempre disponibilizará uma informação coletada para o gerador desta informação;

b) Pode-se lidar com grande heterogeneidade de dispositivos, uma vez que cada um deles não precisa se adaptar para cada mudança de ambiente;

c) Este tipo de informação já é contida nos históricos de cada Ponto de Acesso *Wi-Fi* (AP - *Access Point*), porém:

   – Geralmente sem uso - poucas são as aplicações que usam a localização obtida pelo AP;
   – Com granularidade insuficiente para uso em aplicações contextualizadas;
   – geralmente não disponibilizada pelos APs.

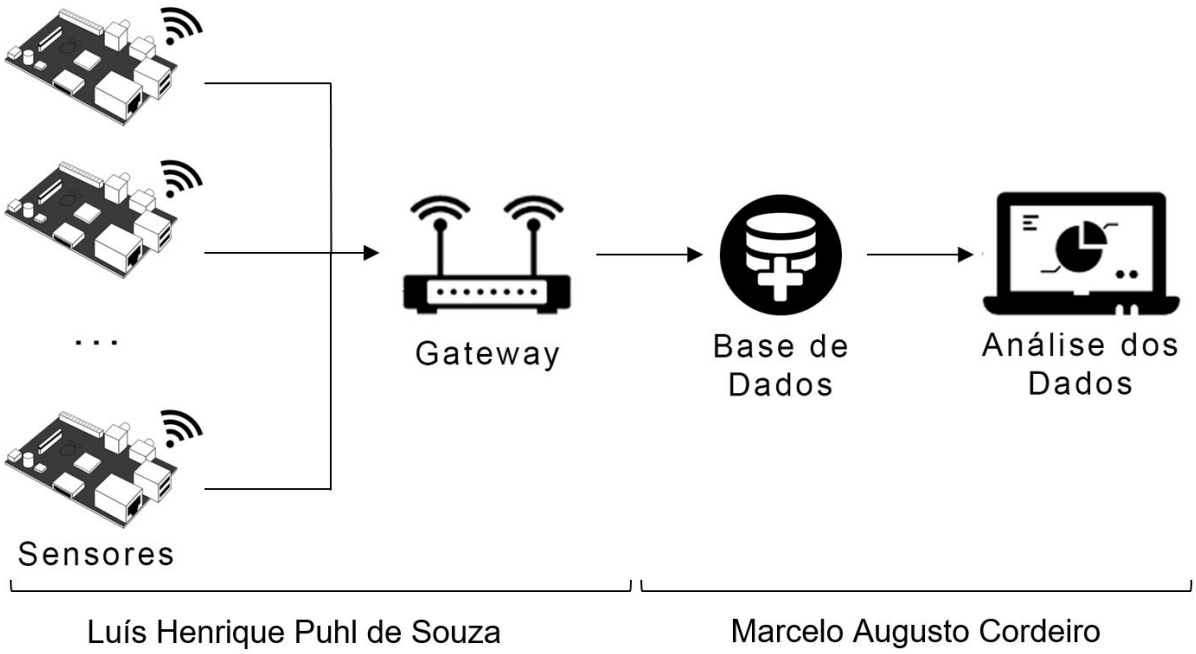
d) Uma vez instalado um PS deste gênero, a quantia de dispositivos que ele pode localizar fica limitada apenas pela rede física anteriormente instalada;

e) Economia de hardware quando menos é exigido de cada dispositivo móvel.

Nota-se também que mesmo com a quantidade prevista de 5 dispositivos IoT por pessoa em média, estes seriam beneficiados sempre que utilizados no ambiente conectado proposto.

A Figura 1 apresenta a arquitetura simplificada de uma aplicação IoT, e no detalhe inferior a relação deste projeto com o do aluno Marcelo Augusto Cordeiro, também do Bacharelado de Ciências da Computação, que é também membro do ambiente de testes LTIA (Laboratório de Tecnologia da Informação Aplicada) da Unesp de Bauru e do mesmo edital para obter o título de bacharel.

Figura 1 – Modelo das camadas



Fonte: Marcelo Augusto Cordeiro (**??**)

1.3 Objetivos

1.3.1 Objetivo Geral

Considerando características locais, propõem-se a construção de uma aplicação para localizar contextualmente dispositivos dentro de um prédio piloto e avaliar sua precisão.

Além da aplicação, é objetivo definir o custo do projeto piloto, incluindo esforço de pesquisa assim como definir um custo para replicação deste localizador contextual em outros prédios utilizando como fonte de ferramentas e recursos o mercado local.

1.3.2 Objetivos Específicos

a) Estabelecer o estado da arte sobre a desenvolvimento de aplicações IoT;

b) Identificar desafios locais para o desenvolvimento;

c) Identificar provedores de serviços, dispositivos e ferramentas para o desenvolvimento;

d) Construir sensores de identificação e localização (distância) de dispositivos cuja comunicação seja baseada em *Wi-Fi*;

e) Posicionar estes sensores;

f) Construir um dispositivo agregador de informações dos sensores (*gateway*) e sua interface web (MQTT - *MQ Telemetry Transport*);

g) Estimar o custo total do projeto piloto incluindo esforço de pesquisa;

h) Estimar o custo de replicação da aplicação em outros prédios utilizando fontes do mercado local.

2 FUNDAMENTAÇÃO TEÓRICA

Para conceituar, fundamentar e dar suporte teórico ao presente trabalho apresentam-se neste capítulo os tópicos e definições dos segmentos: IoT, localização contextual de dispositivos e localização baseada em redes sem fio.

2.1 Internet das coisas (IoT)

Uma das primeiras aplicações e definições de IoT foi feita simultaneamente por Kevin Ashton em 1999 para a *Procter & Gamble* (P&G) (ASHTON, 2009) e pelo laboratório Auto-ID Labs no Instituto de Tecnologia de Massachusetts (MIT - *Massachusetts Institute of Technology*) utilizando identificação por radio-frequência (RFID - *radio-frequency identification*) (ATZORI; IERA; MORABITO, 2010; FRIEDEMANN; FLOERKEMEIR, 2011). Desde então, a IoT cresceu ultrapassando o escopo da tecnologia RFID, porém sempre com as premissas de “uma infraestrutura global para a Sociedade da Informação, habilitando serviços avançados através da interconexão de coisas (físicas e virtuais) baseadas em tecnologias, existentes e evolutivas, de informação e comunicação” descrita por Wortmann e Flüchter (2015 apud International Telecommunication Union, 2012, p. 1, grifo e tradução nossa).

Hoje em dia, quase qualquer tecnologia de comunicação acessível a computadores pode ser utilizada como meio de comunicação entre dispositivos IoT, tornando o RFID mais uma, porém de grande importância, tecnologia info-comunicacional a disposição das coisas para sua conexão. Esta gama de tecnologias possibilita uma variedade equivalente de coisas conectadas. Se a coisa pode usar de uma tecnologia de conexão, considerando suas restrições de volume, custo e utilidade, muito provavelmente vai fazê-lo gerando ao menos uma identidade virtual representando seu objeto físico e seus atributos. Esta identidade virtual e atributos virtuais serão expostos para todos indivíduos, humanos ou coisas, que lhe forem convenientes de qualquer lugar do universo virtual, fazendo efetivamente parte da Internet.

2.2 Localização contextual de dispositivos

Em ciência da computação, os termos *"Contexto"* e *"Consciência de Contexto"* expressam uma ideia recente estudada nos campos de inteligência artificial e ciência cognitiva desde 1991. O tema "Contexto"ainda é considerado atual e promissor a ponto de mudar o cenário de negócios nos próximos 10 anos, mas sem definição simples. Tamanha é a falta de uma definição geral que realmente funcione para casos reais que existe uma

proposta de definir o termo utilizando uma nova metodologia de pesquisa holística através de mineração e agrupamento de texto advindo de publicações científicas (PASCALAU; NALEPA; KLUZA, 2013).

Mesmo sem uma definição permanente em vista, utilizou-se o que é considerado estado da arte para o termo *"Contexto"* que foi introduzido por Dey e Abowd (1999) e reforçado por Dey (2000):

> "Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação."
>
> Dey e Abowd (1999, p. 3) Tradução Nossa.

2.2.1 Localização contextual

Das informações contextuais que uma aplicação de cliente móvel pode obter, a localização é uma das mais importantes. Ajudar pessoas a navegar por mapas, encontrar objetos e pessoas com os quais tem interesse de interagir é sem dúvida uma boa meta a ser alcançada com a coleta da localização do cliente (BELLAVISTA; KÜPPER; HELAL, 2008).

Na categoria de Serviços Baseados em Localização (LBS - *Location-Based Services*) existem duas gerações. A primeira orientada a conteúdo que falhou, pois a informação de localização era armazenada pela rede (que geralmetne era administrada por uma empresa de telecomunicações), podendo até ser vendida pelo provedor a terceiros, causando a sensação de *Spam* (conteúdo não solicitado) no usuário final ao receber conteúdo desta provedora. Já na segunda geração, a posse da informação foi movida para o cliente móvel, deixando a cargo do usuário escolher se ela seria compartilhada e com quem. Esta mudança trouxe maior engajamento do usuário, resultando numa maior aceitação dessa geração (BELLAVISTA; KÜPPER; HELAL, 2008).

Ao contrário das técnicas atuais, neste trabalho os humanos ou tomadores de decisão não estarão em posse do cliente móvel, e sim em posse do prédio. Portanto, a mesma informação, sem degradação em sua importância, passará a ser coletada e armazenada pelo provedor da rede como nos LBSs de primeira geração. Esta decisão garante o foco no usuário uma vez que este mudou, antes ele detinha um cliente móvel, agora ele detem múltiplos. Isso torna a detenção do todo (coisas dentro do prédio) mais precioso do que o das partes (os clientes móveis) além da mudança da propriedade da rede para o usuário final, na comparação celular *versus Wi-Fi*.

Uma vez encontrada a localização de um dispositivo, metadados sobre o prédio são mesclados formando um conjunto rico contextualmente do ponto de vista da aplicação IoT Prédio como fornecedora principal dos dados para a Internet e, portanto, seus

usuários detentores. Essa riqueza é garantida com metadados sobre o dispositivo (identificação, nome, histórico, carecterísticas) e sobre o prédio (ex.: mapa, estrutura de salas, humanos responsáveis e lista de equipamentos) que trazem possibilidades de extração de informação importantes para os detentores deste prédio e seu conteúdo. Esta capacidade do prédio deve-se pelo papel de coordenador de informações e controlador de meta-informações semelhante ao Coordenador em uma aplicação na arquitetura Modelo-Apresentação-Adaptador-Controlador-Coordenador (MPACC - *Model-PresentationAdapter-Controller-Coordinator*) proposto por Román e Campbell (2001).

2.2.2 Contexto de um dispositivo em um prédio

Para metadados agregados à informação de posição pelo prédio defini-se que, para uma aplicação IoT, o modelo de divulgação tem de conter além da posição do dispositivo informação sobre este (nome, histórico), informação da estrutura do prédio, ligação entre a estrutura do prédio e a localização do dispositivo e informação sobre o estado do prédio.

Este modelo visa prover fácil mineração e reutilização de informações por terceiros que é medida pela disponibilidade e relacionamento das informações providas. Essa métrica também será utilizada para avaliar o projeto.

Este foco em reusabilidade vem da definição de Web Semântica (*Semantic Web*) e de uma de suas realizadoras, a Ligação de Dados (*Linked Data*), que sugerem o uso de um formato padrão além de ser acessível e gerenciável pelas ferramentas de exploração. Desta forma a Web de Dados (*Web of Data*) é construída opondo uma simples coleção de dados (BIZER; HEATH; BERNERS-LEE, 2009).

2.3 Localização baseada em redes sem fio

Um sistema de posicionamento pode ser baseado em técnicas *n-lateração?* de distâncias adquiridas com a medição de características eletromagnéticas (ex.: potência de sinal) e dos protocolos (ex.: Tempo de chegada) que já foram explorados anteriormente (ABUSUBAIH; RATHKE; WOLISZ, 2007; BAHILLO et al., 2009; FELDMANN et al., 2003).

Portanto, os sensores seguem as especificações de *WiFi IEEE 802.11* (CROW et al., 1997) e técnicas definidas para *Bluetooth Low Energy (BLE)* (HOSSAIN; SOH, 2007) devido a semelhança da área de cobertura (até 100 metros, geralmente utilizado até 20 metros) e frequência (no caso de 2.4GHz).

Para construir estes sensores uma plataforma de hardware adequada é necessária, para esta escolheu-se o Raspberry Pi (VUJOVIC et al., 2014; VUJOVIĆ; MAKSIMOVIĆ, 2015) que já foi provado funcional no caso de Localização através *Wi-Fi* por Ferreira (2016) especialmente a sua versão 3 que adiciona a capacidade de sensor *Wi-Fi* e *Bluetooth* em

sua placa principal sem necessidade de adaptadores externos destacando ainda mais sua escolha (RASPBERRY PI FOUNDATION, 2016). Em adição, na construção dos sensores foi testada a plataforma ESP8266 bem como outras alternativas que demonstraram afinidade com essas características.

2.4 Trabalhos correlatos

Neste subcapítulo, apresentaremos alguns projetos semelhantes em objetivo ao daqui proposto e que motivaram a construção do sensor resultante deste trabalho.

2.4.1 Zebra

A Zebra é um empresa estadunidense que fabrica e vende tecnologia de marcação, rastreamento e impressão por computador. Dentre os seus produtos, estão: computadores móveis, RFID, software, impressoras, tablets, leitores de códigos de barras, kiosks interativos, entre outros. Já na área de serviços, a empresa oferece desde o planejamento até a execução de projetos.

A Zebra realizou um estudo (Global Shopper Study) que indicou que os varejistas apostaram em recursos online que podem aumentar o envolvimento e fidelidade do consumidor, além, claro, do volume de vendas. Segundo o mesmo estudo, 51compradores tem um forte interesse em serviços baseados em localização e *Wi-Fi* em lojas para cupons *mobile*, mapas de compras e receber assistência. Além disso, 64mais itens se receberem um serviço melhor e mais atenção dos vendedores, enquanto mais da metade prefere que os varejistas usem a tecnologia para criar experiência de compra mais eficiente.

A empresa possui o projeto MPact que é um *indoor location* que unifica *Wi-Fi* e *Bluetooth*. Ele fornece a localização do consumidor em três níves: presença, zona e posição. Com estas informações é possível saber sobre o indíviduo: quem é, onde está, quanto tempo fica em certas áreas e quais produtos está comprando. Esta tecnologia pode ser implementada independente do ambiente, através do *Wi-Fi*, ou do microposicionamento através do *Bluetooth*. Com a união dessas duas plataformas é possível saber o tempo exato e posição exata de onde alguém está.

Em 2016, a empresa implantou no Shopping Cidade Jardim, em São Paulo, uma rede LAN/WAN sem fio de alta velocidade, com a tecnologia MPact que proporciona aos seus clientes acesso gratuito ao *Wi-Fi*, juntamente com uma experiência de compra mais personalizada. Funciona assim, o consumidor assim que possível acessa o *Wi-Fi* do shopping que pede o *login* no Facebook ou Google. Assim que o *login* é feito, o MPact fornece visibilidade instantânea ao operador do shopping em relação aos locais dos compradores no shopping e habilita os operadores a enviar saudações pessoais, oferecer ofertas especiais ou fornecer instruções passo-a-passo para um venda ou promoção

específica. Hoje, o Cidade Jardim conta com 180 lojas numa área de 46.000 metros quadrados.

Segundo Claudio Bessa, diretor de markegin digital do shopping, este tipo de serviço fornece um excelente experiência para os compradores devido a alta velocidade do *Wi-Fi* e a cobertura. Além disso, segundo ele, os operadores e varejistas podem entender melhor o comportamento do consumidores, pois eles podem saber que parte do corredor ou de uma loja o cliente está, quanto tempo permanece na frente de uma loja e quais produtos mais vendem. Oferecer este tipo de serviço é uma maneira de ganhar e manter consumidores, crescer no número de satisfações e ajudar a monitorar os pontos de venda.

2.4.2 Outras tentativas

Outras tentativas bem sucedidas de localizar dispositivos móveis através da rede *Wi-Fi* são o caso do *Decimeter Level Localization with a Single Access Point* (**??**) e do *Radio Frequency Time-of-flight Distance Measurement for low-cost Wireless Sensor localization* (**??**).

No primeiro exemplo, uma placa de rede sem fio Intel 5300 com 3 antenas calcula o tempo de vôo entre uma antena e outra além de utilizar técnicas de mitigação de multi-caminho, mitigação de identificação de pacote entre outras características importantes do protocolo *Wi-fi*, como a frequência e sincronização de clientes para alcançar até 10 centímetros de precisão. Neste caso, as três antenas atuam como três sensores independentes justa posicionados para executar trilateração. Esta aplicação é implementada em uma placa instalada em um computador moderno através do barramento PCI Express com sistema operacional Ubuntu. Ela possui habilidade de injetar pacotes na rede o que difere muito das arquiteturas embarcadas que normalmente são encontradas no ambiente de IoT.

O segundo exemplo de aplicação bem sucedida se utiliza de modificações no hardware de um ponto de acesso do padrão 802.15.4 e alcança precisões de 1 a 3 metros. Este protocolo é mais encontrado em comunicações de longa distância ou sensíveis a uso de energia que são frequentes em aplicações embarcadas.

3 MÉTODO DE PESQUISA

Abordagens para medir distâncias através de redes sem fio *Wi-Fi* (BAHILLO et al., 2009) e *Bluetooth* já existem e, propor novas maneiras não é o foco deste trabalho. Utilizando essas técnicas, constitui-se uma rede de nós sensores colaborativos fixos no ambiente onde deseja-se obter a localização dos dispositivos. As informações de distância são compartilhadas entre os nós para maior precisão da informação.

Para a implementação, pretende-se utilizar os *softwares* de maior destaque recentemente nos ramos de comunicação de baixa energia (*MQTT*), serviços *Web* para geolocalização (*Google Maps*) e publicação (*NodeJS*), além de *softwares* para medição da distância sem interferir na comuncação (*Sniffing*) e das plataformas de *hardware* disponíveis e recomendadas para IoT com capacidade *Wi-Fi* (*Raspberry Pi 3* e *ESP-8266*).

Mesmo com a grande quantidade de dispositivos já conectados são poucos os documentos descrevendo boas práticas para concepção, construção e manutenção de aplicações IoT, especialmente sobre os cuidados tomados quanto a segurança e análise de custos para a implementação e manutenção.

Além disso, a falta de referências neste sentido é agravada quando considera-se a implementação no interior do estado de São Paulo. Nesta região, poucas são as organizações atualizadas neste tema, levando a uma falta enorme de conteúdo escrito na linguagem local além de serviços e produtos disponíveis para construção de uma plataforma completa e competitiva na região.

Devido a falta de conteúdo e instrução, utiliza-se prototipagem ágil neste projeto, uma vez que esta metodologia de desenvolvimento é recomendada para projetos cujas especificações e definições não são claras, demandando muitas modificações das mesmas durante a etapa de execução. Esse método entra em contraste com metodologias clássicas, como a cascata, que apesar de previsíveis, não reagem bem a ambientes de extrema incerteza.

Mais especificamente, utiliza-se uma variante da metodologia *Scrum* (JAMES, 2016) que foi adaptada para o projeto. Nela, foram executadas iterações de uma semana em que a cada iteração, uma nova versão melhorada do produto completo (*hardware*, *software*, documentação e resultados) foi feita.

Dentro de cada iteração, as camadas da aplicação IoT serão escolhidas, implementadas, justificadas e avaliadas, sendo parte do processo registrado sob forma de vídeo (Youtube) e imagens que podem ser encontrados em "Apêndices".

A cada iteração, cumpriu-se parte ou todo de cada objetivo proposto no trabalho,

levando o projeto gradualmente para um estágio de completude. Cada iteração teve como foco os objetivos a seguir, sendo seus resultados utilizados para tomar e justificar decisões durante a execução do projeto bem como servir de posterior documentação. Os objetivos de cada iteração são:

a) Escolha de provedores de serviços, dispositivos e ferramentas para o desenvolvimento;

b) Construir, avaliar, testar e manter os sensores;

c) Construir o dispositivo agregador e sua API;

d) Estimar o custo total do projeto piloto;

e) Estimar o custo de replicação;

f) Identificar os desafios para o desenvolvimento.

Desta forma, espera-se garantir a liberdade necessária para o projeto ser executado com sucesso, mesmo no ambiente de incerteza no qual o mercado local de IoT encontra-se, cumprindo as premissas de funcionamento, manutenção e segurança que são grande importância para os interessados na área.

4 PLATAFORMAS

Para a localização com os resíduos de comunicação *WiFi* são necessários sensores que possam capturar estes resíduos e processar qualquer informação capturada pelo sensor deste trabalho. Esta plataforma de sensor pode ser construída com qualquer plataforma computacional capaz de ser programada com comunicação *WiFi*, porém o *hardware* de *WiFi* e seu *software* controlador deve permitir o Modo Promíscuo.

Este Modo Promíscuo (*promiscuous mode*) é definindo pela capacidade de uma Placa Adaptadora de Rede *WiFi* (*Network Interface Card - NIC*) receber e interpretar todos os pacotes que trafegam em uma rede ou em todas as redes que estão em seu alcance, independentemente do destinatário do pacote. Em seu fucionamento normal, uma *NIC* descarta todos os pacotes que não são destinados para ela o mais cedo possível, evitando reprocessamento de dados indesejáveis, por este motivo não são todas as *NICs* que permitem o Modo Promíscuo. Essa funcionalidade elimina a necessidade de *hardware* ou *software* em cada um dos dispositivos rastreados.

Neste sentido, elegeu-se duas plataformas de notável importância no mercado atual e notável facilidade de acesso para qualquer interessado na área. As plataformas testadas foram o microcomputador *Raspberry Pi* e o microcontrolador *ESP8266*. Ambos foram escolhidos pelo domínio do segmento de Prototipação e Faça Você Mesmo (*Do It Yourself - DIY*) dentro do campo de IoT. Outro líder de segmento, o *Arduino* foi prontamente descartado por não conter nativamente a habilidade de conectar-se à *Internet* sendo constantemte combinado com um dos escolhidos para ganhar esta habilidade, demonstrando claramente menor afinidade a este projeto em comparação aos seus igualmente famosos concorrentes.

Após escolhidas as plataformas de intersse alguns exemplares de cada uma delas foi adquirido para implementar a aplicação proposta. Neste sentido, serão apresentadas cada uma dessas plataformas quanto as suas especificações técnicas e aos produtos utilizados em conjunto para que elas pudessem funcionar e serem programadas e os motivos pela adoção ou não delas.

4.1 ESP8266

O ESP8266 é um SOC (*System On a Chip* - Sistema em um *Chip*), ou seja, é um chip com todos os componentes lógicos eletrônicos necessários e partes para um dado sistema em único cirtuito integrado. Este chip possui:

a) *Wifi* embutido com capacidade de 2,4 GHz (802.11 b/g/n);

b) 16 GPIOs (*general-purpose input/output*) incluindo interfaces I2U, SPI, UART,

entrada ADC, saída PWM;

c) Arquitetura *RISC* de 32 bits;

d) CPU que opera em 80 MHz, com possibilidade de operar em 160 MHz;

e) 64 KB de ROM para *boot*;

f) 64 KB de RAM para instruções;

g) 96 KB de RAM para dados;

h) Memória *Flash SPI* de 512 KB a 4 MB (dependente de módulo externo);

i) Núcleo baseado no *IP Diamand Standard LX3* da *Tensilica*.

Para o mercado de prototipação, fabricantes constroem placas de diferentes configurações com este chip como elemento central, os chamados módulos. Estes módulos usam o ESP8266 com diferenças perceptíveis, por exemplo, quantidade de pinos, dimensões físicas e alguns podem até operar de modo *standalone* (sem outro *hardware* de suporte como reguladores de tensão, conversores serial-USB) e especialmente a *Memória Flash SPI*. Neste trabalho, foram usados os módulos: ESP-01, LoLin, D1 mini e ESP-12F.

Figura X - Módulos ESP Fonte: Elaborada pelo autor

As diferentes especificações implicam em diferentes produtos e mercado para eles, isto resulta em diferentes preços em diferentes regiões.

Tabela 1 – Descrição de custos de módulos ESP8266

Módulo	N de pinos	Memória	Preço
ESP-01	8	1 MB	R$ 16,80
ESP-12F	22	4 MB	R$ 14,90
PCB (sem ESP-12F)	16	4 MB	R$ 3,45
D1 mini (ESP-12F)	16 + microUSB	4 MB	R$ 12,56*
LoLin (ESP-12F)	30 + microUSB	4 MB	R$ 35,87

Fonte: Produzido pelo autor.

Nota: * D1 mini (ESP-12F) foi adquirido do mercado chinês.

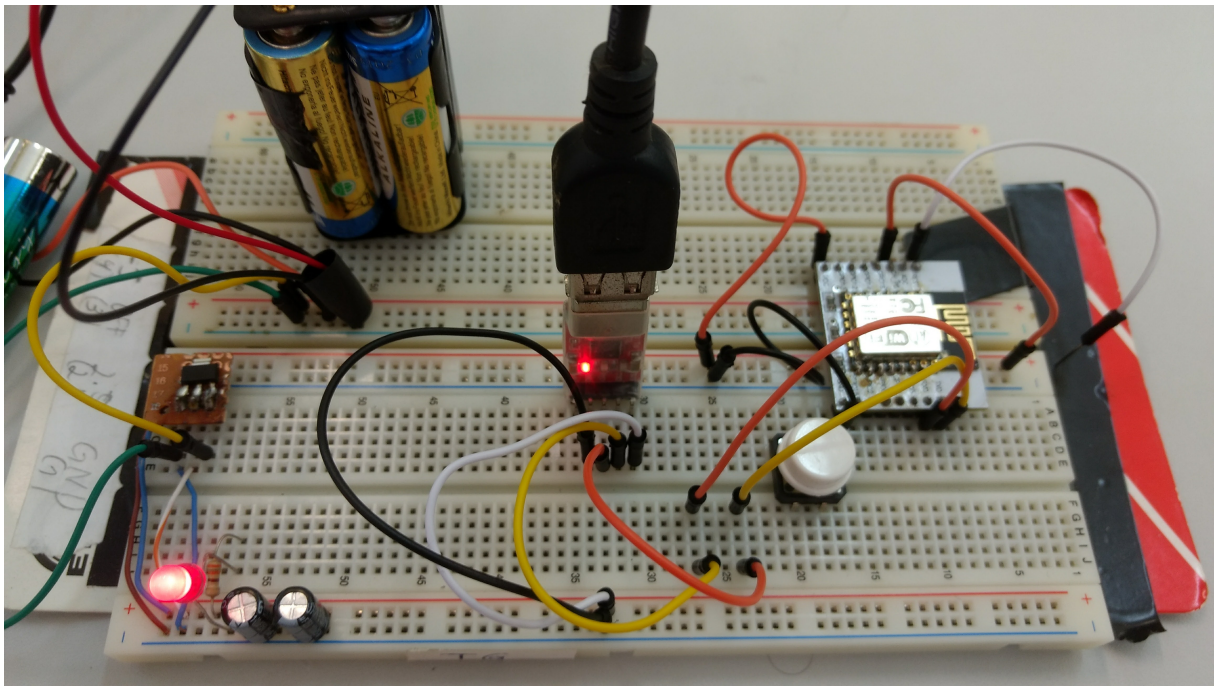
A escolha do ESP8266 como primeira tentativa devido o seu baixo custo e de tamanho reduzido. No exterior, ele pode ser encontrado por de USD $1.76 a 2.2 **??**), e no Brasil, em média, por R$ 15,00 **??**).

[Alibaba]:https://www.alibaba.com/product-detail/ESP-12F-Esp8266-Remote-Serial-Port$_6 0518607068.html?s = p$

Devido ao seu tamanho, ele é de fácil integração com demais dispositivos, bastando o uso de uma comunicação serial. Já sobre a comunidade, há inúmeros projetos DIY (em inglês *do it yourself*, em português "faça você mesmo") que ensinam a como construir e manipular projetos que envolvem diferentes módulos. Além disso, a empresa idealizadora e fabricante do chip, Espressif, disponibiliza no GitHub projetos com documentação e código aberto.

Para ligar um módulo ESP foram utilizadas formas diferentes. Quando o módulo possuía regulador de tensão *onboard*, utilizava-se o próprio conectado a uma porta USB. Quando o módulo não possuía tal, utilizava-se um circuito com fonte externa (pilhas ou USB) e um regulador de tensão conectados aos pinos 3V3 e GND. Depedendo da complexidade do circuito para ligar e ter acesso à serial do módulo, é necessário o uso de uma placa *breadboard*, como a imagem a seguir. Todo ESP precisa de um regulador de tensão de 3.3V. Para este trabalho, foi utilizado o regular AMS1117 3V3.

Figura 2 – ESP-12F com regulador tensão e serial

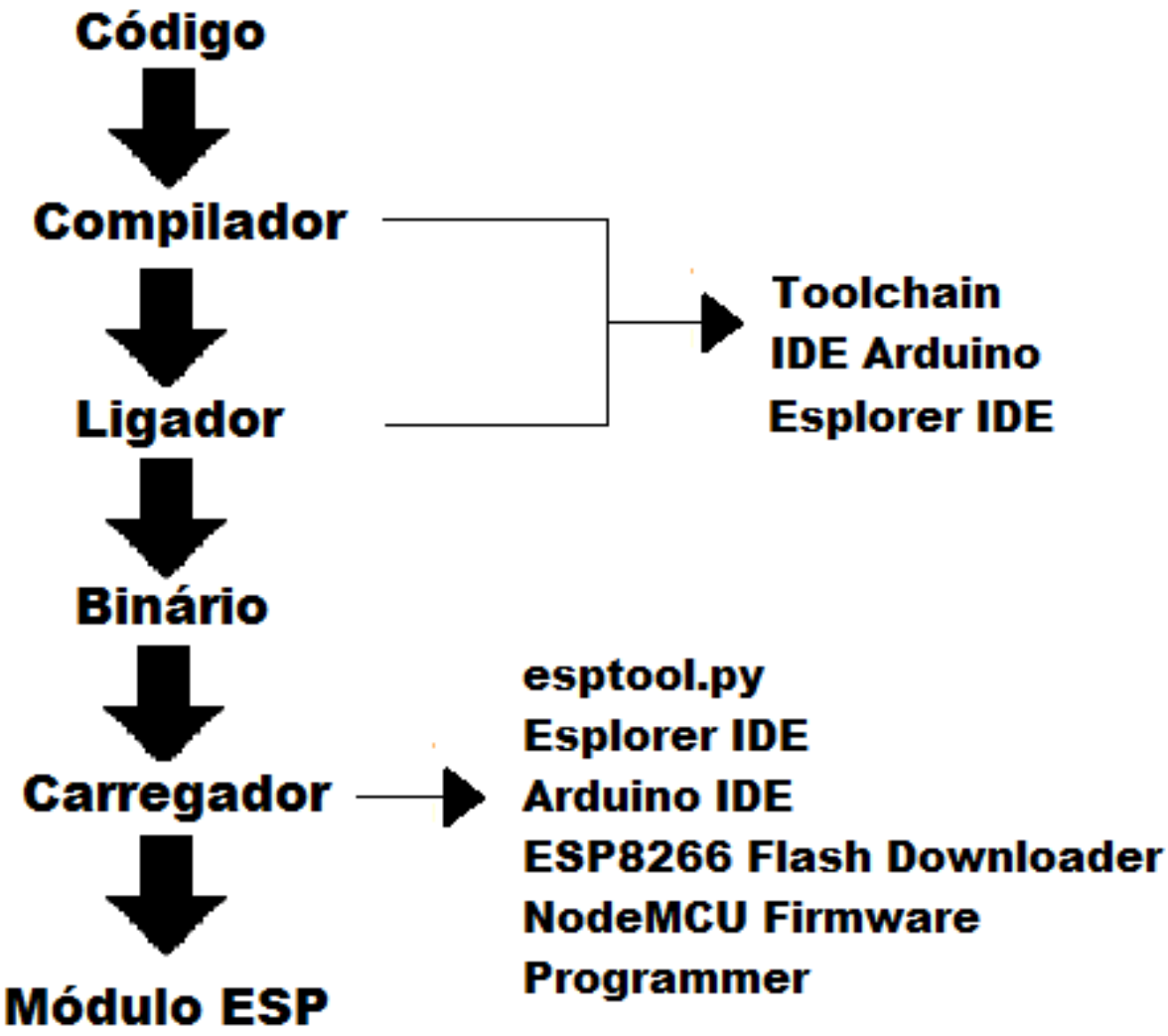


Fonte: Elaborada pelo autor

*Carregadores**

Todo código produzido em uma linguagem de programação é compilado por uma ferramenta e, então, carrega-se os arquivos binários para o ESP8266 através da serial, para que a execução do código seja iniciada. Na figura a seguir, é apresentado um modelo de caminho desde o código até chegar no módulo ESP e, também, a lista de carregadores usados.

Figura 3 – Modelo de processo de desenvolvimento e implantação com ESP8266



Fonte: Elaborada pelo autor

// mudar este paragrafo, simplificar

Todo código produzido é carregado para o módulo ESP através de seu barramento serial. Alguns modelos, como o LoLin e D1 mini, já apresentam conversor serial para micro USB. Para os que não possuem tal interface é necessário utilizar um conversor serial - USB. As imagens a seguir demonstram como é o acesso de alguns módulos utilizados. As GPIOs do ESP12F são acessadas somente através de placas de circuito impresso, então uma foi adquirida para a programação do mesmo. Dos conversores serial-USB adquiridos, o modelo CH340G não funcionou por não ter driver compatível com o Windows 10, então a saída foi utilizar o modelo CP2102.

Os modos de programação descritos nas seções a seguir tem por objetivo acessar o ponto da API de hardware do ESP8266 onde os pacotes destinados a outros dispositivos

são descartados,e habilitar o modo promíscuo.

Com a *IDE Arduino*, a programação foi feita de primeiro modo através de um *firmware* genérico chamado AT. Este é um conjunto de instruções enviados via serial para o módulo ESP que permite configurá-lo. A *IDE Arduino* e o *Cool Term* possuem um emulador de terminal serial que aceita os comandos AT e os envia direto para a serial.Além disso, utilizou-se a linguagem C que foi compilada na *IDE Arduino* e enviada ao ESP.

Nenhuma dessas abordagens funcionou, pois nenhuma delas forneceu uma API que funcionasse a baixo nível suficiente para atingir o modo promíscuo do ESP, que é essencial para a descoberta de pacotes que trafegam entre dispositivos.

Figura 4 – Código em C compilado e implantado em um ESP8266

A esquerda comandos AT no emulador de serial da *Arduino IDE*. // A direita editor da *Arduino IDE* com código C // Abaixo em preto: processo de *upload* do firmware escrito em C // Fonte: Elaborada pelo autor

Toolchains

A segunda tentativa para a programação dos módulos escolhidos foi feita através de *toolchains* (conjunto de ferramentas para desenvolvimento de software) da empresa Espressif e de um usuário do Github, muito utilizado para projetos de ESPs, Paulo Sokolovsky (*pfalcon*). Ambas as *toolchains* são SDKs de código aberto. Os *scripts* foram feitos na linguagem C, compilados nessas SDKs e transferidos para os módulos ESP. O maior problema dessas SDKs foi a configuração delas. Elas requisitavam de uma versão

específica do Ubuntu que a máquina utilizada para fazer os códigos não suporta. Cogitou-se a possibilidade de fazer VMs, mas a máquina também não possui virtualização.

Conclusão sobre o ESP

Apesar do baixo custo e documentação da comunidade aberta, o ESP8266 não foi adotado como sensor, pois não foi possível colocá-lo em modo prosmícuo, essencial para detectar pacotes entre dispositivo e os pontos de acesso.

4.2 Raspberry Pi

Especificações técnicas

O Raspberry PI 3 Model B é um computador *single-board* (única placa) que tem o tamanho próximo ao de um cartão de crédito. Foi desenvolvido pela Raspberry Pi Foundation para promover o ensino da computação nas escolas. Este computador possui:

a) Antena *Wifi* embutida 802.11n;

b) *Bluetooth 4.1* e *Bluetooth Low Energy* (BLE);

c) 1 GB RAM;

d) Processador Gráfico *VideoCore IV 3D*;

e) ARM CPU de 1.2 GHz quad-core 64-bit.

f) 4 portas USB;

g) 40 pinos GPIOs;

h) Porta HDMI;

i) Porta *Megabit Ethernet*;

j) Saída de aúdio e vídeo 3.5 mm;

k) Interface para câmera (CSI) e monitor (DSI);

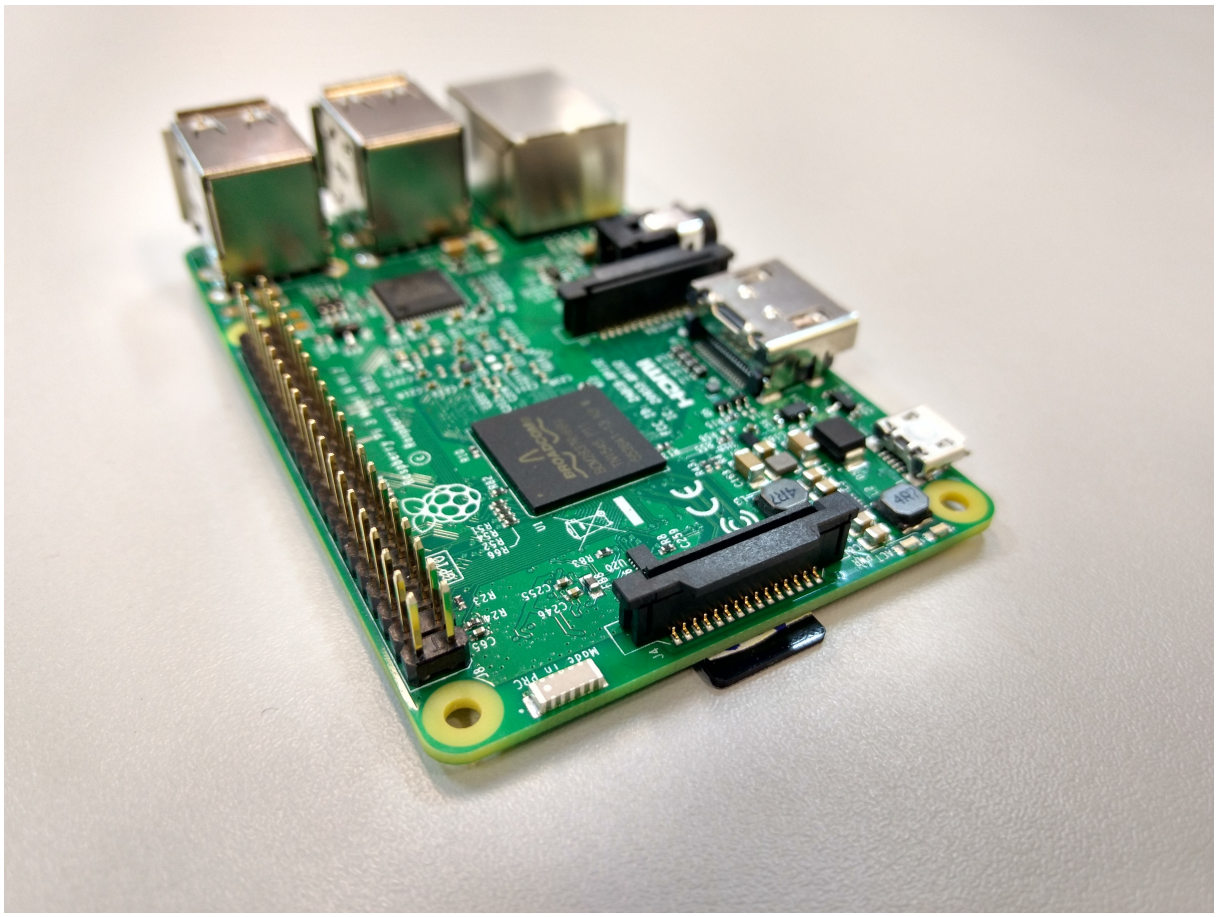
l) Leitor para cartão *micro SD*;

Segunda tentativa

Após da tentativa com o ESP8266, o Raspberry Pi foi escolhido como plataforma para hospedar o sensor. Em média, no exterior, o RPi (Raspberry Pi) é vendido por USD $ 35,00 (Raspberry Foundation) e, no Brasil, por R$ 250,00 (Mercado Livre). Apesar de não ser tão barato como o ESP8266, ele possui recursos que facilitam a programação e justificam seu preço.

A escolha deste dispositivo ocorreu principalmente devido a interface "amigável"com usuário, comunidade *open source* e perfomance de processamento. Como o Raspberry Pi "roda"um sistema operacional, o acesso a seus recursos e recursos externos possui

Figura 5 – Raspiberry Pi 3



Fonte: Elaborada pelo autor

maior nível de abstração, bastando apenas alguns comandos para acessá-los e estabelecer comunicação. Além de facilitar a programação destes dispositivos por esse aspecto, com o sistema operacional IDEs externas podem ser utilizadas.

Além deste recurso a nível de sistema, a comunidade e número de projetos "faça você mesmo"é muito maior que a do ESP8266, devido a sua simplicidade em conectar-se a um monitor e "sair"programando. Por ser um computador completo e, dependendo do projeto, não é necessário mais nada além do RPi, pois ele possui armazenamento, processamento e canal de comunicação (acesso à rede e portas USB).

Outro aspecto é o processamento, como dito anteriormente, este computador possui um poder de processamento para hospedar e processar inúmeras aplicações IoT.

Alimentação

O Raspberry é ligado por uma fonte de 2A, 5V e 10W através de uma entrada micro USB. Para ligá-la, foi adquirido uma fonte USB tipo A para iPad, pois além de poder desconectar o cabo da fonte, facilitando a manutenção, fornece a quantidade exata de

amperagem que o computador precisa. A primeira aquisição foi de um carregador de *smartphone* que não forneceu os amperes necessários.

Figura x.x - Carregador USB

Sistema Operacional

O Raspberry Pi comporta sistemas operacionais que são carregados através no micro cartão SD. Alguns exemplos de sistemas compatíveis: Archlinux, OpenELECE, Raspbian, Risc, Pidora, Kali Linux, Windows 10 IoT, entre outros. Para este trabalho, foi utilizado o Raspbian.

Figura X.X - Raspbian Fonte: Elaborada pelo autor.

Conclusão sobre Raspberry Pi

O Raspberry foi adotado como o sensor para detectar os dispositivos. O modo promíscuo conseguiu ser acessado através de adaptador/módulo USB Wifi. Mais detalhes sobre a construção e adoção deste computador serão apresentados no capítulo "Construção".

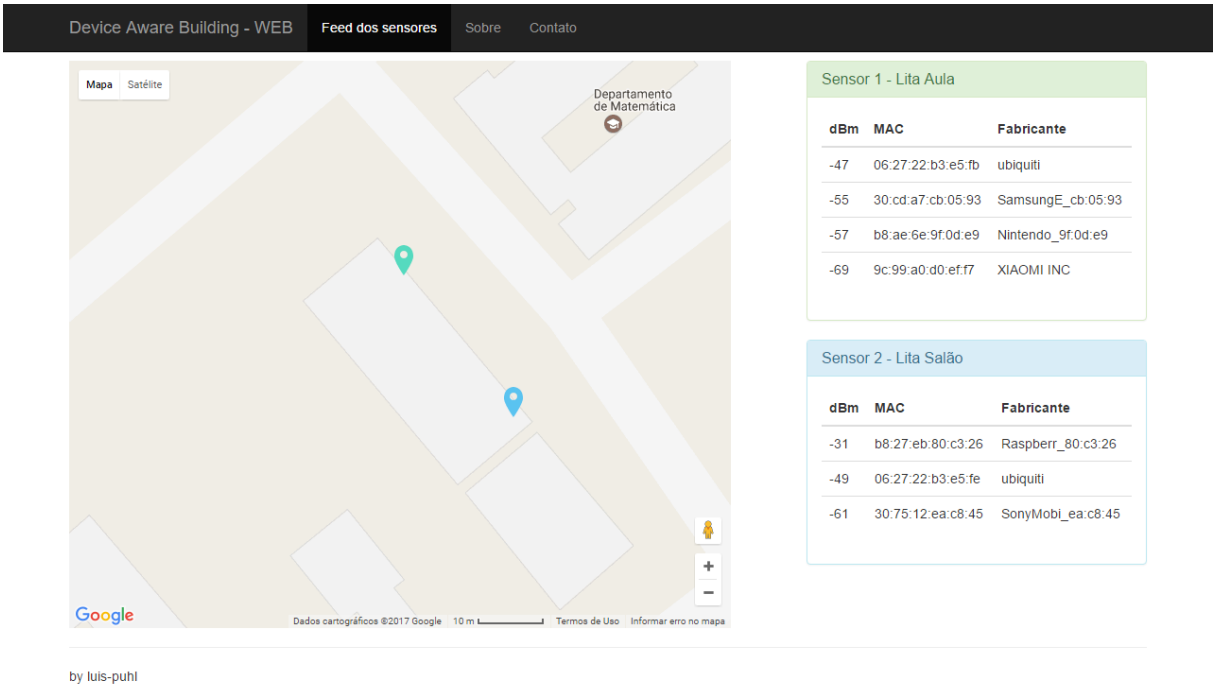
Comparativo RPi X ESP8266

Em comparação com o ESP8266, o Raspberry Pi compensou seu preço mais caro devido a facilidade de programação e acesso aos seus recursos e integração e acesso a recursos externos. Além disso, foi possível chegar ao modo promíscuo facilmente através do Bash e do sistema operacional. A seguir, uma tabela comparando as principais características do RPi e do módulo ESP12F.

Figura X.X - RPi x ESP12F Fonte: Elaborada pelo autor.

5 CONSTRUÇÃO

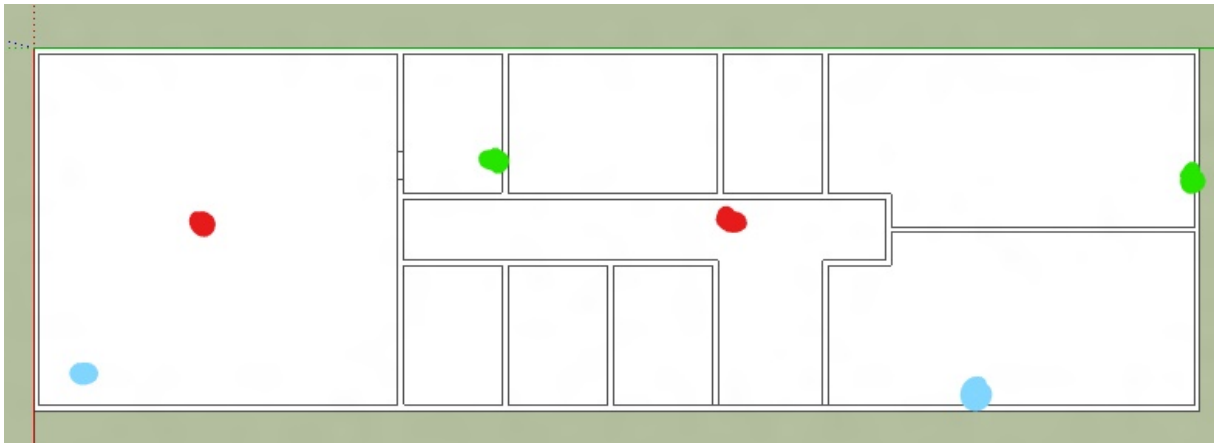
Figura 6 – Web APP



Fonte: Elaborada pelo autor

6 TESTES

Figura 7 – Ambiente de teste



Fonte: Elaborada pelo autor

Para entender o ambiente onde a aplicação foi desenvolvida e testada no âmbito de ruído wi-fi e pontos de referência wi-fi foi executada uma captura de teste durante a noite quando ninguém habitava o prédio prototipo.

Nesta captura dois sensores foram utilizados posicionados a menos de 10 centímetros de distância um do outro sobre uma mesa a um metro do chão a captura ocorreu de 2:50 até aproximadamente 11:25 totalizando aproximadamente 8 horas de captura

A captura foi executada com o comando

[language=bash] tshark -I -i wlan0 -T fields -E header=y -E quote=d -e wlan.sa -e wlan.sa$_{r}esolved-ewlan.ta-ewlan.ta_{r}esolved-eradiotap.dbm_{a}ntsignal-ewlan_{m}gt.ssid >> 2017-01-17--02-48--rpi-02.csv$

A análise de dados foi feita com a função *summary* da ferramenta *Ron's editor* e a filtragem com a função *Filter* da ferramenta *RecCsvEditor*

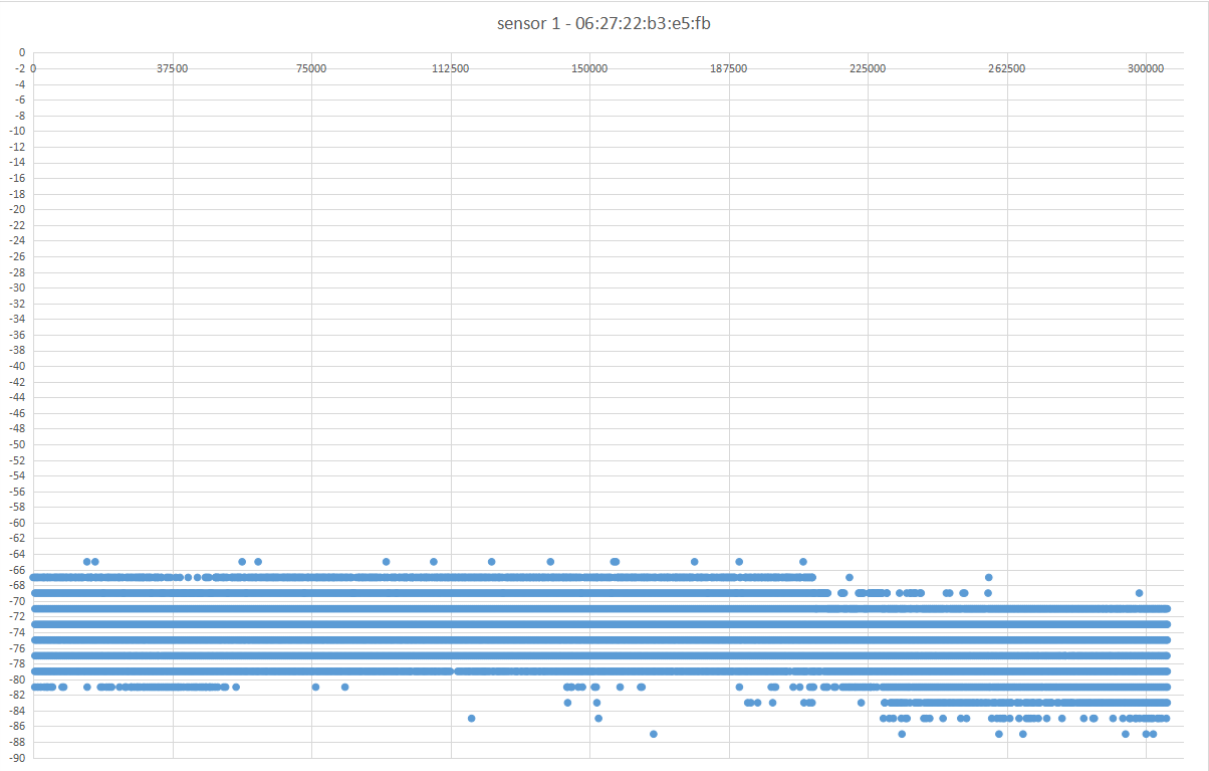
https://www.ronsplace.eu/Products/RonsEditor

http://recsveditor.sourceforge.net/

Para o primeiro sensor foram detidos 1 729 624 pacotes num arquivo de 155MB com 88 transmissores únicos dos quais se destacaram dois endereços MAC que são os pontos de acesso para rede wi-fi do laboratório.

// Modo de uso do tshark na aplicação,utilizar na construção Neste modo de uso os resultados são direcionadas para a saída padrão (stdout) do terminal e podem ser capturados

Figura 8 – Distribuição do dBm pelo tempo - 062722b3e5fb sensor 1



Fonte: Elaborada pelo autor

por outro programa no formato de valores separados por vírgula (csv). os campos escolhidos para captura são *wlan.sa*, *wlan.sa$_resolved*, *wlan.ta*, *wlan.ta$_resolved*, *radiotap.dbm$_antsignal$ e $*wlan_mgt.ssid*$

Teste relação de distância com smartphone como objetivo

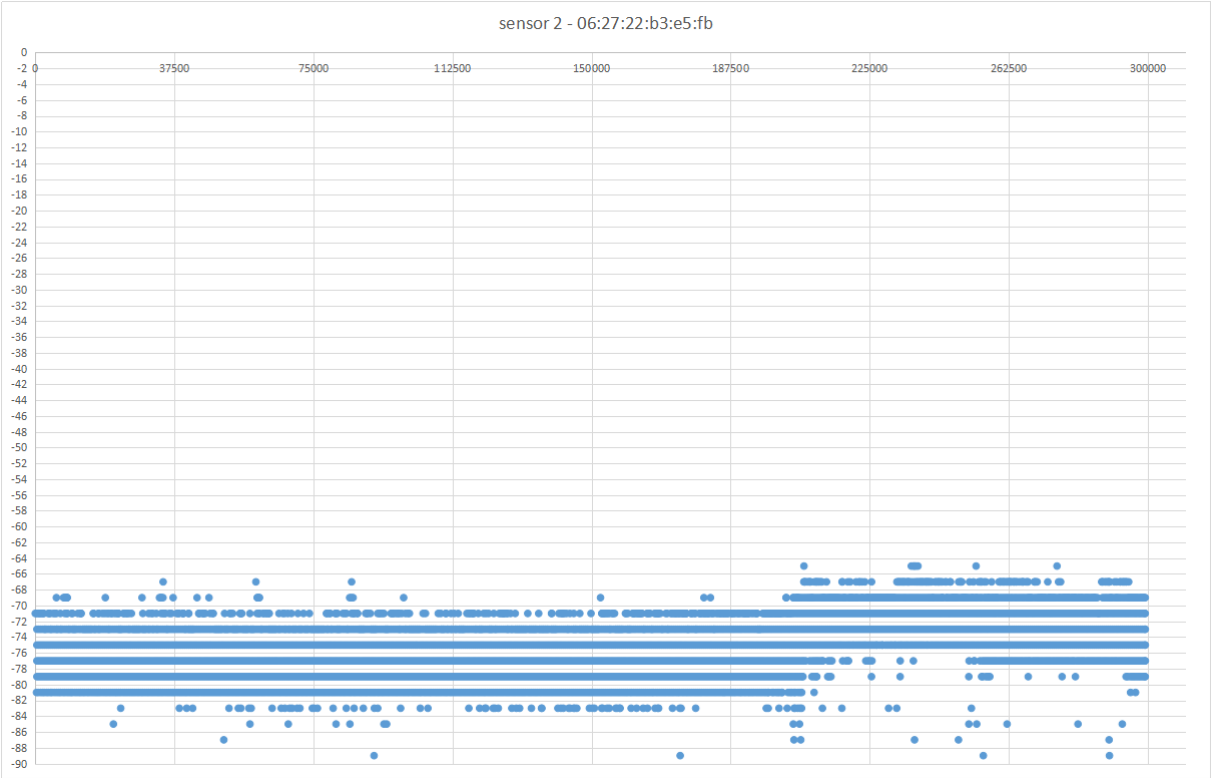
Para verificar a capacidade do sensores de localizar contextualmente um dispositivo móvel, smartphone, foi utilizado. Este foi posicionado em duas salas diferentes, Em cada uma das salas foi executada uma captura de 10 minutos. Para que houvesse tráfego na rede o dispositivo móvel foi configurado para receber um stream de vídeo no aplicativo Netflix.

A captura foi realizada utilizando a ferramenta TShark da mesma maneira que é utilizada no aplicativo. A descrição deste modo de operação pode ser encontrada no capítulo de construção.

$tshark - I - iwlan1 - Tfields - Eseparator\ =, -Equote\ =\ d - ewlan.sa - ewlan.sa_resolved - ewlan.ta - ewlan.ta_resolved - eradiotap.dbm_antsignal - ewlan_mgt.ssid >> sensor - 02 - distance - test - 01.csv$

Para o primeiro caso o dispositivo estava na mesma sala do sensor número 2 que é a maior sala do prédio com 12 metros de comprimento por 10 metros De largura. neste

Figura 9 – Distribuição do dBm pelo tempo - 062722b3e5fb sensor 2



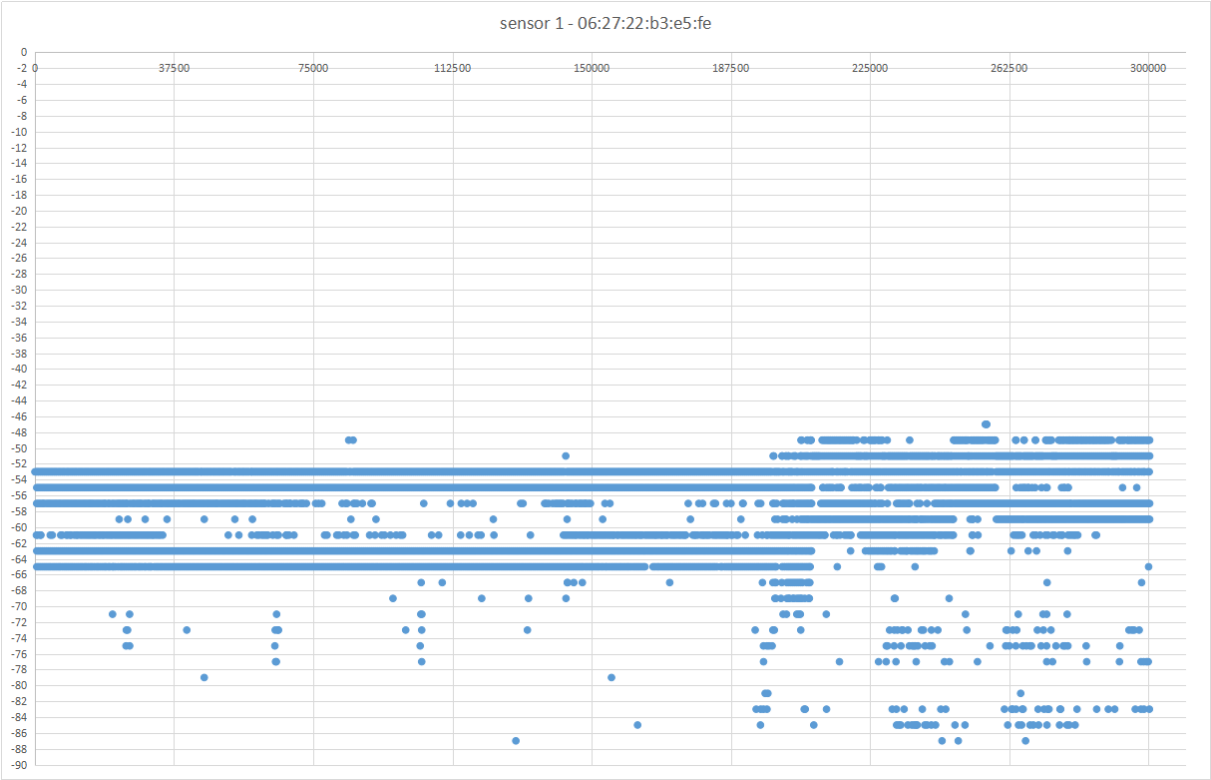
Fonte: Elaborada pelo autor

caso foram capturados 157736 pacotes totalizando 9.7 megabytes pelo sensor 1 21974 pacotes totalizando 1.9 megabytes de captura pelo sensor 2.

No segundo teste o dispositivo móvel estava posicionado no corredor fora da sala do sensor 1 e distante do sensor 2. neste teste o sensor 1 capturou 103555 pacotes totalizando 6.4 megabytes de captura e o sensor 2 capturou 22635 pacotes totalizando 2 megabytes de captura

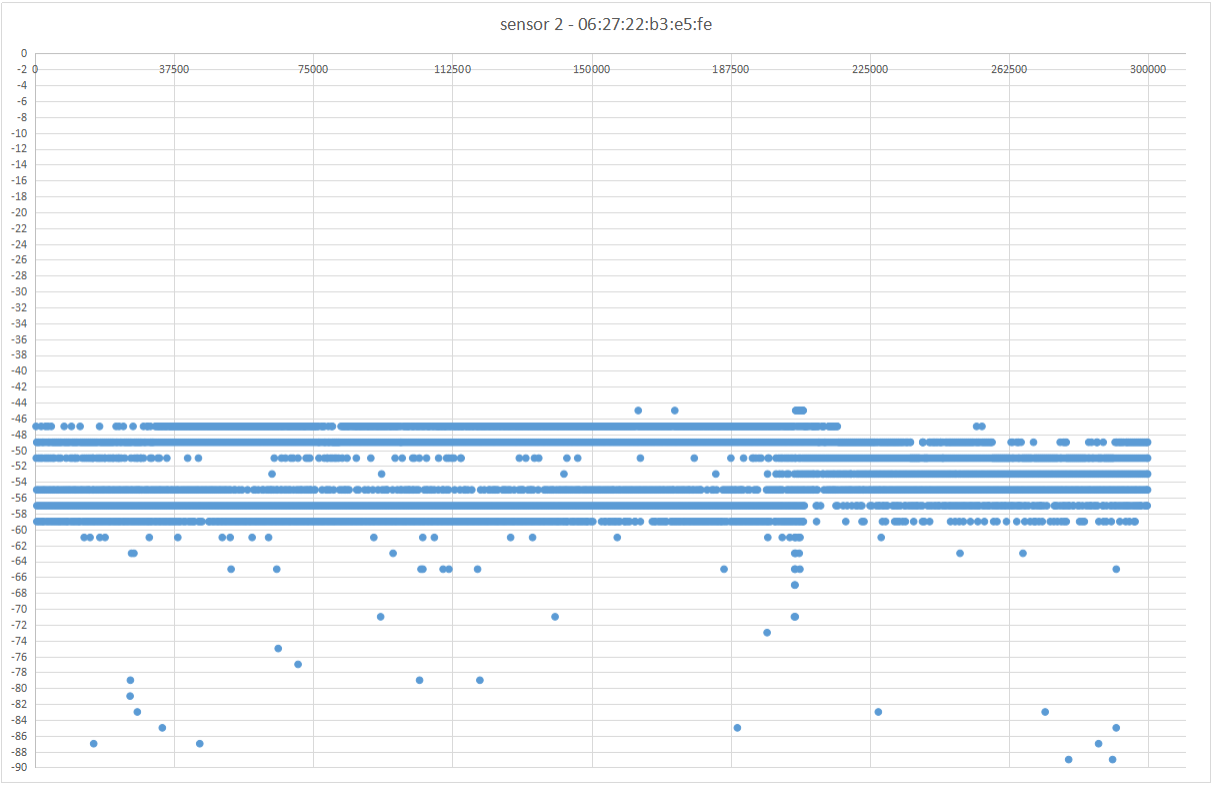
posteriormente os arquivos de captura foram analisados com a ferramenta *Ron's Editor* para que um sumário fosse construído.

Figura 10 – Distribuição do dBm pelo tempo - 062722b3e5fe sensor 1



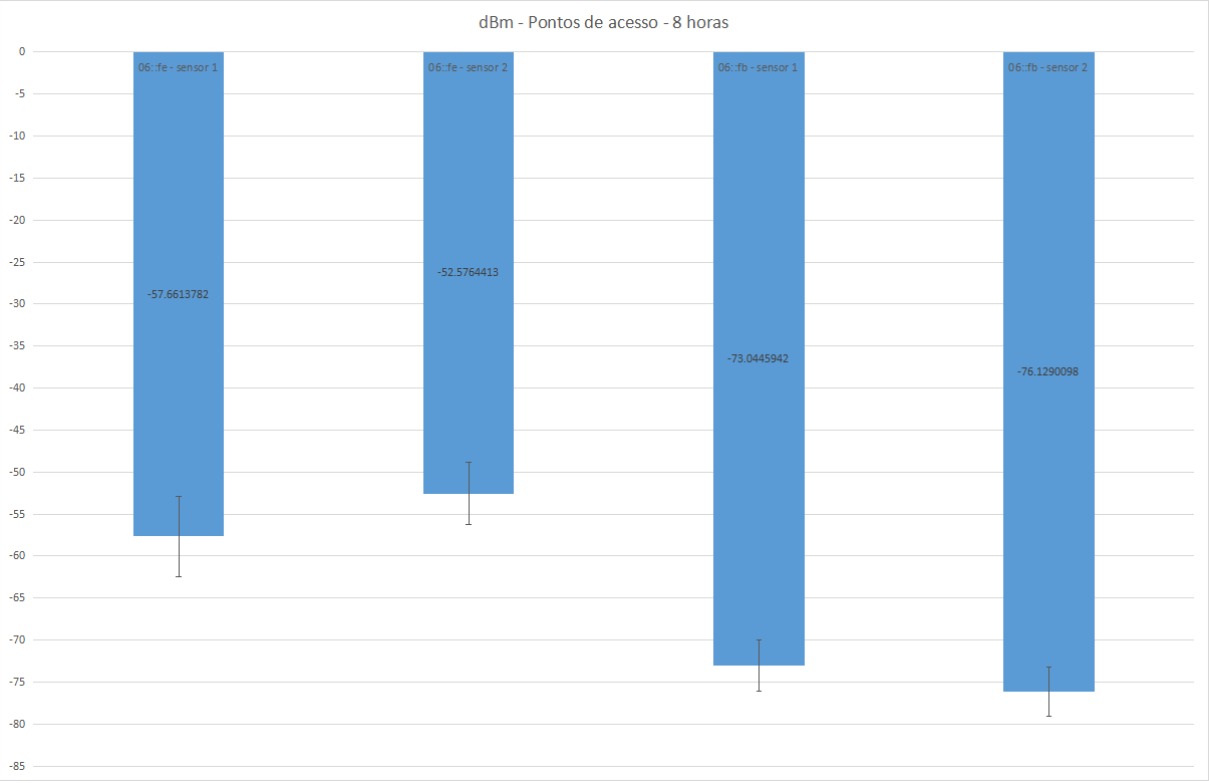
Fonte: Elaborada pelo autor

Figura 11 – Distribuição do dBm pelo tempo - 062722b3e5fe sensor 2



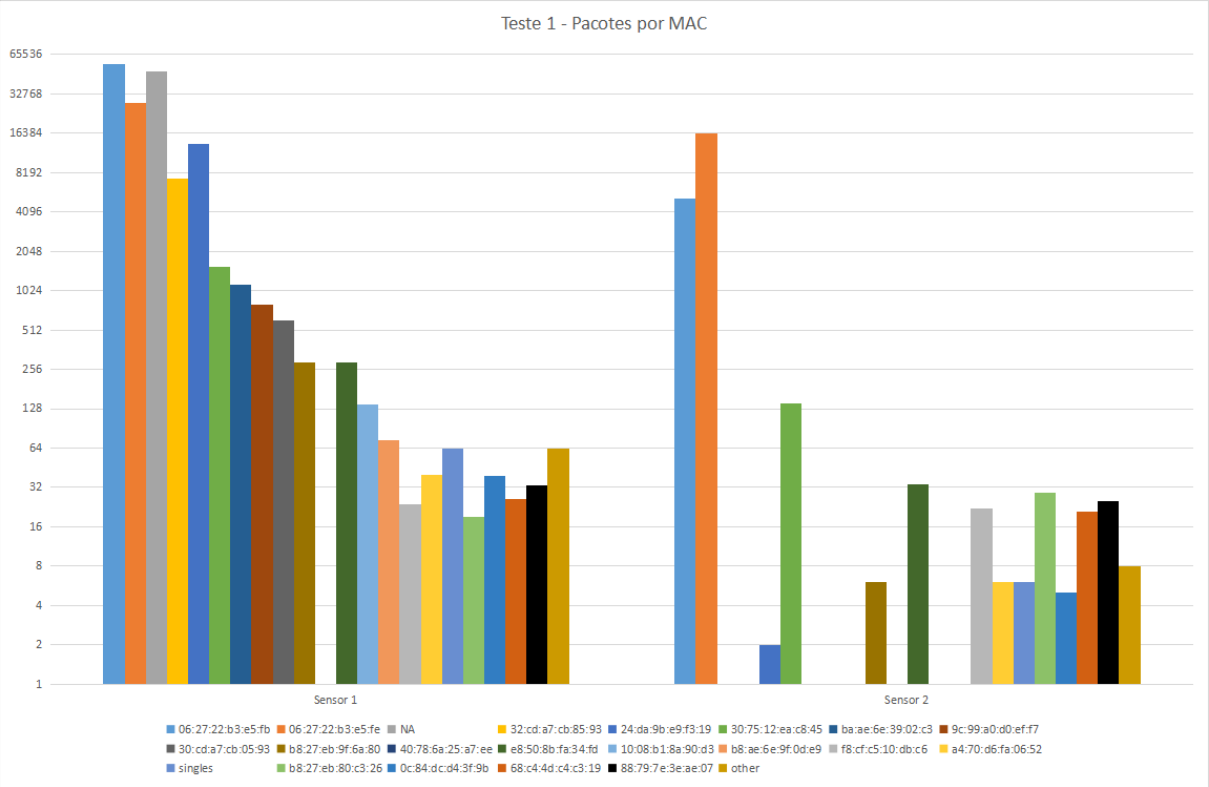
Fonte: Elaborada pelo autor

Figura 12 – dBm Pontos de acesso - Acumulado 8 horas



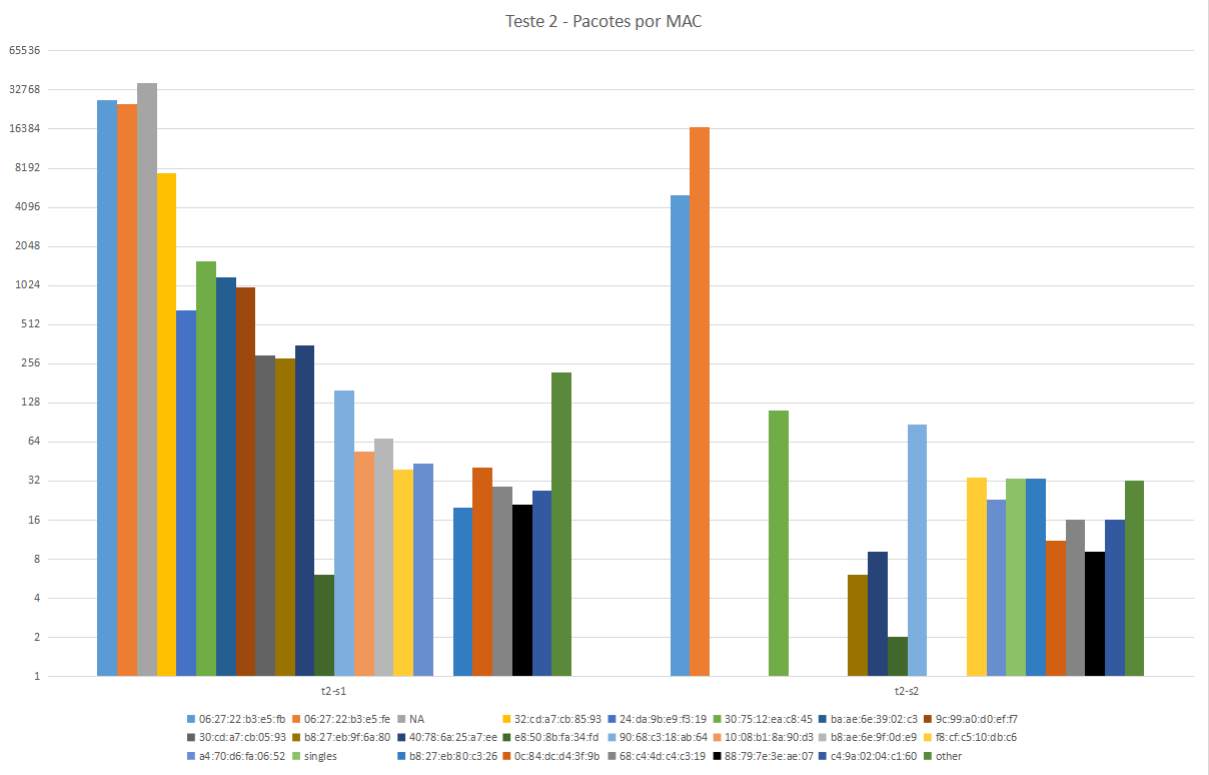
Fonte: Elaborada pelo autor

Figura 13 – Captura total (noise) - Teste 1



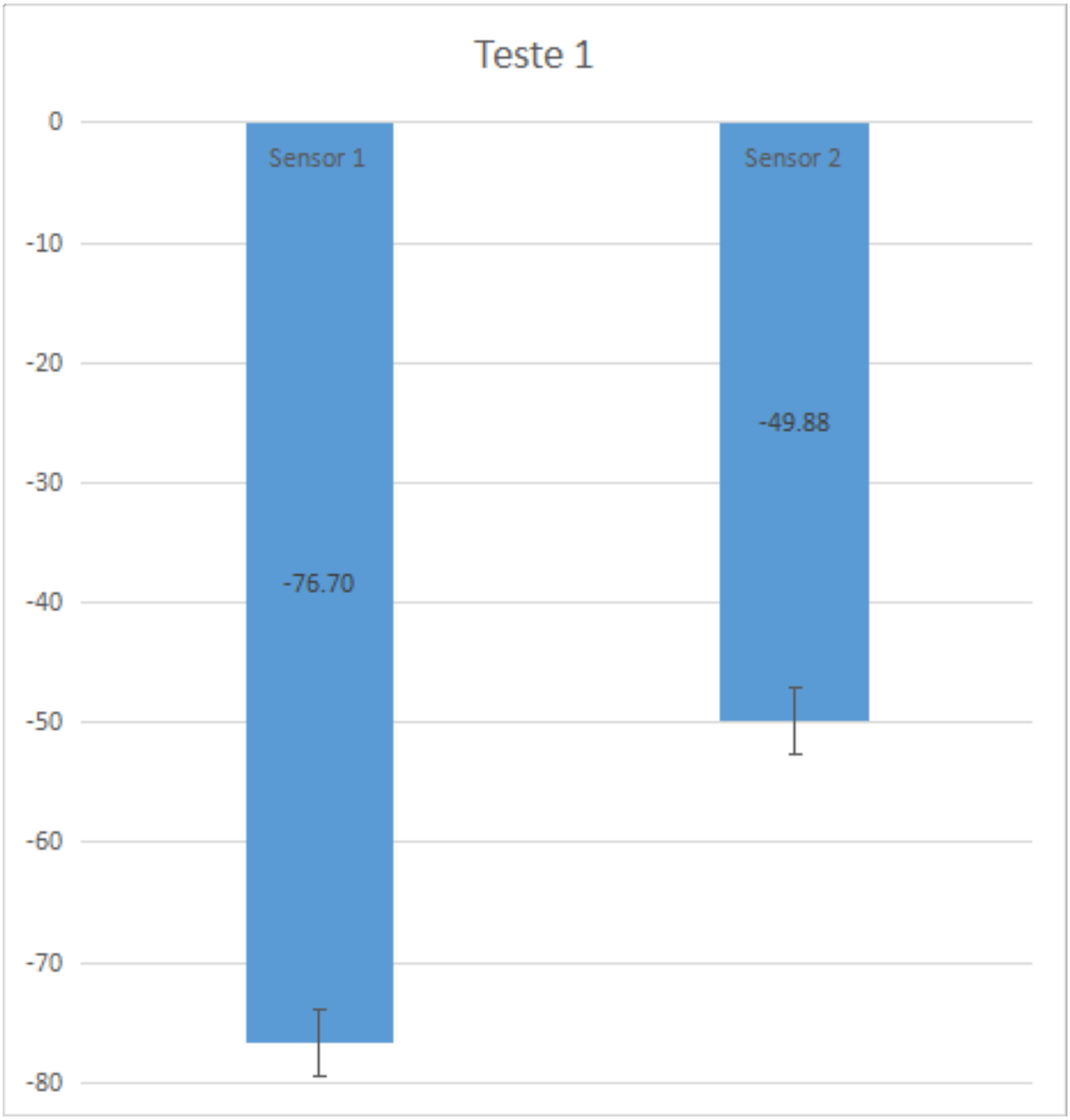
Fonte: Elaborada pelo autor

Figura 14 – Captura total (noise) - Teste 2



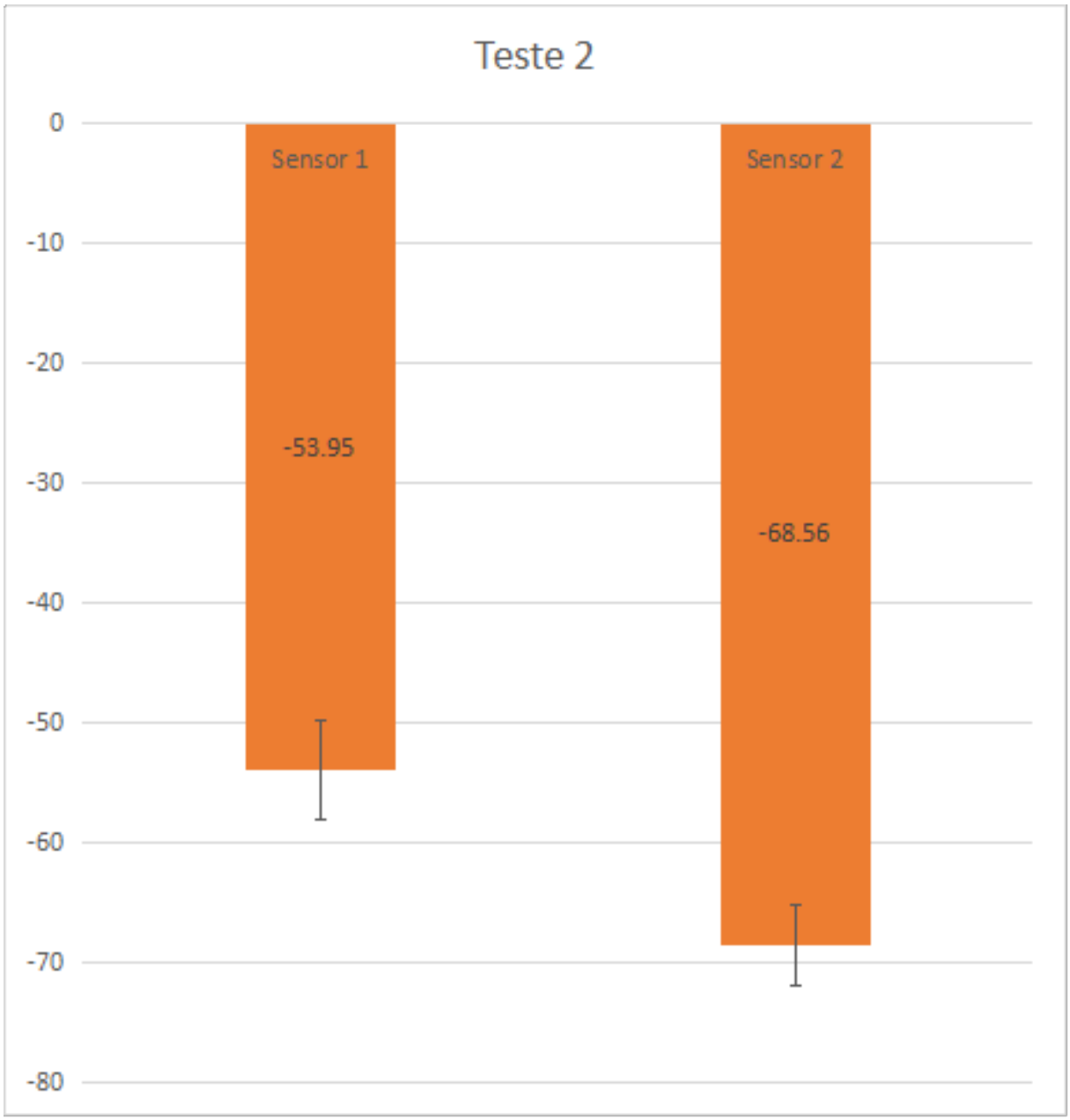
Fonte: Elaborada pelo autor

Figura 15 – dBm Motorola G4+ - Teste 1



Fonte: Elaborada pelo autor

Figura 16 – dBm Motorola G4+ - Teste 2



Fonte: Elaborada pelo autor

REFERÊNCIAS

ABUSUBAIH, M.; RATHKE, B.; WOLISZ, A. A Dual Distance Measurement Scheme for Indoor IEEE 802 . 11 Wireless Local Area Networks *. In: *2007 9th IFIP International Conference on Mobile Wireless Communications Networks*. [s.n.], 2007. p. 121–125. ISBN 9781424417209. ISSN 978-1-4244-1719-3. Disponível em: <http://ieeexplore.ieee.org/document/4668193/>. 16

AMAZON. *AWS IoT*. 2016. 1–8 p. Disponível em: <https://aws.amazon.com/pt/iot/>. 7

AMAZON. *Definição de preço do AWS IoT – Amazon Web Services*. 2016. 2 p. Disponível em: <https://aws.amazon.com/pt/iot/pricing/>. 7

ARM. *Welcome to mbed*. 2016. Disponível em: <https://www.mbed.com/en/>. 7

ASHTON, K. That internet of things thing. *RFiD Journal*, v. 22, n. 7, p. 4986, 2009. Disponível em: <http://www.rfidjournal.com/articles/view?4986>. 14

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. *Computer Networks*, Elsevier B.V., v. 54, n. 15, p. 2787–2805, 2010. ISSN 13891286. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1389128610001568>. 14

BAHILLO, A. et al. IEEE 802.11 distance estimation based on RTS/CTS two-frame exchange mechanism. In: IEEE. *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*. 2009. p. 1–5. Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5073583>. 16, 19

BELLAVISTA, P.; KÜPPER, A.; HELAL, S. Location-based services: Back to the future. *IEEE Pervasive Computing*, v. 7, n. 2, p. 85–89, 2008. ISSN 15361268. 15

BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, IGI Global, v. 5, n. 3, p. 1–22, jan 2009. ISSN 1552-6283. Disponível em: <http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf>. 16

CHEN, G.; KOTZ, D. *A Survey of Context-Aware Mobile Computing Research*. [S.l.], 2000. v. 3755, n. TR2000-381, 1–16 p. Disponível em: <http://www.cs.dartmouth.edu/reports/abstracts/TR2000-381/>. 9

Cisco Blog. *How Many Internet Connections are in the World? Right. Now.* 2013. 2 p. Disponível em: <http://blogs.cisco.com/news/cisco-connections-counter>. 7

CROW, B. et al. IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, v. 35, n. 9, p. 116–126, 1997. ISSN 01636804. Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=620533>. 16

DEY, A. K. Providing Architectural Support for Building Context-Aware Applications. *Sensors Peterborough NH*, v. 16, n. November, p. 97–166, 2000. ISSN 07370024. Disponível em: <http://portal.acm.org/citation.cfm?id=932974>. 15

DEY, A. K.; ABOWD, G. D. Towards a Better Understanding of Context and Context-Awareness. *Computing Systems*, v. 40, n. 3, p. 304–307, 1999. ISSN 00219266. 15

DJUKNIC, G. M.; RICHTON, R. E. Geolocation and assisted GPS. *Computer*, v. 34, n. 2, p. 123–125, 2001. ISSN 0018-9162. Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=901174>. 9

DZONE. THE DZONE GUIDE TO THE INTERNET OF THINGS. *DZone*, p. 32, 2015. Disponível em: <https://dzone.com/guides/internet-of-things-1>. 7

ESP8266.NET. *ESP8266.net home*. 2016. Disponível em: <http://esp8266.net/>. 7

FELDMANN, S. et al. An indoor Bluetooth-based positioning system : concept , Implementation and experimental evaluation. *International Conference on Wireless Networks*, n. JANUARY 2003, p. 109–113, 2003. ISSN 09621105. Disponível em: <https://www.researchgate.net/publication/220719209{_}An{_}Indoor{_}Bluetooth-Based{_}Positioning{_}System{_}Concept{_}Implementation{_}and>. 16

FERREIRA, L. C. P. *Sistema localizador interior de baixo custo*. 2016. Disponível em: <http://repositorio.ipl.pt/handle/10400.21/6162>. 16

FRIEDEMANN, M.; FLOERKEMEIR, C. From the Internet to the Internet of Things. *From Active Data Management to Event-Based Systems and More*, p. 242–259, 2011. ISSN 0302-9743. Disponível em: <http://www.ulb.tu-darmstadt.de/tocs/79304567.pdf>. 14

FUNDATION, R. *Raspberry Pi Zero: the $5 computer*. 2015. 2 p. Disponível em: <https://www.raspberrypi.org/blog/raspberry-pi-zero/>. 7

GARTNER. *Gartner Says the Internet of Things Will Transform the Data Center*. 2014. 5 p. Disponível em: <http://www.gartner.com/newsroom/id/2684616>. 7

GARTNER. *Gartner Says 6.4 Billion Connected "Things"Will Be in Use in 2016, Up 30 Percent From 2015*. 2015. 1 p. Disponível em: <http://www.gartner.com/newsroom/id/3165317>. 7

GOOGLE. *Google for Internet of Things*. 2016. 1–5 p. Disponível em: <https://cloud.google.com/solutions/iot/>. 7

HOSSAIN, A. K. M. M.; SOH, W.-S. A Comprehensive Study of Bluetooth Signal Parameters for Localization. In: *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2007. p. 1–5. ISBN 978-1-4244-1143-6. ISSN 2166-9570. Disponível em: <http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=853>. 16

IBM. *IBM IoT*. 2016. 1–5 p. Disponível em: <http://www.ibm.com/internet-of-things/>. 7

INTEL. *IoT Solutions | IntelDeveloper Zone*. 2016. 1–4 p. Disponível em: <https://software.intel.com/en-us/articles/a-fast-flexible-and-scalable-path-to-commercial-iot-solutions>. 7

International Telecommunication Union. Overview of the Internet of things. *Series Y: Global information infrastructure, internet protocol aspects and next-generation networks - Frameworks and functional architecture models*, p. 22, 2012. Disponível em: <http://handle.itu.int/11.1002/1000/11559>. 14

JAMES, M. The Ultimate Scrum Reference Card. *Dzone*, p. 6, 2016. Disponível em: <https://dzone.com/refcardz/scrum>. 19

KAUFMANN, A.; DOLAN, K. *Price Comparison: Google Cloud vs AWS*. [S.l.], 2015. 16 p. Disponível em: <https://cloud.google.com/files/esg-whitepaper.pdf>. 7

KRANENBURG, R. van. The Sensing Planet: Why The Internet Of Things Is The Biggest Next Big Thing. *Co.CREATE*, p. 1–8, 2012. Disponível em: <http://www.fastcocreate.com/1681563/the-sensing-planet-why-the-internet-of-things-is-the-biggest-next-big-thing>. 7

LEMOS, A. A Comunicação das Coisas: Internet das Coisas e Teoria Ator-Rede. p. 1–23, 2013. 7

MICROSOFT. *The Internet of Your Things*. 2016. Disponível em: <https://dev.windows.com/en-US/iot>. 7

ORACLE. *Oracle IoT*. 2016. 3–5 p. Disponível em: <https://www.oracle.com/solutions/internet-of-things/index.html>. 7

PASCALAU, E.; NALEPA, G. J.; KLUZA, K. Towards a Better Understanding of. *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, p. 959–962, 2013. ISSN 1743-9213. 15

RASPBERRY PI FOUNDATION. *Raspberry Pi 3 Model B - Raspberry Pi*. 2016. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. 17

ROMáN, M.; CAMPBELL, R. H. A Model for Ubiquitous Applications. 2001. Disponível em: <http://www.ncstrl.org:8900/ncstrl/servlet/search>. 16

VUJOVIĆ, V.; MAKSIMOVIĆ, M. Raspberry Pi as a Sensor Web node for home automation. *Computers & Electrical Engineering*, v. 44, p. 153–171, feb 2015. ISSN 00457906. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0045790615000257>. 16

VUJOVIC, V. et al. Raspberry Pi as Internet of Things hardware : Performances and Constraints Raspberry Pi as Internet of Things hardware : Performances and Constraints. n. JUNE, 2014. 16

WEISER, M. The Computer for the 21st Century. *Scientific American*, v. 265, n. 3, p. 3–11, sep 1999. ISSN 0036-8733. Disponível em: <http://www.nature.com/doifinder/10.1038/scientificamerican0991-94http://dl.acm.org/citation.cfm?doid=329124.329126>. 7

WORTMANN, F.; FLÜCHTER, K. *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. [s.n.], 2015. v. 57. 221–224 p. ISSN 1867-0202. Disponível em: <http://dx.doi.org/10.1007/s12599-015-0383-3>. 14