

**UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Faculdade de Ciências - Campus Bauru**

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**DESENVOLVIMENTO DE UM PROTÓTIPO DE RASTREADOR DE
BEACONS UTILIZANDO O RASPBERRY PI**

**BAURU
2015**

GABRIEL LUIZ BASTOS OLIVEIRA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE RASTREADOR DE
BEACONS UTILIZANDO O RASPBERRY PI**

Relatório Parcial apresentado como requisição da disciplina de Projeto de Implementação de Sistemas I do curso de Ciência da Computação.

Orientador: Prof. Dr. Eduardo Martins Morgado

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências - Campus Bauru

Departamento de Computação

BAURU
2015

RESUMO

Atualmente a área da internet das coisas tem recebido um foco enorme. Um dos motivos é devido a ser bem nova e também facilitar a integração das pessoas com o mundo real. Essa área tem motivado várias pessoas pela possibilidade de ter um produto concreto após pouco tempo de trabalho. Por conta disso, alguns dispositivos foram criados para auxiliar essas áreas, entre eles o *Arduino* e *Raspberry Pi*. Os *beacons* também são uma criação nova, comunicando com outros dispositivos maiores como o *RPi* para auxiliar a micro-localização dentro de um pequeno espaço, com um baixo custo. O seu foco de uso é pequeno, somente com leituras via *smartphones*. Esse projeto tem como objetivo criar um rastreador de *beacons* utilizando o *Raspberry Pi* para expandir essa capacidade, não necessitando de um celular para que os *beacons* possam ser encontrados.

Palavras-chave: *Beacon*. *Raspberry Pi*. Internet das Coisas.

SUMÁRIO

1	INTRODUÇÃO	4
2	OBJETIVOS	6
2.1	Objetivos Gerais	6
2.2	Objetivos Específicos	6
3	FUNDAMENTAÇÃO TEÓRICA	7
3.1	Raspberry Pi	7
3.2	Bluetooth Low Energy	9
3.3	Beacon	11
4	PROPOSTA DO PROJETO	15
5	MÉTODO DE PESQUISA	17
6	MATERIAIS E INSTRUMENTOS	18
7	EXPERIMENTOS E RESULTADOS	22
8	CRONOGRAMA	25
9	CONCLUSÃO	26
	REFERÊNCIAS	27

1 INTRODUÇÃO

A internet cresce dia após dia e cada vez mais diferentes tipos de dispositivos são conectados a essa imensa rede. Há uma estimativa de 26 bilhões de "coisas" conectadas a internet até 2020, comparado a 6 bilhões na década de 2000. Isso foi possível graças ao custo do acesso a internet e largura de banda (quantidade de dados trafegados) ter diminuído 40 vezes nos últimos 10 anos. Esse *boom* de crescimento também é influenciado pela IoT (*Internet of Things* - Internet das Coisas). (GOLDMAN SACHS, 2014).

Segundo Ashton (2009) o termo IoT surgiu como título de sua apresentação a Procter & Gamble (P&G) em 1999. Esse termo se dá ao uso de internet em diferentes tipos de dispositivos.

"(...) a Internet das Coisas é um conceito no qual dispositivos de nosso dia a dia são equipados com sensores capazes de captar aspectos do mundo real, como por exemplo, temperatura, umidade, presença, etc, e envia-los a centrais que recebem estas informações e as utilizam de forma inteligente." (NASCIMENTO, 2015).

Como exemplo podemos citar: geladeira ligada a internet informando a falta de condimentos, caixa de remédio conectada a internet prevendo o término da caixa de remédio para avisar ao consumidor, entre diversos outros. Um bom exemplo é interligar uma casa por meio de sensores, como termostatos, sistemas de segurança, iluminação, sistemas de entretenimento com uma inteligência por trás para diversas aplicações. (GOLDMAN SACHS, 2014)

Diversas áreas tiveram o seu crescimento alavancado por conta da IoT. O mais marcante é o ramo do DIY (*Do It Yourself*, ou faça você mesmo), em que pessoas criam ou adaptam coisas para suas necessidades. Isso se dá por conta de ter aparecido no mercado dispositivos e componentes que auxiliam a criação e adaptação de eletrônicos e outros. Um ótimo exemplo é o Arduino, um dispositivo que nos ajuda a criar os projetos de eletrônica que consiste de duas partes: o hardware e o software. Com eles é possível construir praticamente de tudo, desde um LED piscante a um robô que envia um *tweet* quando sua planta está sem água. (BEN, 2015). (SORREL, 2008).

Além do Arduino existem diversos outros *devices* com funcionalidades similares ou até complementares. Podemos citar o *Raspberry Pi*, um computador do tamanho de um cartão de crédito e de baixo custo. (RASPBERRY PI, 2015b). Por ser um computador é possível executar um sistema operacional (como Linux, *RISC OS*, *Windows 10 for IoT*). Dependendo do modelo possui portas USB (1 a 4) para conexão com periféricos

(como mouse, teclado, adaptador WiFi, Bluetooth), e também porta Ethernet para conexão a internet cabeada (exceto modelo A e A+). Também possui de 26 (modelo A e B) a 40 pinos (modelo A+, B+ e 2) para conexões gerais de entrada e saída digitais (GPIO - *General Input and Output*).

Através desses pinos pode-se conectar uma diversidade de componentes eletrônicos como sensores, atuadores, outros dispositivos para comunicação. Dessa forma, suas funcionalidades são expandidas de uma forma absurda, ficando a cargo de cada pessoa montar uma nova aplicação. Uma boa aplicação é a conexão de um adaptador bluetooth pela USB para comunicação com smartphones, tablets, PCs e outros dispositivos tipo sensores sem fio.

Um exemplo desses sensores externos são os *beacons*, pequenos modelos sem fio que se comunicam por meio do Bluetooth 4.0 ou BLE (*Bluetooth Low Energy* - Bluetooth de Baixa Energia). Como o próprio nome diz, esse padrão de comunicação utiliza muita pouca energia. Desta forma, um *beacon* pode funcionar por anos. "Na prática, ela permite localizar objetos (ou pessoas que carregam esses objetos) com muito mais precisão dentro de ambientes fechados." (TEIXEIRA, 2014).

2 OBJETIVOS

2.1 Objetivos Gerais

O objetivo geral desse projeto é planejar e desenvolver um protótipo de rastreador de *iBeacons* utilizando o *Raspberry Pi*.

2.2 Objetivos Específicos

- a) Estudar o funcionamento de *beacons*, comunicação via *bluetooth low energy* e *Raspberry Pi*, assim como suas aplicações.
- b) Identificar os requerimentos básicos de funcionamento de *beacons* e *Raspberry Pi*.
- c) Definir os elementos para desenvolvimento do protótipo.
- d) Planejar a estrutura do sistema.
- e) Implementar o protótipo de acordo com a estrutura e elementos planejados.

3 FUNDAMENTAÇÃO TEÓRICA

A área de IoT está em uma onda crescente, com um grande número de pessoas gastando nesse mercado, e um bom número de profissionais migrando para essa área. Há uma estimativa de que existem 19 milhões de profissionais trabalhando na indústria de desenvolvimento de software, e desses, 19% trabalham em algum projeto relacionado a IoT. (THIBODEAU, 2015).

"A próxima onda na era da computação será fora do domínio do ambiente de trabalho tradicional. No paradigma da IoT, muitos dos objetos que nos rodeiam estarão na rede de uma forma ou de outra. RFID (Radio Frequency Identification - Identificadores via Rádio Frequência) e as tecnologias de redes de sensores crescerão para enfrentar este novo desafio, em que os sistemas de informação e comunicação estão embutidos nos ambientes que nos rodeiam, de forma invisível.". (GUBBI et al., 2013).

É notável o crescimento dessa área. Há uma expectativa de que, em 2020, o número de carros conectados a internet supere o número de carros não conectados, sendo esses carros possíveis de se comunicar com outros veículos e a infraestrutura das ruas, como os semáforos. (GOLDMAN SACHS, 2014). Segundo Press (2014), em 2014 a IoT substituiu a área da *Big Data* como a tecnologia mais empolgante, ou seja, a tecnologia que mais pessoas iriam migrar e se interessar.

3.1 Raspberry Pi

Quando se pesquisa algo relacionado a IoT é praticamente impossível não achar relação e links para *Raspberry Pi*, Arduinos, e toda a área de *D/IY*. Atualmente existem três modelos, conforme a tabela 1 e figura 1.

Tabela 1 – Comparativo entre os modelos de *Raspberry Pi*

Versão	A+	B+	2 Modelo B
Processador	ARMv6 single core	ARMv6 single core	ARMv7 quad core
Velocidade CPU	700 MHz single-core	700 MHz single-core	900 MHz quad-core
Memória RAM	256 MB	512 MB	1 GB
Portas USB	1	4	4
<i>Ethernet</i>	Não	Sim	Sim

Fonte: (MAKER SHED, 2015)

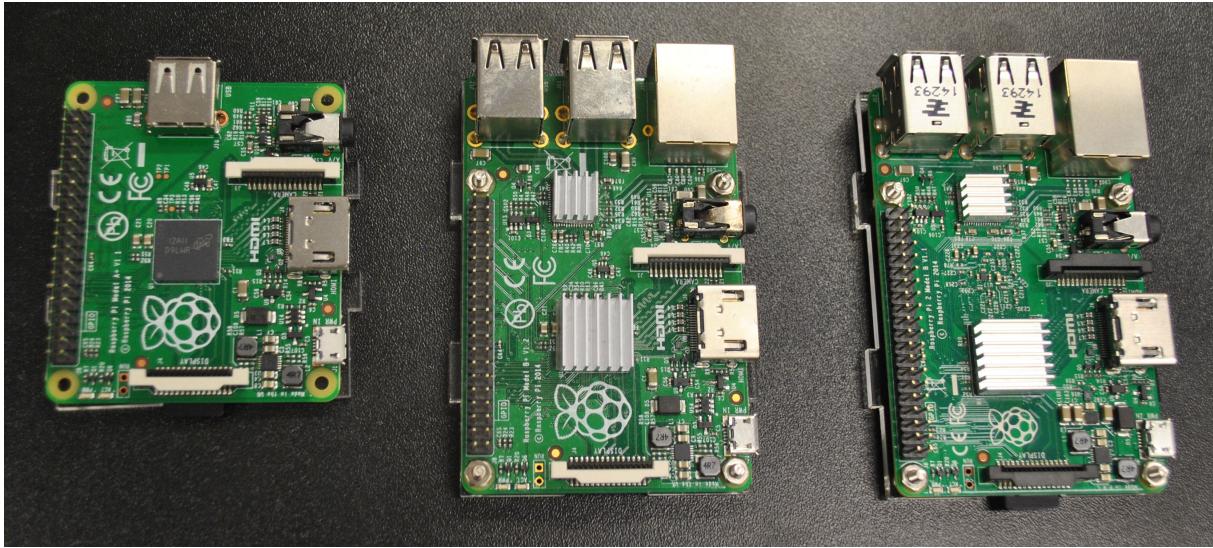


Figura 1 – *RPi* versão A+ (esquerda), versão B+ (centro) e versão 2 Modelo B (direita).
Elaborado pelo autor.

Segundo Raspberry Pi (2015b), a fundação responsável pela criação desse pequeno dispositivo gostaria de ver ele ser usado por crianças e pessoas carentes do mundo todo para aprender a programar e como a computação funciona. É facilmente configurável, utilizando um cartão de memória como HD para salvar dados e o sistema operacional.

Por meio de suas portas de entrada e saída digitais é possível conectar uma gama ampla de sensores, atuadores, componentes eletrônicos, ficando a cargo do programador e criador do projeto a escolher como esses pinos serão conectados e aproveitados. Atualmente existem diversos módulos para expandir os meios de comunicação entre diferentes *devices*, e um deles é via *bluetooth*.

Atualmente existem diferentes sistemas operacionais portados para o *RPi*. Entre eles, temos os seguintes:

- a) **Raspbian**: baseado na distribuição *Linux* chamada *Debian*. Atualmente é a suportada oficialmente pela *RPi Foundation*. (RASPBERRY PI, 2015a).
- b) **Ubuntu Mate**: baseado na distribuição *Linux* chamada *Ubuntu*, juntamente com o software *MATE Desktop* para gerenciamento de janelas. (UBUNTU MATE, 2015).
- c) **Snappy Ubuntu Core**: distribuição *Linux* voltada a *cloud* e dispositivos. (CANONICAL LTD., 2015).
- d) **Windows 10 for IOT Core**: versão do *Windows 10* da *Microsoft* para dispositivos voltados a internet das coisas. (MICROSOFT, 2015).

3.2 Bluetooth Low Energy

A tecnologia do *bluetooth* vem sendo amplamente utilizada, principalmente após a criação da versão 4.0, ou *BLE* (*Bluetooth Low Energy* - *Bluetooth de Baixa Energia*), por conta de ter um baixíssimo gasto energético, podendo preservar a bateria do dispositivo que a utiliza. O *BLE* faz parte da tecnologia nomeada *Bluetooth Smart* inteligente e eficiente energeticamente, voltada para *devices* que usam pequenas fontes de energia. (BLUETOOTH SIG, 2015a).

O *BLE* possui similaridades com a versão clássica do *bluetooth*. Ambos utilizam o espectro de frequências de 2.4 GHz (mesmo utilizado pelas redes *WiFi*), mesma modulação GFSK e velocidade de 1 Mbps, porém a indexação de ambos é diferente. A versão clássica possui 79 canais, e a *BLE* possui 40 canais. Além disso, os canais são espaçados de forma diferente, conforme tabela 2. (ARGENOX, 2015).

Tabela 2 – *BLE Physical Layer*

	BLE	Classic
Modulação	GFSK 0.45 a 0.55	GFSK 0.28 a 0.35
Velocidade de Transferência	1 Mbit/s	1 Mbit/s
Canais	40	79
Espaçamento	2 MHz	1 MHz

Fonte: (ARGENOX, 2015)

O espectro de 2.4 GHz para *bluetooth* se extende de 2402 MHz a 2480 MHz, e os canais 37, 38 e 39 (últimos três) são específicos para anúncio (*advertisement*), conforme figura 2. (ARGENOX, 2015).

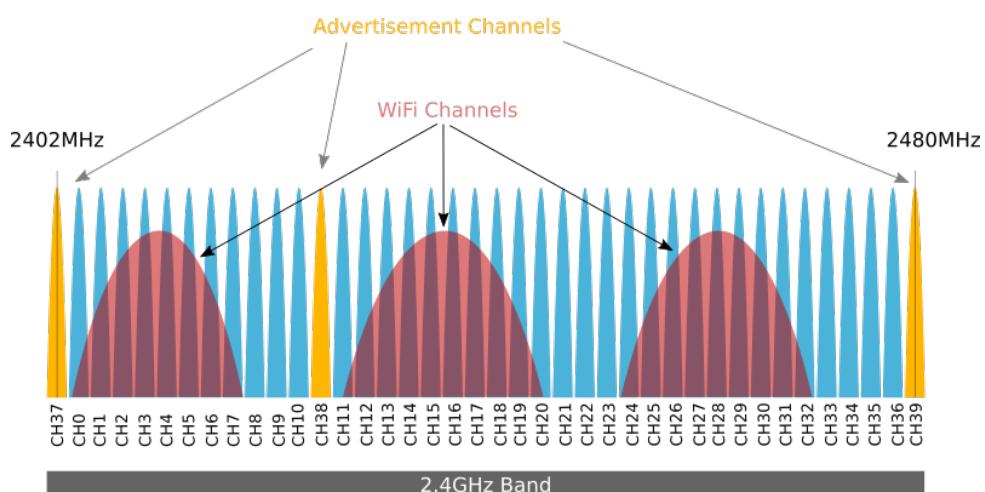


Figura 2 – Separação da banda 2.4 GHz para bluetooth e WiFi. (ARGENOX, 2015).

Interessante notar que esses canais estão posicionados em forma bastante estratégica, no começo, final e meio da banda de 2.4 GHz. Isso se deve para aumentar a eficácia, evitando todos os canais ficarem lotados ou com muita interferência. (ARGENOX, 2015).

Os *BLE Advertisement Packets*, ou pacotes de anúncio *BLE* é uma das formas de conexão do *Bluetooth Smart*. Por meio dos anúncios, um *device* transmite pacotes para todos que estão a sua volta, sem necessariamente necessitar de uma conexão direta entre somente outro dispositivo.

Um *BLE Advertisement Packet* é formado conforme figura 3. O preâmbulo, *access address* e CRC são informações para formação do pacote. Os dados estão de fato dentro do PDU (*Packet Data Unit*). O cabeçalho de 2 bytes informa o tamanho do *payload* (carga de dados), além de informações relevantes como tipo do pacote, tipo de mensagem enviada, entre outros. (ARGENOX, 2015).

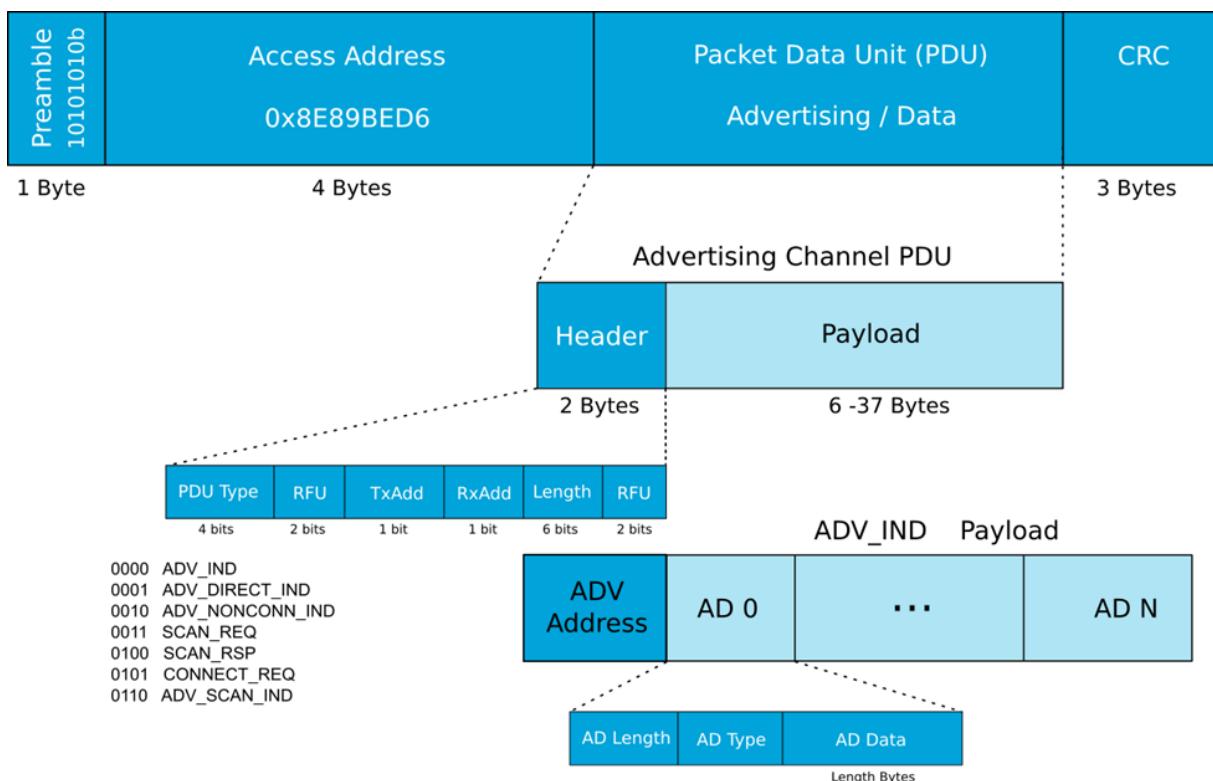


Figura 3 – Modelo de *BLE Advertising Packet*. (ARGENOX, 2015).

O importante do *BLE Advertising Packet* é o tipo de anúncio feito, ou quais são as informações do pacote. Bluetooth SIG (2015b) apresenta uma tabela com os possíveis valores e também o significado de cada uma. Por exemplo, o valor 0xFF significa que o pacote contém dados específicos do fabricante, ou seja, existe a flexibilidade de manipular o pacote da forma que for preciso, contanto que mantenha a estrutura original de 6 a 37 bytes de *payload*, conforme figura 3. (ARGENOX, 2015).

3.3 Beacon

Os *beacons* são pequenos sensores que são capazes de identificar objetos com precisão dentro de ambientes fechados. (TEIXEIRA, 2014).

Como muitos espaços fechados (restaurantes, museus, shopping centers, casas de show) possuem estrutura metálica ou utilizam algum tipo de metal em sua construção, é comum que o sinal de GPS fique enfraquecido quando os usuários estão dentro daquele local. Nesse caso, os *Beacons* são uma ótima solução: um hardware relativamente barato, e pequeno o suficiente para ser plugado na parede ou instalado sobre um balcão. (TEIXEIRA, 2014).



Figura 4 – Modelo de *beacon* proprietário - MPact, da Zebra Technologies Corporation. Elaborado pelo autor.

Segundo Teixeira (2014), os *beacons* utilizam o *BLE* para detectar um dispositivo próximo e transmitir seu identificador único e avisar que está ali presente. Teixeira (2014) também diz que os *beacons* não são inteligentes, toda a interação deve depender do dispositivo que recebe a informação do identificador único.

Atualmente os usos de *beacons* estão restritos a aplicativos em smartphones realizando a leitura e interagindo com o usuário, porém existem ainda diversas áreas a serem exploradas, e um bom exemplo são as casas inteligentes. Segundo Grothaus (2014), a *Apple* está apostando em um kit de desenvolvimento (*HomeKit*) que permita aos desenvolvedores interagirem com *smart devices* presentes no ambiente.

O protocolo *iBeacon* foi apresentado pela *Apple* juntamente com o iOS 7, versão de seu sistema operacional para dispositivos móveis. É uma tecnologia baseada nos *beacons*, porém adaptada para as necessidades e aplicações de seu sistema móvel.

Apple criou uma adaptação do pacote genérico de *beacon* para transmitir um total de três dados:

- UUID**: Identificador único formado de 16 bytes (128 bits). Focado em ser único para cada aplicação, cada aplicativo ou desenvolvimento deve ter o seu.
- Major**: Identificador de 2 bytes que identifica uma sub-região grande. Usado, por exemplo, para dividir as lojas de um grande varejista.
- Minor**: Identificador de 2 bytes que identifica uma sub divisão de região, ou seja, uma região menor que o Minor.

Um exemplo de aplicação é citado na tabela 3. Utiliza-se um único UUID para todas as lojas, com um único Major por loja e um Minor por departamento, podendo ser repetido entre as lojas. Dessa forma amplia-se o número de possíveis aplicações, aumentando a quantidade de números disponíveis.

Tabela 3 – *Exemplo de aplicação*

Localização da Loja	São Francisco	Paris	Londres
UUID	D9B9EC1F-3925-43D0-80A9-1E39D4CEA95C		
Major	1	2	3
Minor - Roupas	10	10	10
Minor - Utilidades Domésticas	20	20	20
Minor - Automotivo	30	30	30

Fonte: (APPLE INC, 2014)

O pacote *BLE* utilizado pela tecnologia iBeacon pode ser visto na figura 5. Segundo Austin (2015), o significado do prefixo iBeacon é:

- Adv Flags**: determinam que é um pacote *BLE* de descobrimento geral, e que somente transmite e não permite conexões.
- Adv Header**: determinam que os próximos 26 bytes serão a carga de dados (*payload* de fato). Sempre será 0x1AFF.
- Company ID**: indica que é o ID da Apple junto com a Bluetooth SIG. Essa informação que faz ser dependente da Apple. Sempre será 0x004C.
- iBeacon Type**: ID secundário utilizado por todos iBeacons que identificam ser um *beacon* de proximidade. Sempre será 0x02.
- iBeacon Length**: identifica quantos bytes terão em seguida. Sempre será 0x15, ou 21 bytes.

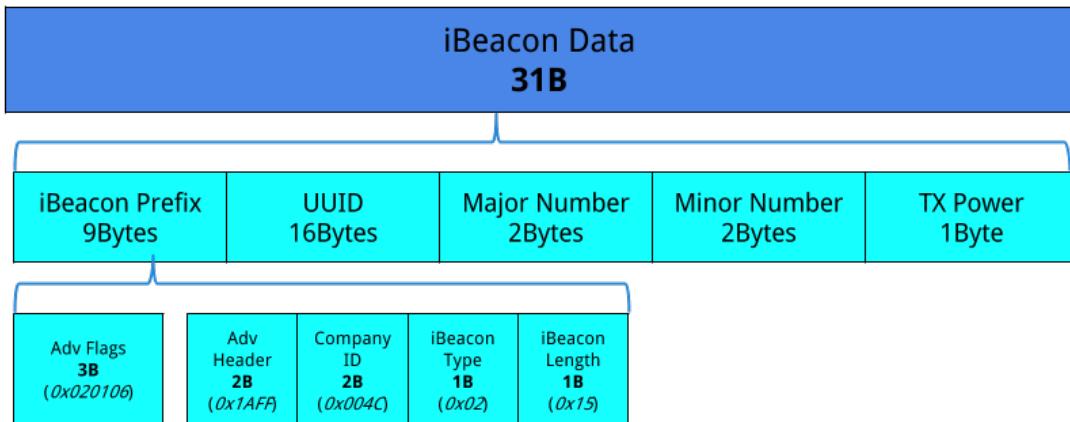


Figura 5 – *Payload* do pacote iBeacon. (AUSTIN, 2015).

Os *iBeacons* foram criados com intuito de serem descobertos por smartphones. Um exemplo de aplicação é uma cafeteria com um *iBeacon* no balcão próximo ao caixa. Quando um consumidor entra na loja e chega próximo ao caixa, um aplicativo em seu celular identifica o *iBeacon* pela sua UUID, identifica pelo Major que se trata da cafeteria número 12 e encontra uma promoção com o Minor de número 26. Em seguida, apresenta uma notificação ao usuário uma promoção e também um cupom válido de desconto para usar no caixa. (AUSTIN, 2015).

Uma outra aplicação interessante citada por Austin (2015) é a possibilidade do smartphone transmitir pacotes *iBeacon*, sem necessidade de um hardware externo. Dessa forma, pode-se por exemplo automatizar o check-in em um evento e rastrear o movimento dos usuários entre os estabelecimentos.

Existem outros protocolos baseados na tecnologia *beacon*. Dois se destacam por ser abertos e passíveis de alterações: *AltBeacon* e *Eddystone*, este último criado pela Google. O *AltBeacon* possui um modelo bastante parecido com o *iBeacon*, porém com possibilidade de alterar o número do fabricante, ter possibilidade de código de *beacons* diferentes, e também a possibilidade do fabricante colocar sua informação ao final do pacote, conforme figura 6 (AUSTIN, 2015).

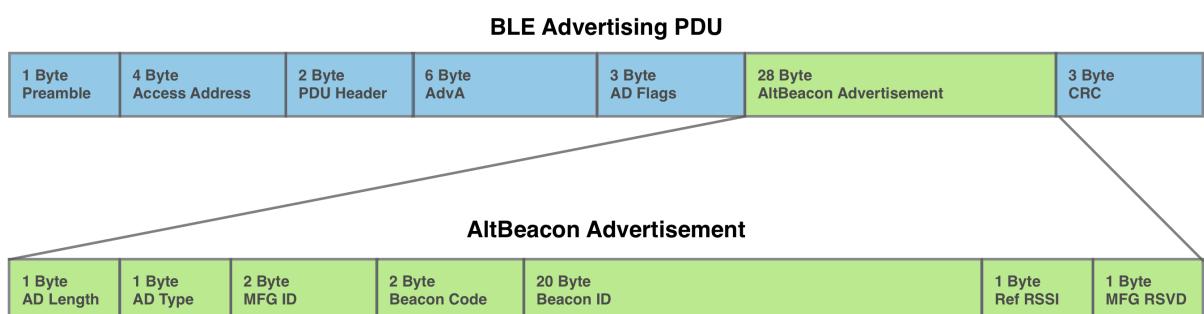


Figura 6 – Pacote do *AltBeacon*. (AUSTIN, 2015).

Segundo Wandschneider Nirdhar Khazanie (2015), o protocolo *Eddystone* possui três modos de funcionamento:

- a) *Eddystone-UID*: transmite um *beacon* ID, composto de 10 bytes identificando um grupo de *beacons* e 6 bytes identificando um único *beacon*.
- b) *Eddystone-URL*: transmite um link comprimido, para que o cliente possa acessar um site na internet.
- c) *Eddystone-TLM*: transmite informações de telemetria sobre o *beacon*, como por exemplo voltagem da bateria, temperatura e quantos pacotes foram enviados.

4 PROPOSTA DO PROJETO

Os *beacons* foram pouco explorados até o momento, seu uso está sendo mais notado na área de grandes lojas do varejo.

"A Apple (...) já está utilizando a tecnologia em 254 lojas nos EUA. As funcionalidades já estão embutidas na versão oficial do aplicativo da Apple Store para iOS [, sistema operacional de seus smartphones e tablets]. (...) Quando o usuário se aproxima de uma loja física, o aplicativo oferece toda uma camada extra de informações e serviços que são específicos para aquela unidade – como por exemplo ofertas locais, tamanho da fila para ser atendido no Genius Bar, eventos e treinamentos que estão agendados ali na loja etc." (TEIXEIRA, 2014).

A Macy's (grande rede norte americana de loja de departamentos) está realizando testes em algumas de suas lojas para enviar alertas a pessoas que entrarem em suas lojas, com promoções e melhores indicações, utilizando o padrão *iBeacon* da Apple. Até o momento esse teste está limitado a usuários de iPhone, e somente quando entrar na loja. Em um teste futuro espera-se que seja possível separar por departamentos, para que quando um usuário percorra a loja apareça as notícias relativo ao local da loja que ela está. (KASTRENAKES, 2013)

Porém as maiores aplicações são voltadas a smartphones e tablets como receptores e identificadores dos *beacons*. Existem diversas empresas desenvolvendo variados tipos de *beacons*, mas a maioria delas utiliza aplicativos para *mobile devices* como identificadores.

Alguns motivos podem ser apresentados para isso, mas o mais relevante é que até o momento poucas pessoas, grupos e empresas dedicaram tempo e dinheiro para investir nessa nova tecnologia. Um grande problema também é a necessidade de ter o *bluetooth* ligado e permitindo conexões para que um smartphone consiga identificar os *beacons*.

Como visto anteriormente, o *RPi* é um computador pequeno e de baixo custo, e também tem portas de entrada e saída digitais e portas USB. Por conta disso é possível expandir suas funcionalidades conectando módulos de *bluetooth*, *WiFi*, entre outros, podendo ser utilizado como um ótimo dispositivo de prototipagem.

Por ter uma comunidade grande e ativa, o seu estudo é facilitado. Um bom número de sites e comunidades estão desenvolvendo projetos com esse dispositivo e publicando suas conquistas. É possível encontrar tutoriais e passo-a-passo de um bom número de projetos, porém somente para aprendizado. Ao desenvolver um projeto mais elaborado isso pode servir como guia de estudo e aprofundamento.

As empresas tem investido um bom dinheiro na área de *DIY* e *IoT*. Existe uma grande expectativa para o futuro, por ser uma tecnologia barata para ser aplicada a grandes volumes, ter bastante gente envolvida e engajada, e também sua grande facilidade de criação.

Esse projeto se propõe a desenvolver um protótipo utilizando outro meio para rastrear os *iBeacons*, neste caso o *RPi*. Desta forma será possível explorar maiores aplicações aos *beacons*.

5 MÉTODO DE PESQUISA

O levantamento bibliográfico relacionado ao tema foi a primeira fase dessa pesquisa, foram realizadas buscas relacionadas aos assuntos: *Raspberry Pi*, comunicação via *Bluetooth Low Energy*, *beacons*. Concomitantemente foi realizado o estudo das tecnologias, suas capacidades, limitações, aplicações, etc.

O segundo passo foi planejar o projeto baseado na análise do levantamento bibliográfico, assim como a definição de sua estrutura. Essa etapa foi necessária para facilitar e agilizar a implementação e testes dos componentes no próximo passo, definindo assim um escopo inicial de funcionalidades que o sistema terá, assim como outras tarefas a serem realizadas. Todo o processo será citado no tópico 6 (Experimentos e Resultados).

Em seguida será iniciada a implementação e testes do protótipo. O projeto será desenvolvido por etapas, sendo que a primeira será o estudo de identificação dos *beacons* via smartphone, a segunda será o desenvolvimento do software de identificação no *RPi*, e a terceira um aprimoramento do software embarcado para rastrear os *beacons*. Em seguida serão realizados testes unitários experimentais em laboratório para que o protótipo possa ser aprimorado. Será testado com mais de um modelo de *RPi* e também mais de um *beacon*, em ambiente aberto e fechado. Ao final espera-se ter um protótipo finalizado.

A etapa de testes será realizada por observação e experimentação nos ambientes. Os dados serão apresentados ao final do projeto pela confecção e apresentação da monografia.

Será desenvolvido no espaço do LTIA (Laboratório de Tecnologia da Informação Aplicada), da Universidade Estadual Paulista "Júlio de Mesquita Filho- Campus Bauru. Os *beacons MPact* e *iPad Mini* são de propriedade do próprio laboratório.

6 MATERIAIS E INSTRUMENTOS

A versão escolhida para início do projeto foi a 2 Modelo B, por ter um melhor processamento e mais memória RAM (figura 7), conforme informado no item 3.2. Para os testes futuros será utilizado também a versão B+.

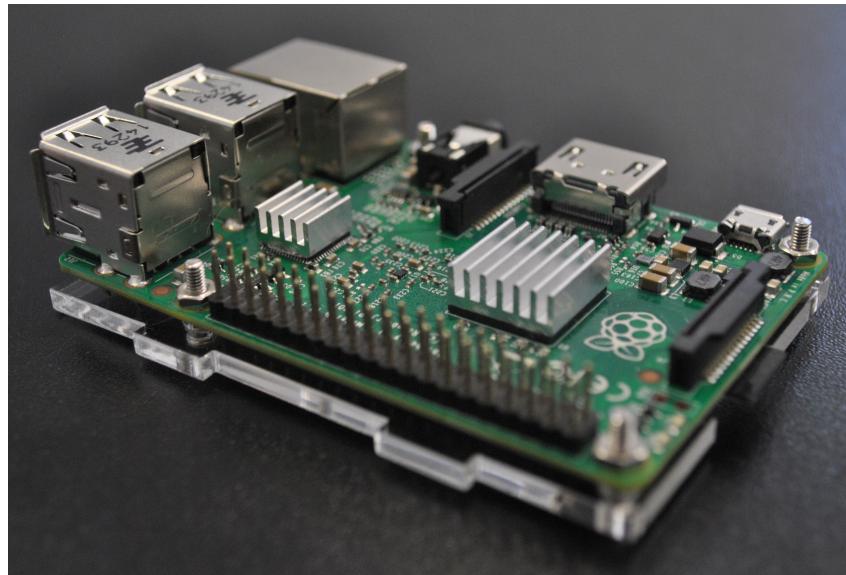


Figura 7 – RPi 2 modelo B utilizado nesse projeto. Elaborado pelo autor.

É necessário o uso de um adaptador *WiFi* para conexão a internet sem necessidade de cabo *Ethernet* e também um adaptador Bluetooth 4.0 com suporte a *BLE* para fazer a busca dos pacotes *beacon*. Os modelos de adaptadores usados serão Orico BTA-406 (bluetooth) e EDUP N8508GS (*WiFi*), conforme figura 8.



Figura 8 – Orico BTA-406 a esquerda e EDUP N8508GS a direita. Elaborado pelo autor.

Todo o sistema operacional fica instalado em um cartão microSD, com no mínimo 4 GB de espaço. O cartão utilizado nesse projeto é um SanDisk Ultra Class 10 de 8 GB, conforme figura 9.



Figura 9 – Cartão microSD SanDisk Ultra Class 10 utilizado no projeto. Elaborado pelo autor.

O sistema Raspbian permite que conecte remotamente via *SSH (Secure Shell)*, sem necessidade de estar conectado a monitor e teclado. O computador utilizado para realização do projeto é um MacBook Pro, com sistema Mac OS X 10.10. Nesse sistema o uso de *SSH* é simples, bastando abrir o aplicativo Terminal e utilizar o comando "*ssh usuario@computador*", conforme imagem 10. Esse tipo de abordagem é bastante utilizado para conexão a servidores na nuvem, para executar comandos, softwares, entre outros.

A screenshot of a Mac OS X terminal window titled "gabrieloliveira — pi@pi-tcc: ~ — ssh — 80x24". The window shows the command "ssh pi@pi-tcc" being run from a MacBook-Pro-de-Gabriel. The output includes the Debian license text, the Debian warranty notice, and the last login information: "Last login: Thu Aug 20 16:45:08 2015 from 192.168.1.53".

```
gabrieloliveira — pi@pi-tcc: ~ — ssh — 80x24
MacBook-Pro-de-Gabriel:~ gabrieloliveira$ ssh pi@pi-tcc
Linux pi-tcc 4.0.9-v7+ #807 SMP PREEMPT Fri Jul 24 15:21:02 BST 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug 20 16:45:08 2015 from 192.168.1.53
pi@pi-tcc ~ $
```

Figura 10 – Conexão com o *RPi* via *SSH*. Elaborado pelo autor.

O *beacon* utilizado para testes foi o *Zebra MPact*, conforme figura 11. Utiliza uma bateria CR2450 para alimentação de energia. A *Zebra* tem um sistema de administração e gerenciamento nomeado *MPact Toolbox*, instalado em um servidor do LTIA com sistema Debian 8.1. Esse sistema foi utilizado somente para conhecimento da

tecnologia *beacon* e atualização do *firmware*, feito somente por essa *Toolbox*. Esse software também apresenta a porcentagem de bateria, conforme imagem 12, permitindo a configuração de vários *beacons* simultaneamente. Permite também a mudança do modo de funcionamento, de *iBeacon* para *MPact*, protocolo criado pela fabricante.

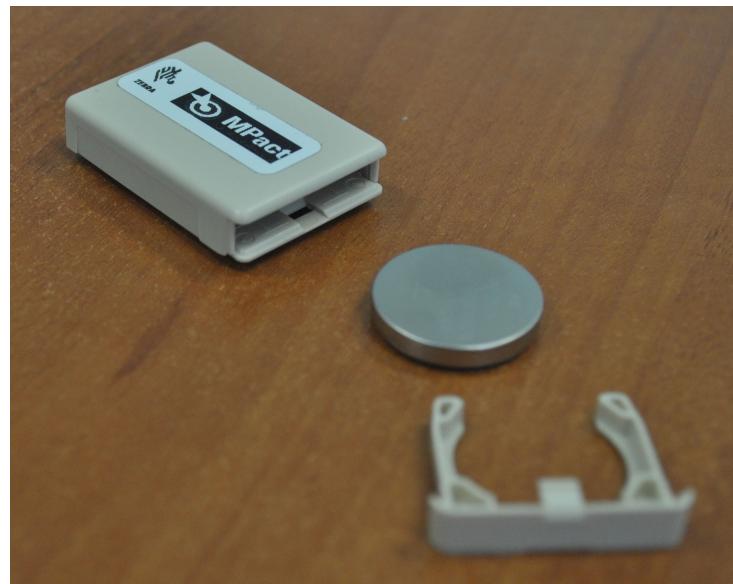


Figura 11 – *Beacon* Zebra MPact utilizado para testes. Elaborado pelo autor.

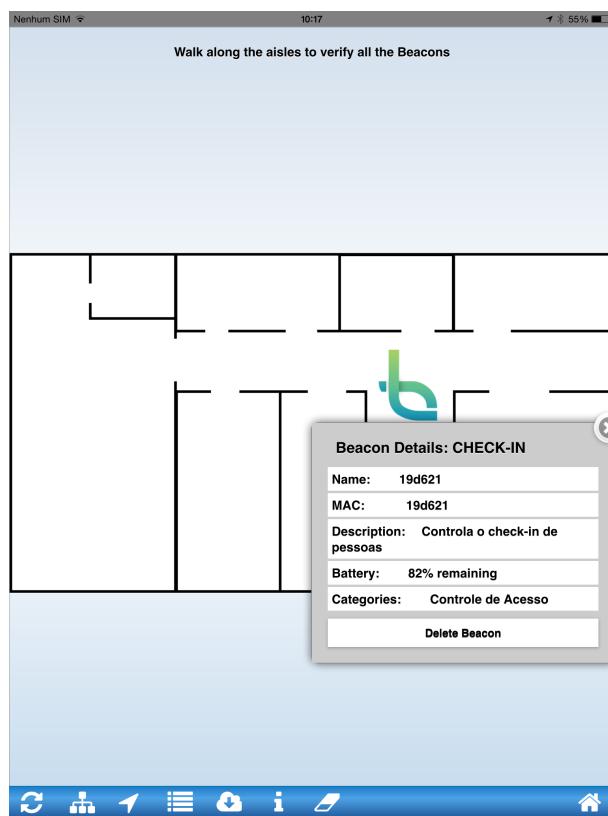


Figura 12 – *Beacon* configurado na *Toolbox* apresentando a porcentagem de bateria. Elaborado pelo autor.

Foi utilizado o smartphone Moto Maxx com sistema Android 5.0.1 e o tablet iPad mini Retina com sistema iOS 8.4 confirme imagem 13. O aplicativo utilizado em ambos foi o *Locate Beacon* da *Radius Networks*, que nos permite identificar os *beacons* e também simular um, para realizar os testes com diferentes tipos, podendo alterar os valores de UUID, Major, Minor e potência de transmissão.



Figura 13 – Moto Maxx (esquerda) e iPad Mini (direita). Elaborado pelo autor.

7 EXPERIMENTOS E RESULTADOS

Os experimentos realizados inicialmente foram a tentativa de identificação de um *beacon* com o *RPi*, para o estudo de funcionamento e comportamento dessas tecnologias. Para isso, o ambiente foi configurado conforme figura 14.

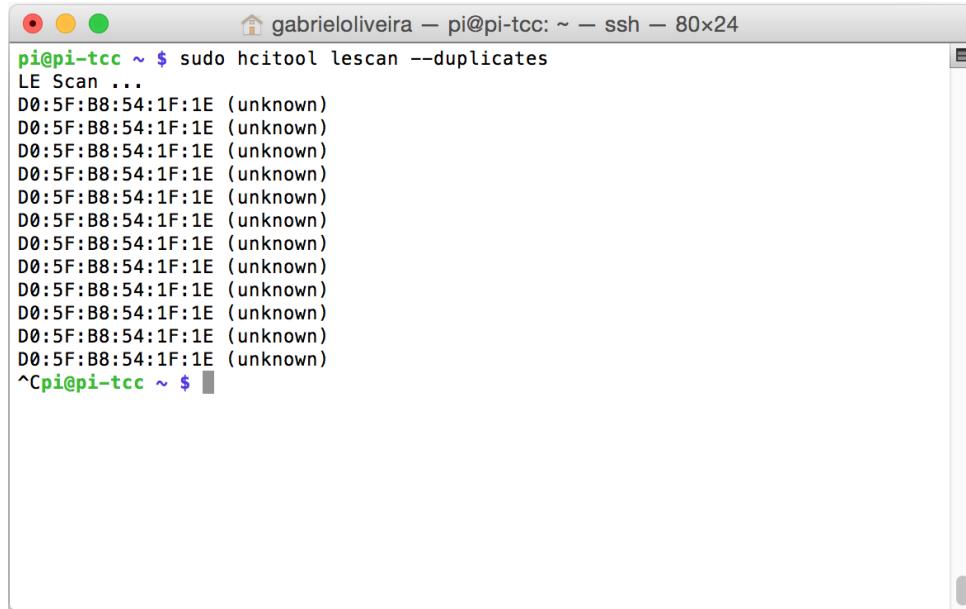


Figura 14 – Primeiro teste realizado, com *RPi* e *beacon MPact*. Elaborado pelo autor.

O computador ficou conectado via SSH com o *RPi*, recebendo as informações de leitura de pacotes *BLE*. Os softwares utilizados para isso foram, conforme Jjnebeker (2014), os seguintes:

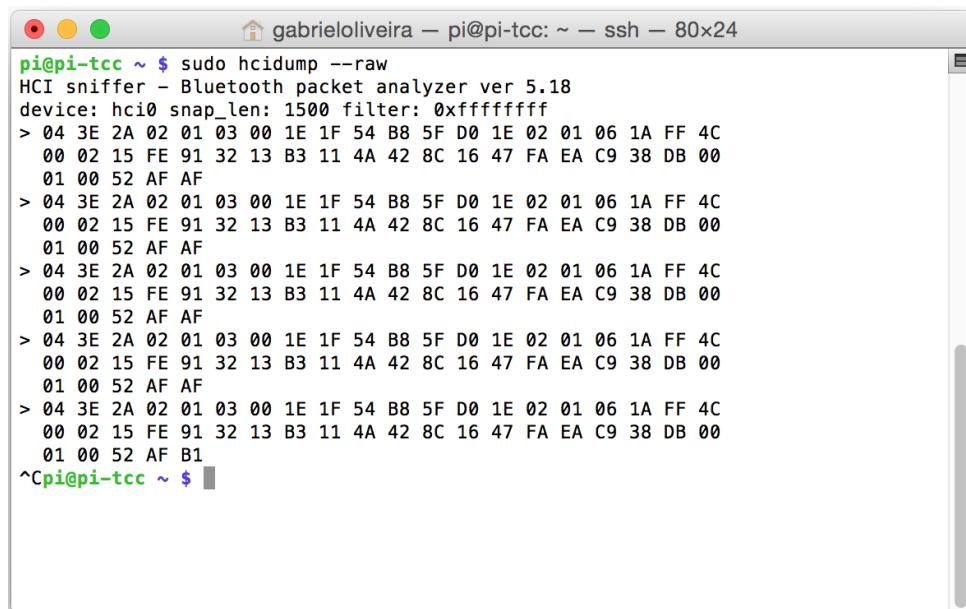
- a) **hcitool**: configurado da maneira *hcitool lescan –duplicates*, faz um scan na frequência *BLE* procurando por dispositivos que estejam transmitindo, conforme figura 15.
- b) **hcidump**: em conjunto com o *hcitool*, apresenta todos os pacotes escaneados na rede. Executado da maneira *hcidump –raw*, conforme figura 16.

Em seguida, com os softwares em execução e os pacotes sendo analisados, o *beacon* foi movido em cima da mesa para ficar a diferentes distâncias do *RPi*, conforme



```
pi@pi-tcc ~ $ sudo hcitool lescan --duplicates
LE Scan ...
D0:5F:B8:54:1F:1E (unknown)
^Cpi@pi-tcc ~ $
```

Figura 15 – Software *hcitool* executando. Elaborado pelo autor.



```
pi@pi-tcc ~ $ sudo hcidump --raw
HCI sniffer - Bluetooth packet analyzer ver 5.18
device: hci0 snap_len: 1500 filter: 0xffffffff
> 04 3E 2A 02 01 03 00 1E 1F 54 B8 5F D0 1E 02 01 06 1A FF 4C
  00 02 15 FE 91 32 13 B3 11 4A 42 8C 16 47 FA EA C9 38 DB 00
  01 00 52 AF AF
> 04 3E 2A 02 01 03 00 1E 1F 54 B8 5F D0 1E 02 01 06 1A FF 4C
  00 02 15 FE 91 32 13 B3 11 4A 42 8C 16 47 FA EA C9 38 DB 00
  01 00 52 AF AF
> 04 3E 2A 02 01 03 00 1E 1F 54 B8 5F D0 1E 02 01 06 1A FF 4C
  00 02 15 FE 91 32 13 B3 11 4A 42 8C 16 47 FA EA C9 38 DB 00
  01 00 52 AF AF
> 04 3E 2A 02 01 03 00 1E 1F 54 B8 5F D0 1E 02 01 06 1A FF 4C
  00 02 15 FE 91 32 13 B3 11 4A 42 8C 16 47 FA EA C9 38 DB 00
  01 00 52 AF AF
> 04 3E 2A 02 01 03 00 1E 1F 54 B8 5F D0 1E 02 01 06 1A FF 4C
  00 02 15 FE 91 32 13 B3 11 4A 42 8C 16 47 FA EA C9 38 DB 00
  01 00 52 AF B1
^Cpi@pi-tcc ~ $
```

Figura 16 – Software *hcidump* executando. Elaborado pelo autor.

figura 17. Esse passo foi necessário para verificar o formato dos pacotes recebidos e também analisar a distância máxima de identificação.

Com o adaptador Orico BTA-406 e posicionamento na mesa conforme figuras 14 e 17, cerca de 1,3 a 1,4 metros de distância entre o *RPi* e o *beacon* os pacotes já começaram a falhar e a leitura não ser tão constante.

O posicionamento do *RPi* foi alterado para testes, conforme figura 18. Com essa nova posição foi notado uma melhoria na recepção dos pacotes. A 1,5 metros entre o *RPi* e o *beacon* os pacotes começaram a falhar, com recepção notada até 1,6 metros.

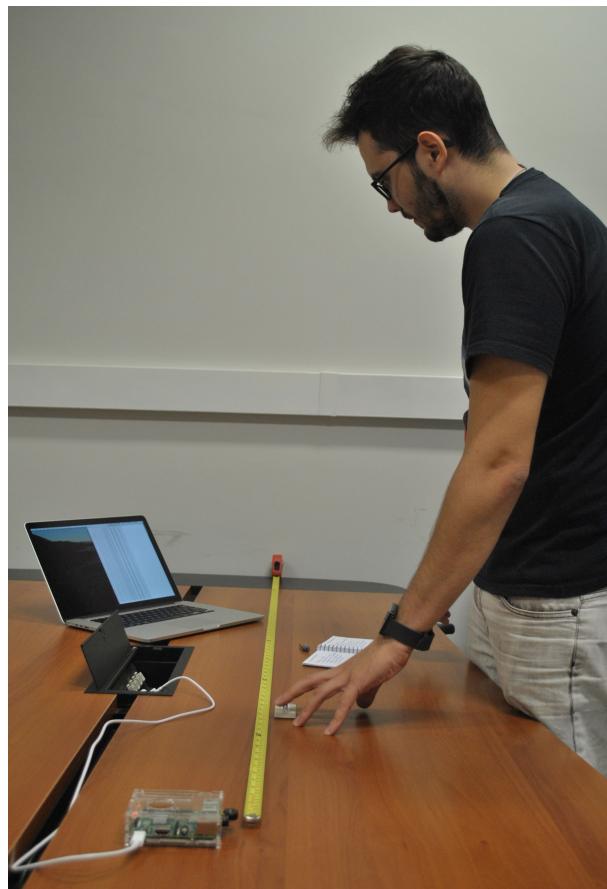


Figura 17 – Teste com movimentação do *beacon*. Elaborado pelo autor



Figura 18 – *RPi* posicionado de outra maneira. Elaborado pelo autor.

Mais experimentos e testes serão feitos durante o andamento do projeto para uma melhor especificação do alcance do *RPi* e *beacon*. Desta forma melhores resultados surgirão.

8 CRONOGRAMA

Até o momento foram realizadas as atividades de Pesquisa Bibliográfica e Estudo das Tecnologias completamente. Estão em andamento as atividades de Análise e Planejamento e Definição da Estrutura. Serão realizadas as atividades de Implementação, Testes e Finalização do Projeto, conforme cronograma apresentado na Tabela 4.

Tabela 4 – Cronograma de Atividades

Atividade	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan
Pesquisa Bibliográfica	X	X	X					
Estudo das Tecnologias		X	X					
Análise e Planejamento			X	X				
Definição da Estrutura			X	X				
Implementação				X	X	X		
Testes					X	X	X	
Finalização do Projeto							X	X

Fonte: Elaborado pelo autor.

9 CONCLUSÃO

Esse projeto tem como objetivo o aprofundamento na área de Internet das Coisas, pelo estudo das tecnologias de *Bluetooth Low Energy*, *Raspberry Pi* e *beacons*. O resultado final será um protótipo de rastreador de *beacons* utilizando o *Raspberry Pi*, ainda em desenvolvimento.

A fundamentação teórica realizada por meio de pesquisa bibliográfica no início do projeto foi essencial para entender como os protocolos *beacons* foram propostas e implementadas, como é o comportamento do *Raspberry Pi*, entre outros. Em conjunto, o estudo das tecnologias foi de extrema importância para entender o funcionamento e utilização do *RPi* e *beacon*.

Certas atividades de extrema importancia ainda serão realizadas, mas o projeto ainda tem muitos frutos para render, caminhando em uma boa direção. A área de Internet das Coisas é uma área extremamente promissora, com áreas de pesquisa e implementação bem vastas.

REFERÊNCIAS

- APPLE INC. *Getting Started with iBeacon*. <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>, 2014.
- ARGENOX. *A BLE Advertising Primer*. <http://www.agenox.com/bluetooth-low-energy-ble-v4-0-development/library/a-ble-advertising-primer/>, 2015.
- ASHTON, K. *That 'Internet of Things' Thing: In the real world, things matter more than ideas*. <http://www.rfidjournal.com/articles/view?4986>, 2009.
- AUSTIN. *Understanding the different types of BLE Beacons*. <https://developer.mbed.org/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/>, 2015.
- BEN. *What is an Arduino?* <https://learn.sparkfun.com/tutorials/what-is-an-arduino>, 2015.
- BLUETOOTH SIG. *Bluetooth Smart Technology: Powering the Internet of Things*. <http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx>, 2015.
- BLUETOOTH SIG. *Generic Access Profile*. <https://www.bluetooth.org/en-us/specification/assigned-numbers/generic-access-profile>, 2015.
- CANONICAL LTD. *Snappy Ubuntu*. <https://developer.ubuntu.com/en/snappy/>, 2015.
- GOLDMAN SACHS. *What is the Internet of Things?* <http://www.goldmansachs.com/our-thinking/outlook/iot-infographic.html>, 2014.
- GROTHAUS, M. *How Apple Thinks About Smart Homes*. <http://www.fastcolabs.com/3034919/how-apple-thinks-about-smart-homes>, 2014.
- GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, <http://www.sciencedirect.com/science/article/pii/S0167739X13000241>, v. 29, n. 7, p. 1645–1660, Sept 2013.
- JJNEBEKER. *Can RaspberryPi with BLE Dongle detect iBeacons?* <http://stackoverflow.com/questions/21733228/can-raspberrypi-with-ble-dongle-detect-ibeacons>, 2014.
- KASTRENAKES, J. *Macy's begins iBeacon shopping test, will send alerts to your iPhone when you enter stores*. <http://www.theverge.com/2013/11/21/5129336/macys-apple-ibeacon-support-herald-union-square-stores-shopkick>, 2013.
- MAKER SHED. *Raspberry Pi Comparison Chart*. <http://www.makershed.com/pages/raspberry-pi-comparison-chart>, 2015.
- MICROSOFT. *A Internet das suas coisas*. <https://dev.windows.com/pt-br/iot>, 2015.

NASCIMENTO, R. *O que, de fato, é internet das coisas e que revolução ela pode trazer?* <http://computerworld.com.br/negocios/2015/03/12/o-que-de-fato-e-internet-das-coisas-e-que-revolucao-ela-pode-trazer>, 2015.

PRESS, G. *It's Official: The Internet Of Things Takes Over Big Data As The Most Hyped Technology.* <http://www.forbes.com/sites/gilpress/2014/08/18/its-official-the-internet-of-things-takes-over-big-data-as-the-most-hyped-technology/>, 2014.

RASPBERRY PI. *Downloads.* <https://www.raspberrypi.org/downloads/>, 2015.

RASPBERRY PI. *What is a Raspberry Pi?* <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>, 2015.

SORREL, C. *Just What Is An Arduino, And Why Do you Want One?* <http://www.wired.com/2008/04/just-what-is-an/>, 2008.

TEIXEIRA, F. *Tudo o que você precisa saber para começar a brincar com iBeacons.* <http://arquiteturadeinformacao.com/ux-em-espacos-fisicos/tudo-o-que-voce-precisa-saber-para-comecar-a-brincar-com-ibeacons/>, 2014.

THIBODEAU, P. *Um em cada cinco desenvolvedores já trabalha em projetos de IoT.* <http://computerworld.com.br/um-em-cada-cinco-desenvolvedores-ja-trabalham-em-projetos-de-iot>, 2015.

UBUNTU MATE. *About Ubuntu Mate.* <https://ubuntu-mate.org/about/>, 2015.

WANDSCHNEIDER NIRDHAR KHAZANIE, M. A. M. *Eddystone Protocol Specification.* <https://github.com/google/eddystone/blob/master/protocol-specification.md>, 2015.