

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**

Trabalho de Conclusão de Curso 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
2015

GABRIEL LUIZ BASTOS OLIVEIRA

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

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação da Faculdade de Ciências da Universidade Estadual Paulista Júlio de Mesquita Filho UNESP, Câmpus de Bauru.

BANCA EXAMINADORA

Aprovado em ____/____/____.

Prof. Dr. Eduardo Martins Morgado
Orientador

Prof. Dr. Aparecido Nilceu Marana

Profa. Dra. Márcia A. Z. Meira e Silva

BAURU
2015

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Eduardo Morgado, por todo o tempo e trajetória no LTIA¹, pela confiança, apoio e incentivo. Agradeço também pelo auxílio em todas as etapas desse projeto, desde a definição do tema até a apresentação final.

A Henrique Lopes Santos, pela ajuda na realização das fotos e também criação do logotipo para o protótipo.

A Elisa Castro, pelo apoio na correção e melhorias na digitação do texto contido nesse documento.

A Giovanna Aguirre, pelo apoio, incentivo, paciência e companhia nos momentos mais importantes de minha vida.

A todos os colegas do laboratório, pela incrível parceria, trabalho em equipe e organização de projetos. Os quatro anos percorridos com eles foram essenciais para meu crescimento profissional e pessoal.

A todos os professores, pois com muito esforço diário passaram não somente as informações técnicas e práticas de cada disciplina, mas lições que servirão para toda vida.

A todos os colegas de curso, que de alguma forma auxiliaram a chegar ao último semestre da melhor maneira possível.

Aos meus amigos e familiares, pelo apoio e incentivo na escolha e decorrer do curso, com muita compreensão e encorajamento.

¹ Laboratório de Tecnologia da Informação Aplicada

*"Que eu não perca a vontade de ajudar as pessoas,
mesmo sabendo que muitas delas são incapazes
de ver, reconhecer e retribuir esta ajuda."
(Francisco Cândido Xavier)*

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.

LISTA DE ILUSTRAÇÕES

Figura 1 – <i>RPi</i> versão A+ (esquerda), versão B+ (centro) e versão 2 Modelo B (direita)	14
Figura 2 – Separação da banda 2.4 GHz para bluetooth e WiFi.	16
Figura 3 – Modelo de <i>BLE Advertising Packet</i>	16
Figura 4 – Modelo de <i>beacon</i> proprietário: MPact, da Zebra Technologies Corporation.	17
Figura 5 – <i>Payload</i> do pacote iBeacon.	19
Figura 6 – Pacote do <i>AltBeacon</i>	20
Figura 7 – <i>RPi</i> 2 modelo B utilizado nesse projeto.	22
Figura 8 – Orico BTA-406 a esquerda e EDUP N8508GS a direita.	22
Figura 9 – Cartão microSD SanDisk Ultra Class 10 utilizado no projeto.	22
Figura 10 – Conexão com o <i>RPi</i> via <i>SSH</i>	23
Figura 11 – Moto Maxx (esquerda) e iPad Mini (direita).	24
Figura 12 – <i>Beacon</i> Zebra MPact utilizado para testes.	24
Figura 13 – <i>Beacon</i> configurado na <i>Toolbox</i> apresentando a porcentagem de bateria.	25
Figura 14 – Caixa de metal utilizada para o protótipo	26
Figura 15 – Parte interna do protótipo	26
Figura 16 – Protoboard na etapa de estudo	26
Figura 17 – Primeiro teste realizado, com <i>RPi</i> e <i>beacon MPact</i>	28
Figura 18 – Software <i>hcitool</i> executando	29
Figura 19 – Software <i>hcidump</i> executando	29
Figura 20 – Teste com movimentação do <i>beacon</i>	30
Figura 21 – <i>RPi</i> posicionado de outra maneira	30
Figura 22 – TeXworks, programa utilizado para escrever o código \TeX	36
Figura 23 – BibDesk, programa utilizado para editar as referências do \TeX	37

LISTA DE TABELAS

Tabela 1 – Comparativo entre os modelos de <i>Raspberry Pi</i>	14
Tabela 2 – <i>BLE Physical Layer</i>	15
Tabela 3 – Exemplo de aplicação	18

LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet of Things</i> - Internet das Coisas
DIY	<i>Do It Yourself</i> - Faça Você Mesmo
RPi	<i>Raspberry Pi</i>
GPIO	<i>General Input and Output</i>
BLE	<i>Bluetooth Low Energy</i>
LTIA	Laboratório de Tecnologia da Informação Aplicada

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVOS	12
2.1	Objetivos Gerais	12
2.2	Objetivos Específicos	12
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Internet das Coisas	13
3.2	Raspberry Pi	13
3.3	Bluetooth Low Energy	15
3.4	Beacon	17
3.4.1	<i>iBeacon</i>	18
3.4.2	Outros Protocolos	19
4	MATERIAIS E MÉTODOS	21
4.1	Métodos e Etapas	21
4.2	Materiais Utilizados	21
4.2.1	Raspberry Pi e Acessórios	21
4.2.2	Computadores e Softwares	23
4.2.3	Smartphones e Tablets	23
4.2.4	Beacons	23
4.2.5	Acessórios para o Protótipo	25
5	TECNOLOGIAS E FERRAMENTAS	27
5.1	Raspberry Pi	27
5.2	Node.js	27
5.3	CouchDB	27
5.4	Git	27
5.4.1	GitHub	27
6	EXPERIMENTOS E RESULTADOS	28
7	CONCLUSÃO	32
	REFERÊNCIAS	33
	APÊNDICE A – LATEX	35
A.1	abnTeX2	35

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 in-

ternet cabeadas (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 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).

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)

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

3.1 Internet das Coisas

A área de IoT está em uma onda crescente, com um grande número de pessoas investindo 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 Frequênci) 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.

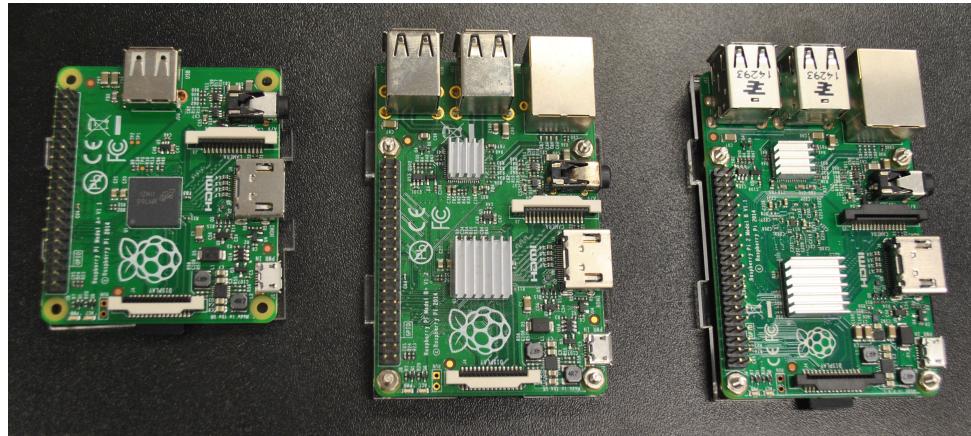
3.2 Raspberry Pi

Quando se pesquisa algo sobre 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 Tabela 1 e figura Figura 1.

Segundo Raspberry Pi (2015b), a fundação responsável por esse pequeno dispositivo criou com intuito de ser utilizado por crianças e pessoas carentes do mundo todo para aprender programação e computação. É facilmente configurável, utilizando um cartão de memória como HD para o sistema operacional e dados.

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 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*.

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



Fonte: elaborado pelo autor

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)

Atualmente existem diferentes sistemas operacionais portados para o *RPi*.

- 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).

3.3 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).

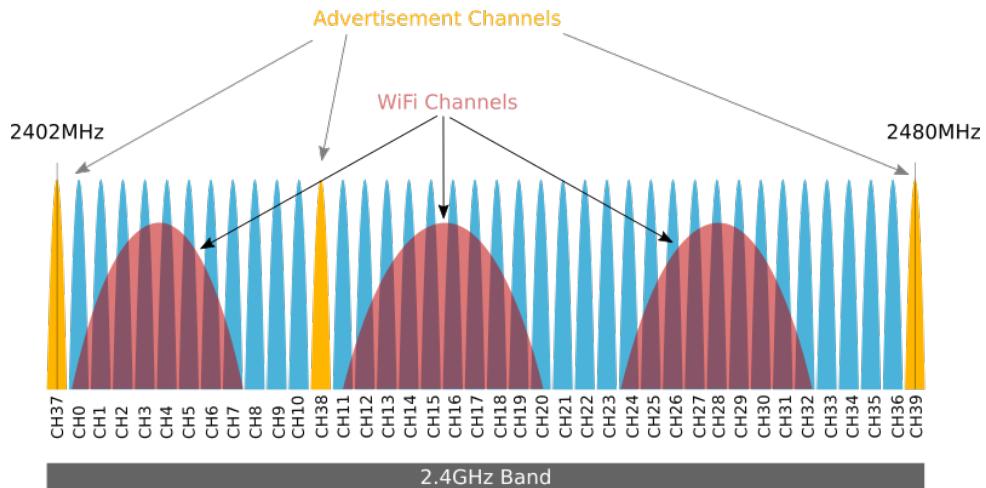
Interessante notar que esses canais estão posicionados em forma bastante estratégica, no começo, final e meio da banda de 2.4 GHz. Isso se deve para aumentar a eficácia, evitando todos os canais ficarem lotados ou com muita interferência. (ARGENOX, 2015).

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.

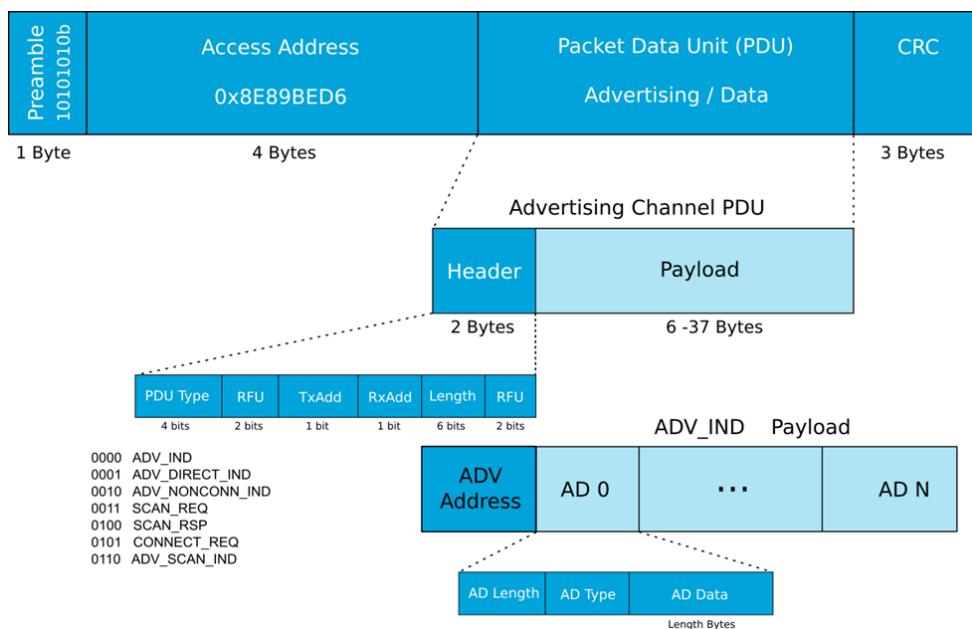
Um *BLE Advertisement Packet* é formado conforme figura 3. O preâmbulo, *access address* e CRC são informações para formação do pacote. Os dados estão de fato dentro do PDU (*Packet Data Unit*). O cabeçalho de 2 bytes informa o tamanho do *payload* (carga de dados), além de informações relevantes como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

O importante do *BLE Advertising Packet* é o tipo de anúncio feito, ou quais são

Figura 2 – Separação da banda 2.4 GHz para bluetooth e WiFi.



Fonte: (ARGENOX, 2015)

Figura 3 – Modelo de *BLE Advertising Packet*.

Fonte: (ARGENOX, 2015)

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).

3.4 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.



Fonte: 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.

3.4.1 iBeacon

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.

A Apple adaptou o pacote genérico de *beacon* para transmitir três dados:

- UUID**: Identificador único formado de 16 bytes (128 bits). Focado em ser único para cada aplicação. Cada aplicação deve ter um único UUID.
- 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 Major.

Um exemplo de aplicação é citado na tabela 3. Utiliza-se um único UUID para todas as lojas, um número Major por loja e um Minor por departamento, podendo este último ser repetido entre as lojas.

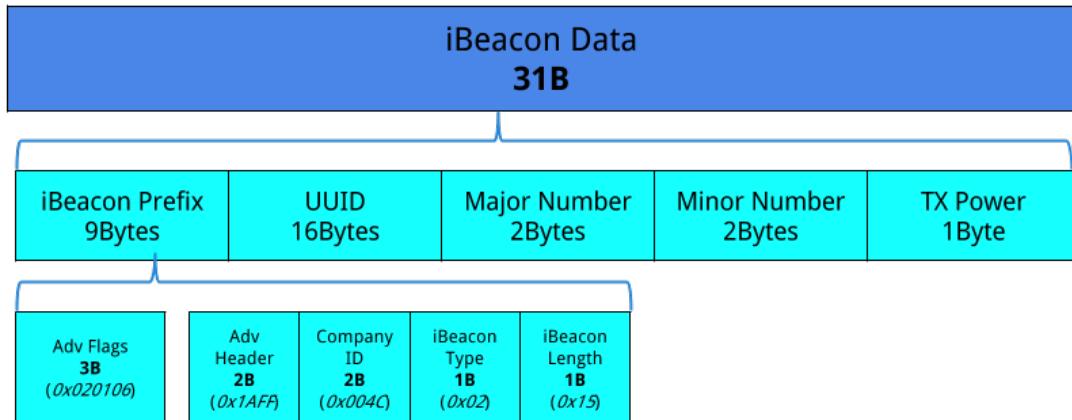
Tabela 3 – Exemplo de aplicação

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 prefixo determinado para o protocolo *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.

Fonte: (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).

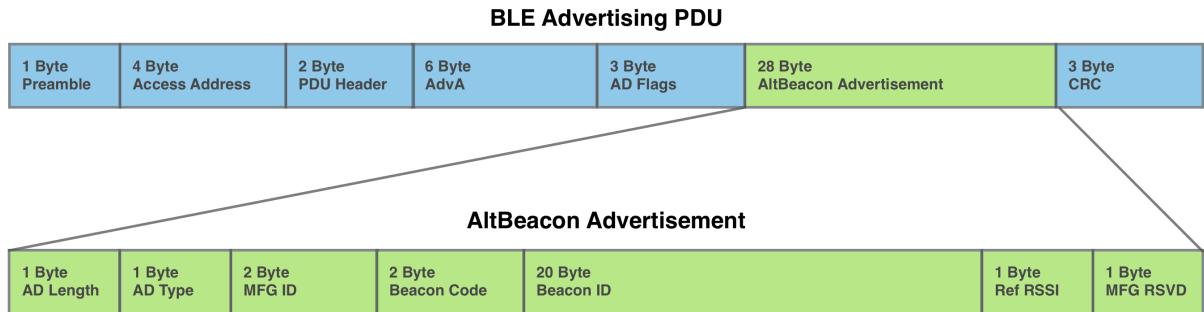
Uma outra aplicação interessante citada por Austin (2015) é a possibilidade do smartphone transmitir pacotes *iBeacon*, sem necessidade de um hardware externo. Dessa forma, pode-se por exemplo automatizar o check-in em um evento e rastrear o movimento dos usuários entre os estabelecimentos.

3.4.2 Outros Protocolos

Existem mais 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).

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*.

Figura 6 – Pacote do *AltBeacon*.

Fonte: (AUSTIN, 2015)

- 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 MATERIAIS E MÉTODOS

4.1 Métodos e Etapas

O projeto foi dividido em etapas para facilitar o desenvolvimento completo. A primeira etapa foi o levantamento bibliográfico relacionado ao tema, realizadas buscas relacionadas aos assuntos: *Raspberry Pi*, comunicação via *Bluetooth Low Energy*, *beacons*. Concomitantemente foi realizado o estudo das tecnologias, levando em conta suas capacidades, limitações, aplicações, etc.

A segunda etapa 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 nas próximas etapas, definindo assim um escopo inicial de funcionalidades que o sistema terá, assim como outras tarefas a serem realizadas. O processo está detalhado no Capítulo 6 - Experimentos e Resultados.

A terceira etapa foi a implementação e testes do protótipo. As funcionalidades foram desenvolvidas por parte e em seguida testadas, seguindo a metodologia de desenvolvimento ágil.

Todo o projeto foi desenvolvido no espaço do LTIA¹, da Universidade Estadual Paulista "Júlio de Mesquita Filho" - Campus Bauru.

4.2 Materiais Utilizados

4.2.1 Raspberry Pi e Acessórios

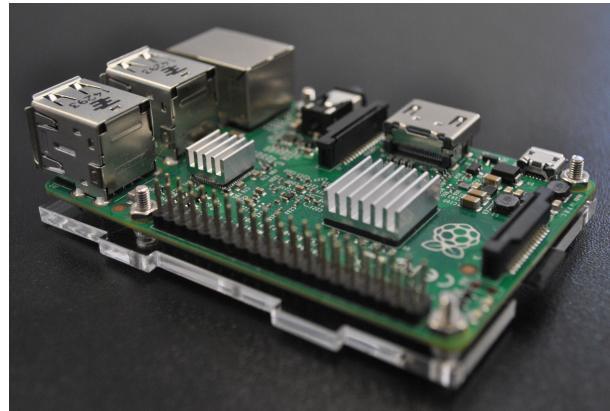
A versão do Raspberry Pi escolhida para o projeto foi a 2 Modelo B, por ter mais processamento e mais memória RAM (Figura 7), conforme informado na seção 3.2.

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 foram Orico BTA-406 (*bluetooth*) e EDUP N8508GS (*WiFi*), conforme Figura 8. O *RPi* e os adaptadores são de propriedade do autor.

O sistema operacional executado no *RPi* é 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. O sistema utilizado foi o Raspbian Wheezy. Durante o desenvol-

¹ Laboratório de Tecnologia da Informação Aplicada - <<http://www.ltia.fc.unesp.br/>>

Figura 7 – RPi 2 modelo B utilizado nesse projeto.



Fonte: elaborado pelo autor

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



Fonte: elaborado pelo autor

vimento do projeto, a versão Raspbian Jessie foi lançada, com inúmeras melhorias, porém preferiu-se por permanecer na versão Wheezy por possíveis incompatibilidades com os softwares utilizados.

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



Fonte: elaborado pelo autor

4.2.2 Computadores e Softwares

A conexão com o *RPi* foi realizada utilizando *SSH*², sem necessidade de monitor e teclado. O computador utilizado no projeto foi um MacBook Pro com sistema Mac OS X 10.10 (posteriormente atualizado para 10.11), de propriedade do autor. Nesse sistema o uso de *SSH* é simples, basta abrir o aplicativo Terminal e utilizar o comando "*ssh usuário@computador*", conforme Figura 10. Esse tipo de abordagem é bastante utilizada para conexão a servidores na nuvem, para executar comandos, softwares, entre outros.

Figura 10 – Conexão com o *RPi* via *SSH*.

```
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 ~ $
```

Fonte: elaborado pelo autor

4.2.3 Smartphones e Tablets

Foi utilizado o smartphone Moto Maxx com sistema Android 5.0.1 e o tablet iPad mini Retina com sistema iOS 8.4 conforme Figura 11. 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. O Moto Maxx é de propriedade do autor, e o iPad é de propriedade do LTIA, foi utilizado nas dependências do laboratório.

4.2.4 Beacons

O *beacon* utilizado para testes foi o *Zebra MPact*, conforme Figura 12. 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

² Secure Shell

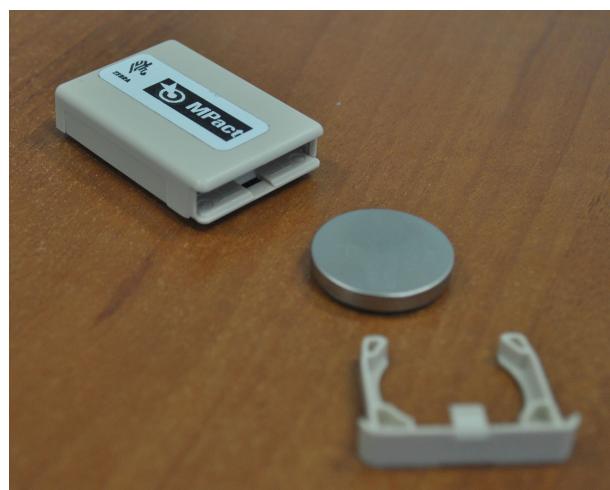
Figura 11 – Moto Maxx (esquerda) e iPad Mini (direita).



Fonte: elaborado pelo autor

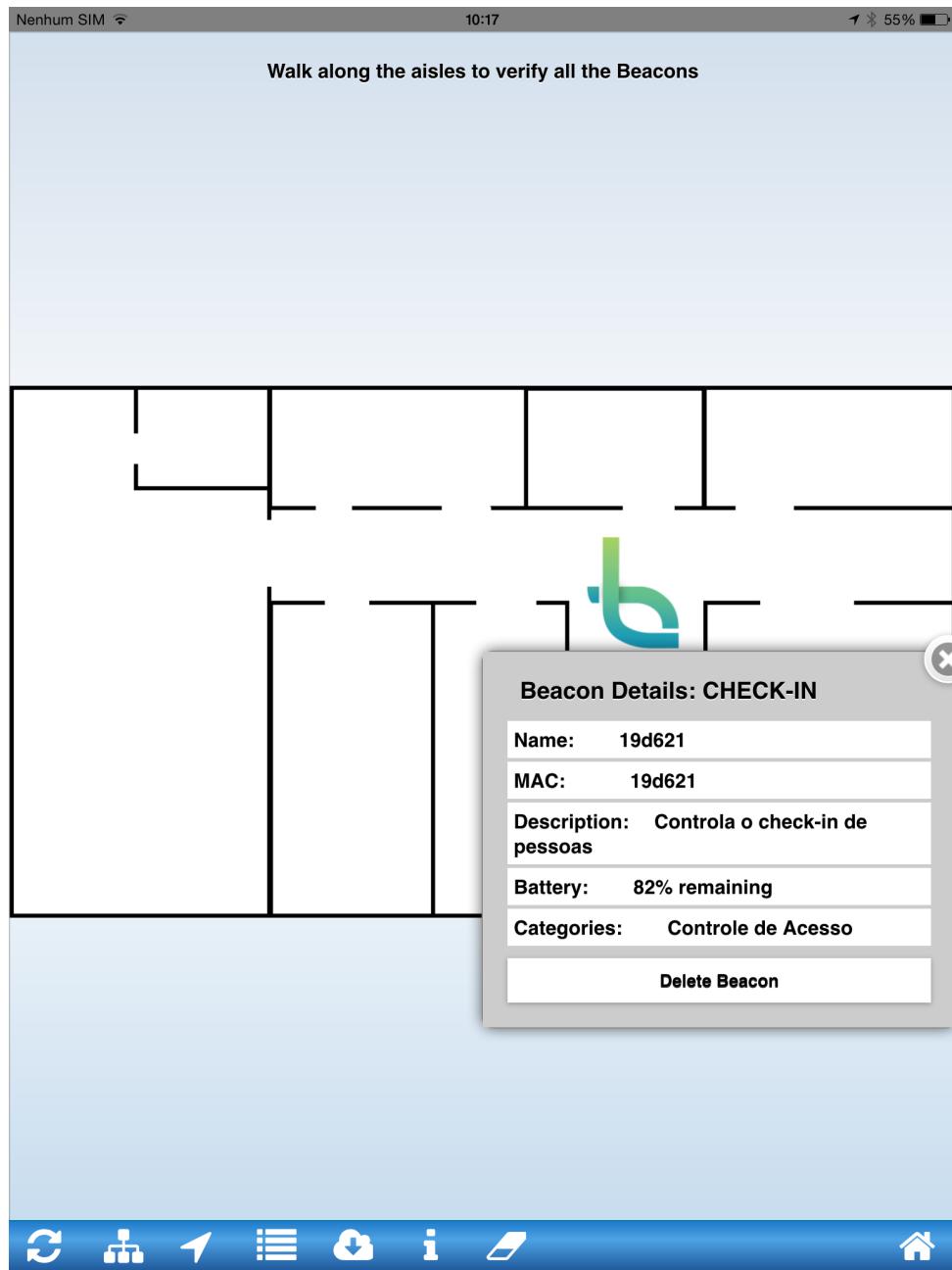
Debian 8.1. Esse sistema foi utilizado somente para conhecimento da tecnologia *beacon* e atualização do *firmware*, somente disponível por essa *Toolbox*. Esse software também apresenta a porcentagem de bateria, conforme Figura 13. Permite a configuração de vários *beacons* simultaneamente, além de alterar o modo de funcionamento, de *iBeacon* para *MPact*, protocolo criado pela fabricante. Os *beacons* são de propriedade do LTIA, e foram utilizados dentro das dependências do laboratório em todas as etapas.

Figura 12 – Beacon Zebra MPact utilizado para testes.



Fonte: elaborado pelo autor

Figura 13 – Beacon configurado na *Toolbox* apresentando a porcentagem de bateria.



Fonte: elaborado pelo autor

4.2.5 Acessórios para o Protótipo

O protótipo foi criado em uma caixa de metal (Figura 14), furada conforme projeto e pintada de preto. Esse processo está detalhado na ??.

Para o protótipo final foram utilizados seis LEDs, resistores, barra de pinos, um display azul de 16 colunas por 2 linhas, três botões do tipo clique, cabos *flat* do tipo IDE e *floppy*, placas com furação para soldagem de componentes, cabos finos removidos de um

Figura 14 – Caixa de metal utilizada para o protótipo

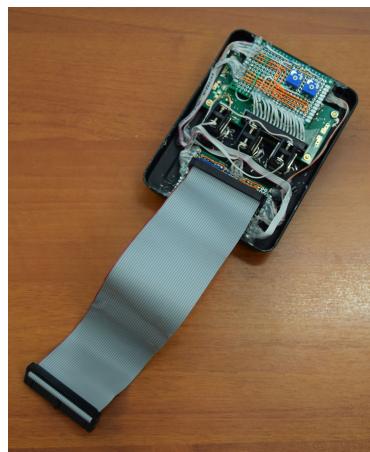


Fonte: elaborado pelo autor

cabo de rede *ethernet* para ligações entre os componentes e cola quente para fixação dos componentes internos. Esses materiais podem ser vistos na Figura 15.

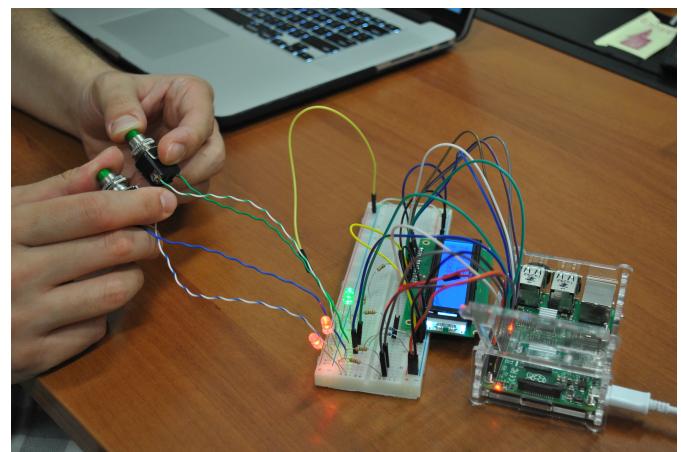
Durante a fase de estudo das tecnologias utilizou-se uma protoboard e fios de conexão (Figura 16) para testes de código, modo de conexão, entre outros, detalhado na ??.

Figura 15 – Parte interna do protótipo



Fonte: elaborado pelo autor

Figura 16 – Protoboard na etapa de estudo



Fonte: elaborado pelo autor

5 TECNOLOGIAS E FERRAMENTAS

5.1 Raspberry Pi

O *RPi* é a tecnologia mais importante desse projeto. Já foi explicado na seção 3.2 e subseção 4.2.1, porém nessa parte será abordado diretamente seu uso.

5.2 Node.js

5.3 CouchDB

5.4 Git

Git é um software de controle de versão para repositórios. Registra as mudanças nos arquivos ao longo do tempo, de forma que as versões possam ser recuperadas. (CHACON, 2014)

5.4.1 GitHub

6 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 17.

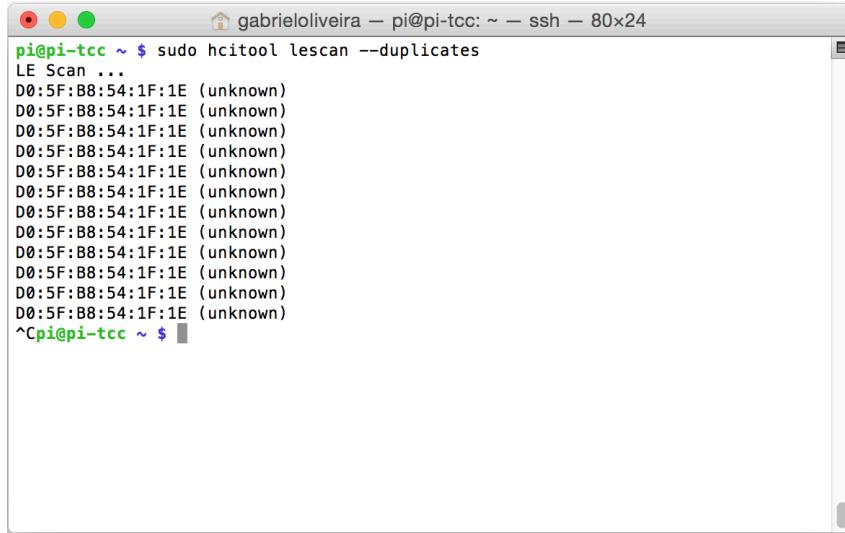
Figura 17 – Primeiro teste realizado, com *RPi* e *beacon MPact*



Fonte: elaborado pelo autor

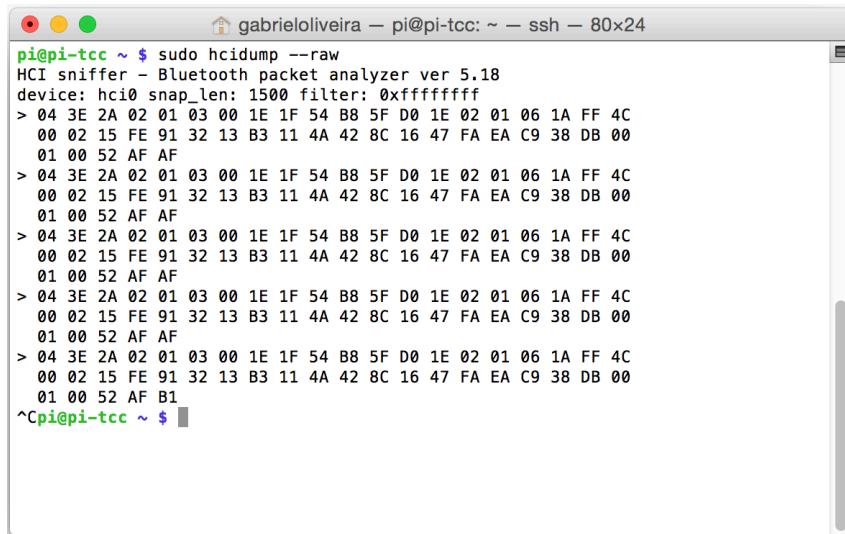
O computador ficou conectado via SSH com o *RPi*, recebendo as informações de leitura de pacotes *BLE*. Os softwares utilizados para isso foram, conforme Jjnebeker (2014), os seguintes:

- a) **hcitool**: configurado da maneira *hcitool lescan -duplicates*, faz um scan na frequência *BLE* procurando por dispositivos que estejam transmitindo, conforme figura 18.
- b) **hcidump**: em conjunto com o *hcitool*, apresenta todos os pacotes escaneados na rede. Executado da maneira *hcidump -raw*, conforme figura 19.

Figura 18 – Software *hcitool* executando


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

Fonte: elaborado pelo autor

Figura 19 – Software *hcidump* executando


```
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 B1
^Cpi@pi-tcc ~ $
```

Fonte: elaborado pelo autor

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 figura 20. 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 17 e 20, 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.

Figura 20 – Teste com movimentação do *beacon*



Fonte: elaborado pelo autor

O posicionamento do *RPi* foi alterado para testes, conforme figura 21. Com essa nova posição notou-se 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 21 – *RPi* posicionado de outra maneira



Fonte: 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 surgião.

7 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.argenox.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.
- CHACON, B. S. S. *Pro Git*. [S.I.]: Apress, 2014.
- 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-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.

APÊNDICE A – LATEX

Para o desenvolvimento desse documento foi utilizado uma ferramenta denominada TeX, um processador de texto baseado em comandos. Normalmente é grafado usando o comando (ou macro) `\TeX`, resultando em `\TeX`.

Não é comum o uso do TeX por si só. Utiliza-se uma outra ferramenta denominada LaTeX (ou `\LaTeX`), uma série de macros definidas para auxiliar o desenvolvimento.

Essa abordagem é muito utilizada no meio acadêmico e científico, pois o ponto mais importante dessa ferramenta é o conteúdo. O design final ficará sempre para o compilador gerar. Isso evita erros de formatação, como por exemplo devido a diferenças entre versões de programas.

Congressos e revistas científicas já disponibilizam arquivos no formato `.sty`¹ para que os pesquisadores enviem os artigos preenchidos em `.pdf` formatado conforme modelo próprio. Dessa forma todos os artigos seguirão o mesmo formato, auxiliando também os pesquisadores a escreverem os documentos rapidamente.

Por ser um documento de texto no formato `.tex`, qualquer editor comum como o Bloco de Notas do Windows pode ser utilizado. Dessa forma a compilação é feita por meio da linha de comando, com algumas opções de compiladores. As mais utilizadas são `pdflatex` e `xelatex`.

O meio mais utilizado para escrita de documentos `\LaTeX` de forma a facilitar a compilação são programas próprios. Existem diversas opções, mas as mais utilizadas são MiKTeX² para Windows e MacTeX³ para Mac OSX. O programa utilizado para esse documento foi o TeXworks, um editor de texto para `\TeX` multiplataforma e de código aberto, conforme imagem 22.

A.1 abnTeX2

Devido a esse Trabalho de Conclusão de Curso seguir as normas ABNT⁴ para escrita e formatação de documentos científicos, foi utilizado a biblioteca de macros abnTeX2 (abnTeX2)⁵ para formatação do documento conforme as normas vigentes. Essa biblioteca é livre, com código fonte aberto e disponível para qualquer um auxiliar.

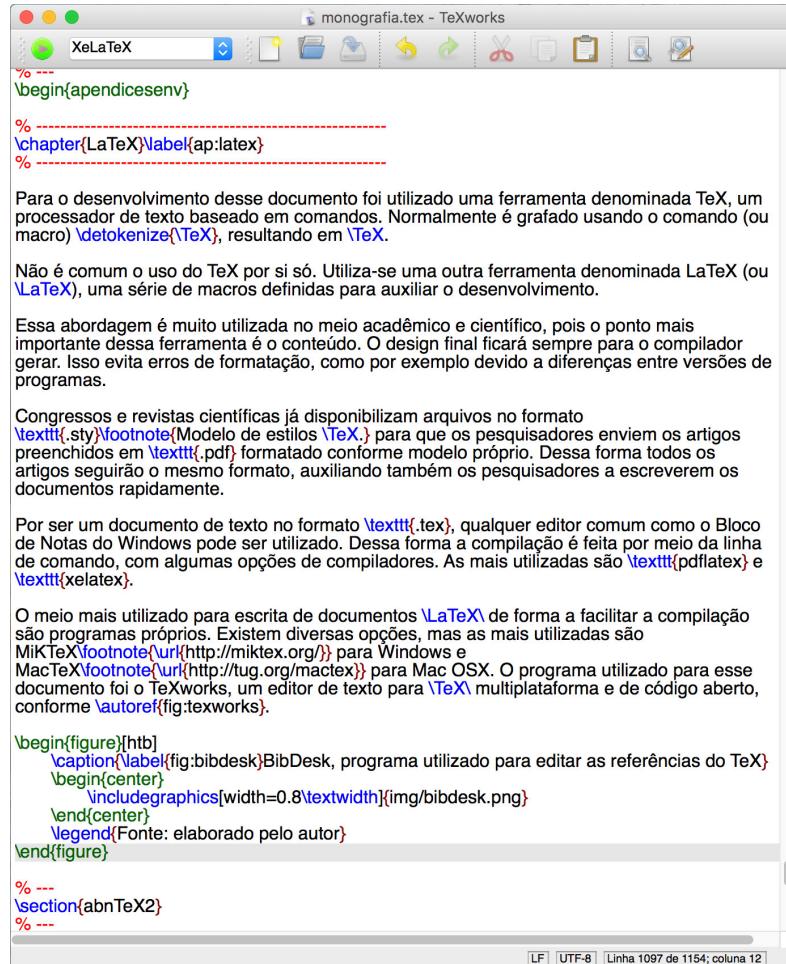
¹ Modelo de estilos `\TeX`.

² <<http://miktex.org/>>

³ <<http://tug.org/mactex>>

⁴ Associação Brasileira de Normas Técnicas.

⁵ <<https://github.com/abntex/abntex2>>

Figura 22 – TeXworks, programa utilizado para escrever o código \TeX 

Fonte: elaborado pelo autor

Uma ferramenta muito útil disponível com o abn $\text{\TeX}2$ é a criação das Referências Bibliográficas no formato ABNT de forma automática. Para tal, é criado um arquivo `.bib` com todas as informações de citações e referências utilizadas no texto.

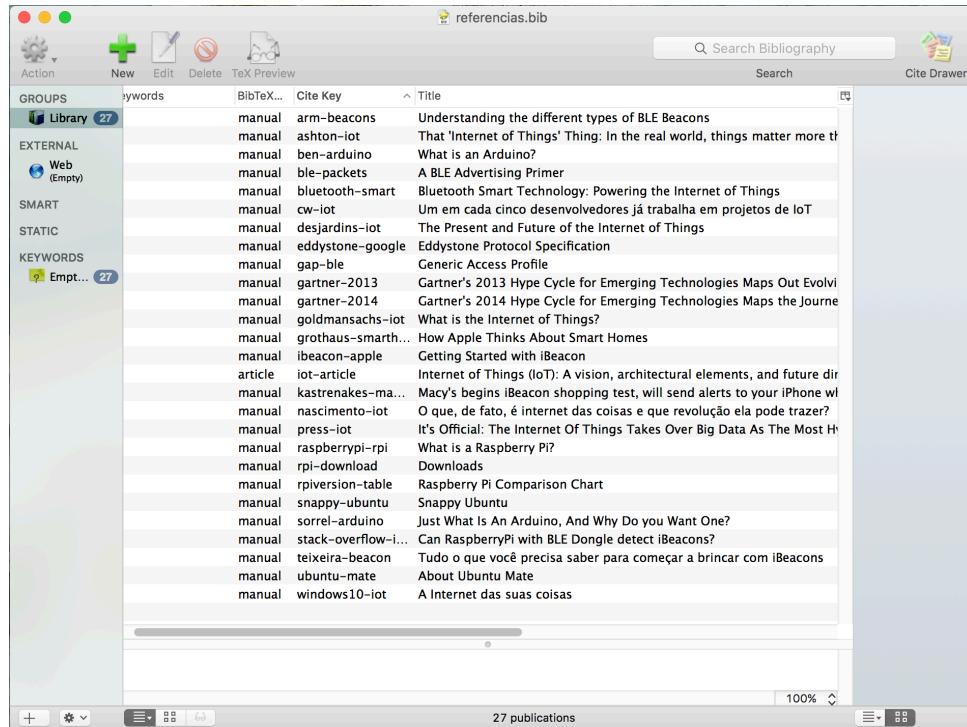
As chaves de citação são os valores que serão utilizados nas macros do abn $\text{\TeX}2$ para que as referências sejam corretamente configuradas. Por exemplo, se desejamos referenciar um artigo de Microsoft (2015), adicionamos todas as informações como nome dos autores, organização, data, edição, entre outros, e adicionamos a chave `windows10-iot`.

Utilizando a macro `\cite {windows10-iot}`, no local ficará o texto (MICROSOFT, 2015), e nas referências será adicionado conforme modelo ABNT: MICROSOFT. A Internet das suas coisas. <https://dev.windows.com/pt-br/iot>, 2015. Para citações *inline*, ou na mesma linha, o comando `\citeonline {windows10-iot}` é utilizado, gerando o texto Microsoft (2015).

O arquivo `.bib` pode ser criado manualmente ou por meio de programas. Para

esse artigo foi utilizado o programa BibDesk⁶ para Mac OSX, conforme Figura 23. Esse programa já cria o arquivo no formato correto com as chaves de citação configuradas.

Figura 23 – BibDesk, programa utilizado para editar as referências do TeX



Fonte: elaborado pelo autor

Além de formatar o documento, abnTeX2 também disponibiliza macros para criação de conteúdo citados na ABNT. Por exemplo, tabelas que seguem o formato do IBGE⁷ podem ser facilmente criadas com macros disponíveis, como exemplo da Tabela 2:

```
\begin{table}[htb]
\IBGETab{%
  \caption{\textit{BLE Physical Layer}}%
  \label{table:ble-physical}%
}%
\begin{tabular}{ccc}
\toprule
& BLE & Classic \\
\midrule
Modulação & GFSK 0.45 a 0.55 & GFSK 0.28 a 0.35 \\
\midrule

```

⁶ <<http://bibdesk.sourceforge.net/>>

⁷ Instituto Brasileiro de Geografia e Estatística

```
    Velocidade de Transferência & 1 Mbit/s & 1 Mbit/s \\
    \midrule
    Canais & 40 & 79 \\
    \midrule
    Espaçamento & 2 MHz & 1 MHz \\
    \bottomrule
\end{tabular}%
}{%
\fonte{\cite{ble-packets}}%
}
\end{table}
```

O código desse documento, assim como a versão em .pdf pode ser acessado pelo repositório no GitHub⁸. Está disponível para livre consulta, tanto de professores quanto de alunos que tenham interesse em saber mais sobre o \LaTeX .

⁸ <<https://github.com/gabrielb Oliveira/Monografia-Pearcon>>