

# SISTEMAS DE LOCALIZAÇÃO: EXPLORANDO A IPS - BEACONS

Djonathan Krause, Dalton Solano dos Reis – Orientador

**Resumo:** Este artigo descreve o desenvolvimento e testes realizados com beacons bluetooth onde o objetivo é criar uma plataforma de localização indoor utilizando este hardware. Os principais resultados obtidos foram..... As conclusões são.....

**Palavras-chave:** Ciência da computação. IPS. Localização Indoor. Beacons. Bluetooth.

## 1 INTRODUÇÃO

Atualmente, o sistema mais utilizados para localização é o Global Position System (GPS) (BRAMHE, 2017). Segundo Mackey (2017, p. 823, tradução nossa) “O surgimento da internet das coisas (IoT) e, especificamente, edifícios e casa inteligentes gerou um aumento no desejo por serviços de localização interna”. Entretanto, ambientes internos podem ser muito mais complexos e desafiadores devido ao grande número de obstáculos e *layouts* que podem ter (HAN, 2013). O GPS tem limitações como precisão, consumo de bateria e interferência de sinal não conseguindo fornecer a localização com precisão necessária para uma navegação interna (ROCHA, 2015).

As aplicações de um sistema de Indoor Positioning System (IPS) podem se dar em estabelecimentos comerciais onde seria possível quantificar os locais visitados dentro da loja podendo melhorar a disposição dos produtos. Ainda na área comercial, propagandas personalizadas poderiam ser mostradas em monitores para cada cliente visto que a localização do mesmo seria conhecida. Em hospitais, seria possível saber a localização de enfermeiros, médicos e pacientes permitindo o rápido acionamento destes quando necessário.

Visto que existe a necessidade de localizar pessoas e objetos em ambientes internos e a principal tecnologia de posicionamento utilizada, o GPS, pode não atender alguns dos requisitos necessários para a sua utilização num contexto IPS, é proposto o desenvolvimento de uma aplicação que faça possível a localização em ambientes internos. Os objetivos específicos são:

- permitir localizar o usuário em um ambiente interno;
- utilizar beacons bluetooth como ferramenta;
- analisar a precisão da posição obtida.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 CONCEITOS

A seguir são apresentados os principais conceitos utilizados como base de fundamentação teórica para este trabalho.

#### 2.1.1 BEACON

Dudhane e Pitambare (2015) explicam que beacons bluetooth são transmissores que usam Bluetooth Low Energy 4.0 (BLE) para emitir sinais que podem ser ouvidos por dispositivos compatíveis. Ainda segundo os mesmos autores, *Bluetooth Low Energy* é uma rede wireless usada para transmitir dados em pequenas distâncias.

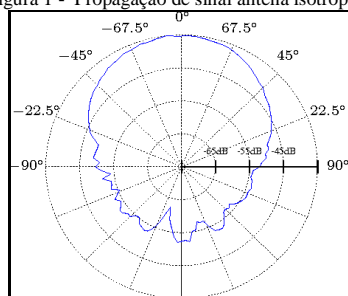
Um conceito bastante utilizado referente a beacons é o *measured power*. Esse é o valor do sinal recebido pelo receptor a um metro de distância. O *measured power* tem um valor individual para cada beacon independentemente de seu modelo ou marca visto que diversos fatores podem interferir nesta medição.

Beacons da Estimote, fabricante de beacons e de plataformas de desenvolvimento para utilização dos dispositivos, são utilizados em museus como o Guggenheim nos Estados Unidos. Lá, os beacons são utilizados junto com um aplicativo desenvolvido para o museu que auxiliam os visitantes a se localizarem e também fornecendo informações sobre as obras que estão expostas próximas a ele (ESTIMOTE, 2018).

#### 2.1.2 RSSI

Segundo Xu, Yang e Jiang (2011, p. 1) o Received Signal Strength Indicator (RSSI) é uma métrica da qualidade do sinal de rádio emitido por um dispositivo. Conforme Larson (2015, p. 11) afirma, praticamente todos os dispositivos Bluetooth Low Energy (BLE) utilizam antenas não isotrópicas. Antenas não isotrópicas são as que não transmitem ondas de rádio igualmente em todas as direções do espaço (GIACOMIN, 2006). A Figura 1 mostra a propagação do sinal transmitido por uma antena não isotrópica.

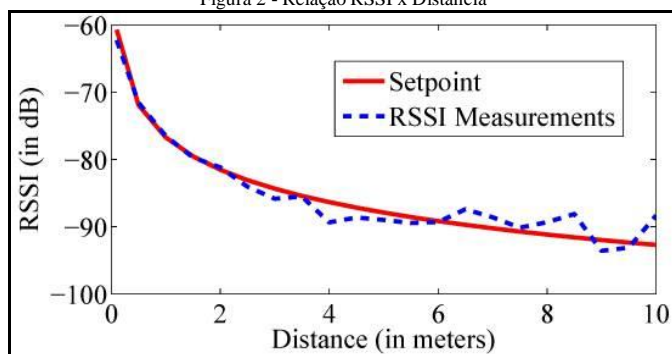
Figura 1 - Propagação de sinal antena isotrópica



Fonte: Bruinjes (2015).

Além da característica isotrópica, outro fator que pode interferir no sinal é o ambiente onde o emissor está. Como Rôbesaat (2017, p. 6) explica em seu trabalho, o RSSI é influenciado por interferências do ambiente como reflexões de ondas do sinal. Rôbesaat (2017, p. 8) também mostra que quanto maior a distância entre o emissor e o receptor, maior o valor do RSSI, entretanto, Parameswaran, Husain e Upadhyaya (2009, p. 3) afirma que a relação entre distância e valor do RSSI não é linear. Isso pode ser visto na Figura 2.

Figura 2 - Relação RSSI x Distância

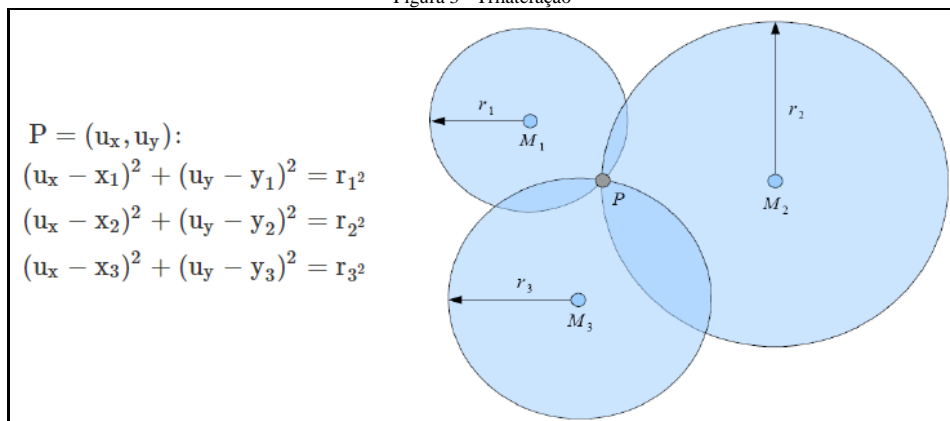


Fonte: Rôbesaat (2017).

### 2.1.3 TRILATERAÇÃO

Como Awad, Frunzke e Dressler (2007, p. 6) afirmam, a trilateração é o algoritmo de localização mais utilizado quando é necessário encontrar a localização de um vértice baseando-se em outros vértices conhecidos. Ainda segundo os autores, se as distâncias de pelo menos três objetos de referência são conhecidas, é possível encontrar a posição do vértice com a equação demonstrada na Figura 3 - Trilateração.

Figura 3 - Trilateração

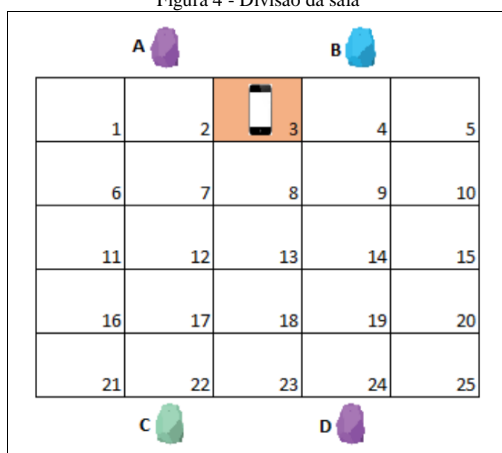


Fonte: Awad, Frunzke e Dressler (2007).

#### 2.1.4 TÉCNICA FINGERPRINT

A técnica de *fingerprint* consiste em duas etapas. A primeira sendo a calibração ou treinamento onde é necessário dividir o ambiente que será mapeado em subáreas e efetuar a coleta da intensidade de sinal RSSI em cada uma dessas subáreas (RECK, 2016). Reck (2016) ilustra em seu trabalho uma sala de cem metros quadrados divididas em subáreas de dois metros quadrados conforme é possível ver na Figura 4.

Figura 4 - Divisão da sala



Fonte: Reck (2016).

Cada subárea deverá ter uma medição por beacon. A complexidade do ambiente pode fazer que sejam necessárias várias medições para obter-se uma média. Os valores recebidos deverão ser mantidos em um banco de dados.

A segunda etapa da técnica de *fingerprint* é feita durante o uso da aplicação. O sinal recebido deverá ser comparado com os medidos anteriormente assim podendo determinar em qual subárea o receptor está. É possível adotar uma abordagem determinística para caracterizar cada posição do mapa, assim cada subárea terá indicadores como força do sinal, valor médio e desvio padrão de todas as medições feitas na primeira fase (RECK, 2016).

2.2 TRABALHOS CORRELATOS

A seção a seguir apresenta os trabalhos correlatos que possuem características e funcionalidades semelhantes ao que está sendo apresentado neste artigo. O Quadro 1 apresenta o aplicativo TÔ AQUI: Aplicativo Para georreferenciamento em ambientes restritos.

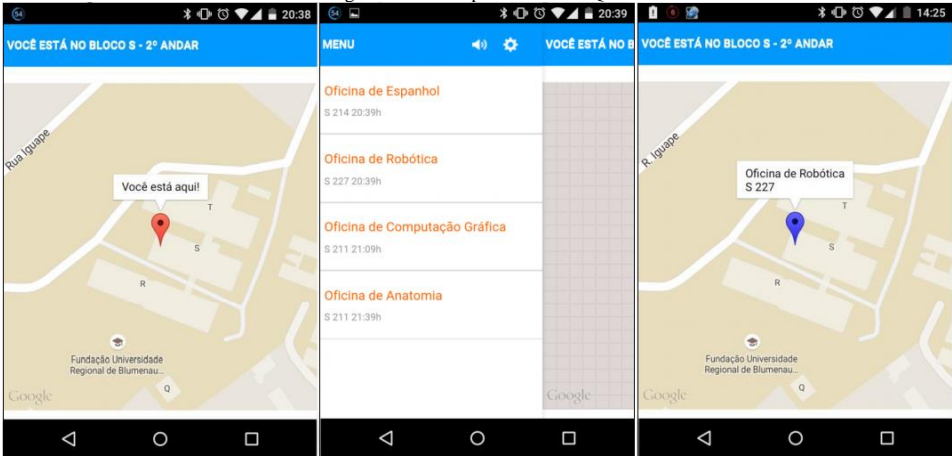
Quadro 1 - Trabalho Correlato 1	
Referência	Rocha (2015)
Objetivos	Possibilitar que o usuário se localize na FURB por meio de um mapa que indique sua posição atual assim aprimorando a localização em ambientes restritos utilizando beacons e AGPS.
Principais funcionalidades	Visualizar eventos e ambientes próximos e visualizar posição atual no mapa.
Ferramentas de desenvolvimento	Foi utilizado o PhoneGap para o desenvolvimento do aplicativo, o que permitiu que este fosse disponibilizado para Android e iOS. Beacons foram utilizados para melhorar a precisão da localização.
Resultados e conclusões	Foi possível instruir o usuário a ir do ponto A ao ponto B dentro de um prédio de forma acurada mesmo sem sinal de GPS.

Fonte: elaborado pelo autor.

Observa-se que Rocha (2015) utilizou ferramentas parecidas com as utilizadas neste trabalho, como beacons e a plataforma PhoneGap que permite o desenvolvimento de aplicativos multiplataforma tal qual o Ionic. O protocolo de comunicação utilizado foi o iBeacon.

As funcionalidades incluem visualizar eventos e ambientes próximos e visualizar a posição atual no mapa além de notificações serem mostradas quando o usuário passar por um evento próximo. A Figura 5 mostra telas que mostram a posição atual do usuário, uma lista de eventos próximo e os detalhes e localização de um evento específico. Também é possível ouvir as informações da localização atual.

Figura 5 - Prints aplicativo TÔ AQUI



Fonte: Rocha (2015)

O autor conclui que é possível aprimorar a localização de um usuário de forma mais precisa do que o GPS dentro de um ambiente restrito utilizando beacons bluetooth. Após testes de utilização os participantes foram questionados sobre a eficiência da aplicação, 90% avaliou o aplicativo como bom e 10% como muito bom provando que a utilização dos beacons pode aprimorar a percepção do usuário referente sua localização.

No Quadro 2 é apresentado o FURB-MOBILE: Sistema móvel multiplataforma para navegação em rotas internas. Trabalho de conclusão de curso desenvolvido por Rocha (2016).

Quadro 2 - Trabalho Correlato 2

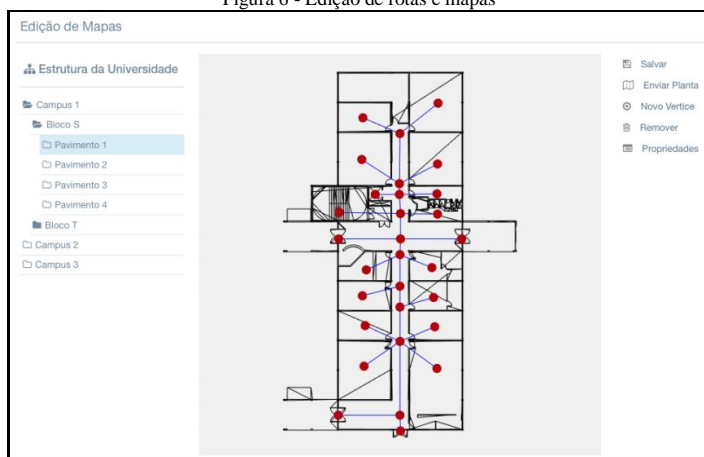
Referência	Rocha (2016)
Objetivos	Apresentação de rotas virtuais para auxiliar a movimentação do usuário no mundo real.
Principais funcionalidades	Navegar pelo mapa, selecionar origem e destino e mostrar rota.
Ferramentas de desenvolvimento	O aplicativo foi desenvolvido com a plataforma PhoneGap. A linguagem utilizada no servidor foi Java junto com o banco de dados PostgreSQL. A aplicação web foi desenvolvida com o framework AngularJS.
Resultados e conclusões	Foi desenvolvido um aplicativo que possibilitasse a apresentação de uma rota virtual para auxiliar a movimentação do usuário no mundo real. Entretanto, não foi possível apresentar a posição real do usuário em relação aos blocos dos prédios da universidade.

Fonte: elaborado pelo autor.

A aplicação web desenvolvida por Rocha (2016) consiste em um sistema onde é possível carregar uma planta baixa com o formato OBJ que são geradas pelo editor gráfico SketchUp e editar rotas tendo como base a planta carregada. Após carregar a planta baixa no sistema, é possível efetuar a edição dos mapas e rotas. Na edição, vértices podem ser adicionados ao mapa. Um vértice pode ter três tipos, conector, terminal ou acesso. Segundo o autor (2016), “O tipo conector é normalmente o mais utilizado, pois deve ser utilizado para representar uma conexão, esquina ou curva no caminho”. Já o tipo terminal é “[...] um ponto de início ou parada, uma sala, um banheiro, entre outros, ou seja, um lugar para onde, ou de onde, o usuário pode ir ou esta” (ROCHA, 2016), ainda segundo o autor. Por fim, Rocha (2016, p.46) explica que o vértice de tipo acesso cria uma aresta, ou seja, uma ligação entre o vértice de origem e o escolhido como destino. Um acesso permite o cadastro de uma custo em metros que represente a distância entre os pontos. As coordenadas geográficas são propriedades tanto dos vértices quanto dos blocos da construção.

A Figura 6 mostra a tela de edição de mapas e criação de rotas. Os pontos vermelhos são os vértices que podem ser conectores, terminais ou acessos e as linhas azuis são as arestas ou ligações entre os vértices. É possível ver no lado esquerdo que o sistema permite o cadastro de um edifício com mais de um pavimento além de organizar construções que tenham mais de um prédio em blocos ou setores como foi exemplificado com os blocos e campus da FURB.

Figura 6 - Edição de rotas e mapas



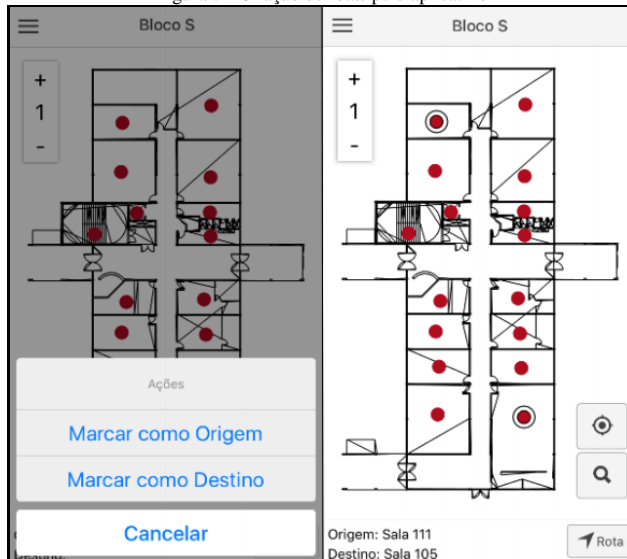
Fonte: Rocha (2016).

O trabalho de Rocha (2016) também disponibilizou um aplicativo móvel para que o usuário possa navegar por um mapa carregado e configurado na aplicação web. Ao iniciar o aplicativo o usuário pode escolher um vértice de origem e outro de destino clicando sobre ele como demonstrado na Figura 7.

Commented [DK1]: Citação OK?

Commented [DK2]: Precisa da página?

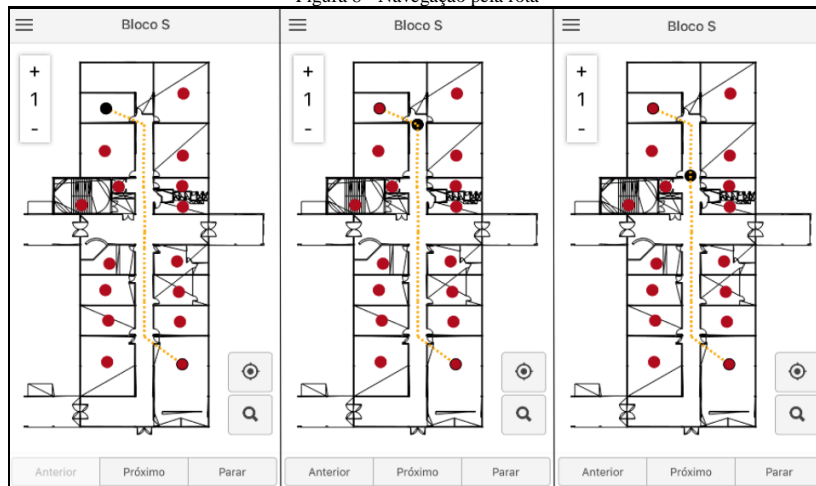
Figura 7 - Criação de rotas pelo aplicativo



Fonte: Rocha (2016).

Feita a seleção da rota desejada, o usuário poderá navegar por ela clicando no botão “Rota” no canto inferior direito. A rota de menor custo entre os pontos será calculada e apresentada. Para navegar o usuário irá utilizar os botões disponíveis na parte inferior da tela como demonstrado na Figura 8.

Figura 8 - Navegação pela rota



Fonte: Rocha (2016).

Em sua conclusão Rocha (2016, p. 53) explica que o aplicativo desenvolvido apresenta a rota virtual que permite o auxílio de navegação ao usuário. Porém, não foi possível utilizar o georeferenciamento para determinar a localização real do usuário não sendo possível apresentar sua posição dentro do prédio.

3 DESCRIÇÃO

3.1 HARDWARE E ARQUITETURA

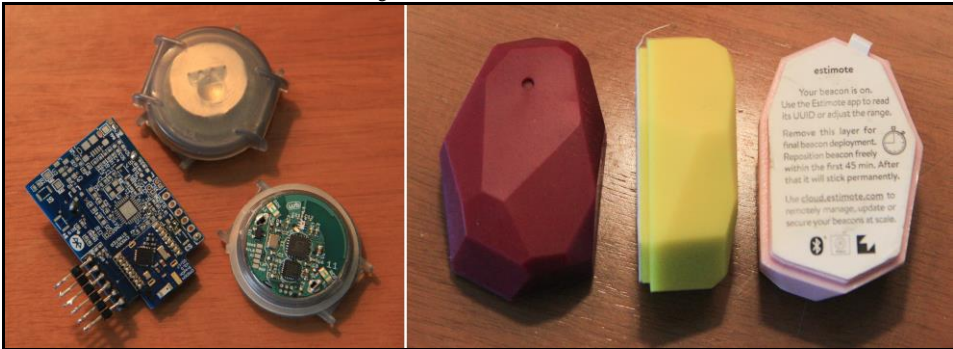
Foram usados seis beacons durante o desenvolvimento, sendo eles: três Location Beacons da Estimote, um beacon desenvolvido na universidade utilizando o EM9304 que será referenciado como MiBeacon, dois beacons X que foram utilizados no trabalho de Reichert (2017). Também foram usados dois dispositivos como receptores. Um celular LG G3 e um tablet Samsung SM-T113NU utilizados para executar o aplicativo que será descrito mais adiante no artigo. A Quadro 3 mostra algumas características dos beacons utilizados no trabalho.

Quadro 3 - Comparação entre beacons				
Beacon	Eddystone	iBeacon	Alcance Máximo	Intervalo de Emissão
Estimote	Sim	Sim	15 metros	300 ms
MiBeacon	Sim	Não	3 metros	1000 ms
Beacon X	Sim	Não	X	

Fonte: elaborado pelo autor

Todos os beacons utilizados trabalham com o protocolo Eddystone, o que facilitou o desenvolvimento do aplicativo. Segundo Google (2018, p. 1, tradução nossa), “Eddystone é um formato de beacon aberto desenvolvido pela Google e planejado com transparência e robustez em mente. Eddystone pode ser detectado por dispositivos Android e iOS”. Na Figura 9 o beacon retangular mais a esquerda é o MiBeacon, os dois redondos são os os XXXX e os três a direita são os beacons da Estimote.

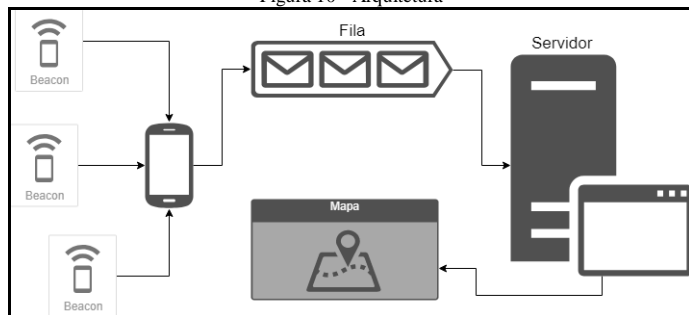
Figura 9 - Beacons utilizados



Fonte: elaborado pelo autor.

A arquitetura foi montada conforme a representação demonstrada na Figura 10 - Arquitetura. Um celular ou tablet com o aplicativo instalado recebe os dados dos beacons dispostos na sala. Os dados são publicados em uma fila que é consumida pelo servidor. O servidor por sua vez processa os dados e disponibiliza no mapa.

Figura 10 - Arquitetura

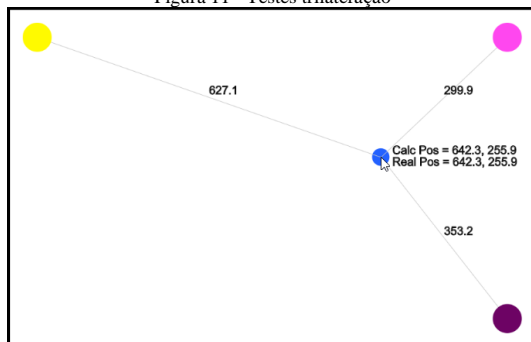


Fonte: elaborado pelo autor

### 3.2 PLATAFORMAS DE TESTES

Foram criadas duas plataformas de testes. Uma delas para analisar os cálculos de trilateração e outra para testar a medição de distância em porcentagem entre um beacon e o receptor. A Figura 11 mostra a plataforma desenvolvida para comprovar a acurácia do método de trilateração.

Figura 11 - Testes trilateração



Fonte: elaborado pelo autor.



A distância euclidiana entre cada âncora e o ponteiro do mouse é calculada, após isso a trilateração aplicada. O valor resultante do cálculo sempre é o mesmo da posição real do ponteiro do mouse. A posição real é obtida através das constantes *mouseX* e *mouseY* que a biblioteca de desenho p5.js disponibiliza. A função de trilateração utilizada é demonstrada no Quadro 4.

Quadro 4 - Função de trilateração

```
var xa = p1.x
var ya = p1.y

var xb = p2.x
var yb = p2.y

var xc = p3.x
var yc = p3.y

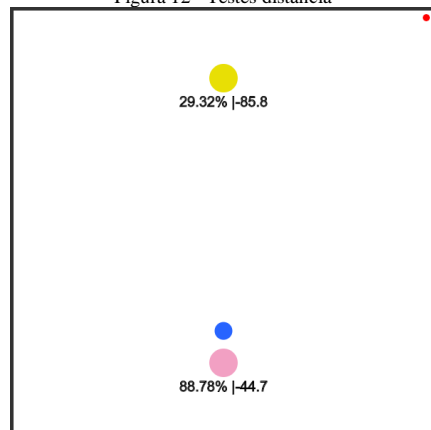
var ra = calcDist(p1, mouse)
var rb = calcDist(p2, mouse)
var rc = calcDist(p3, mouse)

var S = (Math.pow(xc, 2) - Math.pow(xb, 2) + Math.pow(yc, 2) - Math.pow(yb, 2) + Math.pow(rb, 2) - Math.pow(rc, 2)) / 2.0
var T = (Math.pow(xa, 2) - Math.pow(xb, 2) + Math.pow(ya, 2) - Math.pow(yb, 2) + Math.pow(rb, 2) - Math.pow(ra, 2)) / 2.0
var y = ((T * (xb - xc)) - (S * (xb - xa))) / (((ya - yb) * (xb - xc)) - ((yc - yb) * (xb - xa)))
var x = ((y * (ya - yb)) - T) / (yb - xa)
```

Fonte: elaborado pelo autor

A Figura 12 mostra a plataforma desenvolvida para testar o cálculo de aproximação percentual. Neste teste foram utilizados dois beacons posicionados em linha reta. Ao calibrar os beacons, o usuário deve chegar o mais próximo e o mais longe possível de cada beacon, assim será possível saber qual o valor RSSI máximo e mínimo para o ambiente atual. Ao calibrar o beacon A (amarelo na parte superior da figura), por exemplo, foi possível saber que o valor RSSI mínimo para o ambiente era de -103dBm e o máximo -58dBm.

Figura 12 - Testes distância



Fonte: elaborado pelo autor.

Assim é possível calcular em porcentagem qual a distância entre o beacon e o receptor com base no RSSI atual e com os valores calibrados. Para isso foi utilizada a função *map()* da biblioteca p5.js que funciona da seguinte forma. São informados o RSSI atual, o RSSI mínimo, RSSI máximo, novo limite mínimo e novo limite máximo. Será retornado um valor de 0 a 100%.

### 3.3 COLETA DE DADOS

Para a coleta de dados foi desenvolvido um aplicativo móvel com a utilização do *framework* Ionic. O Ionic é um *framework* focado em desenvolvimento de aplicativos móveis multiplataforma (IONIC, 2018). Este permite a utilização da linguagem TypeScript junto com HTML e CSS para o desenvolvimento de aplicativos que podem ter o mesmo código compilado para várias plataformas como Android e iOS. O *framework* Ionic com o auxílio do Cordova também faz possível a comunicação com o *hardware* do dispositivo como câmera, acelerômetro, GPS e *bluetooth* (CORDOVA, 2018).

O Ionic disponibiliza diversos plugins que adicionam funcionalidades ao aplicativo. Um dos principais *plugins* utilizados neste trabalho foi o Native BLE que faz com que o aplicativo se comunique com dispositivos *bluetooth*. Após a instalação do plugin, é possível receber os dados dos *beacons* conforme demonstrado no Quadro 5.

Quadro 5 - Função que procura os beacons

```

1 scan() {
2   this.ble.startScanWithOptions([], { reportDuplicates: true }).subscribe(
3     beaconFound => {
4       console.log(beaconFound) // Printa as informações recebidas
5     })
6 }

```

Fonte: elaborado pelo autor.

A opção *reportDuplicates* faz com que um beacon já encontrado pelo aplicativo seja reportado a cada nova leitura. Se está opção não for especificada, o padrão para ela será falso e então o beacon encontrado será reportado apenas uma vez sendo ignorado nas demais leituras. O plugin Native BLE tem sua estrutura de dados disponibilizada em um JSON com o identificador único do beacon, que é seu *MAC Address* e o valor do RSSI atual, além do valor de *advertising* que não é utilizado neste trabalho. Abaixo o Quadro 6 mostra um exemplo de leitura recebida pelo aplicativo.

Quadro 6 – Exemplo de JSON recebido ao ler beacon

```

1 {
2   "id": "3A:F5:49:71:70:63",
3   "advertising": {},
4   "rssi": -48
5 }

```

Fonte: elaborado pelo autor

### 3.4 PROCESSAMENTO DE DADOS

Ao iniciar o aplicativo, serão instanciados os beacons já pré-definidos. A classe que representa estes objetos é a Beacon que tem atributos como identificador, RSSI e *txPower*. O identificador único dos beacons já deve ser conhecido e descrito no código fonte previamente a compilação.

Antes de começar a procurar pelos beacons, é necessário efetuar a calibração dos dispositivos no ambiente. Para isso o usuário irá escolher a opção “Calibrar” no aplicativo e deverá andar por todo o ambiente chegando o mais perto e o mais longe possível de cada um dos beacons que deverão estar dispostos nos cantos da sala. Ao efetuar a calibração o atributo *minRSSI* e *maxRSSI* de cada um dos beacons será atualizado. Após finalizar a calibração, a opção “Procurar Beacons” será disponibilizada.

Após encontrar um *beacon*, a função que faz a busca irá verificar se este é um dos beacons esperados, ou seja, se o identificador do beacon encontrado é um dos pré-definidos. Essa verificação é necessária pois o plugin Native BLE irá receber os dados de todos os beacons alcançáveis que estiverem transmitindo dados. Se for um dos beacons esperados, o atributo RSSI do beacon é atualizado bem como o valor do RSSI filtrado.

A cada 500 milissegundos todos os beacons instanciados serão enviados para a fila via POST. Disponibilizando assim os dados atualizados para o servidor. A fila será populada pelo aplicativo Ionic e consumida pelo mapa. A ordem dos elementos é *first-in-first-out* (FIFO), portanto, o primeiro elemento inserido será o primeiro a sair para o processamento.

O servidor irá consumir os dados da fila conforme sua ordenação atualizando o valor do RSSI atual e o valor do RSSI filtrado cada objeto de beacon instanciado no mapa. O percentual de distância entre o receptor e o beacon será calculado da mesma forma descrita na seção 3.2. As classes de Beacon no aplicativo Ionic e no servidor/mapa são parecidas mas com alguns atributos e métodos diferentes. A especificação de ambas as classes podem ser consultadas no apêndice A.

Com as distâncias calculadas é possível aplicar o cálculo de trilateração e encontrar a posição do receptor no ambiente e representa-la no mapa com o valor x, y calculado. A trilateração é aplicada conforme exemplificado na seção 3.2, entretanto, foi necessário aplicar a normalização de alguns dados a mais. **EXPLICAR AQUI COMO FOI FEITA A TRILATERAÇÃO**

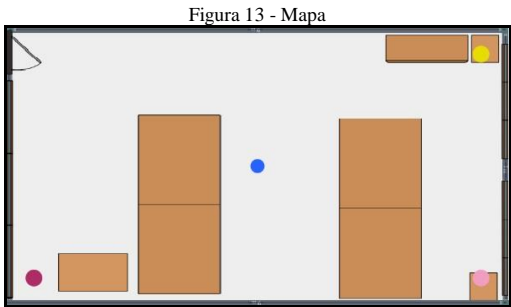
### 3.5 REPRESENTAÇÃO GRÁFICA DO MAPA

Foi utilizada a biblioteca p5.js para a representação gráfica. P5.js é uma biblioteca JavaScript que possui um conjunto de funcionalidades gráficas para desenho (P5.js, 2018). A biblioteca foi criada criada por Lauren McCarthy e desenvolvida pela comunidade com o suporte da Processing Foundation e NYU ITP.

O p5.js tem *setup()* e *draw()* como as principais funções. A função *setup()* é onde fica todo o código de preparação como por exemplo a criação do canvas, o carregamento das imagens necessárias e onde os Beacons são instanciados. A função *draw()* será executada infinitamente trinta vezes por segundo durante a execução do programa. As funções que consomem os dados da fila e os processam são chamadas dentro desse loop. Ali também é onde a lógica

Commented [DK3]: Rever como fazer esta citação

de desenho é especificada. Como demonstrado na **Error! Reference source not found.**, o mapa consiste em um desenho que representa a sala onde os testes foram efetuados. Os beacons dispostos nos cantos do canvas e o dispositivo receptor no centro.



Fonte: elaborado pelo autor.

4 RESULTADOS

- Abordagens:
- Discreto (fingerprint)
- Grafos (tcc rocha)
- Trilateração

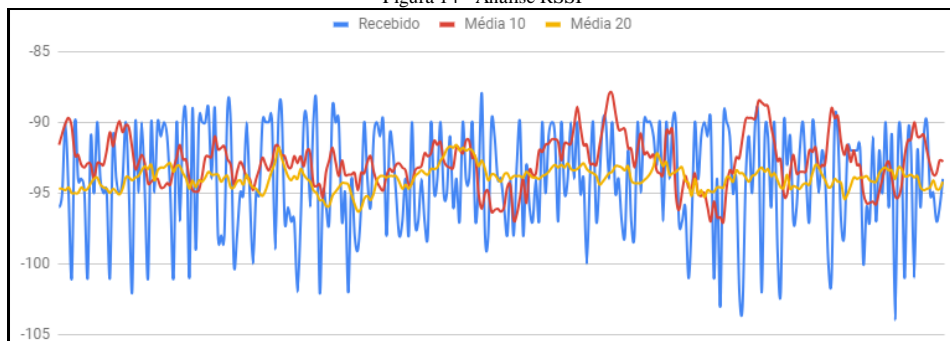
Foi confirmado que o valor do *measured power* (RSSI a um metro) diverge para cada um dos beacons, independentemente da marca ou modo de fabricação. Alguns dos fatores que podem causar essa variação são a potência da bateria do emissor e as oscilações e reflexões de sinal. A Tabela 1 mostra o valor médio do RSSI a um metro para cada um dos beacons utilizados no trabalho.

Tabela 1 - Medições a um metro	
Beacon	dBm
Estimote Amarelo	-78,19
Estimote Rosa	-77,84
Estimote Roxo	-80,29
MiBeacon	?
GenBeacon 1	?
GenBeacon 2	?

Fonte: elaborado pelo autor.

Também foi possível observar que o valor do RSSI não é constante em um ambiente sem interferências diretas mesmo quando o beacon e o receptor parados. Uma medição do sinal efetuada a um metro variou em média seis decibéis. Para contornar oscilação do sinal foram testados dois filtros, o filtro de média móvel. Nas medições efetuadas, o filtro de média móvel com uma janela de 20 valores demonstrou um resultado satisfatório, como é possível verificar na Figura 14 - Análise RSSI.

Figura 14 - Análise RSSI



Fonte: elaborado pelo autor.

Durante o desenvolvimento foram efetuados cálculos para estimar a distância entre o beacon e o receptor em metros. Para isso, foi utilizada a equação demonstrada por Qathrady (2017, p. 2)  $d = A \times (r / t)^B + C$ . Onde as constantes A, B e C representam os valores 0.42093, 6.9476 e 0.54992 respectivamente. O valor de  $r$  é o RSSI atual e  $t$  é o valor do RSSI a um metro. Qathrady explica que essa é a equação e constantes sugeridas pela Android Beacon Library. Entretanto, os resultados não foram satisfatório e optou-se por calcular a distância em porcentagem como demonstrado na seção 3.

Commented [DK4]: Pode deixar a equação assim?

## 5 CONCLUSÕES

### REFERÊNCIAS

- XU, Lisheng; YANG, Feifei; JIANG, Yuqi. **Variation of Received Signal Strength in Wireless Sensor Network**. 2011 3rd International Conference On Advanced Computer Control, Harbin, p.1-1, jan. 2011. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/6016387/>>. Acesso em: 05 nov. 2018.
- DUDHANE, Nilima A.; PITAMBARE, Sanjeevkumar T.. **Location Based and Contextual Services Using Bluetooth Beacons: New Way to Enhance Customer Experience**. Lecture Notes On Information Theory, Pune, v. 3, n. 1, p.31-32, jun. 2015. Disponível em: <<http://www.lnit.org/uploadfile/2016/0115/20160115052139243.pdf>>. Acesso em: 06 nov. 2018.
- ROCHA, Diego T. **Tô Aqui**: Aplicativo para georreferenciamento em ambientes restritor. 2015. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- ROCHA, Marcus Otávio. **FURB-Mobile**: sistema móvel multiplataforma para navegação em rotas internas. 2016. 61 f. Trabalho de Conclusão de Curso (Graduação) Curso de Ciência da Computação. Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2016.
- REICHERT, Leonardo A. **Mapa de calor baseado em geolocalização interna com utilização de beacons**. 2017. 72 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- LARSSON, Johan. **Distance estimation and positioning based on Bluetooth low energy technology**. 2015. 37 f. Tese (Mestrado) - KTH Royal Institute Of Technology, Estocolmo, Suécia.
- PARAMESWARAN, Ambili Thottam; HUSAIN, Mohammad Iftekhar; UPADHYAYA, Shambhu. **Is RSSI a Reliable Parameter in Sensor Localization Algorithms** – An Experimental Study. 2009. 5 f. Pesquisa – Departamento de Ciências da Computação e Engenharia, Buffalo.
- GIACOMIN, João Carlos; VASCONCELOS, Flávio Henrique. **Qualidade da Medição de Intensidade de Sinal nas Comunicações de uma Rede de Sensores Sem Fios: uma Abordagem da Camada Física**. 2006. 10 f. Artigo - Departamento de Ciência da Computação, Universidade Federal de Lavras (UFLA), Lavras.
- BRUINJES T. M.; KOKKELER A. B. J.; KARAGIANNIS G.; SMIT G. J. M. **Shaped pattern synthesis for equispaced linear arrays with non-isotropic antennas**. 2015. 5 f. Departamento de Engenharia Elétrica, Matemática e Ciência da Computação, Universidade de Twente, Enschede, Países Baixos.

**RÖBESAAT**, Jenny et al. **An Improved BLE Indoor Localization with Kalman-Based Fusion**: An Experimental Study. Sensors, Oldenburg, v. 17, n. 5, p.951-961, 26 abr. 2017. MDPI AG.

**AWAD**, A.; FRUNZKE, T.; DRESSLER, F. **Adaptive Distance Estimation and Localization in WSN using RSSI Measures**. 10th Euromicro Conference On Digital System Design Architectures, Methods And Tools (dsd 2007), Lubeck, p.7-9, ago. 2007.

**QATHRADY**, Mimonah Al; HELMY, Ahmed. **Improving BLE Distance Estimation and Classification Using TX Power and Machine Learning**. Proceedings Of The 20th Acm International Conference On Modelling, Analysis And Simulation Of Wireless And Mobile Systems - Mswim '17, Gainesville, p.2-2, 2017.

GOOGLE. **Eddystone format**. [S.I.], 2018. Disponível em: <<https://developers.google.com/beacons/eddystone>>. Acesso em 20 de out de 2018.

IONIC. **Chapter 1**: all about ionic. [S.I.], 2018. Disponível em: <<https://ionicframework.com/docs/v1/guide/preface.html>>. Acesso em 02 abr. 2018.

ESTIMOTE. **The icon of modern art puts Estimote beacons on display**. [S.I.], 2018. Disponível em: <https://blog.estimote.com/post/157200820650/the-icon-of-modern-art-puts-estimote-beacons-on>. Acesso em 13 nov. 2018.

RECK, Marcelo S. **Beacons BLE – BLUETOOTH LOW ENERGY – Design e análise de um sistema de localização indoor**. 2016. 84 f. Trabalho de Conclusão de Curso (Engenharia de Controle e Automação) -Universidade de Caxias do Sul, Caxias do Sul.

## APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

**Este apêndice é obrigatório.** Nele devem constar todos os diagramas de especificação desenvolvidos no trabalho. Os diagramas devem conter legendas numeradas na sequência do artigo.

## ANEXO A – DESCRIÇÃO

Elemento opcional, **anexos são documentos não elaborados pelo autor**, que servem de fundamentação, comprovação ou ilustração, como mapas, leis, estatutos, entre outros. Os anexos são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada anexo. Colocar sempre um preâmbulo no anexo. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.