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. Foram encontrados diversos problemas em relação a reflexão e sinuosidade dos sinais emitidos pelos beacons não sendo possível aplicar uma solução de contorno adequada. A conclusão é que não é possível medir de forma acurada a distância entre um dispositivos BLE e um receptor sem haver calibrações e estudos mais aprofundados sobre propagação de sinais, antenas e filtros de sinais.*

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:

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

2 FUNDAMENTAÇÃO TEÓRICA

A seção 2 apresenta conceitos importantes para o entendimento da pesquisa. Também são descritos dois trabalhos correlatos utilizados como base para os desenvolvimentos e estudos efetuados. Ambos os correlatos são trabalhos de conclusão curso (Bacharelado em Ciência da Computação) pela FURB sendo eles o aplicativo TÔ AQUI: Aplicativo Para georreferenciamento em ambientes restritos e FURB-MOBILE: Sistema móvel multiplataforma para navegação em rotas internas.

2.1 CONCEITOS

A seguir são apresentados os principais conceitos utilizados como base de fundamentação teórica para este trabalho. Serão descritos dispositvos bluetooth *beacon,* indicador RSSI, cálculo de trilateração e técnica *fingerprint* para localização interna utilizando *beacons*.

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, BLE é uma rede *wireless* usada para transmitir dados em pequenas distâncias. Um conceito bastante utilizado referente a *beacons* é o *measured power* ou *tx power*, esse é o valor do sinal recebido pelo receptor a um metro de distância. O *measured power* tem uma valor individual para cada dispositivo indiferentemente de seu modelo ou marca visto que diversos fatores podem interferir nesta medição.

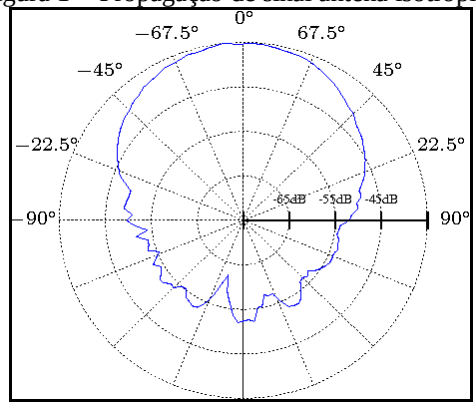
Atualmente existem algumas opções de *beacons* no mercado, entre elas os da Estimote, fabricante de *beacons* e de plataformas de desenvolvimento para utilização dos dispositivos. Os produtos da Estimote são utilizados em museus

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 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.

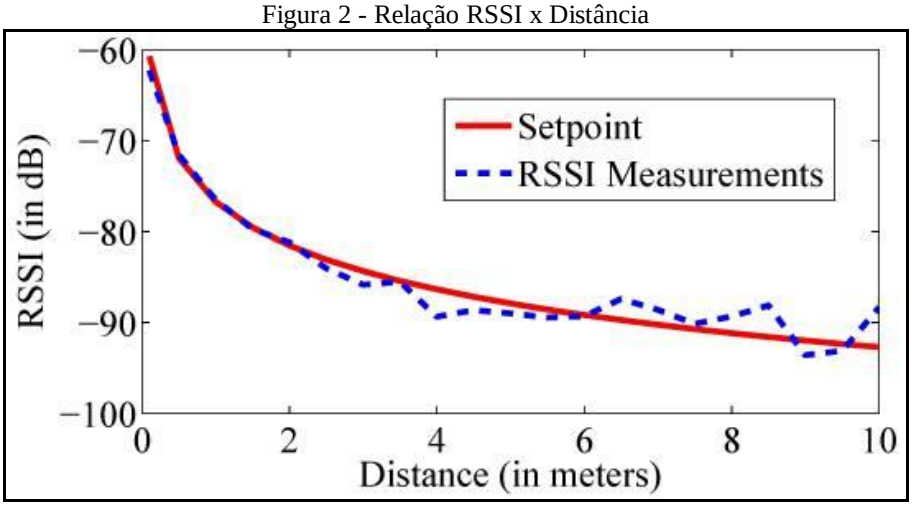
Figura 1 - Propagação de sinal antena isotrópica



Fonte: Bruintjes (2015).

Além da característica isotrópica, outro fator que pode interferir no sinal é o ambiente onde o emissor está. Como Röbesaat (2017, p. 6) explica em seu trabalho, o RSSI é influenciado por interferências do ambiente como reflexões de sinal. Röbesaat (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

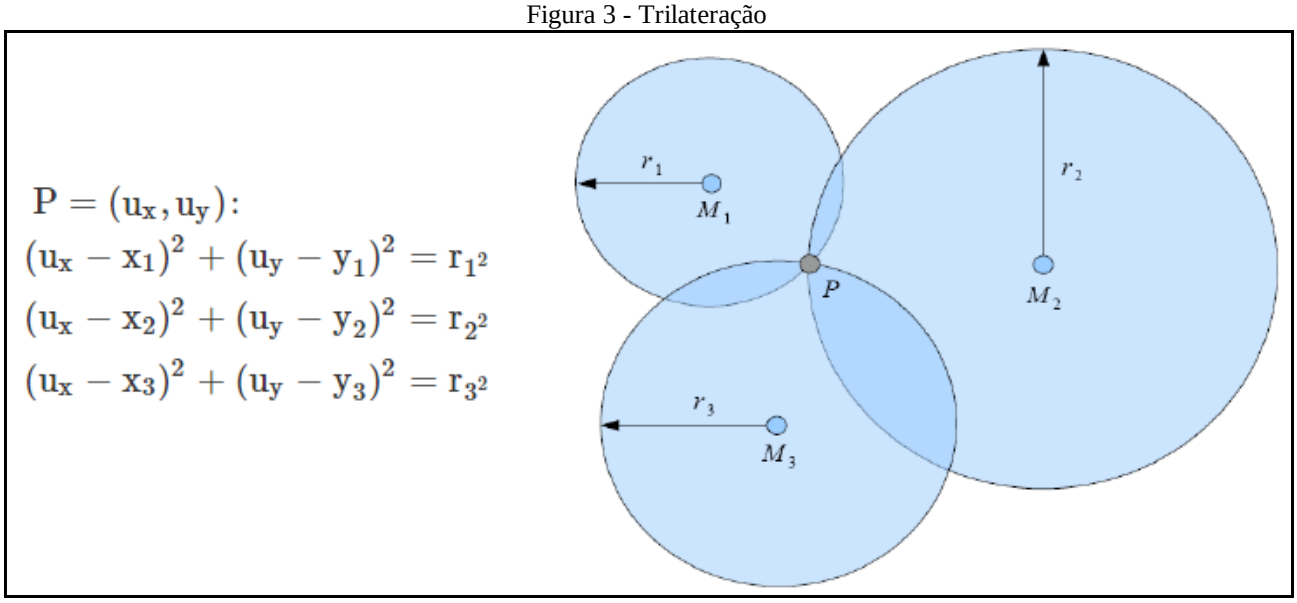


Fonte: Röbesaat (2017).

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 o raio das 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

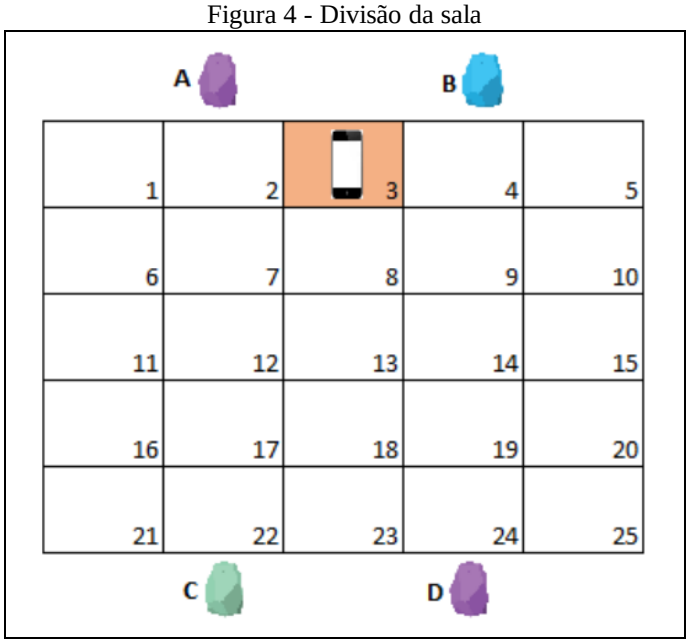


Fonte: Awad, Frunzke e Dressler (2007).

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 subáreas de dois metros quadrados conforme é possível ver na Figura 4. Cada subárea deverá ter uma medição feita com cada um dos quatro *beacons* utilizados. 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.

Figura 4 - Divisão da sala



Fonte: Reck (2016).

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 – Aplicativo Tô Aqui

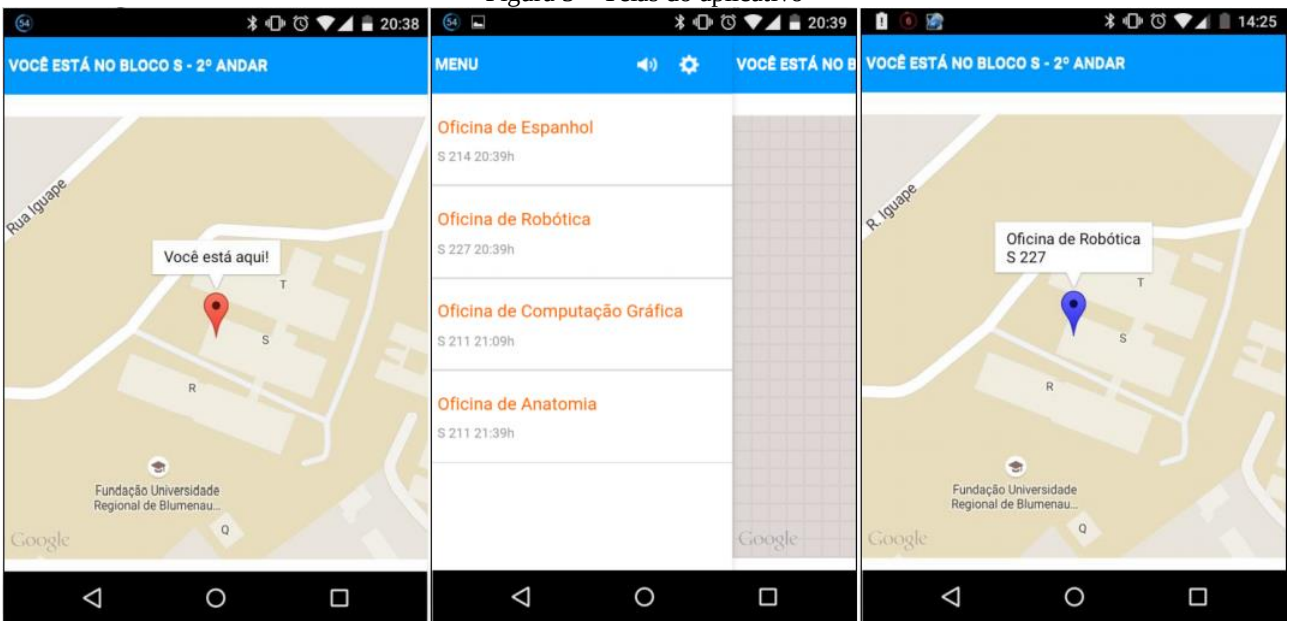
Referência	Rocha (2015)
Objetivos	Possibilitar que o usuário se localize na FURB por meio de um mapa que indique sua posição atual assim aprimorando a localização em ambientes restritos utilizando *beacons* e AGPS.
Principais funcionalidades	Visualizar eventos e ambientes próximos e visualizar posição 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.

É possível observa 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 funcionaliadades incluem notificar o usuário de eventos e ambientes próximos bem como visualizar sua posição atual no mapa. 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 – Telas do aplicativo



Fonte: Rocha (2015).

Rocha (2015) 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. 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 do trabalho, "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, p. 45), 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.

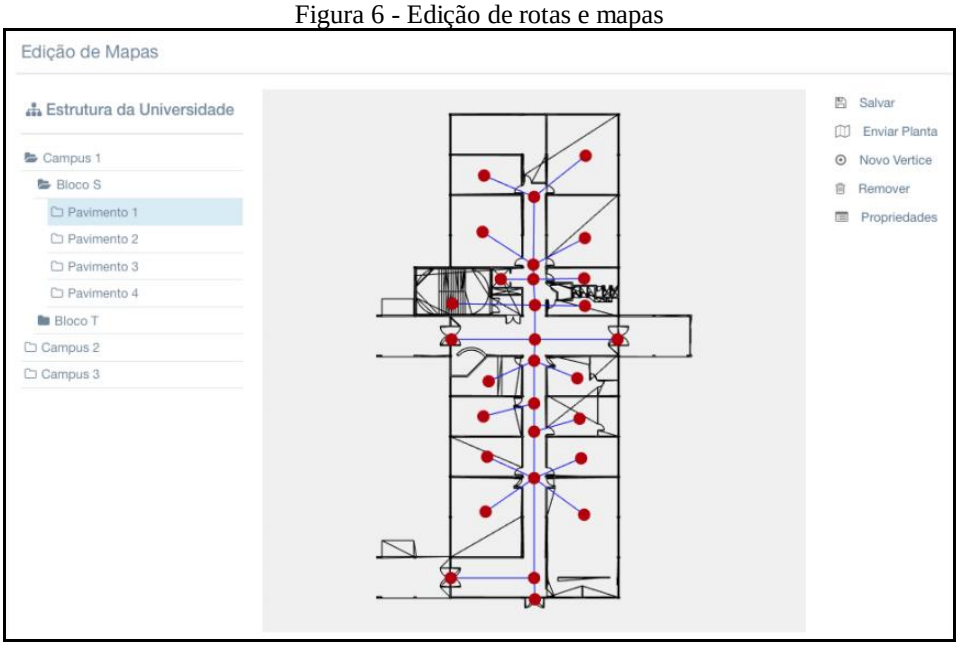
Quadro 2 – FURB-MOBILE

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 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.

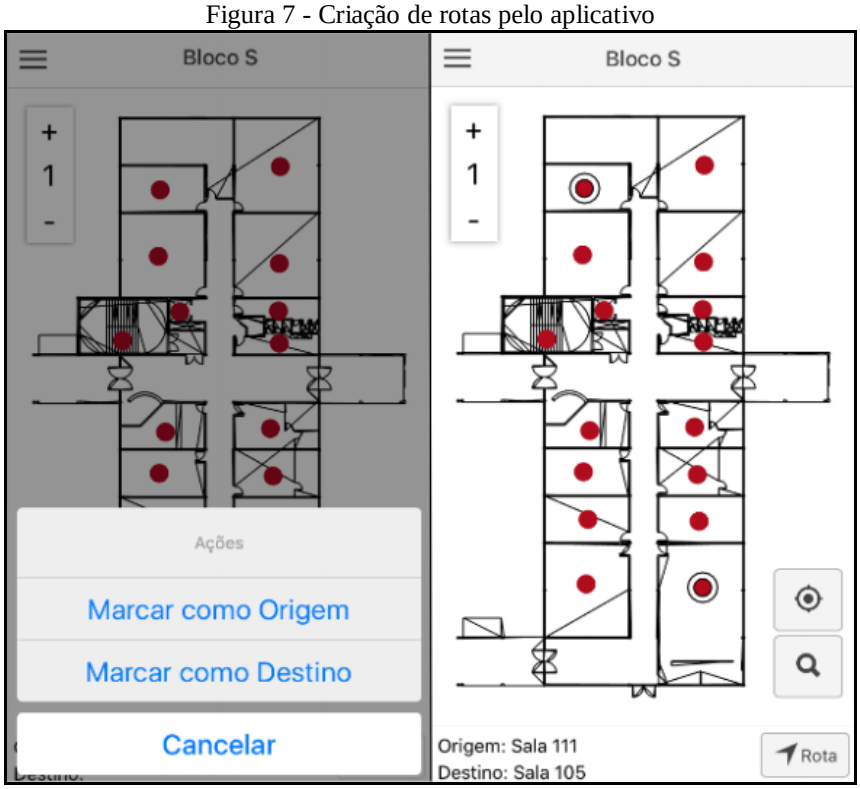
Figura 6 - Edição de rotas e mapas



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.

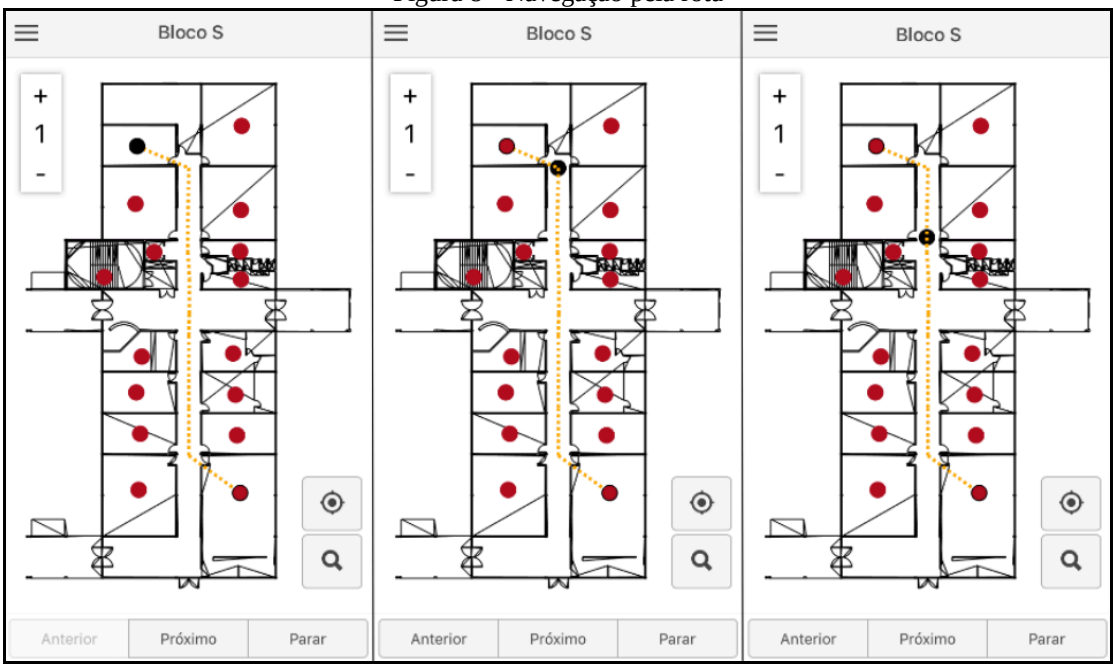
Figura 7 - Criação de rotas pelo aplicativo



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

A seção 3 demonstra detalhes da pesquisa como quais os *hardware*, bibliotecas e arquitetura utilizadas além de descrições detalhadas de como a aplicação foi desenvolvida. Também são descritas todas as plataformas de testes utilizadas para a coleta e análise de dados.

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 e dois *beacons* dos qual não foi possível identificar detalhes sobre a fabricação e que serão referenciados como GenBeacon 1 e 2. Os dispositivos GenBeacon 1 e 2 foram utilizados no trabalho de Reichert (2017). Também foram usados dois dispositivos como receptores. Um celular LG G3 e um tablet Samsung SM-T113NU, ambos foram utilizados para executar o aplicativo que será descrito mais adiante no artigo. O Quadro 3 mostra algumas características dos dispositivos BLE utilizados.

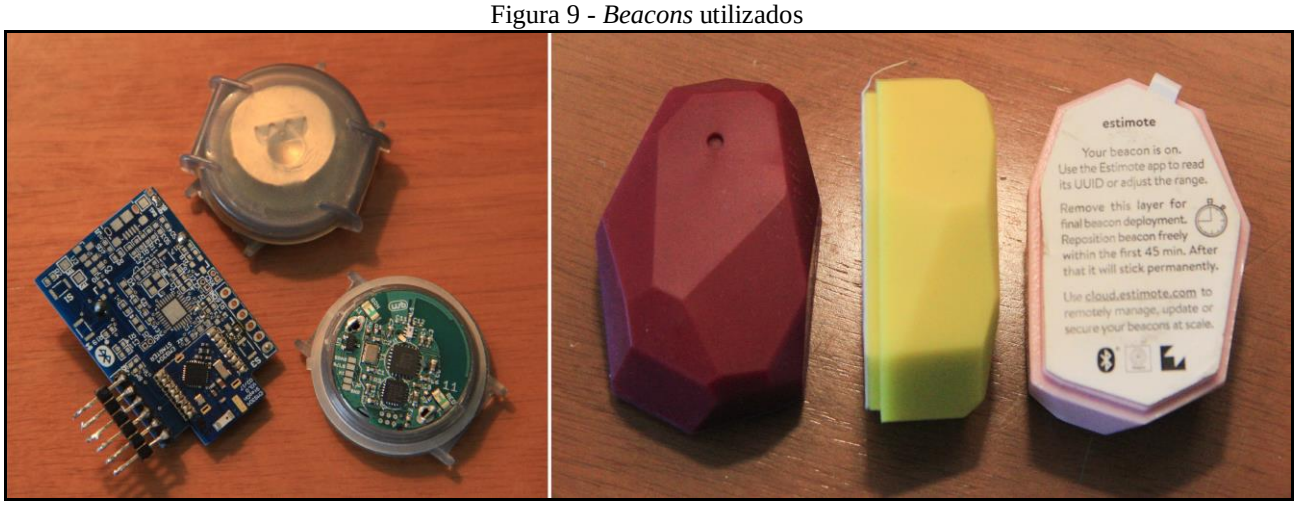
Quadro 3 - Comparação entre *beacons*

Beacon	Eddystone	iBeacon	Alcance Máximo	Intervalo de Emissão
Estimotes	Sim	Sim	15 metros	300 ms (configurável)
MiBeacon	Sim	Não	3 metros	300 ms (configurável)
GenBeacons	Sim	Não	8-15 metros	300 ms (configurável)

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 robustes em mente. Eddystone pode ser detectado por dispositivos Android e iOS”. Na Figura 9 o *beacon* retangular azul mais a esquerda é o MiBeacon, os dois redondos são os GenBeacons e os três a direita são os Estimote.

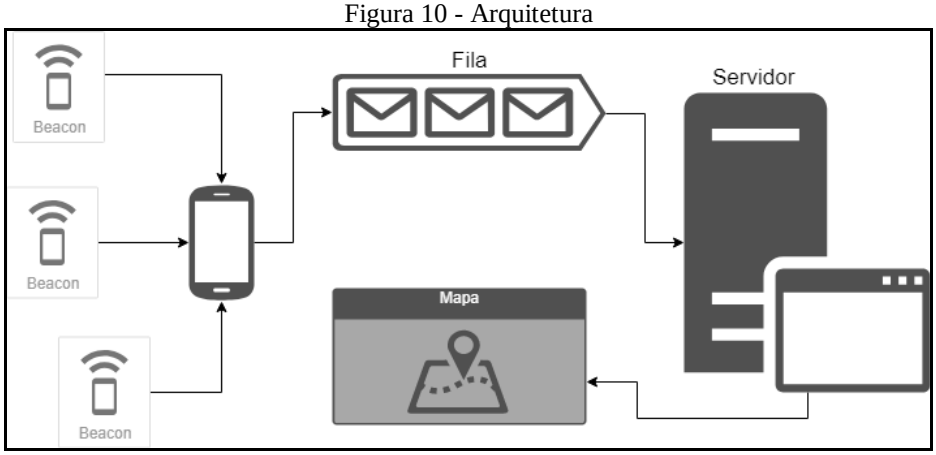
Figura 9 - *Beacons* utilizados



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 graficamente no mapa.

Figura 10 - Arquitetura

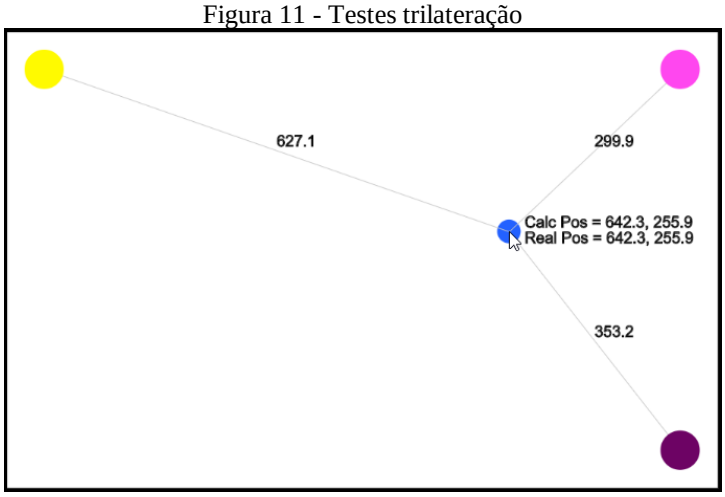


Fonte: elaborado pelo autor.

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 acertividade 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 é aplicada. O valor resultante do cálculo sempre é o mesmo da posição real do ponteiro do mouse. A posição real é obtida através das variáveis `mouseX` e `mouseY` que a biblioteca de desenho p5.js disponibiliza. A distância entre os pontos é calculada pela função `dist()`, também nativa da biblioteca p5.js. A função de trilateração utilizada é demonstrada no Quadro 4.

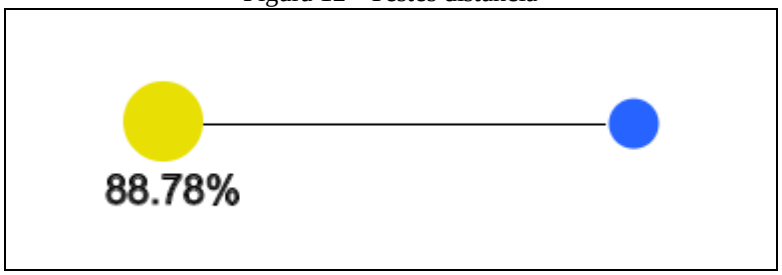
Quadro 4 - Função de trilateração

```
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) / (xb - xa)
```

Fonte: elaborado pelo autor.

A Figura 12 mostra a plataforma desenvolvida para testar o cálculo de aproximação percentual. Neste teste foi utilizado apenas um *beacon*. Ao calibrar o *beacon* o usuário deve chegar o mais próximo e o mais longe possível do dispositivo. Com isso o valor RSSI máximo e mínimo considerando as interferências que o ambiente atual possa ter será conhecido. No exemplo ao caibrar o *beacon* (círculo amarelo a esquerda da Figura 12 - Testes distância), 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 pode-se 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 na função o RSSI atual, o RSSI mínimo, RSSI máximo, novo limite mínimo e novo limite máximo. Será retornado um valor de 0 a 100% dependendo da distância do usuário em relação ao emissor.

3.3 COLETA DE DADOS

Para a coleta de dados foi desenvolvido um aplicativo móvel com a utilização do *framework* Ionic. O Ionic disponiviliza 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 demonstratrado 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)
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 específicada, o padrão para ela será falso e então o dispositivo 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 identificado 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 instânciados 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 até chegar o mais perto e o mais longe possível de cada um dos três *beacons* que deverão estar dispostos nos cantos da sala. Ao efetuar a calibração o atributo `minRSSI` e `maxRSSI` de cada um dos objetos instanciados será atualizado. Após finalizar a calibração, a opção `Procurar Beacons` deverá ser escolhida para iniciar o *scan* pelos dispositivos alcançáveis.

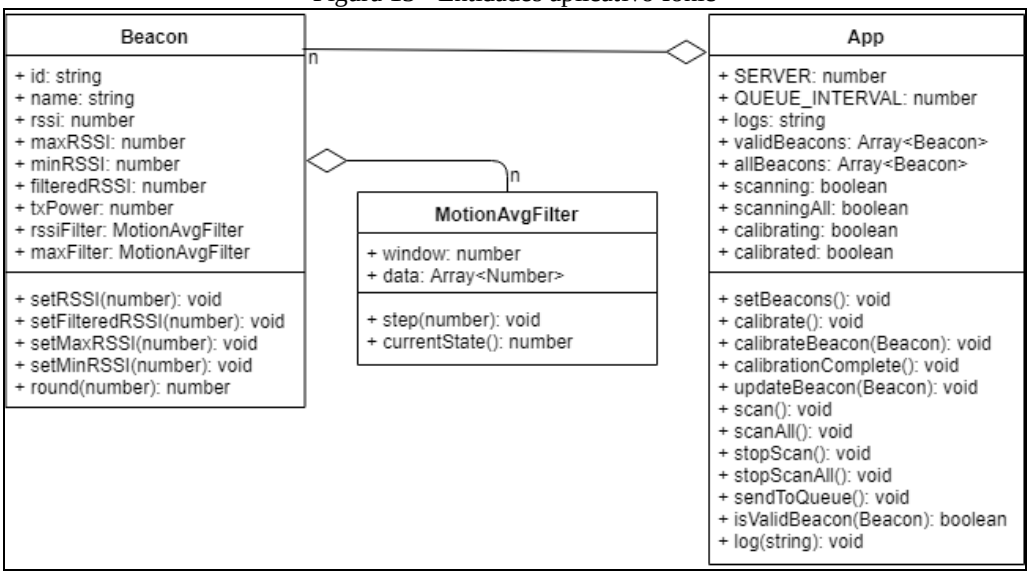
Ao encontrar um *beacon* a função `scan()` que faz a busca irá verificar se o dispositivo encontrado é um dos esperados, ou seja, se algum dos instanciados tem o mesmo identificador do encontrado. Essa verificação é necessária

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 é atualizado bem como o valor do RSSI filtrado.

A cada 500 milisegundos um `json` com os dados de todos os *beacons* instânciados é enviado para a fila via POST, disponibilizando assim os dados atualizados para o servidor. A fila será populada pelo aplicativo Ionic e consumida pelo mapa (servidor). A ordem dos elementos é First-In-First-Out (FIFO), portanto, o primeiro elemento inserido será o primeiro a sair para o processamento.

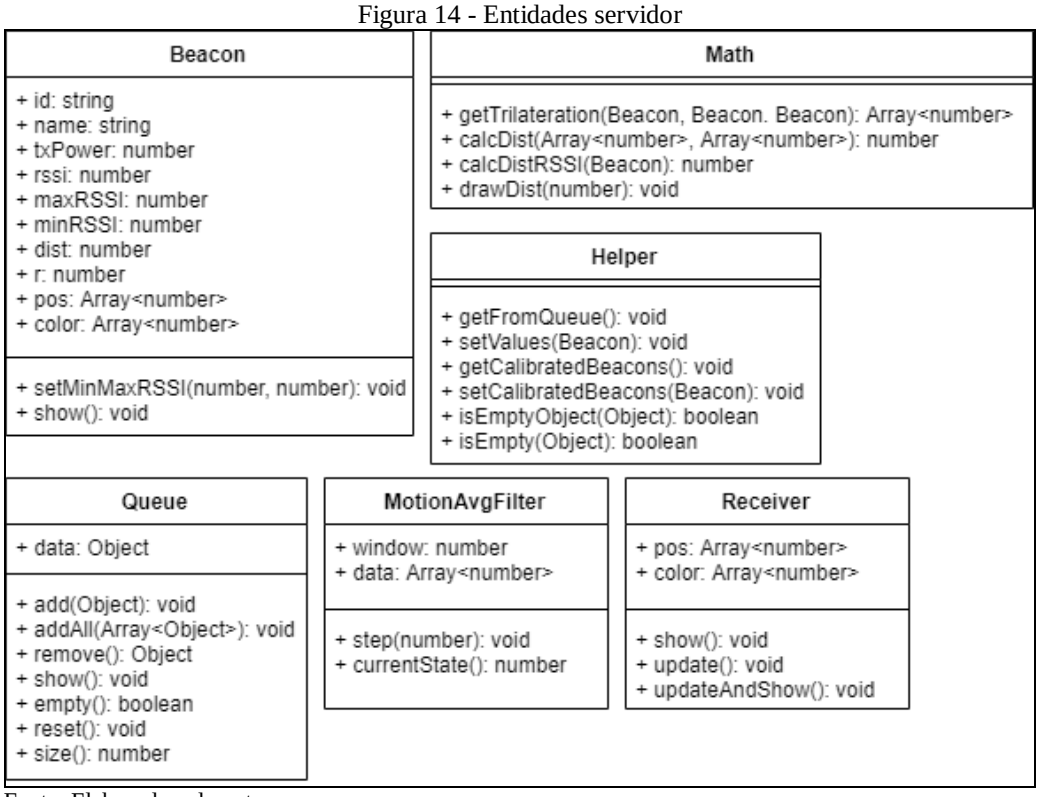
O servidor irá consumir os dados da fila conforme sua ordenação atualizando o valor do RSSI atual e o valor do RSSI filtrado para cada objeto de `Beacon` instânciado no mapa (servidor). O percentual de distância entre o receptor e o emissor será calculado da mesma forma descrita na seção 3.2. As classes de `Beacon` no aplicativo Ionic e no servidor/mapa são semelhantes mas com alguns atributos e métodos diferentes. A especificação das entidades utilizadas no aplicativo podem ser vistas na Figura 13 e as utilizadas no servidor na Figura 14

Figura 13 - Entidades aplicativo Ionic



Fonte: Elaborado pelo autor.

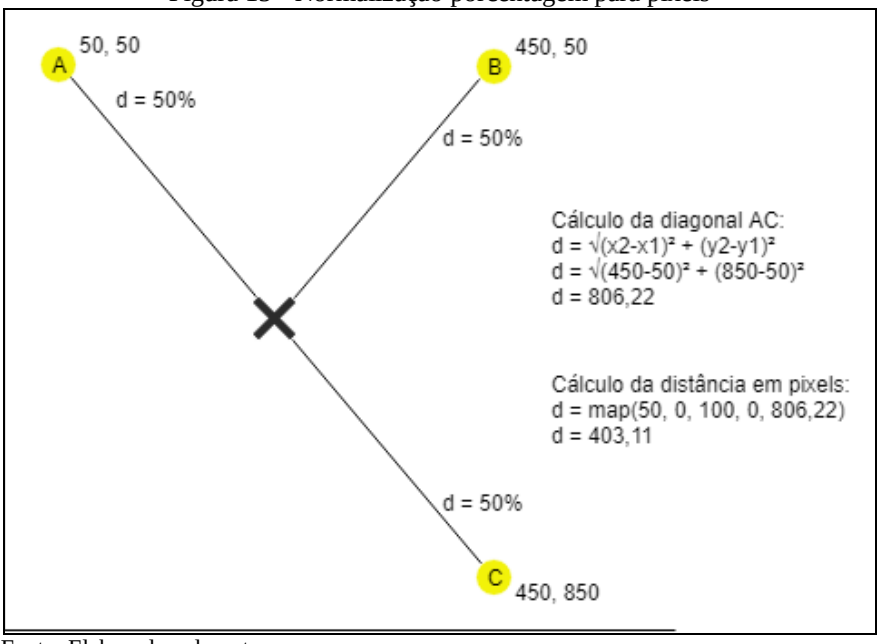
Figura 14 - Entidades servidor



Fonte: Elaborado pelo autor.

Com as distâncias estimadas é possível aplicar o calculo de trilateração, encontrar a posição do receptor no ambiente e representa-lá no mapa. A trilateração é aplicada conforme exemplificado na seção 2.1.3, entretanto, foi necessário aplicar a normalização de alguns dados como explicado a seguir. Com o valor em porcentagem, é necessário normalizar para pixels e para isso foi utilizada a função `map()` que neste caso recebe como parâmetro a distância em porcentagem e transforma em um valor em pixels. Como é possível ver na Figura 15, o tamanho da diagonal entre os pontos A e C é calculado utilizando a fórmula da distância euclidiana entre dois pontos, no exemplo a diagonal tem 806,22 pixels. Sabendo o valor mínimo e o máximo do intervalo atual da distância (0 a 100%) e o novo intervalo (0 a 806,22 pixels), é possível converter o valor percentual em pixels. No exemplo demonstrado na Figura 15 o valor de 50% seria equivalente a 403,11 pixels de distância.

Figura 15 - Normalização porcentagem para pixels

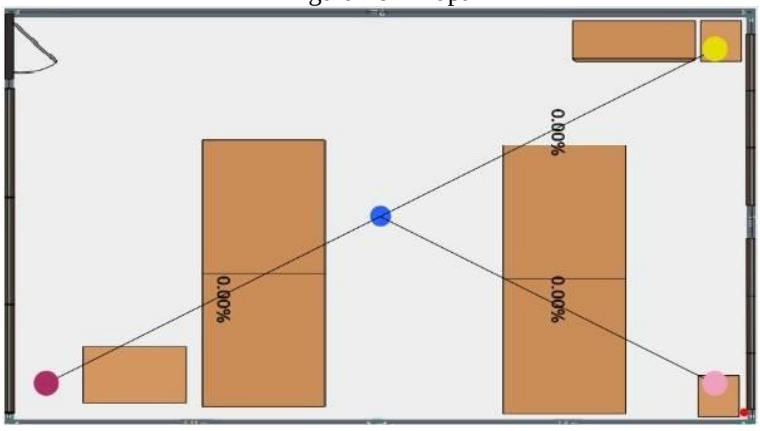


Fonte: Elaborado pelo autor.

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 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 princiapis 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 instânciados. A função `draw()` será executada infinitamente no máximo 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 de desenho é especificada. Como demonstrado na Figura 16, o mapa consiste em um desenho que representa a sala onde os testes foram efetuados onde os círculos dispostos nos cantos do canvas representam os *beacons* e o disposivo receptor representado em azul no centro. Existe também um controle para saber se a fila contém dados para serem consumidos ou não. Quando não ouverem mais dados, um ponto vermelho é mostrado no mapa (canto inferior direito da Figura 16).

Figura 16 - Mapa



Fonte: elaborado pelo autor.

4 RESULTADOS

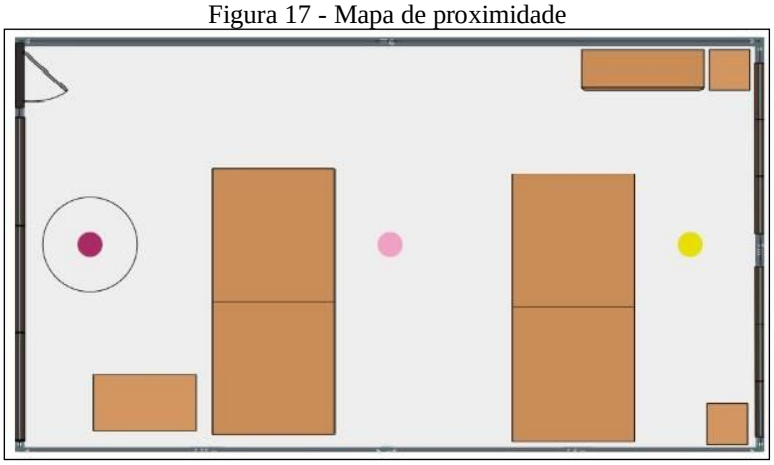
A seção de resultados foi dividida em três partes. A primeira trata de explicar técnicas e abordagens analisadas e/ou aplicadas durante o desenvolvimento do trabalho. A segunda parte demonstra dados coletados durante o estudo efetuado. Por fim são apresentadas propostas para a extensão desta pesquisa.

4.1 TÉCNICAS E ABORDAGENS

Foram identificados diferentes tipos de abordagem para determinação da localização de um receptor *bluetooth* utilizando *beacons*. A abordagem *fingerprint* descrita na seção 2.1.4, tem como principal problema a dificuldade na calibração ou treinamento. Nesta parte se faz necessário dividir o ambiente em subáreas e medir a intensidade do sinal recebido em cada subárea demarcada. Além do demasiado trabalho e dificuldade da medição, outro problema encontrado nesta abordagem é a precisão das medições efetuadas. Além dos fatores que podem influnciar no sinal emitido pelos *beacons* descrito na seção 2.1.2, as caracteristicas da antena receptora podem influenciar na medição do sinal (ALMANGUER, 2018). Ou seja, os valores obtidos pela antena utilizada na calibração podem ser diferentes da antena do receptor durante o uso da aplicação interferindo assim no resultado esperado.

Outra abordagem analisada é a utilização de grafos como feito por Rocha (2016) descrito no trabalho correlado 2 na seção 2.2. O desenvolvimento dessa técnica teria o ambiente representado por um grafo com os *beacons* sendo os vértices. Ao decorrer da pesquisa observou-se que o sinal emitido pelos *beacons* não é precisamente convertido para distância em metros, entretanto, para determinar a proximidade, os *beacons* podem ser utilizados com mais confiabilidade. É possível determinar se um beacon está próximo ou distante de um receptor, assim sendo plausível determiar qual é o vértice (*beacon*) mais próximo e efetuar a navegação pelo ambinte. Um breve ensaio foi desenvolvido com está técnica como é possível ver na Figura 17. Com os *beacons* dispostos de maneira diferente na sala, o servidor fica constantemente recebendo o RSSI atualizado e verifica qual é o dispositivo com o maior valor. Como visto na Figura 2 seção 2.1.2, quanto maior o valor do RSSI, menor a distância, sendo possível dizer qual o *beacon* mais próximo do receptor e então determinar sua posição.

Figura 17 - Mapa de proximidade

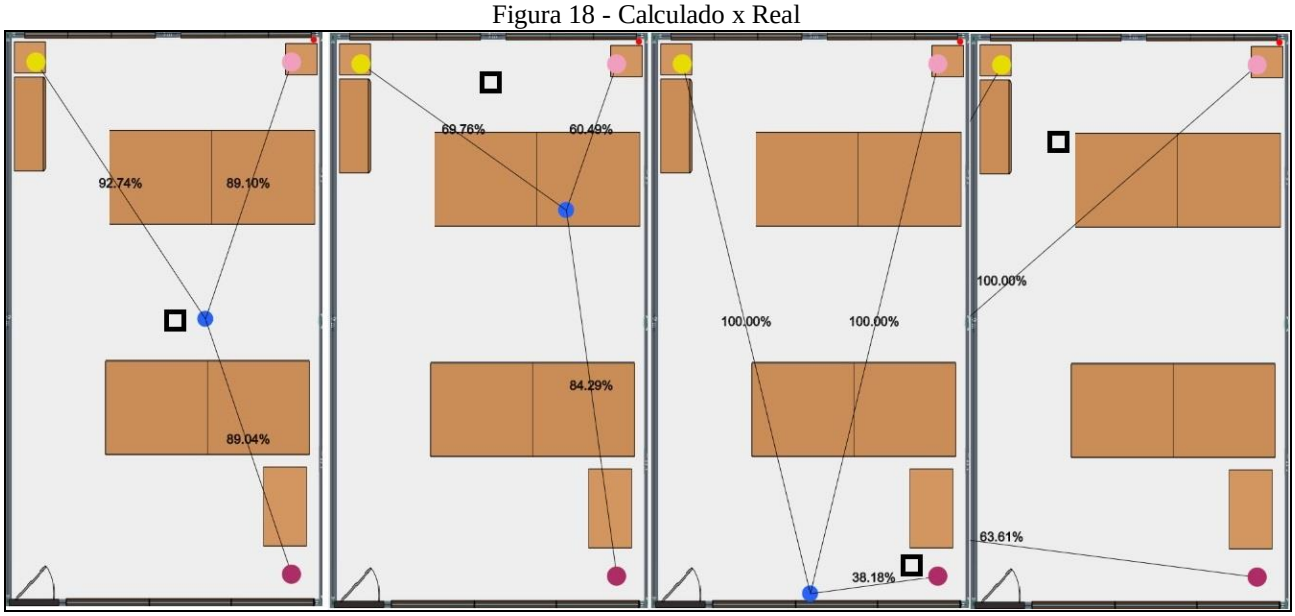


Fonte: Elaborado pelo autor.

Por fim, a abordagem implementada nesta pesquisa foi a de trilateração. Está baseia-se na ideia básica do sistema GPS. Onde é sabido a posição de três pontos conhecidos e a distância de cada ponto até o receptor, assim podendo-se aplicar cálculos para determinar a posição atual do receptor. A trilateração descrita na seção 2.1.3 foi utilizada para calcular a posição do receptor. A posição dos três *beacons* sempre fixa e conhecida foi utilizada para o cálculo junto com a distância estimada pela intensidade de sinal emitida. Entretanto, como a distância estimada não é precisa o suficiente, o resultado da trilateração não é a posição correta como é possível verificar na Figura 18 onde a distância real é representada pelo quadrado de cor preta e a posição calculada o círculo de cor azul.

Os principais fatores que não permitiram determinar a posição do receptor foram a não linearidade do sinal emitido pelos dispositivos bem como a reflexão do sinal. O fato do sinal não ser isotrópico como demonstrado na Figura 1 influenciou no cálculo de trilateração que espera um raio de distância isotrópico como demonstrado na Figura 3. Esses fatores foram impeditivos de obter-se a distância entre o dispositivo emissor e receptor de forma acurada resultando em posições não acertivas durante o cálculo de trilateração.

Figura 18 - Calculado x Real



Fonte: Elaborado pelo autor.

4.2 DADOS COLETADOS

Foi confirmado que o valor do *measured power* (sinal medido 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 ocilações/reflexões de sinal. A Tabela 1 mostra o valor médio do sinal a um metro para cada um dos *beacons* analisados no trabalho.

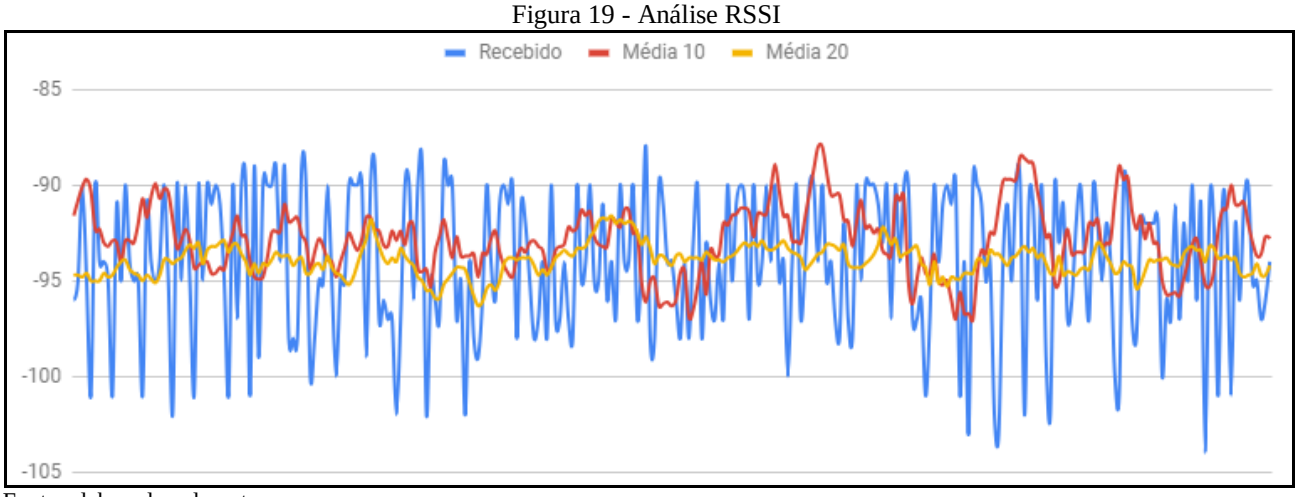
Tabela 1 - Medições a um metro

Beacon	dBm
Estimote Amarelo	-78,19
Estimote Rosa	-77,84
Estimote Roxo	-80,29
MiBeacon	-87,04
GenBeacon 1	?
GenBeacon 2	?

Fonte: elaborado pelo autor.

Também foi possível observar que o valor do sinal não é constante em um ambiente sem interferências diretas mesmo com o *beacon* e o receptor parados. Uma das medições efetuadas a um metro variou em média seis decibéis. Para contornar ocilação 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 19 - Análise RSSI. Quando maior a janela do filtro de média móvel, maior a estabilização do sinal mas também será maior o tempo necessário para efetuar a atualização deste valor visto que serão necessários mais dados para atualizar a média.

Figura 19 - Análise RSSI



Fonte: elaborado pelo autor.

Durante o estudo foram efetuados cálculos para estimar a distância em metros entre o *beacon* e o receptor. Com base nas equações demonstradas por Dong e Dargie (2012, p. 3) é possível dizer que $\mathbf{d = 10^{txPower - rssi / 20}}$. Entretanto, os resultados não foram satisfatórios como demonstrado na Tabela 2. Outras equações foram testadas como a demonstrada por Qathrady (2017, p. 2) ($\mathbf{d = A \times (r / t)B + C}$), mas também sem resultados satisfatórios.

Tabela 2 - Distância Calculada x Real

Real	Calculado
1 metro	0.8 metros
6 metros	4.2 metros
9 metros	6.9 metros

Fonte: Elaborado pelo autor.

4.3 PROPOSTAS DE EXTENSÃO

Ao decorrer da pesquisa foram encontrados diversos desafios como a falta de estabilização e não linearidade dos sinais captados pelo receptor. Foi possível constatar que o cálculo de trilateração não resulta na posição esperada pelo motivo de que as distâncias estimadas não são precisas o suficiente. Visto isso, é proposto como extensão o estudo mais aprofundado de possíveis técnicas ou modelos para aprimorar a estimação de distância entre o emissor e o receptor do sinal. Além disso, mais filtros podem ser estudados e verificados se possuem melhor estabilização do que o filtro de média móvel que foi utilizado.

É possível também aprofundar o desenvolvimento para a abordagem com a utilização de grafos como discutido na seção 4.1. O ensaio desenvolvido permite apenas identificar qual o *beacon* mais próximo do receptor assim sendo possível determinar sua posição. A extensão do desenvolvimento pode se aproximar das técnicas utilizadas por Rocha (2016) permitindo criar rotas e navegar pelo ambiente mapeado.

5 CONCLUSÕES

O objetivo de localizar o usuário em um ambiente interno não foi atingido utilizando a técnica de trilateração visto que a posição resultada pode ser muito diferente da real. Como demonstrado ao longo do trabalho, os principais fatores que foram deterministicos para que não se pudesse calcular a localização do usuário utilizando a trilateração foi a falta de confiabilidade e acuracidade nas distâncias calculadas entre os *beacons* e o dispositivo receptor. Diversos são os motivos que culminaram para esse resultado como o fato do sinal RSSI não ser linear, a reflexão do sinal e a falta de uma equação mais acertiva para converter o sinal recebido numa distância em metros.

A técnica que utiliza grafos para a navegação e localização mostrou-se ser bastante receptiva para a implementação utilizando *beacons*. Principalmente pelo fato de que calcular a proximidade entre um *beacon* e um receptor demonstrou ser muito mais efetiva do que calcular a distância entre eles. A tecnologia BLE pode ter grande serventia em diversas áreas, entretanto, sua utilização para mensurar distâncias e com isso calcular a posição do receptor em um ambiente interno comum não comprovouse-se acertiva com as técnicas utilizadas e descritas nesta pesquisa.

REFERÊNCIAS

ALMANGUER, Hugo Armando Dominguez. Entrevista sobre o comportamento de sinais e antenas. Entrevistador: Djonathan Krause. Blumenau. 2018. Entrevista feita através de conversação – não publicada.

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.

DONG, Qian; DARGIE, Waltenegus. **Evaluation of the reliability of RSSI for indoor localization**. 2012 International Conference On Wireless Communications In Underground And Confined Areas, Alemanha, p.2-3, ago. 2012. IEEE. Disponível em: <https://www.rn.inf.tu-dresden.de/dargie/papers/icwcuca.pdf>. Acesso em: 27 nov. 2018.

ROCHA, Diego T. **Tô Aqui:** Aplicativo para georreferenciamento em ambientes restrito**r**. 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.

BRUINTJES T. M.; KOKKELER A. B. J.; KARAGIANNIS G., et al. **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.

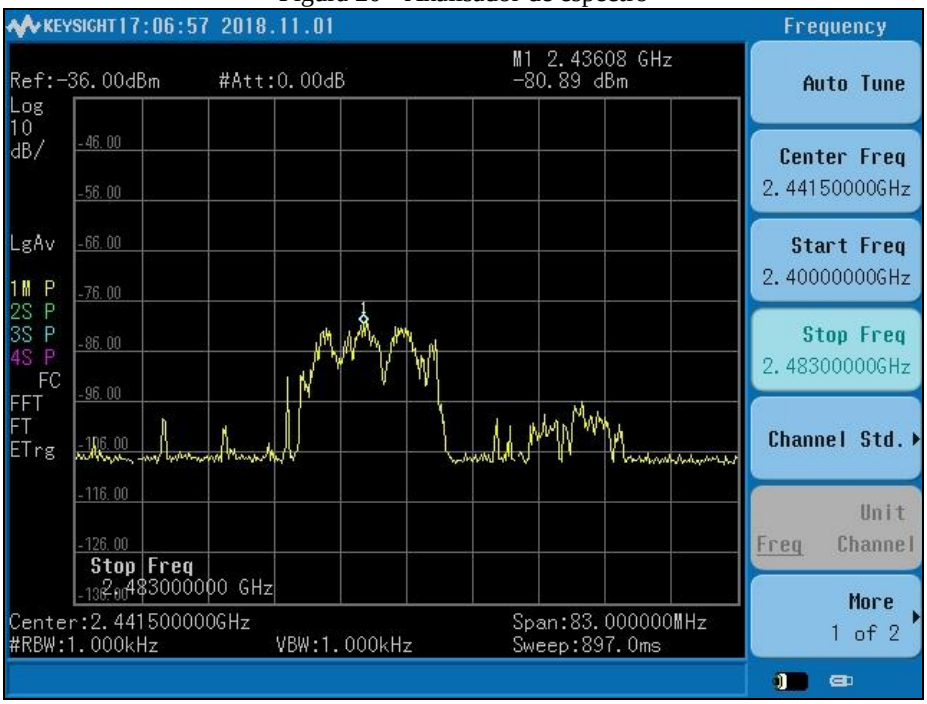
RECK, Marcelo S. ***Beacons* BLE – BLUETOOTH LOW ENERGY – Design e análise de um sistema de localização indoor**. 2016. 84 f. Trabalho de Conclusão de Curso (Engenheiro de Controle e Automação) -Universidade de Caxias do Sul, Caxias do Sul.

ANEXO A – MEDIÇÃO SINAL BEACONS

Em conversa com professor Fábio Luis Perez doutor em Engenharia Elétrica pela UFSC foi sugerigo efetuar medições do sinal emitido pelos dispositivos com um analisador de espectro. As medições foram feitas com o auxílio e

orientação do professor Hugo Armando Dominguez Almaguer, doutor em Engenharia Elétrica pela UFSC. A Figura 20 mostra os resultados obtidos.

Figura 20 - Analisador de espectro

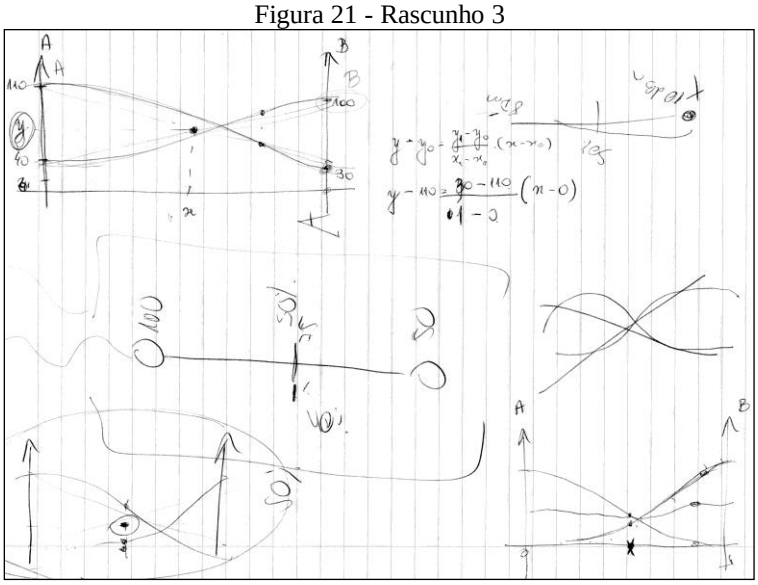


Fonte: Almanguer (2018).

ANEXO B – RASCUNHOS MODELOS MATEMÁTICOS

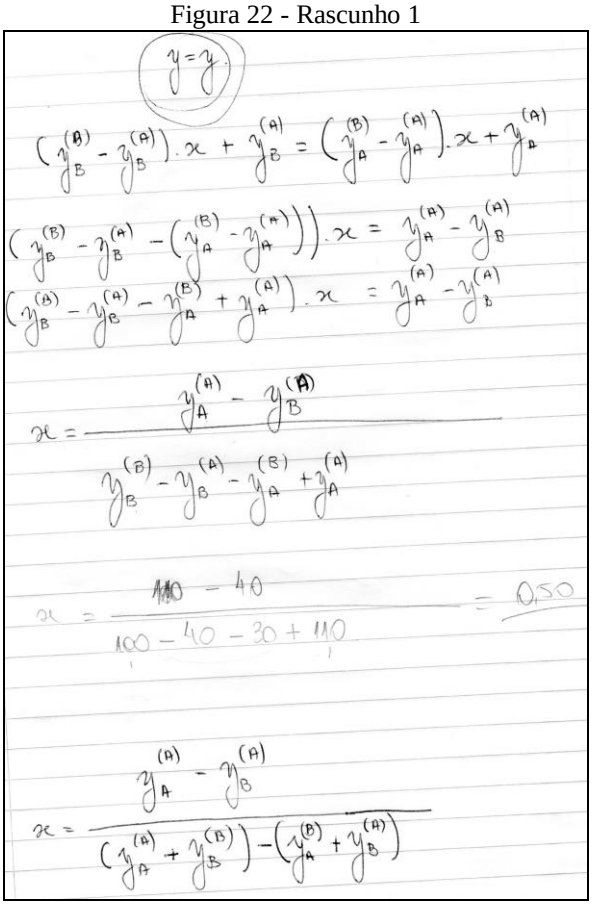
Durante a pesquisa foi discutido com o professor Moacir Manoel Rodrigues Junior, doutor em Métodos Numéricos em Engenharia (Programação Matemática (Teoria dos Jogos)) do curso de matemática da FURB alguns possíveis modelos que permitisse a linearização do sinal dos *beacons*. A seguir a Figura 21, Figura 22 e Figura 23 mostram os rascunhos projetados. Nenhum foi utilizado no desenvolvimento.

Figura 21 - Rascunho 3



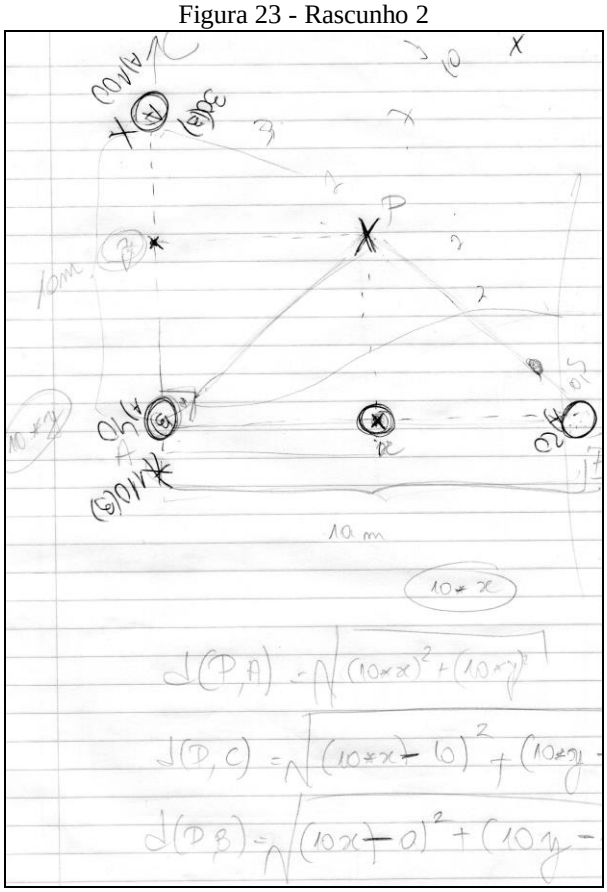
Fonte: Rodrigues Junior (2018).

Figura 22 - Rascunho 1



Fonte: Rodrigues Junior (2018).

Figura 23 - Rascunho 2



Fonte: Rodrigues Junior (2018).