

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO DE JOINVILLE  
CURSO DE ENGENHARIA MECATRÔNICA

GUSTAVO HENRIQUE OLIVEIRA DE PAULA

AVALIAÇÃO E PROJETO DE LOCALIZAÇÃO INDOOR UTILIZANDO BLUETOOTH

Joinville  
2023

GUSTAVO HENRIQUE OLIVEIRA DE PAULA

AVALIAÇÃO E PROJETO DE LOCALIZAÇÃO INDOOR UTILIZANDO BLUETOOTH

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Dr. Anderson Wedderhoff Spengler

Joinville  
2023

## **AGRADECIMENTOS**

Ao meu pai por prestar apoio e me permitir seguir por esse caminho.

Á minha vó que sempre continuou me acompanhando mesmo de longe.

Aos meus irmãos por sempre estarem disponíveis pra me ouvir.

Ao Prof. Dr. Anderson Wedderhoff Spengler, pelo suporte e compartilhamento de conhecimento ao longo desta jornada acadêmica.

Aos meus amigos pelos momentos que dividimos e me possibilitaram chegar nessa etapa.

Á Universidade Federal de Santa Catarina, pela experiência única.

*"There's no such thing as perfect. You're beautiful as you are, Courage. With all your imperfections, you can do anything."*(Peixe do Coragem, 2002).

## RESUMO

Este trabalho aborda a avaliação de localização em ambientes internos (*indoor*) utilizando a tecnologia *bluetooth*. A implementação de *bluetooth* em dispositivos eletrônicos aumenta constantemente com a Internet das Coisas (IoT), principalmente em equipamentos que buscam evitar a utilização de fiação. Levando isso em conta, foi idealizado um cenário em ambiente *indoor* onde se faz necessário localizar dispositivos específicos. Para isso, um sistema capaz de localizar algo a partir do *bluetooth* já presente no dispositivo seria oportuno. Inicialmente foram analisados cinco métodos de obtenção de distância: *Fingerprinting*, *Time of Arrival (ToA)*, *Time Difference of Arrival (TDoA)*, *Angle of Arrival (AoA)* e por RSSI. O primeiro método, chamado de *Fingerprint*, consiste no mapeamento de um ambiente estático. O segundo método foi o Tempo de Chegada (ToA), que estima a distância baseada no tempo que o sinal demora a chegar. O terceiro, o Diferença de Tempo de Chegada (TDoA), obtém a medida pela diferença do tempo de chegada entre os sinais. O quarto, Ângulo de Chegada (AoA), estima a angulação do dispositivo em relação a cada antena para assim obter sua distância. E o quinto caso consiste em basear a distância no valor de RSSI, que indica a potência do sinal *bluetooth*. Todos esses métodos são utilizados para a medição da distância entre o receptor e o emissor, mas para a localização foi considerado somente o método de triangulação. O método de *Fingerprinting* foi descartado para o uso *indoor*, pois precisa de um ambiente estático. O AoA precisa de uma comunicação direcional, o que não se enquadra no *Bluetooth*. O projeto então testou essa aplicação em um cenário onde o dispositivo não está pareado às antenas que o procuraram, o que resultou na eliminação dos métodos de ToA e TDoA. Mas teve sucesso na aplicação da triangulação por RSSI. Mostrando que é possível utilizar como ferramenta de auxílio, porém falta precisão para usos mais específicos.

**Palavras-chave:** *Bluetooth*. RSSI. *Fingerprint*. ToA. TDoA. AoA. Triangulação.

## ABSTRACT

This work addresses the evaluation of indoor location using Bluetooth technology. The implementation of Bluetooth in electronic devices is constantly increasing with the Internet of Things (IoT), particularly in equipment that aims to avoid the use of wiring. Taking this into account, a scenario was envisioned in an indoor environment where it is necessary to locate specific devices. For this purpose, a system capable of locating something based on the Bluetooth already present in the device would be opportune. Initially, five distance acquisition methods were analyzed: Fingerprinting, Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and RSSI-based. The first method, called Fingerprinting, consists of mapping a static environment. The second method was Time of Arrival (ToA), which estimates the distance based on the time it takes for the signal to arrive. The third method, Time Difference of Arrival (TDoA), obtains the measurement by the difference in arrival time between the signals. The fourth method, Angle of Arrival (AoA), estimates the angle of the device in relation to each antenna to obtain its distance. And the fifth case consists of basing the distance on the RSSI value, which indicates the power of the Bluetooth signal. All of these methods are used to measure the distance between the receiver and the transmitter, but for localization purposes, only the triangulation method was considered. The Fingerprinting method was discarded for indoor use, as it requires a static environment. AoA requires directional communication, which does not fit with Bluetooth. The project then tested this application in a scenario where the device is not paired with the antennas that searched for it, which resulted in the elimination of the ToA and TDoA methods. However, it was successful in applying triangulation using RSSI, demonstrating that it is possible to use it as an auxiliary tool, although it lacks precision for more specific uses.

**Keywords:** Bluetooth. RSSI. Fingerprint. ToA. TDoA. AoA. Triangulation.

## LISTA DE FIGURAS

Figura 1 – Representação da conexão <i>Bluetooth</i> e sua versatilidade. . . . .	13
Figura 2 – Quadro de diferenças entre <i>Bluetooth</i> clássico e BLE. . . . .	15
Figura 3 – Representação de uma aplicação do método <i>Fingerprint</i> . Cada triângulo representaria uma medição feita naquele local e cada círculo é uma antena receptora. Assim é obtido o método de <i>fingerprinting</i> . . . . .	16
Figura 4 – Um objeto com dados semelhantes a outros no banco de dados, revelando com uma maior precisão sua localização. . . . .	17
Figura 5 – Localização utilizando o método ToA. . . . .	18
Figura 6 – Representação da diferença temporal entre sinais recebidos (TDoA). . . . .	19
Figura 7 – Representação dos ângulos encontrados pelo método AoA. . . . .	20
Figura 8 – Circunferência representando possíveis localizações do alvo . . . . .	21
Figura 9 – Localizações obtidas com três referenciais. . . . .	22
Figura 10 – Possíveis variáveis no método de triangulação. . . . .	22
Figura 11 – Os três dispositivos posicionados. . . . .	26
Figura 12 – Especificações do local de realização dos testes. . . . .	27
Figura 13 – Pontos aleatórios no espaço do projeto. . . . .	32
Figura 14 – Gráfico com os resultados. . . . .	34

## LISTA DE TABELAS

Tabela 1 – TDoA . . . . .	31
Tabela 2 – Pontos onde o dispositivo foi posicionado. Valores em [m] (metro). .	33

## LISTA DE SIGLAS

AoA Angle of Arrival

BLE Bluetooth Low Energy

CTS Clear to Send

GPS Global Position System

IoT Internet of Things

ISM Industrial, scientific, and medical.

RSSI Received Signal Strength Indicator

SIG Special Interest Group

TDoA Time Difference of Arrival

ToA Time of Arrival

## LISTA DE SÍMBOLOS

$m$	Metros
$r$	Raio
$mA$	Miliampere
$\mu A$	Microampere
$mW$	Miliwatt
$DBm$	Decibel miliwatt
$db/m$	Decibéis por metro
$m/s$	Metro por segundo

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Objetivo	12
<b>1.1.1</b>	<b>Objetivo Geral</b>	<b>12</b>
<b>1.1.2</b>	<b>Objetivos Específicos</b>	<b>12</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
2.1	<i>Bluetooth</i>	13
<b>2.1.1</b>	<b>BLE</b>	<b>15</b>
2.2	Métodos de Obtenção de Distância	15
<b>2.2.1</b>	<b><i>Fingerprinting</i></b>	<b>16</b>
<b>2.2.2</b>	<b><i>Time of Arrival (ToA)</i></b>	<b>17</b>
<b>2.2.3</b>	<b><i>Time Difference of Arrival (TDoA)</i></b>	<b>18</b>
<b>2.2.4</b>	<b><i>Angle of Arrival (AoA)</i></b>	<b>19</b>
<b>2.2.5</b>	<b>Por RSSI</b>	<b>20</b>
2.3	Triangulação	21
<b>3</b>	<b>METODOLOGIA</b>	<b>24</b>
3.1	Métodos de obtenção da distância	24
<b>3.1.1</b>	<b><i>Time of Arrival</i></b>	<b>24</b>
<b>3.1.2</b>	<b><i>Time Difference of Arrival</i></b>	<b>24</b>
<b>3.1.3</b>	<b>Por RSSI</b>	<b>26</b>
3.2	Triangulação	30
<b>4</b>	<b>ANÁLISE DE DADOS</b>	<b>31</b>
4.1	Cenário sem pareamento	31
<b>4.1.1</b>	<b><i>Time Difference of Arrival</i></b>	<b>31</b>
<b>4.1.2</b>	<b>Por RSSI e Triangulação</b>	<b>31</b>
<b>5</b>	<b>CONCLUSÕES</b>	<b>35</b>
	<b>REFERÊNCIAS</b>	<b>37</b>
	<b>APÊNDICE A</b>	<b>39</b>
	<b>APÊNDICE B</b>	<b>41</b>
	<b>APÊNDICE C</b>	<b>42</b>
	<b>APÊNDICE D</b>	<b>43</b>

## 1 INTRODUÇÃO

A noção de localização é importante para a civilização e seu desenvolvimento. A forma mais comumente utilizada, o GPS, foi criado como um sistema militar dos Estados Unidos em meados dos anos 1970. Posteriormente, foi liberado para acesso aos civis, resultando no sistema atual (EL-RABBANY, 2006). A utilização dessa tecnologia acarretou benefícios tanto em lazer quanto em serviços essenciais, e também tornou possível a análise e operação para a criação de diversos utensílios facilitadores.

Porém, um sistema considerado eficiente como esse, também deixa a desejar quando é necessário uma localização *indoor* (*indoor* remete a lugares em ambientes internos) onde a precisão torna-se essencial. Observando o modo de operação do GPS, onde o dispositivo que está sendo localizado envia um sinal aos satélites localizadores, que então mandam sinais de volta e baseado no tempo levado por cada satélite a localização é determinada. Isso resulta em uma precisão que pode variar 22 *m* no melhor caso e mais de 300 *m* no pior (EL-RABBANY, 2006). Essa precisão não é o objetivo esperado para situações *indoor*, onde 22 *m* pode englobar diversas salas.

Com a virada para os anos 2000, a invenção da tecnologia *Bluetooth* dispersou-se para diversos produtos. Mais comumente presente em celulares modernos, atualmente é comum a utilização desta até em eletrodomésticos, onde é usado para englobar todos esses dispositivos em uma rede própria provendo um gerenciamento ao usuário. Isso pode ser observado em *Smarts Homes*, onde por meio do seu próprio dispositivos ou até controle por voz, o cliente é capaz de iniciar processos em seus eletrodomésticos ou até de mudar a decoração do cômodo desejado.

Com esse avanço tecnológico, revelou-se uma necessidade mais precisa de localização para esses dispositivos. Os exemplos deste fator vão desde um cidadão comum esquecendo onde seu dispositivo móvel foi colocado, até grandes empresas com salas lotadas de diferentes dispositivos em busca de um específico. A resolução criada por algumas empresas varia desde o uso de *chips* ou *tokens* que compartilham a localização, ou até utilizando de efeitos sonoros para chamar atenção. Porém podemos utilizar de uma tecnologia já presente nesses dispositivos para localizá-los sem ter que alterar ou anexar nada no dispositivo desejado, apenas precisando configurá-lo.

Com esses fatores, surge a ideia da utilização dessa tecnologia presente em grande parte da tecnologia atual como um método de localização *indoor* para resultados mais precisos necessários em algumas situações.

Em empresas, é muito comum a utilização de equipamentos que possuem conexão utilizando a tecnologia *Bluetooth*, graças ao custo, logo, utilizar essa tecnologia para descobrir o posicionamento resulta em diversos benefícios, indo de organização à

segurança.

Isso abre para as empresas, companhias e fábricas a possibilidade de implantação de sistemas mais diversos baseando-se na localização. É possível mapear tarefas e funções, saber quando algum funcionário acessou alguma área ou deixou de acessar, e assim, obtém-se um maior controle sobre as operações desejadas.

Existem alguns métodos de localização de dispositivos que serão abordados neste trabalho. Em seguida será analisado sua aplicabilidade na abordagem de tecnologia *Bluetooth* e seu funcionamento. E por fim serão aplicados os métodos escolhidos em um ambiente controlado para efetuar-se a comparação de seus resultados.

## 1.1 OBJETIVO

### 1.1.1 Objetivo Geral

Avaliar e projetar um sistema de localização *indoor* baseado na tecnologia *Bluetooth*.

### 1.1.2 Objetivos Específicos

- Levantar métodos de localizações possíveis;
- Analisar suas aplicabilidades em um ambiente *indoor*;
- Projetar a localização de um dispositivo *Bluetooth* utilizando os métodos propostos em um ambiente *indoor*.
- Comparar os resultados dos métodos.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 BLUETOOTH

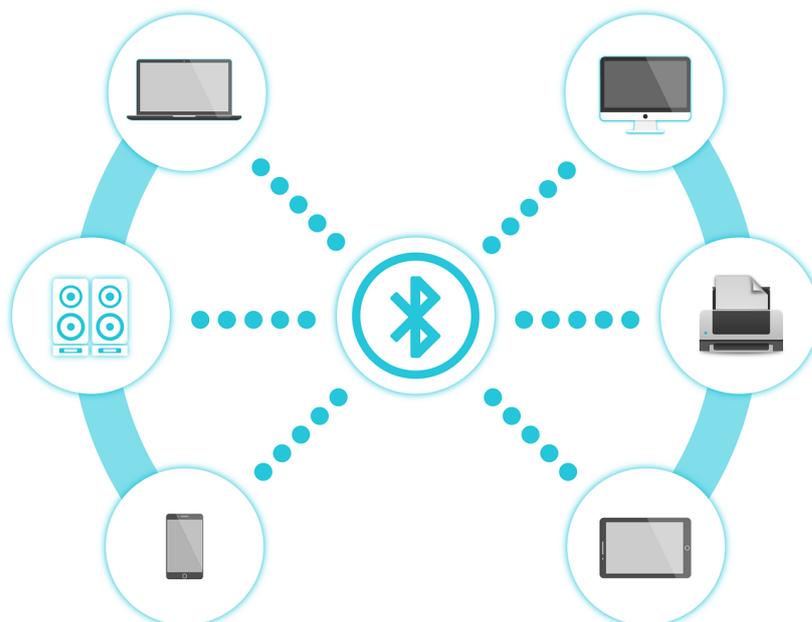
O *Bluetooth* é um protocolo de comunicação sem fio que utiliza ondas de rádio de baixa energia e é compatível com diversos dispositivos. Seu nome provém de Harald Bluetooth Gormsson, rei da Dinamarca que tinha um dente com uma coloração azulada e acinzentada (SIG, 2022).

Em 1996, Intel, Ericsson e Nokia uniram-se para discutir essa tecnologia de transferência de dados a curta distância. O nome *Bluetooth* foi sugerido por um deles fazendo uma comparação entre o rei que unificou a Dinamarca e essa tecnologia que uniria a indústria de computadores e celulares (SIG, 2022).

Em 1998, o *Bluetooth Special Interest Group* (SIG) foi criado para promover a tecnologia. Atualmente, o grupo conta com mais de 33.000 membros (SIG, 2022).

O estabelecimento de conexão *Bluetooth* depende do dispositivo que envia o primeiro pacote de dados e inicia a ligação entre eles. Eles são referidos como cliente e servidor, porém essa conotação é somente um meio de distinguir entre si, não sendo a relação de cliente-servidor normalmente utilizada em programação de rede (HUANG; RUDOLPH, 2007).

Figura 1 – Representação da conexão *Bluetooth* e sua versatilidade.



Fonte: Banco de imagens em pixabay <sup>1</sup>.

<sup>1</sup> Disponível em: <https://encr.pw/1aRec>. Acesso em 29 de março de 2023.

O dispositivo que inicia a comunicação e tem dados de saída escolhe um dispositivo de destino e um protocolo de comunicação antes de estabelecer a conexão. Enquanto isso, o dispositivo que receberá os dados, sendo a entrada destes, deve também estabelecer um protocolo de comunicação e procurar por pareamento antes de aceitar a requisição de conexão e os dados transferidos.

Cada dispositivo *Bluetooth* tem um endereço físico próprio, que é único e permanece durante toda a vida útil do *chip*. Para uma transferência de dados bem-sucedida, um dispositivo precisa conhecer o endereço do outro, que é usado em toda a operação de envio de dados. No entanto, é comum que dispositivos *Bluetooth* tenham nomes mais simples do que um endereço físico complexo. Isso ocorre porque é possível associar um apelido ao dispositivo, que é visto dessa forma quando está visível para outros dispositivos (HUANG; RUDOLPH, 2007).

Para que um dispositivo de saída possa descobrir outros próximos, ele envia uma mensagem de "descoberta" e espera a resposta. Diferentes tipos de dispositivos respondem de formas diferentes a essa mensagem, o que permite a identificação do tipo de dispositivo, como telefones, fones de ouvido ou computadores. Quando solicitada a conexão, ambos os dispositivos pareiam e a transferência de dados é realizada (HUANG; RUDOLPH, 2007). Essa comunicação inicial é vital para o objetivo de localização, pois com as respostas e dados que obtemos nessa mensagem podemos aplicar alguns métodos de localização, que serão detalhados mais a frente.

O *Bluetooth* clássico opera em uma faixa de frequência não licenciada para uso industrial, científico e médico, conhecida como *ISM Band* (Industrial, scientific, and medical band). Essa faixa opera em torno de 2,4 GHz e tem mais de 79 canais para transferência de dados, o que o torna compatível com muitos dispositivos (SIG, 2022).

O *Bluetooth* possui alcance e potência dependendo de sua classe (SIG, 2022), onde:

- Classe 1, 20 DBm: potência máxima de 100 mW, alcance até 100 metros.
- Classe 2, 4 DBm: potência máxima de 2,5 mW, alcance até 10 metros.
- Classe 3, 1 DBm: potência máxima de 1 mW, alcance até 1 metro.

Quando o dispositivo tenta enviar um pacote pela rede, ele verifica se o meio ou canal já está ocupado ou se já há alguma transmissão ocorrendo no canal. Se não detectar nenhuma transmissão ou nenhuma energia de radiofrequência no canal, ele emitirá um CTS (*Clear To Send*) ou "Limpo para Enviar". Isso significa que o adaptador de rede sem fio começará a transmitir o pacote. Usando a mesma técnica, quando outra rede *Bluetooth* co-localizada tenta enviar o pacote, ela adiará a transmissão, causando uma interferência (MATHEW et al., 2009).

### 2.1.1 BLE

O *Bluetooth Low Energy* (BLE), também conhecido como *Bluetooth* de baixa energia, foi introduzido na atualização 4.0 do *Bluetooth*. Este modo apresenta um consumo de energia significativamente menor do que seu predecessor. O modelo de funcionamento do BLE baseia-se em um modo de espera durante a maior parte do tempo, saindo desse modo apenas para fazer conexões com durações de alguns milissegundos. Isso resulta em um consumo de energia com picos de 6 mA, mas uma média de apenas 1 uA (PESSOA, 2016), tornando-o uma escolha popular para dispositivos IoT (*Internet of Things*) e outros dispositivos de baixo consumo.

Figura 2 – Quadro de diferenças entre *Bluetooth* clássico e BLE.

#### Classic Bluetooth vs. Bluetooth low energy

The table below shows a high level comparison between classic *Bluetooth* (also known as *Bluetooth* BR/EDR) and *Bluetooth* low energy technologies.

Technical specification	Classic <i>Bluetooth</i> technology	<i>Bluetooth</i> low energy technology
Radio frequency	2.4GHz	2.4GHz
Distance/Range	~10-100 meters	~10-100 meters
Symbol rate	1-3Mbps	1Mbps
Application throughput	0.7 – 2.1Mbps	305kbps
Nodes/Active slaves	7	Unlimited
Security	56 to 128 bit	128-bit AES
Robustness	FHSS	FHSS
Latency (from not connected state to send data)	100+ ms	<6ms
Government regulation	Worldwide	Worldwide
Certification body	Bluetooth SIG	Bluetooth SIG
Voice capable	Yes	No
Network topology	Point-to-point, scatternet	Point-to-point, star
Power consumption	1 (reference value)	0.01 to 0.5 (use case dependent)
Service discover	Yes	Yes
Profile concept	Yes	Yes
Primary use cases	Mobile phones, headsets, stereo audio, automotive, PCs etc.	Mobile phones, gaming, PCs, sport & fitness, medical, automotive, industrial, automation, home electronics etc.

Fonte: Blog NeilPatel <sup>2</sup>.

## 2.2 MÉTODOS DE OBTENÇÃO DE DISTÂNCIA

Uma aproximação possível para a localização de dispositivos por meio do *Bluetooth* é com a utilização do valor recebido do RSSI (*Received Signal Strength*

<sup>2</sup> Disponível em: <https://neilpatel.com/br/blog/beacon/>. Acesso em 15 de abril de 2023.

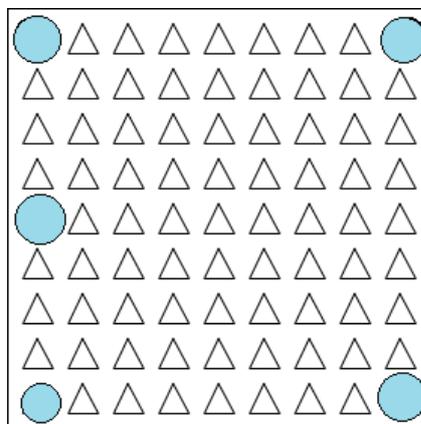
*Indicator*). O RSSI indica o nível de intensidade do sinal recebido, e é medido em decibéis, quanto mais perto da fonte, maior o valor lido, variando de 0 a 100 e sendo negativo (LORENC; SZARATA; CZUBA, 2023). Um fator problemático em sua utilização é a flutuação do sinal recebido, que para ser usado apropriadamente requer a utilização de filtros (DALCE; VAL; BOSSCHE, 2011).

Seguindo os métodos propostos por Adrien Van Den Bossche, Rejane Dalce e Thierry Val em suas publicações, será analisado a implementação dos métodos de *Fingerprinting*, *Time Difference of Arrival* (TDoA), *Time of Arrival* (Toa), *Angle of Arrival* (AoA). É importante salientar que todos os métodos resultam na distância entre a antena e o dispositivo, porém para obter a localização, é necessário obter as coordenadas por meio dessas distâncias. Neste trabalho para a obtenção da localização será abordado o método de Triangulação, abordado na seção 2.3.

### 2.2.1 *Fingerprinting*

O método de *Fingerprinting* consiste em utilizar diversas medições de um mesmo dispositivo em diferentes lugares no mesmo ambiente (Figura 3). Essas medidas em locais diferentes servem para normalizar os efeitos adversos que as ondas de rádio sofrem no ambiente. Com essas medições em locais de localização conhecida, é criado um banco de dados que mapeia o local e proporciona informações mais precisas de possíveis dispersões. Esse método se mostra pouco eficaz em locais dinâmicos, onde ocorrem muitas mudanças no ambiente, pois o mapeamento considerará a adversidade registrada no momento da medição (DALCE; VAL; BOSSCHE, 2011).

Figura 3 – Representação de uma aplicação do método *Fingerprint*. Cada triângulo representaria uma medição feita naquele local e cada círculo é uma antena receptora. Assim é obtido o método de *fingerprinting*.

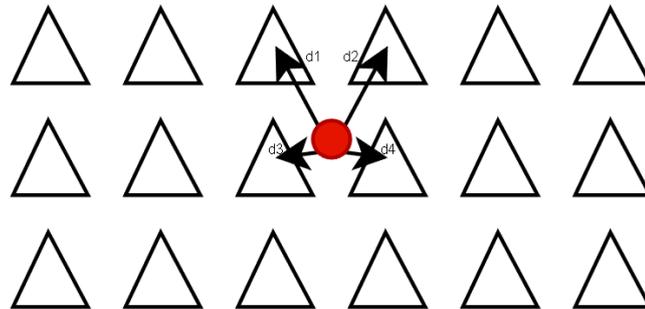


Fonte: Autor.

Com a base de dados obtida, para obter uma posição mais precisa de um dispositivo novo, os seus dados são comparados com os dados do banco de dados,

os que ele possuir uma informação mais próxima vai dar indícios de sua verdadeira posição com mais precisão (QUAN; NAVARRO; PEUKER, 2010), como pode ser observado na Figura 4.

Figura 4 – Um objeto com dados semelhantes a outros no banco de dados, revelando com uma maior precisão sua localização.



Fonte: Autor.

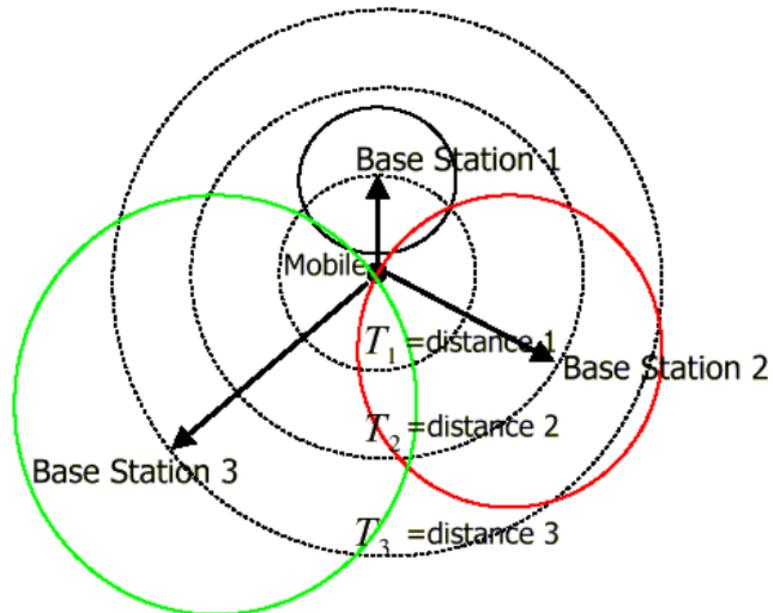
Sua distância é baseada comparando os dados medidos com os que tem-se no banco de dados. Logo, o objeto terá dados semelhantes a pontos medidos próximos, e esta correlação é feita para estimar sua localização. Uma relação que deve ser considerada é a quantidade de antenas no sistema. Quanto mais antenas utilizadas, maior é a precisão da localização. Todavia, uma quantidade elevada de antenas aumenta proporcionalmente os dados a serem tratados no banco de dados, tornando o mapeamento complexo e exigindo alto processamento. Em seus estudos no aprofundamento dos diferentes métodos de *Fingerprint*, (ZHANG et al., 2013) utilizaram de cinco antenas para comparações mais amplas.

A principal dificuldade de aplicação desse método em um ambiente interno é que sua precisão depende do ambiente permanecer estático, caso contrário muita interferência é criada, corrompendo os dados internos (DALCE; VAL; BOSSCHE, 2011). Por estes motivos, esse método não será testado.

### 2.2.2 Time of Arrival (ToA)

A técnica de ToA baseia-se no tempo levado entre o envio de uma mensagem e a captação dela por parte do receptor. Alguns fatores se fazem necessários para o funcionamento deste método, já que seu procedimento é baseado no tempo. Tanto os receptores quanto o emissor devem estar com seus relógios sincronizados para que o cálculo da distância seja feito com base na velocidade de propagação. Efetivamente, por basear-se no tempo, não há degradação com a distância (GUEDES, 2003). Em um exemplo onde tem-se três receptores, a técnica resulta no esquema mostrado na Figura 5.

Figura 5 – Localização utilizando o método ToA.



Fonte: (GUEDES, 2003).

Pela necessidade síncrona temporal entre os receptores, é indispensável que seja garantido que eles possuam o mesmo referencial. Sistemas com horários configurados por uma mesma rede em comum seria um exemplo de utilização certa desse método.

A posição obtida é determinada pela intersecção entre as esferas, podendo ser descritas pela equação 1.

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = m_i^2, (i = 1, 2, \dots, n) \quad (1)$$

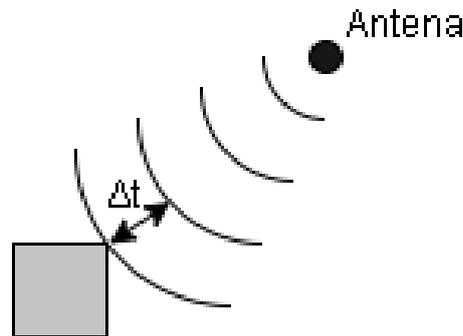
Onde  $(x_i, y_i, z_i)$  e  $m_i, (i = 1, 2, \dots, n)$  são as coordenadas conhecidas das antenas receptoras e a estimativa de alcance.  $n$  é a quantidade de antenas utilizadas. As coordenadas que serão estimadas baseada no tempo de resposta para cada antena são referidas como  $(x, y, z)$  (SHEN; ZETIK; THOMA, 2008).

Serão feitos testes de aplicação desse método na seção 3.

### 2.2.3 Time Difference of Arrival (TDoA)

O TDoA é semelhante ao ToA, mas em substituição a analisar o tempo de um sinal recebido, são enviados dois sinais em um intervalo de tempo qualquer, e então a diferença temporal entre suas chegadas é analisada. A partir dessa diferença de tempo, a distância é calculada (BRANCO; PALA, 2017). Essa representação pode ser demonstrada na Figura 6.

Figura 6 – Representação da diferença temporal entre sinais recebidos (TDoA).



Fonte: Autor.

A posição é então obtida usando uma fórmula semelhante ao método anterior. Porém a estimativa de alcance  $m_i$  é denotada pela diferença entre o dado referência e o dado seguinte registrado, como mostrado na equação 2.

$$m_{i,1} = r_i - r_1, (i = 2, 3, \dots, n) \quad (2)$$

Onde  $m_{i,1}, (i = 2, 3, \dots, n)$  representa a estimativa de alcance do TDoA, e  $r_i (i = 1, 2, \dots, n)$  são os parâmetros desconhecidos da verdadeira distância entre o sinal de referência e o próximo sinal, e  $n$  sendo o número de sinais de referências. E então as coordenadas do receptor são obtidas por meio da equação 3.

$$(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2 = (r_1 + m_{n,1})^2 \quad (3)$$

Onde há quatro variáveis a ser encontradas,  $(x, y, z)$  e  $r_1$  (SHEN; ZETIK; THOMA, 2008).

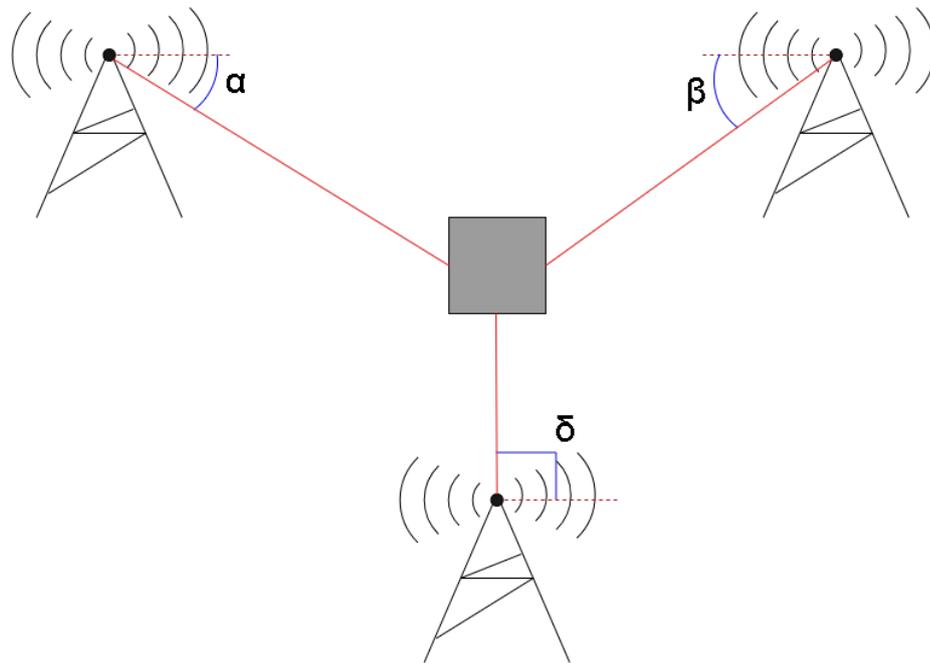
A aplicação desse método também será melhor detalhada na seção 3.

#### 2.2.4 Angle of Arrival (AoA)

O método AoA utiliza método semelhante ao ToA e TDoA, com o receptor recebendo os sinais dos emissores e tratando para obtenção das coordenadas. Neste método, o dispositivo receptor deve estar ao alcance de todos os emissores.

Essas antenas emissoras terão coordenadas já especificadas e conhecidas, logo as antenas devem ser fixadas durante o processo. Em seus trabalhos, (LORENC; SZARATA; CZUBA, 2023) propõem a criação de uma matriz baseada nos dados recebidos pelo receptor. Nessa matriz, as pequenas variações de amplitude e suas defasagens revelam um ângulo aproximado que o emissor está da antena, juntando os três ângulos, é possível estabelecer sua localização de forma aproximada. Esse método é demonstrado na Figura 7.

Figura 7 – Representação dos ângulos encontrados pelo método AoA.



Fonte: Autor.

Em complemento a isso, (DALCE; VAL; BOSSCHE, 2011) propõe a obtenção das coordenadas por meio do cálculo direto com os dados. Considerando então um dispositivo  $M$  com localização desconhecida, supõe-se que seja recebido sinal de duas antenas, sendo estas  $B_1(x_1, y_1)$  e  $B_2(x_2, y_2)$ . Os ângulos  $\alpha$  e  $\beta$  são obtidos entre os eixos diretos de  $MB_i$  com as linhas diretamente perpendiculares a direção de cada antena, utilizando a matriz mencionada anteriormente. Com a descoberta dos ângulos, a intersecção de  $(B_1, M)$  e  $(B_2, M)$  será as coordenadas de  $M$ , como demonstrado na Equação 4.

$$M \left( \frac{y_2}{\tan\alpha + \tan\beta}; \frac{y_2 \tan\alpha}{\tan\alpha + \tan\beta} \right) \quad (4)$$

Por esse método se basear na comunicação direcional, recurso pouco aplicado ao *Bluetooth* (WANG, 2013), esse método não será testado neste trabalho.

### 2.2.5 Por RSSI

Primeiramente é necessário obtermos a distância do objeto até a antena, para isso, o seu valor de RSSI será lido, e então o cálculo será feito baseado na Equação 5.

$$d = 10^{-\frac{RSSI - A}{10n}} \quad (5)$$

Onde  $d$  é a distância em metros da antena para o objeto.  $RSSI$  é o valor lido de RSSI do objeto.  $A$  representa o nível de sinal recebido de um objeto que estava a

uma distância de um metro, servindo assim de referencial. E  $n$  é um coeficiente de auxílio de calibração, sendo calculado pela Equação 6.

$$n = -\frac{RSSI - A}{10\log_{10}(d)} \quad (6)$$

Para utilizar essa equação, deve-se fazer a medição do objeto em uma distância conhecida, justamente para calibrar em relação ao ambiente. Nela,  $n$  é o fator de correção do ambiente, sendo medido em db/m.  $RSSI$  é o valor lido na distância já conhecida.  $A$  representa a potência do sinal RSSI lido na distância de um metro. E  $d$  representa a distância conhecida onde foi colocado o objeto (LORENC; SZARATA; CZUBA, 2023).

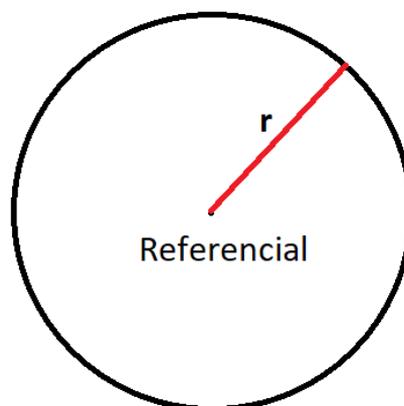
### 2.3 TRIANGULAÇÃO

Os métodos abordados na seção anterior resultam nas distâncias entre as antenas e os dispositivos. Porém a fim de obter a localização, a triangulação foi utilizada.

O método da triangulação tornou-se uma das principais técnicas de localização. Nele, busca-se um ponto específico em relação a três referenciais, e então, utilizando propriedades geométricas dos triângulos, esse ponto é localizado (YANG; LIU, 2010).

Considerando a comunicação entre o referencial e o alvo, é possível calcular uma medida de separação entre eles. No entanto, apenas com essa informação isolada, não é suficiente para determinar a posição exata do alvo. Isso ocorre porque, ao considerar todos os possíveis locais que estão àquela distância fixa do referencial, forma-se uma linha circular em torno dele (Figura 8), na qual o raio  $r$  representa a medida obtida anteriormente.

Figura 8 – Circunferência representando possíveis localizações do alvo

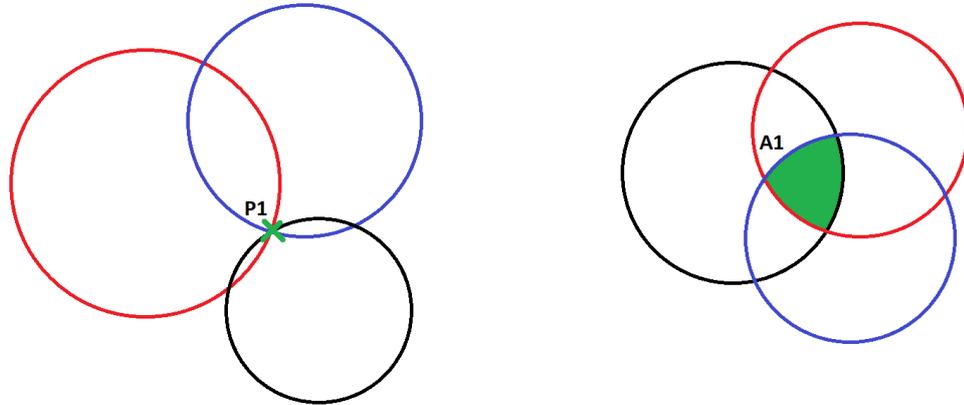


Fonte: Autor.

Para contornar esse problema, é adicionado mais referenciais. Com somente um referencial a mais, é obtido uma interseção entre duas circunferências, o que daria dois pontos. Logo, é adicionado mais um referencial, obtendo assim a triangulação

com três referências que resultam nas duas possibilidades mostradas na Figura 9.

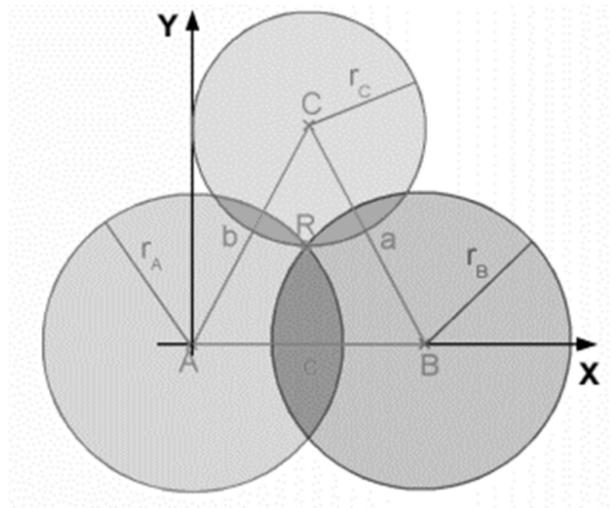
Figura 9 – Localizações obtidas com três referenciais.



Fonte: Autor.

A primeira possibilidade ocorre quando a intersecção entre os três círculos resultam em um ponto  $P1$ , esse ponto é a localização desejada do alvo com a maior precisão. A segunda possibilidade, é a mais frequente, é quando a intersecção resulta em uma área  $A1$ , embora não seja preciso, sua margem de diferença é suficientemente próxima da localização desejada. Para calcularmos, considera-se as variáveis presentes na Figura 10.

Figura 10 – Possíveis variáveis no método de triangulação.



Fonte: (LORENC; SZARATA; CZUBA, 2023).

Onde tem-se três antenas, intituladas de  $A$ ,  $B$  e  $C$ . Cada uma delas possui uma distância até o objeto  $R$ , sendo essas medidas representadas como  $r_A$ ,  $r_B$  e  $r_C$ . Nisso, um triângulo com lados  $a$ ,  $b$  e  $c$  é formado. Utilizando propriedades de círculos, (LORENC; SZARATA; CZUBA, 2023) chegaram nas seguintes equações para definir a coordenada de  $R$ :

$$x_R = \frac{x_B^2 + r_A^2 - r_B^2}{2x_B} \quad (7)$$

$$y_R = \frac{y_C^2 + x_C^2 - 2x_C x_R + r_A^2 - r_C^2}{2y_C} \quad (8)$$

$$z_R = \sqrt{r_A^2 - y_R^2 - x_R^2} \quad (9)$$

Esse método será aplicado utilizando de três antenas e um receptor, sendo abordado na seção 3.

### 3 METODOLOGIA

Para análise dos métodos apresentados, foi considerado um cenário onde pretende-se obter a localização de um dispositivo cuja única informação disponível é o seu nome, mas seu *Bluetooth* não está pareado às antenas. Logo, a localização é obtida pelos dados que são disponibilizados somente na varredura por dispositivos ao redor.

Nesse cenário, foi utilizado os seguintes componentes para a tentativa de localização:

- 3 notebooks com *Bluetooth*, para servirem como antenas.
- 1 *smartphone* Mi A3 que será o dispositivo com *Bluetooth* a ser rastreado.
- 1 *smartwatch* Mi Band 7, que será o dispositivo com BLE a ser rastreado.

Posteriormente foi feita a tentativa de aplicação dos métodos escolhidos sob essas circunstâncias.

#### 3.1 MÉTODOS DE OBTENÇÃO DA DISTÂNCIA

##### 3.1.1 *Time of Arrival*

Como mencionado na seção 2.2.2, o método ToA possui alguns pré-requisitos. O primeiro deles é o fator sincronismo entre as antenas e o dispositivo.

Esse fator não pode ser ajustado, pois não há pareamento entre as antenas e o relógio. O segundo fator também se perde pela falta de conexão, já que seria necessário possuir o horário em que o sinal foi recebido, e sem conexão ao dispositivo esse valor não pode ser alcançado.

No cenário proposto, esse método não é aplicável.

##### 3.1.2 *Time Difference of Arrival*

Embora a situação anterior acabe por se repetir nesse método, podemos tentar contorná-lo de outra forma. Sua lógica utiliza a diferença de tempo entre dois sinais enviados, a hipótese considerada será a de que podemos utilizar a diferença de tempo entre o momento de envio do sinal de varredura e o momento de detecção do dispositivo.

Para isso então, foi criado um *script* em *Python* que tentará reproduzir este efeito. Primeiramente é feita a inclusão das bibliotecas necessárias, foi incluído então as bibliotecas *bluetooth* e *time*. A primeira serve para utilização das suas funções que permitem a descoberta de dispositivos com *Bluetooth* clássico próximos. A segunda

permite retornar o tempo do relógio interno da máquina, podendo estimar o tempo entre uma operação e outra pela diferença entre esses tempos.

---

```
import bluetooth
import time
```

---

Então criam-se algumas variáveis de escopo para utilização nas funções. Para uma precisão melhor de resultados, cria-se uma variável chamada *NUM\_MEASUREMENTS* que será o número de varreduras feita. O resultado final então será a média da diferença de tempo entre essas varreduras.

---

```
NOME_DISPOSITIVO = "Mi A3"
ENDERECO_DISPOSITIVO = None
NUM_MEASUREMENTS = 100
```

---

A seguir, é criada uma função para escanear dispositivos próximos e obter seu endereço. A intenção desse passo é obter uma resposta qualquer do dispositivo para assim o tempo de pergunta e resposta ser estimado.

---

```
def escaneia_dispositivos():
    dispositivos_proximos = bluetooth.discover_devices(duration
        =8, lookup_names=True, flush_cache=True)
    for addr, name in dispositivos_proximos:
        if name == NOME_DISPOSITIVO:
            return addr
    return None
```

---

Então, na *main*, cria-se uma interação com duração da quantidade de medidas imposta no código. Uma requisição de endereço é feita, para verificar se há uma resposta.

---

```
if __name__ == '__main__':
    for _ in range(NUM_MEASUREMENTS):
        hora_envio = time.time()
        print(f"Sinal de procura enviado em: {hora_envio}")

        ENDERECO_DISPOSITIVO = escaneia_dispositivos()

        hora_recebe = time.time()
        print(f"Sinal de procura encontrado em: {hora_recebe}")
```

---

Se encontrado, é feito o cálculo da diferença de resposta do tempo e

armazenado em um vetor. Caso contrário, uma mensagem de aviso alerta que o dispositivo não foi encontrado.

---

```

if ENDERECO_DISPOSITIVO:
    print(f"Dispositivo '{NOME_DISPOSITIVO}'
          encontrado. Endereço: {
          ENDERECO_DISPOSITIVO}")
    TDoA = hora_recebe - hora_envio
    print(f"Diferença de tempo: {TDoA} segundos
          ")
    diferencas_tempo.append(TDoA)
else:
    print(f"Dispositivo '{NOME_DISPOSITIVO}' nã
          o encontrado.")

```

---

Fora da iteração, quando o número de medidas estimadas foi alcançado, é feita uma verificação para caso o vetor *diferencas\_tempo* esteja vazio. Se não estiver, sua média é calculada e exibida. Caso contrário, um aviso é dado ao usuário.

---

```

if diferencas_tempo:
    media = sum(diferencas_tempo) / len(diferencas_tempo)
    print(f"Média das diferenças de tempo: {media}")
else:
    print("Nenhuma diferença de tempo encontrada.")

```

---

O código pode ser verificado completo na seção 5, os resultados obtidos com esse método são detalhados na seção 4.

### 3.1.3 Por RSSI

Nesse caso apresentado, o RSSI pode ser facilmente obtido durante a varredura.

Figura 11 – Os três dispositivos posicionados.



Fonte: Autor.



Nele, é utilizado a biblioteca *bleak* para fazer a varredura de dispositivos com BLE próximos. Essa foi a biblioteca utilizada, pois o dispositivo a ser localizado (*Xiaomi Mi Band 7*) se comunica com BLE. A biblioteca *asyncio* é utilizada também para a criação de funções assíncronas, para que assim seja possível executar o armazenamento dos dados sem interromper as varreduras por dispositivos, como mostrado no código a seguir.

---

```
import asyncio
from bleak import BleakScanner
```

---

É criado duas variáveis no escopo, sendo elas *nome\_disp*, que contém o vetor de caracteres que formam o nome do dispositivo que será localizado, e *num\_medicoes* que contém a quantidade de varreduras que é feita.

---

```
nome_disp = 'Xiaomi Smart Band 7 25AA'
num_medicoes = 100
```

---

Então, é definida uma função assíncrona de nome *scan\_devices()*. Essa função cria uma variável *scanner*, sendo esta do tipo *BleakScanner*, um recurso da biblioteca *bleak* que cria uma estrutura capaz de realizar varreduras por dispositivos BLE (BLIDH, 2020). Uma outra variável de nome *devices* é criada, e recebe os dispositivos detectados na varredura. É feita uma verificação, e se o dispositivo com o nome definido anteriormente foi encontrado, o valor de seu RSSI é retornado.

---

```
async def scan_devices():
    rssi_encontrado = 0
    scanner = BleakScanner()
    devices = await scanner.discover()
    for device in devices:
        if device.name == nome_disp:
            print('Encontrado')
            rssi_encontrado = device.rssi
            print(rssi_encontrado)
        else:
            print('Não encontrado')
    return rssi_encontrado
```

---

Uma função *main* é criada, também assíncrona em razão da função anterior possuir um retorno assíncrono. A quantidade de varreduras definidas no escopo é feita, e os RSSI's encontrados são armazenados em um vetor. Ao final das varreduras, é feita uma média dos valores encontrados e esse valor é exibido no terminal. A função

também possui uma alternativa caso nenhum valor seja encontrado, exibindo uma mensagem de aviso.

---

```

async def main():
    rssi_valores = []
    for _ in range(num_medicoes):
        valor_rssi = await scan_devices()
        if valor_rssi != 0:
            rssi_valores.append(valor_rssi)

    if rssi_valores:
        media_rssi = sum(rssi_valores) / len(rssi_valores)
        print(f"Média dos valores de RSSI: {media_rssi}")
    else:
        print("Nenhum valor de RSSI encontrado.")

```

---

Por fim, essas duas funções são chamadas em seus respectivos *loops*, devido sua característica assíncrona. Deste modo, elas funcionam independente do tempo de processo uma da outra.

---

```

loop = asyncio.get_event_loop()
loop.run_until_complete(main())

```

---

O código completo está disponível na seção 5.

Com essa execução, é possível obter o valor de  $A$  e  $n$  da expressão 6. Então o dispositivo foi deixado a exato um metro da antena, e 300 medições foram feitas, para uma maior precisão do RSSI médio. Esse valor foi então armazenado como o  $A$  dessa antena. Esse processo é então repetido nos outros notebooks. Em seguida o dispositivo foi deixado a uma distância aleatória do emissor, seu RSSI médio foi calculado e a distância foi medida manualmente para encontrar o fator de calibração  $n$ . Esse processo foi também repetido nas outras antenas.

Com os valores essenciais da expressão 5, já pode-se estimar a distância entre cada antena e o dispositivo.

Para essa aplicação, considerando o referencial definido na Figura 12 e as coordenadas das antenas, a posição do dispositivo é calculada por meio de um segundo *script* de *Python*, que com ao receber os parâmetro  $A$ ,  $rssi\_conhecido$ , que seria o valor do RSSI na distância aleatória,  $d\_conhecido$  que seria o valor da distância aleatória e  $rssi$  que é o valor do RSSI medido na posição atual do dispositivo, a distância entre ele e a antena é estimada na função *calc\_dist*, como demonstrado a seguir.

---

```

def calc_dist(rssi_conhecido, rssi, A, d_conhecido):

```

```

n = -(rssi_conhecido - A)/(10*math.log10(d_conhecido))
d = math.pow(10,-(rssi - A)/(10*n))
return d

```

---

O *script* está disponível inteiramente na seção 5. Finalizando, as distâncias são obtidas e então a triangulação é aplicada.

### 3.2 TRIANGULAÇÃO

Para obtenção das coordenadas, basta realizar as medições nas três antenas e aplicar o método de triangulação, utilizando as equações 7, 8 e 9. Com essa finalidade, um terceiro código é criado, onde tem-se uma função responsável por aplicar as fórmulas citadas e obter as coordenadas, como pode-se observar a seguir. Para um tratamento mais simples dos dados e melhor visualização, considera-se somente os eixos  $X$  e  $Y$ .

```

def obtem_coordenadas(Antena_A, Antena_B, Antena_C):
    x_r = (pow(Antena_B[0],2)+pow(Antena_A[3],2)-pow(Antena_B
        [3],2)) / (2*Antena_B[0])
    y_r = (pow(Antena_C[1],2)+pow(Antena_C[0],2) - 2*Antena_C
        [0]*x_r +pow(Antena_A[3],2)-pow(Antena_C[3],2)) / (2*
        Antena_C[1])

```

---

O código e sua estrutura estão disponíveis inteiramente na seção 5. Utilizando esse método, foram obtidos alguns dados que serão discutidos na seção 4.

## 4 ANÁLISE DE DADOS

Com a metodologia definida, iremos avaliar os resultados obtidos nos cenários propostos.

### 4.1 CENÁRIO SEM PAREAMENTO

#### 4.1.1 *Time Difference of Arrival*

A principal dificuldade nesse método está no fator que quanto mais distante do dispositivo, maior deve ser a diferença de chegada entre os sinais, para que então possa ser aplicada uma conversão direta do tempo com a velocidade de propagação. Foi então obtido a diferença de tempo nas medidas de 1, 2 e 3 metros para verificar se há uma escala linear do tempo, alguns dos resultados podem ser observados na Tabela 1.

Tabela 1 – TDoA

Distância (m)	Tempo total (s) - Experimento 1	Tempo total (s) - Experimento 2
1	10.297420954704284	10.257900977134705
2	10.295750229358672	10.258889803886413
3	10.254848637580870	10.251271247863770

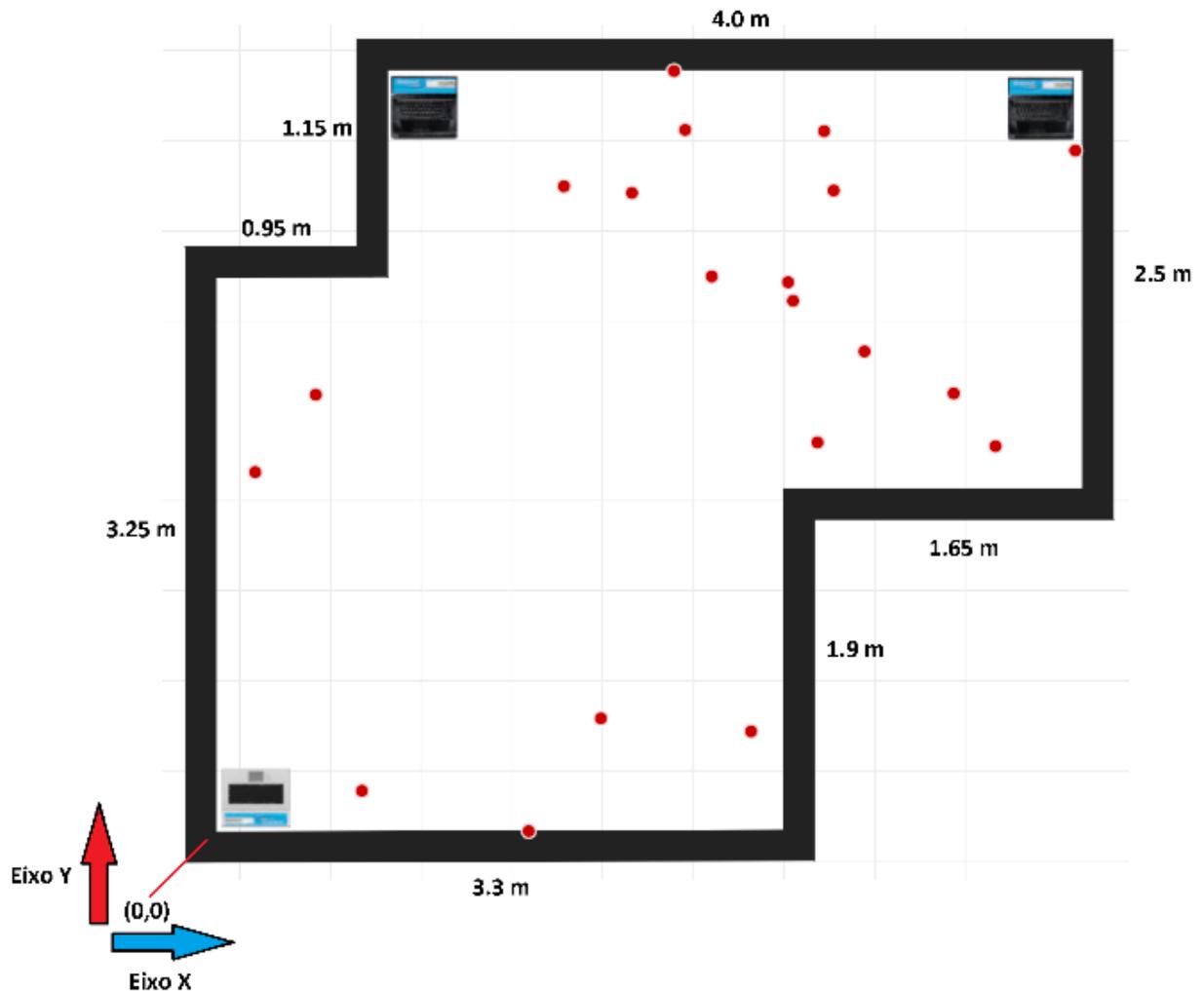
Como mencionado na seção 2, o *Bluetooth* utiliza ondas de rádio, que viajam na velocidade da luz, ou seja, 299 792 458 m/s (BIPM, 2022). Com esse valor grande, uma diferença pequena de tempo no resultado poderia resultar em grandes distâncias em ambientes *indoor*. Mas para o funcionamento do método no cenário idealizado, uma diferença de um e três metros deve ser bem definida, idealmente então quer-se que o tempo levado aumente com a distância. Mas os resultados obtidos mostram que as medições se entrelaçam a todo momento. Primeiramente, se fosse considerado só o tempo do sinal, o tempo não seria tão longo. Porém como não está pareado, não tem-se um controle sobre um sinal específico para monitorar.

Por conseguinte, nesses aproximados dez segundos está toda a preparação do processo de varredura. Este processo culmina na análise de todos os dispositivos no ambiente para procurar o especificado.

#### 4.1.2 Por RSSI e Triangulação

Para analisar os resultados, criou-se um *script* simples para randomizar 20 pontos no espaço, como mostrado na Figura 13.

Figura 13 – Pontos aleatórios no espaço do projeto.



Fonte: Autor.

Os pontos representam locais onde o dispositivo será posicionado para as antenas iniciarem as medidas. Quando posicionado, o processo mencionado na seção 3 é realizado, e então obtêm-se as coordenadas estimadas pelo método, esses dados são apresentados na Tabela 2. Para melhorar a precisão, o dispositivo passou por 100 medições no ponto escolhido e então a média dessas foi passada para o cálculo da distância.

Com os pontos estimados e os pontos reais, um dos referenciais é escolhido para que a distância até ele seja calculada, e as diferenças entre as medidas dos pontos originais e os estimados seja mais evidenciado. O referencial do notebook 1 foi o escolhido, pois ele possui coordenadas na origem (0.00, 0.00) facilitando a visualização.

Com a distância do ponto real até a origem e a distância do ponto estimado, pode-se estimar o erro de cada medição por meio da diferença entre elas. Com esses valores, a análise da eficiência do método pode ser feita. A Tabela 2 apresenta todos esses dados, onde:

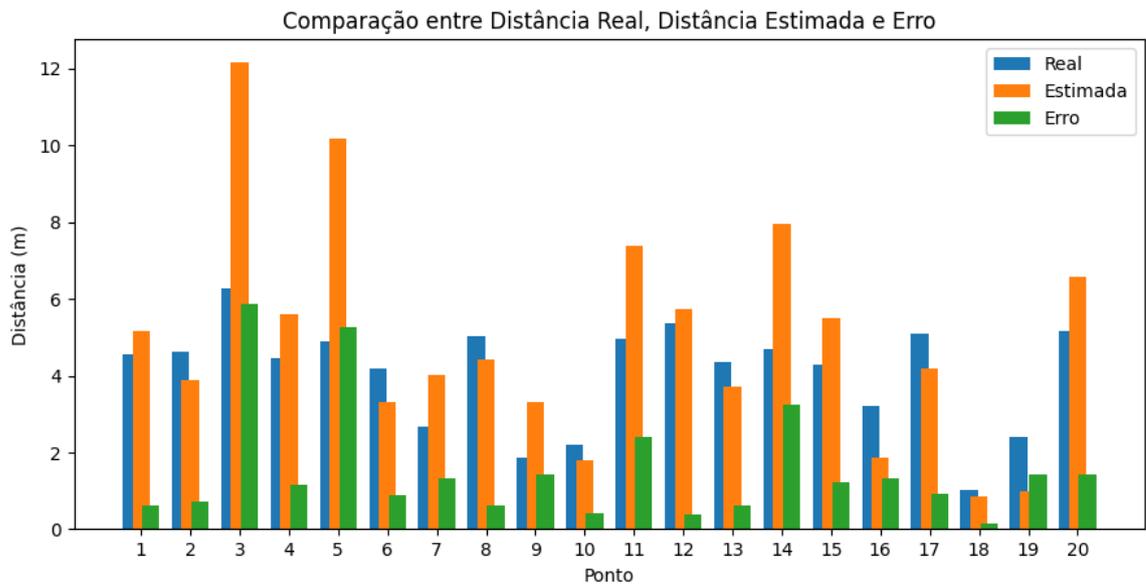
- *Ponto*: indica o ponto que está sendo analisado.
- *Eixo X*: indica o valor do eixo *X* real daquele ponto.
- *Eixo Y*: indica o valor do eixo *Y* real daquele ponto.
- *Eixo X'*: indica o valor do eixo *X* estimado daquele ponto.
- *Eixo Y'*: indica o valor do eixo *Y* estimado daquele ponto.
- *Distância O*: indica o valor da distância do ponto real à origem daquele ponto.
- *Distância O'*: indica o valor da distância do ponto estimado à origem daquele ponto.
- *Erro*: indica o valor da diferença entre a distância do ponto real à origem e a distância do ponto estimado à origem.

Tabela 2 – Pontos onde o dispositivo foi posicionado. Valores em [m] (metro).

<b>Ponto</b>	<b>Eixo X</b>	<b>Eixo X'</b>	<b>Eixo Y</b>	<b>Eixo Y'</b>	<b>Distância O</b>	<b>Distância O'</b>	<b>Erro</b>
1	3.34	4.00	3.12	3.29	4.57	5.18	0.61
2	3.31	3.69	3.22	1.27	4.62	3.90	0.72
3	4.90	9.09	3.95	8.06	6.29	12.15	5.86
4	2.44	3.51	3.72	4.36	4.45	5.60	1.15
5	2.73	6.49	4.07	7.84	4.90	10.18	5.28
6	3.48	2.96	2.33	1.47	4.19	3.30	0.88
7	0.68	1.88	2.60	3.56	2.69	4.03	1.34
8	4.47	3.69	2.31	2.42	5.03	4.41	0.62
9	1.88	1.25	0.05	-3.05	1.88	3.30	1.42
10	0.35	-0.81	2.17	1.59	2.20	1.79	0.41
11	4.23	5.29	2.60	5.14	4.96	7.38	2.41
12	3.51	5.20	4.06	2.41	5.37	5.74	0.37
13	2.88	3.35	3.25	1.63	4.34	3.73	0.61
14	3.74	5.34	2.84	5.90	4.70	7.96	3.26
15	2.06	3.55	3.76	4.18	4.28	5.49	1.20
16	3.12	1.53	0.73	1.09	3.20	1.88	1.32
17	2.67	3.19	4.35	2.71	5.11	4.19	0.92
18	0.95	0.66	0.40	0.58	1.03	0.87	0.16
19	2.28	-0.99	0.80	0.09	2.42	0.99	1.43
20	3.56	5.26	3.73	3.96	5.16	6.58	1.42

Com a discrepância obtida, podemos produzir um gráfico de barras (ESTRELA, 2017) para analisar a diferença obtida em cada ponto, como mostrado na Figura 14. Nela podemos realizar uma análise detalhada da eficiência dos resultados.

Figura 14 – Gráfico com os resultados.



Fonte: Autor.

Avaliando principalmente a diferença entre eles tem-se dados interessantes. O valor máximo de erro foi 5,86 m e seu valor mínimo foi de 0,16 m, possuindo uma média de 1,561 m para esses vinte pontos.

## 5 CONCLUSÕES

Conforme apresentado neste trabalho, buscou-se a avaliação de localização *indoor* utilizando *Bluetooth* de forma simples, mas que finda por utilizar conhecimentos específicos da tecnologia, não sendo trivial de implementar. Como um dos objetivos presentes neste trabalho inclui a obtenção da localização de um dispositivo utilizando os métodos levantados, pode-se observar a grande dificuldade acerca da precisão nessa tecnologia.

Para o cenário onde não há pareamento, e a única informação obtida é o nome do dispositivo, os testes foram feitos. O método de ToA já não foi possível nesse cenário, devido a impossibilidade de sincronização e obtenção do tempo entre os dispositivos emissores e receptor.

Essa limitação pode ser superada se tanto as antenas quanto os dispositivos estiverem interligados em uma mesma rede, sincronizados pelo mesmo relógio interno.

A tentativa de TDoA mostrou-se falha diante dos dados coletados, por conta do fator randômico acerca do processo de varredura por dispositivos, o tempo resulta em não ser constante, causando essa grande variância que impede uma precisão para um ambiente *indoor*. Dado então esse cenário proposto, o método também não seria aplicável. Um fator a ser analisado em trabalhos futuros é a influência de outros processos no sistema operacional utilizado, e como isso pode ser manipulado de uma forma que o escaneamento fique preciso.

A utilização do método guiado pelo valor de intensidade do sinal RSSI e aplicação da triangulação mostrou-se utilizável, mas não preciso, a sua aplicabilidade é aceitável para ambientes onde a precisão não se faz importante. A noção de uma taxa de erro que varia entre 16 centímetros e quase 6 metros implica ainda mais no quanto o ambiente altera constantemente o sinal de *Bluetooth*.

No gráfico criado para análise, podemos concluir que possui um erro aceitável. Realçando porém que sua precisão é baixa, e possui uma alta tendência de receber interferência do ambiente a qualquer momento, acarretando em erros grandes, vide dois valores de erros acima de 4 metros presentes no gráfico.

O *Bluetooth* mostrou-se uma tecnologia de difícil utilização para precisão, mesmo com aplicação de métodos para auxiliar e com um grande número de medidas, a variância entre as possibilidades de medições em um mesmo ponto acaba por complicar sua exatidão, mas não é totalmente descartável, possuindo uma margem decente para usos recreativos, como auxílio na procura de um equipamento específico, já que reduziria a área de busca.

Assim, os objetivos deste trabalho foram cumpridos, os métodos foram

levantados, analisados e filtrados. Além disso, foi feito um projeto para aplicação dos métodos na tentativa de obter a localização de um dispositivo usando o *Bluetooth*.

Com o objetivo do trabalho alcançado, indica-se como possíveis trabalhos futuros o aprofundamento nos métodos sugeridos para aplicações com *Bluetooth*. Analisar a aplicabilidade do método de *Fingerprint* em ambientes estáticos e as vantagens que podem ser retiradas deste cenário. Utilizar dispositivos conectados a uma mesma rede de referência para sincronismo temporal e aplicar os métodos de ToA e TDoA. Verificar a utilização de outros métodos de obtenção de distância não citados neste trabalho e suas eficiências. Considerar a utilização de outros métodos de localização e abordagens para o cálculo da posição do dispositivo.

## REFERÊNCIAS

- BIPM. **SI brochure, Section 2.1.1.1. BIPM.** 2022. Disponível em: <https://www.bipm.org/documents/20126/41483022/SI-Brochure-9-EN.pdf>. Acesso em: 03 jun. 2023.
- BLIDH, H. **BleakScanner class.** 2020. Disponível em: <https://bleak.readthedocs.io/en/latest/api/scanner.html>. Acesso em: 02 jun. 2023.
- BRANCO, G. D.; PALA, S. V. V. Avaliação de técnicas de localização de dispositivos ble para a physical web e prova de conceito. **Universidade de Brasília, Instituto de Ciências Exatas, Departamento de Ciência da Computação**, 2017.
- DALCE, R.; VAL, T.; BOSSCHE, A. van den. Comparison of indoor localization systems based on wireless communications. **Wireless Engineering and Technology**, v. 2, p. 240–256, 01 2011.
- EL-RABBANY, A. **Introduction to GPS: the global positioning system.** Norwood, EUA: Artech House, 2006.
- ESTRELA, J. V. N. C. B. D. S. M. T. C. Boxplot: Um recurso gráfico para a análise e interpretação de dados quantitativos. In: **Rev Odontol Bras Central**. [S.l.: s.n.], 2017. v. 26, n. 76.
- GUEDES, E. M. P. Estudo de técnica híbrida de localização de estações móveis baseada em tdoa e aoa. **Instituto Militar de Engenharia, IME**, 2003.
- HUANG, A. S.; RUDOLPH, L. **Bluetooth Essentials for Programmers.** [S.l.]: Cambridge University Press, 2007.
- LORENC, A.; SZARATA, J.; CZUBA, M. Real-time location system (rtls) based on the bluetooth technology for internal logistics. **Sustainability**, v. 15, n. 6, 2023. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/15/6/4976>.
- MATHEW, A. et al. Ieee 802.11 bluetooth interference: Simulation and coexistence. In: **2009 Seventh Annual Communication Networks and Services Research Conference**. [S.l.: s.n.], 2009. p. 217–223.
- PESSOA, L. **Introdução ao Bluetooth Smart (BLE).** 2016. Disponível em: <https://embarcados.com.br/bluetooth-smart-ble/>. Acesso em: 15 nov. 2022.
- QUAN, M.; NAVARRO, E.; PEUKER, B. Wi-fi localization using rssi fingerprinting. 2010.
- SHEN, G.; ZETIK, R.; THOMA, R. S. Performance comparison of toa and tdoa based location estimation algorithms in los environment. In: **2008 5th Workshop on Positioning, Navigation and Communication**. [S.l.: s.n.], 2008. p. 71–78.
- SIG, B. **Bluetooth Special Interest Group.** 2022. Disponível em: <https://www.bluetooth.com>. Acesso em: 22 jul. 2022.

WANG, Y. X. Y. Z. Y. L. C. L. Bluetooth positioning using rssi and triangulation methods. In: **2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)**. [S.l.: s.n.], 2013. p. 837–842.

YANG, Z.; LIU, Y. Quality of trilateration: Confidence based iterative localization. **IEEE Transactions on Parallel and Distributed Systems**, v. 21, n. 5, p. 631–640, 2010.

ZHANG, L. et al. A comprehensive study of bluetooth fingerprinting-based algorithms for localization. **2013 27th International Conference on Advanced Information Networking and Applications Workshops**, p. 300–305, 2013.

## APÊNDICE A - TDOA.PY

---

```
import bluetooth
import time

NOME_DISPOSITIVO = "Mi A3"
ENDERECO_DISPOSITIVO = None
NUM_MEASUREMENTS = 100

diferencas_tempo = []

def escaneia_dispositivos():
    dispositivos_proximos = bluetooth.discover_devices(duration
        =8, lookup_names=True, flush_cache=True)
    for addr, name in dispositivos_proximos:
        if name == NOME_DISPOSITIVO:
            return addr
    return None

if __name__ == '__main__':
    for _ in range(NUM_MEASUREMENTS):
        hora_envio = time.time()
        print(f"Sinal de procura enviado em: {hora_envio}")

        ENDERECO_DISPOSITIVO = escaneia_dispositivos()

        hora_recebe = time.time()
        print(f"Sinal de procura encontrado em: {hora_recebe}")

        if ENDERECO_DISPOSITIVO:
            print(f"Dispositivo '{NOME_DISPOSITIVO}' encontrado
                . Endereço: {ENDERECO_DISPOSITIVO}")
            TDoA = hora_recebe - hora_envio
            print(f"Diferença de tempo: {TDoA} segundos")
            diferencas_tempo.append(TDoA)
        else:
```

```
print(f"Dispositivo '{NOME_DISPOSITIVO}' não  
      encontrado.")  
  
time.sleep(1) # Aguarda 1 segundo antes de fazer a pró  
             xima medição  
  
# Calcular a média das diferenças de tempo  
if diferencas_tempo:  
    media = sum(diferencas_tempo) / len(diferencas_tempo)  
    print(f"Média das diferenças de tempo: {media}")  
else:  
    print("Nenhuma diferença de tempo encontrada.")
```

---

## APÊNDICE B - SCAN.PY

---

```
import asyncio
from bleak import BleakScanner

nome_disp = 'Xiaomi Smart Band 7 25AA'
num_medicoes = 100

async def scan_devices():
    rssi_encontrado = 0
    scanner = BleakScanner()
    devices = await scanner.discover()
    for device in devices:
        if device.name == nome_disp:
            print('Encontrado')
            rssi_encontrado = device.rssi
            print(rssi_encontrado)
        else:
            print('Não encontrado')
    return rssi_encontrado

async def main():
    rssi_valores = []
    for _ in range(num_medicoes):
        valor_rssi = await scan_devices()
        if valor_rssi != 0:
            rssi_valores.append(valor_rssi)

    if rssi_valores:
        media_rssi = sum(rssi_valores) / len(rssi_valores)
        print(f"Média dos valores de RSSI: {media_rssi}")
    else:
        print("Nenhum valor de RSSI encontrado.")

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

---

## APÊNDICE C - CALCULA\_DISTANCIA.PY

---

```
import math
rssi_conhecido = -64.24503311258277
rssi = -63.310344827586206
A = -56.85034013605442
d_conhecido = 2.5

def calc_dist(rssi_conhecido , rssi , A, d_conhecido):
    n = -(rssi_conhecido - A)/(10*math.log10(d_conhecido))
    d = math.pow(10,-(rssi - A)/(10*n))
    return d

print('A distância para o objeto é: \n')
print(calc_dist(rssi_conhecido , rssi , A, d_conhecido))
```

---

## APÊNDICE D - CALCULA\_COORDENADAS.PY

---

```
import math
#criando antenas com as variaveis
# Antena_A = (x,y,z,r)
def obtem_coordenadas(Antena_A, Antena_B, Antena_C):
    x_r = (pow(Antena_B[0],2)+pow(Antena_A[3],2)-pow(Antena_B
        [3],2)) / (2*Antena_B[0])
    y_r = (pow(Antena_C[1],2)+pow(Antena_C[0],2) - 2*Antena_C
        [0]*x_r +pow(Antena_A[3],2)-pow(Antena_C[3],2)) / (2*
        Antena_C[1])

    print('As seguintes coordenadas foram obtidas:\n')
    print('X: ' + x_r)
    print('Y: ' + y_r)

Antena_A = [0,0,0,0]
Antena_B = [0,0,0,0]
Antena_C = [0,0,0,0]

obtem_coordenadas(Antena_A, Antena_B, Antena_C)
```

---