

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA BIOMÉDICA**

PIETRO GOULART DAL PIZZOL

Sistrack

Sistema de Rastreamento de Equipamentos Médico-Hospitalares

São Leopoldo

2022

PIETRO GOULART DAL PIZZOL

Sistrack

Sistema de Rastreamento de Equipamentos Médico-Hospitalares

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia Biomédica, pelo Curso de Engenharia Biomédica da Universidade do Vale do Rio dos Sinos (UNISINOS).

Orientador Prof.^o Dr. Rodrigo Marques de Figueiredo

São Leopoldo

2022

RESUMO

O Sistrack – sistema de rastreamento de equipamentos médicos-hospitalares – é um sistema baseado na linguagem de programação Python, que faz uso de etiquetas de identificação por *Bluetooth Low Energy* (BLE) e leitores de identificação também por *Bluetooth Low Energy* (BLE), para geolocalização dentro de uma área pré-definida em um hospital. O trabalho tem como objetivo central desenvolver um sistema auxiliar que trabalha simultaneamente com o de gerenciamento de uma instituição, atuando no rastreamento dos equipamentos dentro do ambiente hospitalar, visando melhorar a eficiência do gerenciamento de equipamentos médico-hospitalares. Esta melhora de eficiência de gerenciamento tem como base que um equipamento quando está parado, gera custos negativos para a instituição, sendo assim, quanto menos tempo for necessário para resolver o problema, menor é o custo. Utilizando os BLEs, foram implementados dois métodos de geolocalização no projeto, um de proximidade e um de trilateração, os resultados obtidos em ambos os métodos foram satisfatórios, estando dentro do esperado. Para as comprovações desses métodos foram elaborados quadros com 30 ensaios para cada método. Os resultados obtidos foram de 97% de acerto para o método 1 e de 90% para o método 2, sendo o método 2 mais preciso, levando a uma taxa de erro maior. Realizando os ajustes finais, melhorando e refinando o necessário, o projeto poderá ser implementado em um ambiente hospitalar, podendo reduzir os custos que um equipamento parado gera, muitas vezes por não serem encontrados nas unidades, assim como a melhora da eficácia para a realização de manutenções preventivas, visto que quando são feitas corretamente, diminuem significativamente as manutenções corretivas, diminuindo assim os custos que podem ser convertidos em novos equipamentos e novas tecnologias.

Palavras-chave: Rastreador de Equipamentos; Trilateração de Sinais; *Bluetooth Low Energy*; *BLE*; Equipamentos Médicos-hospitalares; Python; Supervisório.

LISTA DE FIGURAS

Figura 1 - Exemplo de aplicação de RFID	17
Figura 2 - Modelo básico de funcionamento de um RFID	17
Figura 3 - Exemplo de BLE	19
Figura 4 - Exemplo de ESP32	22
Figura 5 - Interface IDE Arduino	23
Figura 6 - Protocolo MQTT	24
Figura 7 – Exemplo Visual Studio Code	25
Figura 8 - Exemplo Spyder	26
Figura 9 – Fluxograma inicial do projeto	30
Figura 10 - Fluxograma da primeira evolução da ideia do projeto	31
Figura 11 - Fluxograma da segunda evolução da ideia do projeto	32
Figura 12 - Diagrama detalhado das etapas do programa	33
Figura 13 - Exemplo de inscrição em tópico	35
Figura 14 - Interface MQTTX	36
Figura 15 - Exemplo de Trilateração	39
Figura 16 - Exemplo de arquivo txt criado	41
Figura 17 – Interface	43
Figura 18 - Exemplo de resultados método 2	46
Figura 19 – Etapa 1.1 IDE Arduino	54
Figura 20 - Etapa 1.2 IDE Arduino	54
Figura 21 - Etapa 1.3 IDE Arduino	54
Figura 22 - Etapa 1.4 IDE Arduino	55
Figura 23 - Etapa 1.5 IDE Arduino	55
Figura 24 - Etapa 1.6 IDE Arduino	55
Figura 25 - Etapa 2.1 IDE Arduino	56
Figura 26 - Etapa 2.2 IDE Arduino	56
Figura 27 - Etapa 2.3 IDE Arduino	56

LISTA DE FOTOGRAFIAS

Foto 1 - Etiquetas BLE	33
--	----

LISTA DE GRÁFICOS

<u>Gráfico 1 Potência x Distância</u>	37
<u>Gráfico 2 - Erros BLE1</u>	48
<u>Gráfico 3 - Erros BLE2</u>	48

LISTA DE QUADROS

<u>Quadro 1 - Comparação ESP32 com ES8266</u>	20
<u>Quadro 2 - Trabalhos correlatos</u>	26
<u>Quadro 3 - Resultados método um</u>	43
<u>Quadro 4 - Resultados método dois</u>	46
<u>Quadro 5 - Quantidade de erros</u>	46

LISTA DE SIGLAS

BLE	<i>Bluetooth Low Energy</i> – Bluetooth de Baixa Energia
dB	<i>decibel</i>
HTTP	<i>Hypertext Transfer Protocol</i> - Protocolo de Transferência de Hipertexto
IBM	<i>International Business Machines Corporation</i>
IDE	<i>Integrated Development Environment</i> - Ambiente de Desenvolvimento Integrado
IoT	<i>Internet of Things</i> - Internet das coisas
IP	<i>Internet Protocol</i> - Protocolo de Internet
ISO	Organização Internacional de Padronização
M2M	<i>Machine-to-Machine</i> – Máquina-para-máquina
MAC	<i>Media Access Control</i> - Controle de Acesso de Mídia
MQTT	<i>Message Queuing Telemetry Transport</i> - Transporte de Filas de Mensagem de Telemetria
NFC	<i>Near Field Communication</i> - Comunicação de Campo Próximo
OMS	Organização Mundial da Saúde
OS	Ordem de Serviço
RFID	<i>Radio Frequency Identification</i> - Identificação por Radiofrequência
RSSI	<i>Received Signal Strength Indicator</i> - Indicador de Intensidade do Sinal Recebido
TCP	<i>Transmission Control Protocol</i> - Protocolo de Controle de Transmissão
TXT	Arquivo de texto

SUMÁRIO

<u>1 INTRODUÇÃO</u>	100
<u>2 FUNDAMENTAÇÃO TEÓRICA</u>	13
<u>2.1 O que é e qual a função da engenharia clínica/biomédica</u>	13
2.1.1 <u>Manutenção preventiva e sua periodicidade</u>	14
2.1.2 <u>Comodatos</u>	14
<u>2.2 Sistemas de Gestão de Manutenção Hospitalar</u>	15
<u>2.3 Chip de localização</u>	16
2.3.1 <u>RFID</u>	16
2.3.2 <u>NFC</u>	18
2.3.3 <u>Beacon Bluetooth</u>	18
<u>2.4 Linguagens de programação</u>	19
2.4.1 <u>C++</u>	19
2.4.2 <u>Java</u>	20
2.4.3 <u>Python</u>	20
<u>2.5 ESP32</u>	21
<u>2.6 IDE Arduino</u>	22
<u>2.7 Protocolo MQTT</u>	23
<u>2.8 Editores de código</u>	25
2.8.1 <u>Visual Studio Code (VSCode)</u>	25
2.8.2 <u>Spyder</u>	26
<u>2.9 Trabalhos correlatos</u>	26
<u>3 METODOLOGIA</u>	30
<u>3.1 Etiquetas BLE</u>	34
<u>3.2 ESP32</u>	34
3.2.1 <u>Configuração ESP32</u>	35
3.2.2 <u>Programa ESP32</u>	35
<u>3.3 Server MQTT</u>	36
<u>3.4 Programa em Python – Parte 1</u>	37
3.4.1 <u>Conexão MQTT</u>	37
3.4.2 <u>Cálculo</u>	37
3.4.3 <u>Gravação em TXT</u>	40

<u>3.5 Programa em Python - Parte 2</u>	41
<u>3.5.1 Leitor TXT</u>	41
<u>3.5.2 Interface</u>	42
<u>4 ANÁLISE DOS RESULTADOS</u>	44
<u>4.1 Método 1</u>	44
<u>4.2 Método 2</u>	45
<u>5 CONSIDERAÇÕES FINAIS</u>	50
<u>REFERÊNCIAS</u>	51
<u>ANEXO A - CONFIGURAÇÃO IDE ARDUINO PARA ESP32</u>	54

1 INTRODUÇÃO

Como se sabe, um dos grandes problemas enfrentados dentro das unidades de saúde é a localização de equipamentos, onde em vários casos, simplesmente desaparecem, podendo apenas ter sido guardados em locais diferentes dos comuns, ou levados para outro setor ou unidade, fugindo do controle dos gestores. Tornando-se a rastreabilidade uma ferramenta de grande impacto nas gestões, pois permite saber a localização, o histórico de uso, o estado em que está o equipamento e o estado das manutenções preventivas, evitando assim, esquecimentos, erros e, por sua vez, trazendo diversos benefícios para o gerenciamento.

Além desses benefícios todos, esse sistema também pode ser utilizado para evitar que equipamentos sejam extraviados, conforme a reportagem da GauchaZH, onde:

A Polícia Civil, por meio da Delegacia de Polícia de Repressão às Ações Criminosas Organizadas (Draco), prendeu nesta quinta-feira (12) uma técnica de enfermagem de 24 anos que teria furtado e revendido 34 equipamentos hospitalares destinados a pacientes com covid-19. A investigada revenderia os itens, avaliados em mais de R\$ 100 mil, em um site de comercialização de produtos na internet (GauchaZH, 2022).

Percebe-se que com a falta da confiabilidade da informação sobre a localização dos equipamentos, notícias como está, da Gaúcha Zero Hora, se tornam cada vez mais frequentes, aumentando cada vez mais o comércio ilegal destes equipamentos.

Equipamentos de baixa, média e alta complexidade são danificados pelos usuários e pelos pacientes, seja pelo manuseio incorreto ou pelo excesso de uso do mesmo e muitas vezes perdidos dentro de um vasto ambiente hospitalar. Não tendo registros, o conserto ou sua recuperação se torna ainda mais difícil.

Com a melhora da rastreabilidade dos equipamentos, é possível diminuir o tempo que ele fica parado até a identificação do problema, ou a identificação do setor em que o equipamento foi utilizado por último, reduzindo custos de manutenção e, também facilitando na identificação dos equipamentos que necessitam de manutenção preventiva e sua respectiva localização.

Além de que com o controle mais rígido dos equipamentos, é possível ter a documentação em ordem, melhorando a confiabilidade do processo e de suas etapas, ajudando também nas futuras validações.

O presente trabalho propõe um sistema onde é utilizando chip de localização nos ativos das instituições, e com antenas dispostas em locais estratégicos será possível identificar quais equipamentos estão em determinados setores, facilitando o acesso aos mesmos.

Este sistema irá utilizar de um chip BLE, inicialmente implantados nos equipamentos médico-hospitalares, pertencentes a um hospital ou clínica, como respiradores pulmonares, bombas de infusão, entre outros. Haverá pontos estrategicamente dispostos na instituição, onde serão instalados leitores, microcontrolador ESP32 (CURVELLO, 2018), para que os equipamentos sejam controlados à distância, para que haja também, um monitoramento mais eficaz de suas manutenções preventivas e para facilitar o rastreamento desses equipamentos na hora de realizar uma manutenção corretiva ou quando são perdidos pela unidade.

2 FUNDAMENTAÇÃO TEÓRICA

Para a realização e validação das propostas, foram realizadas pesquisas para ter como base teórica os motivos das escolhas e caminhos tomados durante o trabalho. Essas pesquisas são referentes aos tipos de sistemas presentes hoje no mercado, os tipos de tecnologias e sensores, métodos de comunicação, linguagens para programação e diversos outros assuntos apresentados a seguir.

2.1 O que é e qual a função da engenharia clínica/biomédica

O objetivo da Engenharia Clínica é disponibilizar equipamentos funcionais e seguros e contribuir para a melhoria dos cuidados e atendimento aos pacientes. Ela se formou através da junção das engenharias Biomédica e Elétrica, focando na gestão de tecnologias de saúde, visando melhorar o tratamento oferecido aos pacientes, viabilizando procedimentos e agilizando a disponibilidade de equipamentos médicos (MICROMED, 2019).

Apesar de ser uma função onde muitas vezes os profissionais não são reconhecidos pelos pacientes e passam despercebidos, eles são fundamentais em todas as etapas dos processos que envolvem o cuidado com o paciente. Quando se fala em equipamentos, o interesse destes profissionais envolve todo o seu ciclo de vida, desde a pesquisa, desenvolvimento, produção, comercialização, instalação, operação (uso no paciente), manutenção até a obsolescência, quando um novo avanço é disponibilizado por fabricantes (MICROMED, 2019).

Para Bajur e Lunardi (2016), muitas vezes, a engenharia clínica é vista como um gasto supérfluo aos olhos dos gestores e diretores, mas é essencial na saúde, pois: permite otimizar os recursos financeiros, facilitar processos, reduzir chances de erros, entre outros. Rastrear dados é muito importante para a Engenharia Clínica. Essas informações são cruciais na hora de, por exemplo, avaliar se é preciso desativar um determinado equipamento e descobrir as razões por trás disso: se é por um aumento de custo na manutenção, pela redução de confiabilidade ou pela disponibilidade no mercado de tecnologias mais efetivas.

Segundo a OMS (Organização Mundial da Saúde), 20% a 40% dos gastos em instituições de saúde são decorrentes de desperdício. O que na grande maioria das

vezes impede que medidas preventivas e corretivas sejam tomadas é a falta de dados sobre os processos faz com que desperdícios não sejam detectados (BAJUR/LUNARD, 2016).

2.1.1 Manutenção preventiva e sua periodicidade

Segundo Mello (2021), embora a manutenção dos equipamentos médico-hospitalares seja extremamente importante, essa ainda não é a realidade da maioria dos hospitais brasileiros. Esse é um risco enorme que pode tanto trazer prejuízos aos profissionais de saúde, quanto aos pacientes e à saúde financeira do hospital. Alguns benefícios de realizar a manutenção preventiva são:

- Redução de custos, visto que muitos equipamentos médicos são caros e usam tecnologia de ponta, ou seja, ao realizar manutenções periódicas, diminui os gastos com quebras, trocas de peças, paralisações de serviços além de custos indiretos ocasionados por defeitos das máquinas;
- Aumento da segurança, pois equipamentos que passam por inspeções periódicas têm maiores chances de estarem calibrados e de funcionarem de acordo com os padrões de qualidade exigidos, significando um aumento de eficácia dos dados e mais segurança deles;
- Aumento da vida útil, já que é uma informação de extrema importância para o gestor que planeja a compra de equipamentos novos. Quando não é feita a manutenção preventiva essa vida útil tende a diminuir e a ficar inserta.
- A periodicidade das manutenções preventivas em um ambiente hospitalar pode variar, dependendo da complexidade do equipamento, dos dados/laudos coletados e outras variáveis. Geralmente as manutenções são anuais (de 12 em 12 meses), semestrais (6 em 6 meses) ou trimestrais (3 em 3 meses) (MINISTÉRIO DA SAÚDE, 2002).

2.1.2 Comodatos

A palavra comodato refere-se a um regime de aluguel ou empréstimo que não envolve recursos financeiros. Como prevê nos artigos 579 a 585 do Código Civil Brasileiro, o comodato é um empréstimo gratuito de coisas não fungíveis. Ou seja, significa que qualquer instituição só cede em comodato bens que não podem ser substituídos, seja em qualidade, espécie ou quantidade.

O regime de comodato é comum em diversas áreas como a agrícola, a industrial e a médica (hospitais, clínicas e consultórios) que desejam ofertar serviços, mas não dispõem do orçamento necessário para comprar aparelhos. Como empregam alta tecnologia, esses equipamentos são caros, o que impacta nas finanças de forma negativa, comprometendo o capital de giro e até inviabilizando o negócio. Graças ao comodato, o investimento inicial é baixo, tornando possível a oferta de novos exames e promovendo lucros interessantes para hospitais e clínicas (MORSCH, 2020).

2.2 Sistemas de Gestão de Manutenção Hospitalar

Na atualidade, há diversos sistemas de gestão hospitalares. Não são todos que possuem a capacidade de gestão de manutenção, mas ainda assim, há um número grande de opções no mercado, entre os mais famosos tem-se o exemplo do Philips TASY.

Segundo Soares (2019), o sistema de gestão da Philips é o mais completo do mercado, auxiliando desde a entrada em um hospital, funções de suporte a procedimentos, assim como suporte a manutenções. Contribuindo para que tanto prestadores de serviços, como planos de saúde tenham uma maior integração e simplificação do processo.

Neste sistema, quando um ativo apresenta defeito, a enfermeira, por exemplo, realiza a abertura de uma ordem de serviço (OS), com o número de patrimônio e/ou número de série, geralmente ambos já cadastrados no sistema. Esta OS, deverá conter descrito o problema apresentado e o setor em que se encontra, porém o equipamento mesmo estragado, nem sempre é encaminhado para a engenharia clínica ou fica aguardando no setor em que apresentou defeito, o que dificulta a sua localização e o atendimento. Após ser localizado o equipamento e realizado o atendimento, o prestador de serviços, indica a quantidade de horas trabalhadas e descreve o que foi realizado. Este é um procedimento padrão do sistema Philips Tasy e que segue como padrão para muitos outros.

2.3 Chip de localização

Atualmente, há diversas opções de chips de localização, geolocalização e de rastreamento, todos com funções semelhantes, algumas mais específicas que outras. Quando se fala de um ambiente hospitalar, logo se opta por uma tecnologia mais de localização mais precisa, eficaz e barata, porém, que abrange uma área menor, as três tecnologias escolhidas para fins de comparação foram a RFID, NFC e *Beacon Bluetooth*.

2.3.1 RFID

A tecnologia de RFID (identificação por radiofrequência) é um termo genérico para as tecnologias que utilizam a frequência de rádio para captura de dados. Existem vários métodos, mas o mais comum é armazenar em um microchip um número de série que identifique uma pessoa ou um objeto, ou outra informação (QUEBECK, 2021).

Essa tecnologia permite a captura automática de dados, para a identificação de objetos com etiquetas eletrônicas, *tags*, *RF tags* ou *transponders*. A sua principal função hoje é ajudar a reduzir desperdício, limitar roubos, gerir inventários, simplificar a logística e aumentar a produtividade (UFRJ, 2010).

Como mostrado na Figura 1 o RFID emite sinais de radiofrequências para leitores que captam estas informações e são decodificadas por um software, geralmente conectado a um banco de dados. Esse sistema é basicamente composto por uma etiqueta de RFID (*tag*) e leitor de RFID (*reader*).

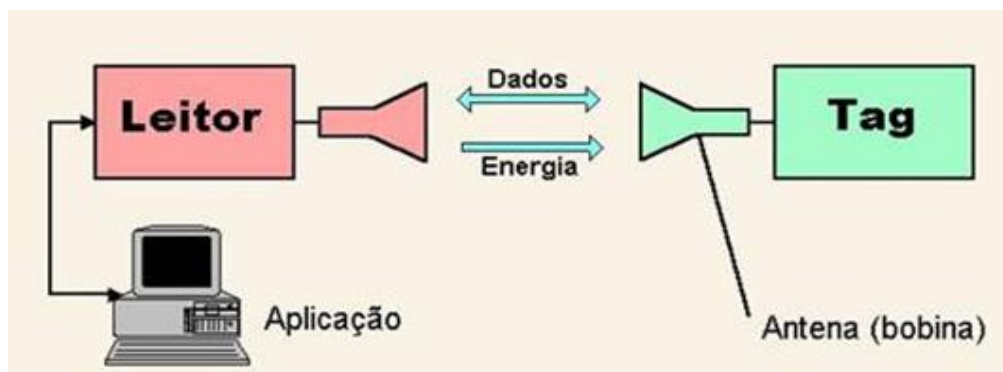
Figura 1 - Exemplo de aplicação de RFID



Fonte: https://www.gta.ufrj.br/grad/07_1/rfid/RFID_arquivos/o%20que%20e.htm

A RFID *tag* é um transponder, um pequeno objeto que pode ser colocado em qualquer lugar, pessoa ou animal, contém um ou mais chips de silício e antenas que lhe permitem responder aos sinais de rádio enviados por uma base transmissora. São divididas em ativas e passivas, sendo que as ativas enviam um sinal próprio de identificação e possuem um alcance de dezenas ou centenas de metros, e geralmente requerem uma bateria que gera energia. Já as etiquetas passivas apenas respondem ao sinal enviado pela base transmissora, e geralmente não requerem bateria ou fonte própria de energia, porém seu alcance é limitado e menor que as ativas. O RFID *reader* é o equipamento que coleta as informações armazenadas nas etiquetas RFID e interpreta o sinal (UFRJ, 2010). Esse sistema é exemplificado na Figura 2.

Figura 2 - Modelo básico de funcionamento de um RFID



Fonte: https://www.gta.ufrj.br/grad/07_1/rfid/RFID_arquivos/como%20funciona.htm

2.3.2 NFC

Segundo Alves (2019), NFC ou *Near Field Communication* (Comunicação de Campo Próximo), é considerado uma versão mais sofisticada do RFID, que opera em frequência constante de 13,56 MHz. O NFC é considerado mais seguro por só operar em curtas distâncias evitando intrusos na conexão, além de dispensar configurações de pareamento. Permite que um dos dispositivos não precise de energia e ainda energize o outro ao estabelecer uma conexão.

Dispositivos NFC podem ser de dois tipos: ativos ou passivos. Os ativos têm fonte de energia própria e podem enviar ou receber informações, já os passivos precisam de um chip ativo próximo para obter energia e só podem transmitir dados. (ALVES, 2019)

2.3.3 Beacon Bluetooth

Como assinala Carneiro (2016), *Beacons* são dispositivos de geolocalização, criados pela Apple Inc., para ambientes fechados que utilizam uma tecnologia chamada *Bluetooth Low Energy* (BLE). Sendo assim, o *Beacon* permite que objetos sejam localizados com alta precisão em uma distância pequena. Hoje em dia, há uma imensa variedade de tipos de BLEs e de marcas. Tem a vantagem de consumir menos energia do dispositivo que o Bluetooth comum.

O *Beacon* funciona como emissor de sinais, o receptor, por sua vez, pode apenas guardar a informação de que a conexão ocorreu, ou pode iniciar alguma ação no dispositivo do usuário. Podem ser ações simples, como enviar notificações, ou mais complexas, como ativar um dispositivo remotamente. Sua funcionalidade é transmitir a localização por meio de um sinal de potência (dB) para um aplicativo que verifica se há alguma ação já determinada para a localização, caso haja, será executada a ação pré-determinada (DANTAS, 2021).

Figura 3 - Exemplo de BLE



Fonte: <https://www.tecmundo.com.br/seguranca/216658-smart-tag-entenda-etiqueta-inteligente.htm>

Como mostra a figura 3, há uma diversidade grande de tipos e marcas de BLE, podendo ser usados facilmente no dia a dia. O valor de aquisição também é um atrativo, quando levado em conta que seu valor unitário é o melhor custo-benefício entre os métodos aqui apresentados, além de ser muito versátil quando utilizado com os ESP32.

2.4 Linguagens de programação

Para a escolha da linguagem de programação a ser usada no projeto, foram analisadas três: C++, Java e Python. Abaixo segue uma breve explicação sobre suas utilidades e o motivo da escolha de uma delas.

2.4.1 C++

É uma linguagem de programação multiplataforma, multiparadigma e de médio nível, desenvolvida em meados da década de 80, com objetivo de expandir e atualizar a linguagem C. Como diz Dias (2021), é uma das linguagens mais populares do mundo, se destacando como líder nos cenários de jogos, mercado financeiro, grandes aplicações, sistemas operacionais e microcontroladores, sendo muito utilizada também na área acadêmica para compor grade de disciplinas de curso superior.

Algumas características do C++:

- Possibilidade em programação de alto e baixo nível;
- É padronizado pela ISO;
- Compatibilidade com a linguagem C;
- Possui paradigmas de programação funcional, genérica, orientada a objetos e imperativa;
- Pode ser utilizada sem a necessidade de um ambiente de desenvolvimento sofisticado.

2.4.2 Java

Java é uma linguagem de programação e plataforma computacional lançada pela Sun Microsystems em 1995. Consiste em uma linguagem de programação orientada a objetos, baseada em classes. Permite a criação de programas que são executados no Mac, Linux e Windows, além de ser rápida, segura e confiável. Essas propriedades fazem com que o Java seja usado em muitos aplicativos, sites, de laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares e muitas outras aplicações. Possui uma sintaxe similar a C e C++, porém em um modelo mais simples (JAVA, 2019).

2.4.3 Python

Para Pereira (2021), o Python é uma das melhores alternativas para se começar a programar, visto que é simples e fácil de aprender. Ele se parece muito com a língua inglesa. É uma fonte livre de alto nível e tem uma grande comunidade que compartilha dados. As propriedades portáteis e extensíveis do Python permitem que sejam executadas operações de idioma cruzado sem problemas. É suportado pela maioria das plataformas presentes no mercado hoje, desde o Windows, Linux, Macintosh, Solaris, Playstation, entre outros. Os recursos de extensibilidade do Python permitem integrar componentes Java e NET. Pode invocar bibliotecas C e C++. É amplamente utilizado em projetos pequenos e grandes, online ou offline. Ele é usado para construir aplicativos GUI e de desktop. Ele usa a biblioteca "Tkinter" para fornecer uma maneira rápida e fácil de criar aplicativos. O Python também lida com

dados tabulares, matriciais e estatísticos e até mesmo visualizá-los com bibliotecas populares como “Matplotlib” e “Seaborn” (PEREIRA, 2021).

2.5 ESP32

Segundo Curvello (2018), o ESP8266, foi um divisor de águas no mundo dos embarcados e na Internet das Coisas (IOT), que se refere a conexão de objetos do dia a dia na rede mundial de computadores. Seu sucessor o ESP32, não fica para trás, é uma placa de desenvolvimento mais potente, como mostra no quadro 1, com processador dual core, com até 240MHZ de velocidade, contando ainda com conexão WiFi e Bluetooth BLE.

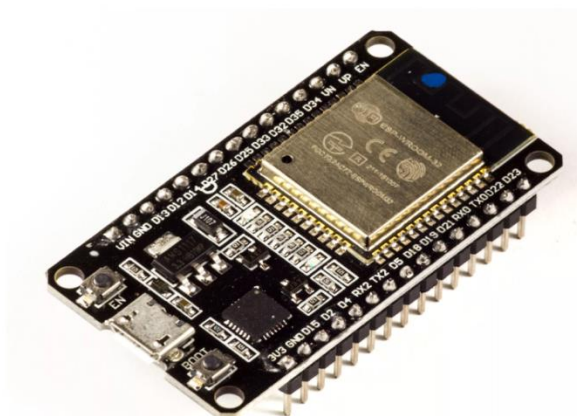
Quadro 1 - Comparação ESP32 com ES8266

Especificações	ESP8266	ESP32
Processador	Xtensa Single-Core 32-bit L106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Sim, HT20	Sim, HT40
Bluetooth	-	Bluetooth 4.2 e abaixo
Frequência Típica	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash até 16MBytes	SPI Flash até 16MBytes
GPIO	17	36
Hardware / Software PWM	Nenhum / 8 Canais	1 / 16 Canais
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	-	1
Interface MAC	-	1
Sensor de toque	-	Sim
Sensor de temperatura	-	Sim
Temperatura de trabalho	-40°C ~ 125°C	-40°C ~ 125°C

Fonte: <https://www.cnx-software.com/2016/03/25/esp8266-and-esp32-differences-in-one-single-table/>

. É uma placa relativamente pequena e de fácil manuseio, como pode ser observado na figura 4.

Figura 4 - Exemplo de ESP32



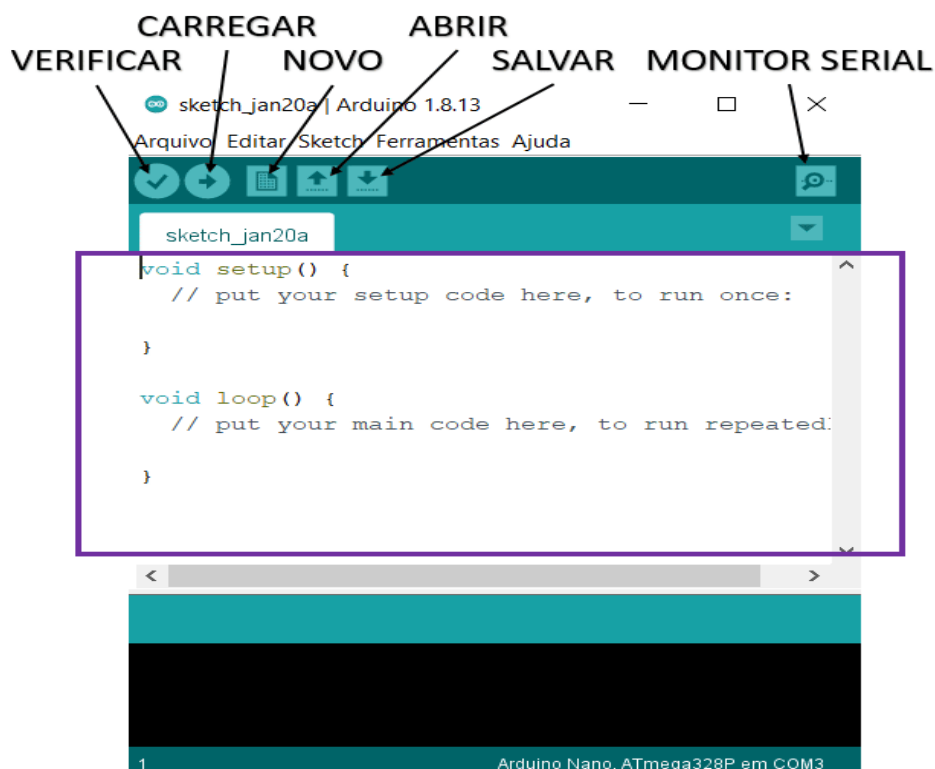
Fonte: <https://lobodarobotica.com/blog/o-que-e-esp32-para-que-serve-quando-usar/>

É compatível com a IDE do Arduino, compartilhando diversas bibliotecas, acaba tornando sua programação mais simples. O fato de possuir wifi e BLE, torna esta placa uma grande aliada quando se trabalha com IOT. Apesar de alguns modelos possuírem até 32 pinos físicos, uma porta micro USB e dois botões, no projeto apresentado será usado a porta micro USB para a gravação do código, o wifi para se conectar à rede, onde serão transferidos os dados e o BLE para realizar a aquisição dos dados das etiquetas (CURVELLO, 2018).

2.6 IDE Arduino

IDE vem do inglês que significa Ambiente de Desenvolvimento Integrado, segundo Quintino (2021), é o ambiente desenvolvido para que o programador tenha tudo (ou quase tudo), que precisar para programar a sua placa. Tendo um layout simples e de fácil manuseio, como mostrado na figura 5.

Figura 5 - Interface IDE Arduino



Fonte: <https://www.filipeflop.com/blog/o-que-e-ide-arduino/>

Esta IDE possui compatibilidade com diversas placas de diversos fabricantes, entre eles a Espressif, fabricante oficial do ESP32. Ou seja, essa IDE possui suporte para praticamente todos os modelos dessa placa, necessitando apenas baixar as bibliotecas específicas para o tipo de placa e tipo de programa que será desenvolvido.

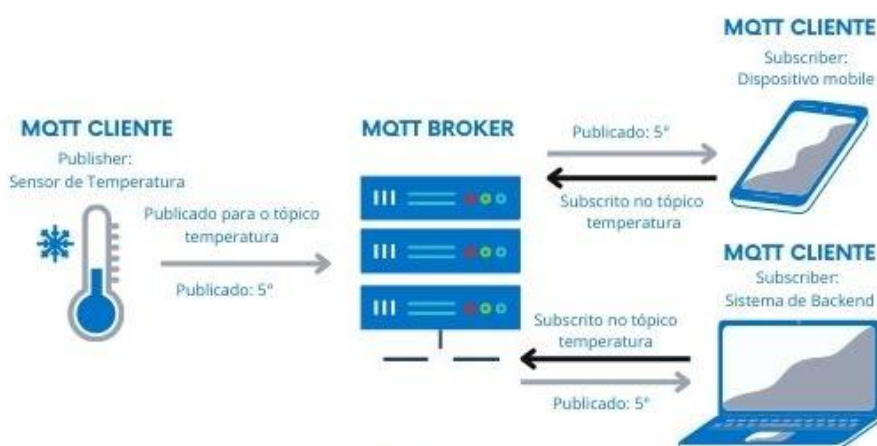
2.7 Protocolo MQTT

Como relata Santos (2022), o Transporte de Filas de Mensagem de Telemetria (MQTT), foi desenvolvido em meados de 1990 pela IBM e Eurotech, com o propósito de substituir o HTTP, oferecer baixo consumo de rede, banda e software, além de uma conexão mais segura. Este protocolo é de extrema utilidade em M2M (*Machine To Machine*) e em IOT (*Internet Of Things*), utiliza formato de Cliente/Servidor é comumente implementado sobre a pilha TCP/IP, embora não existam restrições quanto a esse quesito na especificação do MQTT. Este protocolo funciona de acordo com algumas regras e palavras-chave, como:

- **Tópico:** é o endereço no qual os dados serão enviados pelo publicador;
- **Client (Cliente):** tem como função interagir com o broker e se conectar a um tópico onde irá realizar a publicação;
- **Broker:** é o intermediário entre o publicador e o subscrito, podendo ser um interno no computador ou na nuvem, em outras palavras, um servidor;
- **Subscriber (Subscrito):** é a pessoa ou equipamento que estará inscrito em algum tópico e terá como função ler, ou seja, receber os dados do publicador;
- **Publisher (Publicador):** é a pessoa ou equipamento que estará conectado a um tópico e terá como função publicar, ou seja, enviar os dados para o subscrito;
- **Payload:** é o conteúdo da mensagem que será enviada.

Para entender o funcionamento deste protocolo, é necessário compreender como funciona cada um dos itens citados acima. Para isto será utilizado um exemplo prático, como demonstra a professora Fernandes (2021), onde em uma linha produção de caixas de leite, é necessário armazenar o produto em um local que haja o monitoramento da temperatura. Para que essa temperatura seja monitorada, há duas alternativas, a primeira necessita de um profissional específico para controlar e verificar periodicamente essa temperatura, ou usar um sistema automatizado como o da figura 6.

Figura 6 - Protocolo MQTT



Fonte <https://materiais.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/>

Neste sistema, as informações de temperatura são coletadas pelo sensor e publicadas no broker, através do tópico "temperatura", onde fica armazenada. Após

isto, é possível cadastrar a quantidade desejada de dispositivos neste tópico, onde cada um deles irá receber as informações que nele foram publicadas, facilitando o acesso para análise e a tomada de medidas, caso seja necessário.

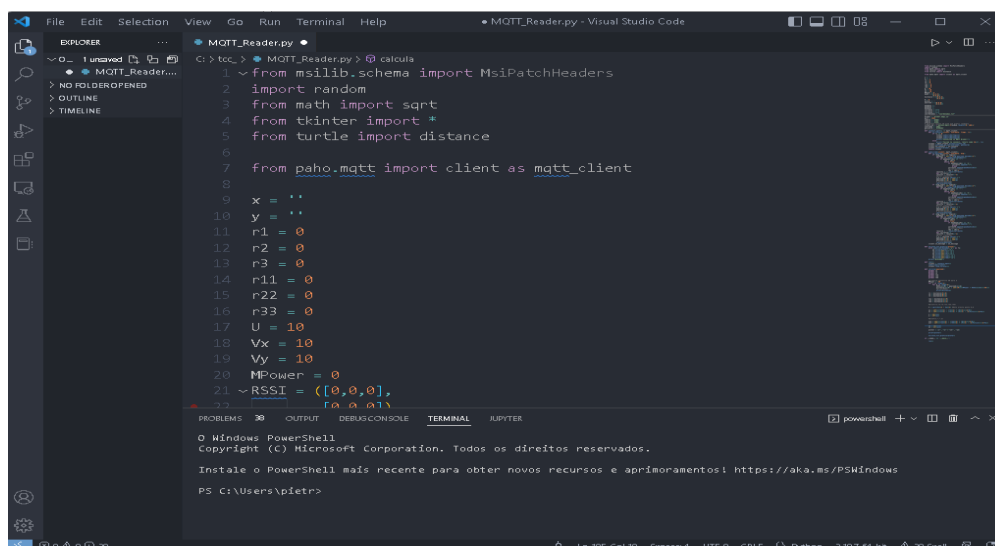
2.8 Editores de código

Para o desenvolvimento do programa para a leitura dos dados do broker e seu processamento, é necessário um editor de código. Entre as milhares de opções nesta área, foram escolhidos o Visual Studio Code (VSCode) e o Spyder, por quesitos de simplicidade de manuseio, ser completo e gratuito.

2.8.1 Visual Studio Code (VSCode)

O Visual Studio Code, produzido pela Microsoft, é uma IDE que possui diversas funcionalidades, como a edição de códigos com suporte para várias linguagens de programação. Possui também terminal de comandos e Poweshell.

Figura 7 – Exemplo Visual Studio Code



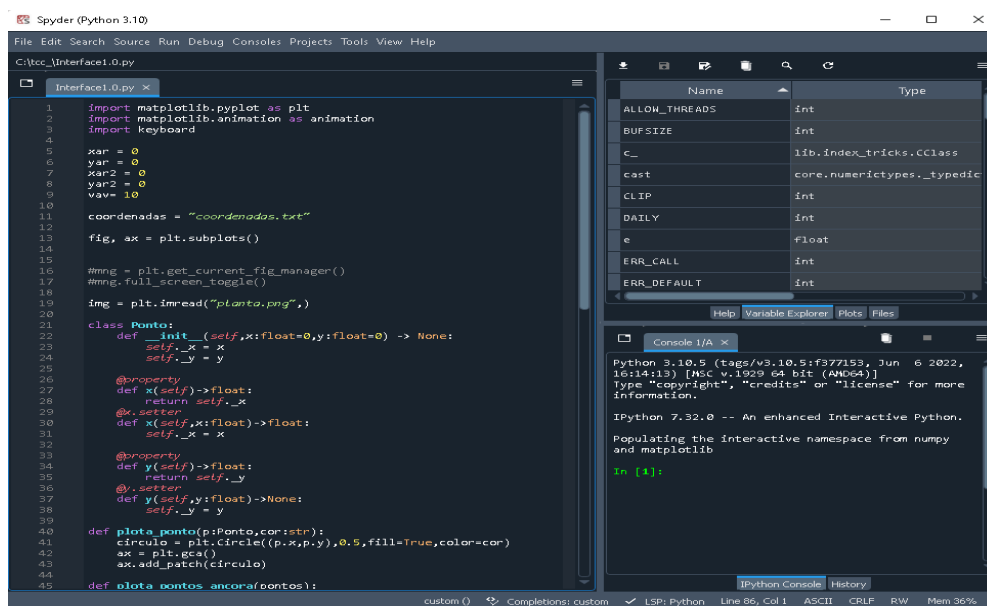
Fonte: o autor

Os pontos positivos dessa IDE, como mostra na figura 7, é o fato de ser customizável, onde fica mais agradável ao uso, loja de extensões extremamente ampla e gratuita, além disso tudo, é mais leve que a versão padrão do Visual Studio. (HANASHIRO, 2021)

2.8.2 Spyder

Pydee, ou como é chamado atualmente Spyder, é uma IDE focada em Python. Foi a interface mais usada para teste ao longo do projeto, principalmente por mais leve.

Figura 8 - Exemplo Spyder



Fonte: o autor

Em comparação com o VSCode, não tem tantas extensões quanto seu concorrente, porém vem com bibliotecas previamente instaladas, como a Numpy e a Matplotlib. Além, é claro, de ser mais leve e ter uma interface mais intuitiva para iniciantes, como pode-se notar na figura 8. (MANTOVANI, 2014)

2.9 Trabalhos correlatos

Foi realizada uma análise de mercado, no intuito de aperfeiçoar a ideia de acordo com a lacuna que há na área. Com os dados coletados foi montado o quadro 2, onde consta a empresa responsável, nome do produto, ano que foi desenvolvido e uma breve relação com o projeto até então proposto.

Quadro 2 - Trabalhos correlatos

AUTOR(ES)	ANO	TÍTULO	RELAÇÃO COM O TCC
Portal Saúde Business	2005	Hospitalar 2005: GS1 Brasil lança sistema de rastreamento	1*
Lucas Almeida – Cofundador e CRO da NEXXTO	2020	RTLS – Rastreamento em tempo real de insumo de saúde	2*
SmartX Technology Inc	2021	Soluções de RTLS e inteligência de negócios para setor de saúde	3*
Empresa Brasileira de Serviços Hospitalares - gov.br	2021	Sistema de rastreamento por chip controla circulação de roupas em unidade da Rede Ebserh	4*
NOVIDÁ	2022	Rastreamento de equipamentos em um hospital	5*
TR Service	2022	Etiquetas e insumos hospitalares com rastreamento	6*

Fonte: o autor

1. É um sistema utilizado para rastreamento de pacientes, profissionais e insumos, através de um código de barras. Porém, diferentemente da ideia do

projeto até então apresentado, não localiza em tempo real os equipamentos, apenas mantém um banco de dados dos “produtos” bipados pelo código de barras.

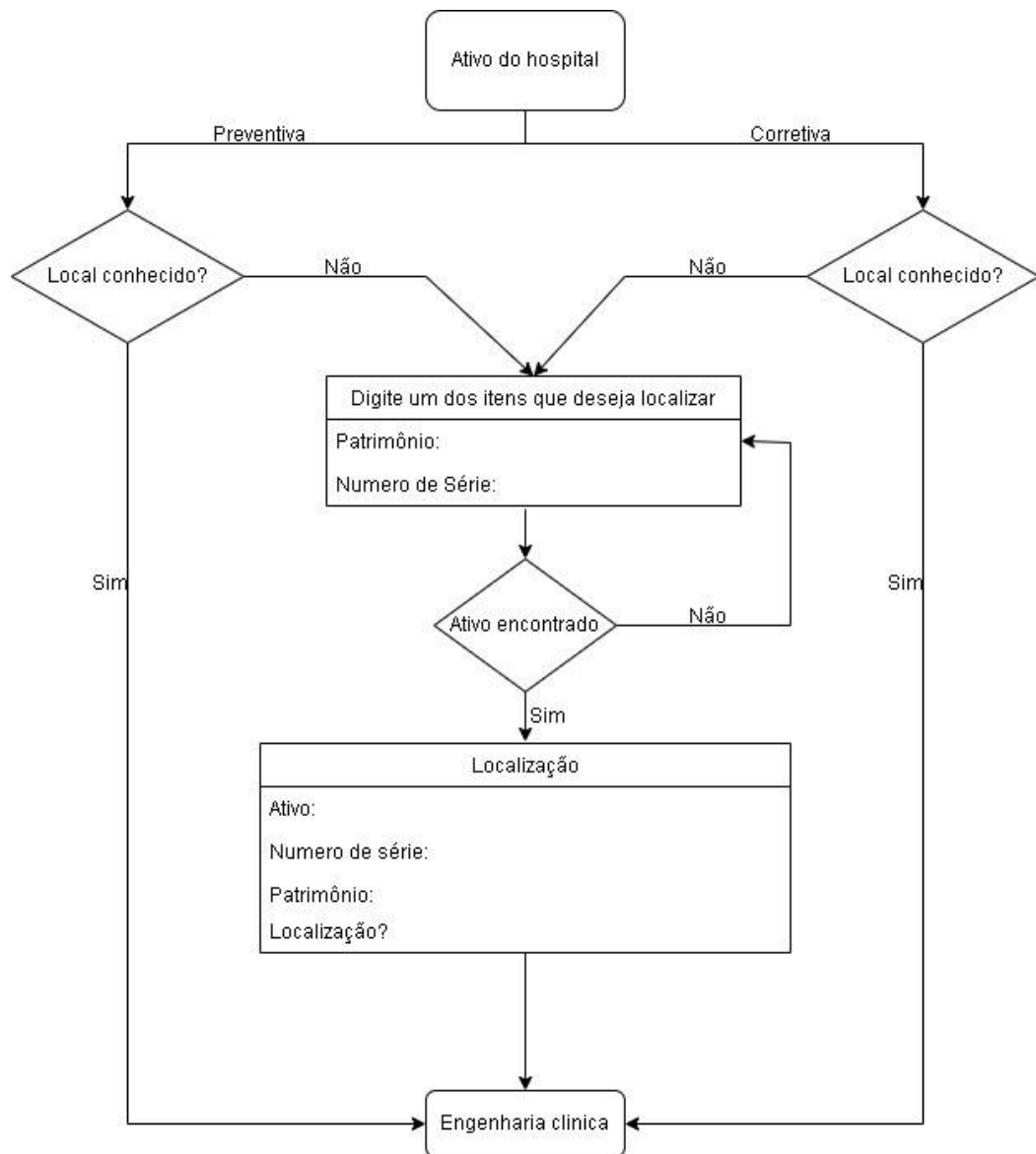
2. É uma empresa especializada no desenvolvimento de sistemas de monitoramento para ambientes hospitalares. Usando tecnologia de ponta, possui dois produtos no mercado, um para controle e monitoramento de temperatura e umidade dos insumos hospitalares, outro para controle e gestão da energia utilizada na instituição. Possui algumas ideias de sistema de gestão para esse monitoramento, assim como ideias de acoplamento para outros sistemas de gestão, que são parecidas com o projeto até então proposto.
3. SmartX HUB é o produto mais parecido com o projeto até então proposto, já que esta empresa vende o módulo que possui a mesma ideia de usar transmissão de sinais para localizar um ativo. Entretanto, além de utilizar etiquetas e leitores RFID, para utilizar este módulo, é necessário adquirir o sistema de gerenciamento da própria, o que em muitas vezes se torna inviável para muitas instituições.
4. É um sistema do governo federal, aplicado em Santa Catarina, onde é utilizada a tecnologia RFID para rastrear as roupas de ambientes hospitalares para ter uma administração melhor de fluxo. Apesar de usar RFID, é basicamente a mesma ideia até então proposta, porém aplicada em roupas, ao invés de equipamentos médico-hospitalares.
5. Um sistema parecido com o proposto, utiliza o BLE (Bluetooth de baixo consumo). Sua precisão só é boa ao chegar próximo. Visto que o alcance é pequeno, a utilização do método de proximidade não um método mais eficaz.
6. Esta empresa usa a mesma ideia do projeto, porém em insumos hospitalares, para ter um controle de estoque, desde o mais simples papel toalha, até uma bolsa de sangue.

Como demonstra o quadro 2, há diversos produtos com tecnologia e ideias iguais ou similares no mercado. O produto mais próximo tanto em ideia como em tecnologia, é o da empresa NOVIDA, onde utiliza BLEs para a geolocalização de equipamentos em clínicas médicas, utiliza um dos métodos que serão apresentados, o de aproximação, entretanto este não é o método mais eficaz para grandes instituições, este método tem uma acurácia menor que o de trilateração, que é utilizado no trabalho simultaneamente com o de proximidade, conseguindo melhores resultados.

3 METODOLOGIA

Inicialmente a ideia proposta consistia em desenvolver todo um sistema de gerenciamento e monitoramento dos equipamentos de um hospital ou clínica médica, como mostra na figura 9.

Figura 9 – Fluxograma inicial do projeto

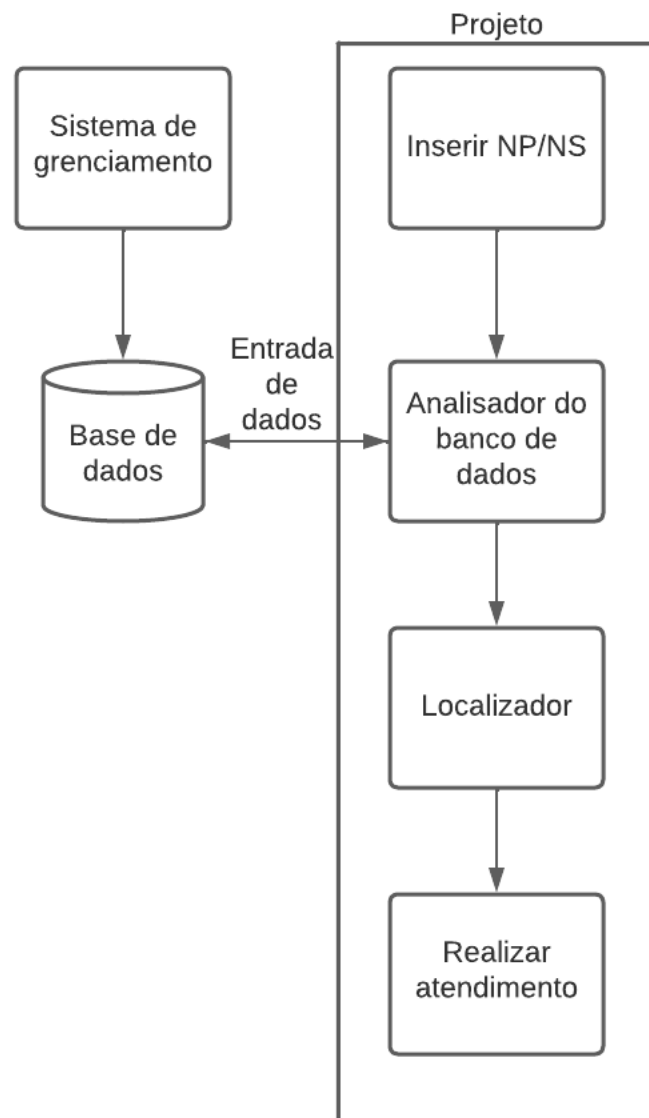


Fonte: o autor

Ao ser amadurecida a ideia, foi encontrada a barreira de que muitos hospitais e clínicas já possuem algum sistema de gerenciamento, próprio ou terceirizado, porém

ambos sem ter a opção de localização em tempo real dos equipamentos. Com isso, a ideia evoluiu para a Figura 10, onde o projeto será um módulo a parte, usado em conjunto com o sistema de gerenciamento que a instituição já possui, para seu fim principal, que seria a localização dos ativos.

Figura 10 - Fluxograma da primeira evolução da ideia do projeto



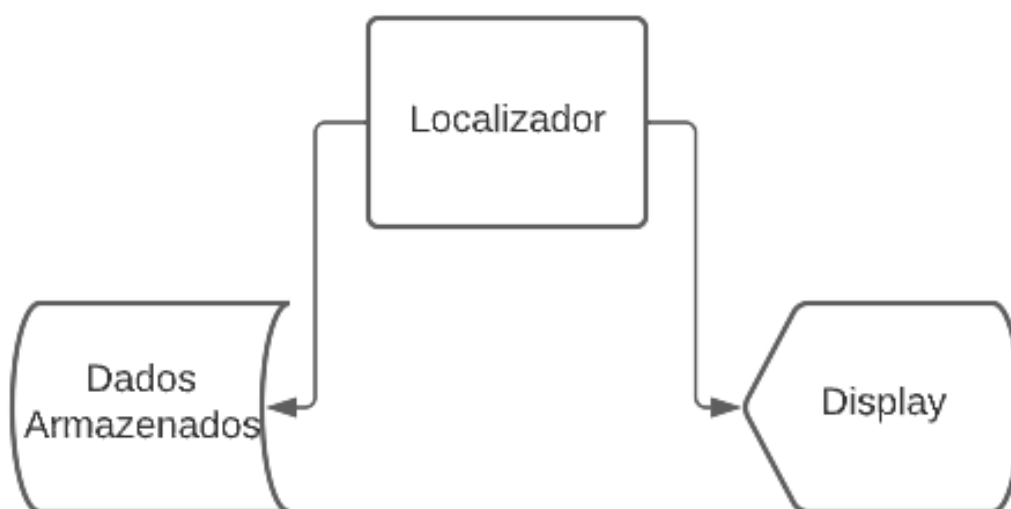
Fonte: o autor

Na Figura 10, pode-se ver que está dividido em duas partes, a esquerda é o sistema de gerenciamento do local, onde há o banco de dados, que a informações

como o número de série, patrimônio, modelo, marca, data da última preventiva, entre outras especificações dos ativos. Já à direita, ficaria o projeto que teria uma interface de busca, onde o operador pesquisaria pelo número de série ou de patrimônio, o programa pesquisaria no banco de dados as informações sobre o equipamento, em seguida localizaria em que unidade estaria o equipamento.

A fim de agilizar e simplificar o processo, a ideia foi novamente refinada, como mostra na Figura 11.

Figura 11 - Fluxograma da segunda evolução da ideia do projeto



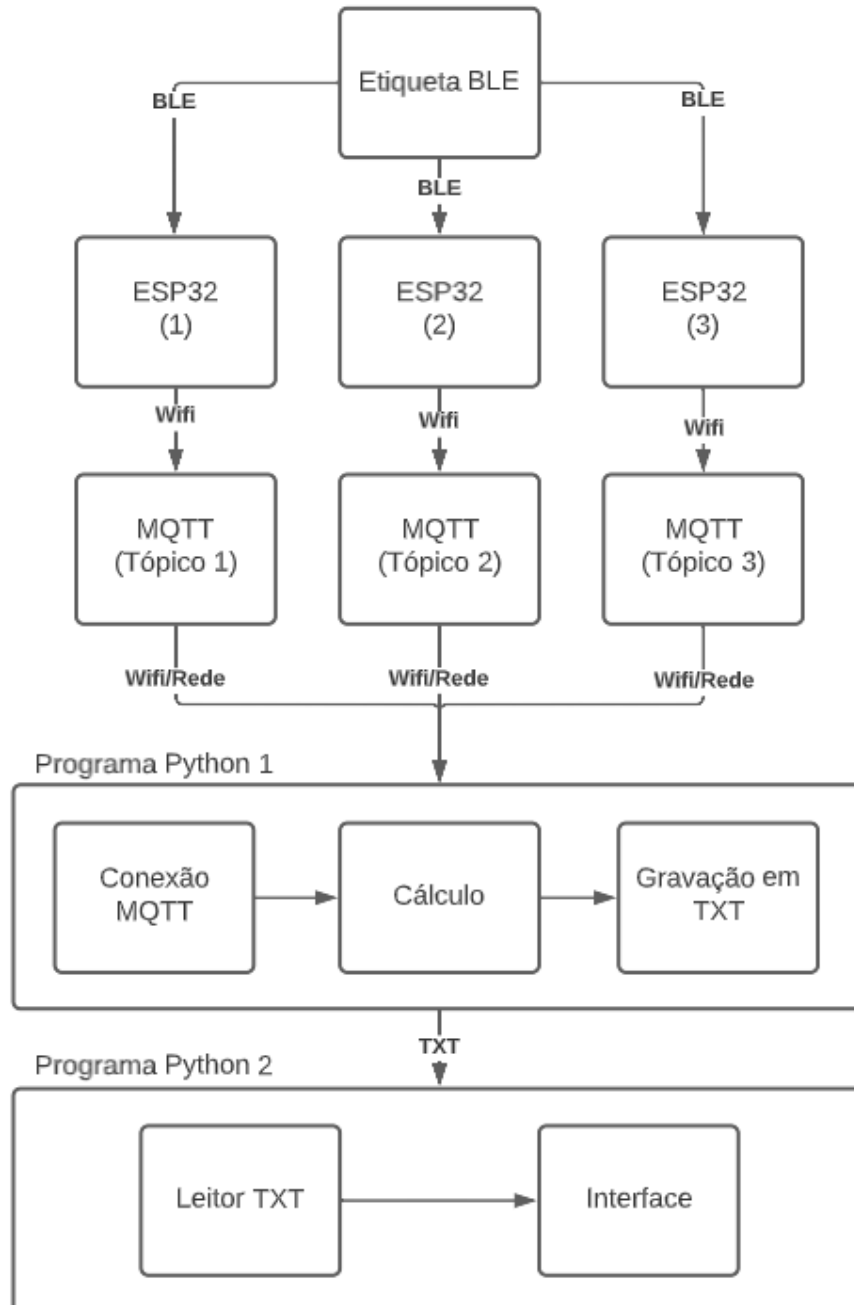
Fonte: o autor

A figura acima demonstra o funcionamento da proposta, onde não é atrelado a outro sistema de gerenciamento. Com isso, simplifica o processo e o desenvolvimento, visto que a variável inserta dos sistemas autônomos, principalmente de hospitais menores, é descartada. O operador ao abrir o sistema terá no display a localização de todos os dispositivos cadastrados. No caso do projeto foram utilizados dois para fins de teste.

A ideia funciona com as placas ESP32 captando o sinal das etiquetas, enviando os dados contidos nos sinais de cada uma para o servidor MQTT. Após ser publicado, um primeiro programa se inscreve para receber esses dados do servidor, realiza os cálculos necessários para a localização e envia para o segundo programa e, este é responsável por ler este arquivo e criar uma interface.

O diagrama da figura 12 foi criado para explicar de forma mais detalhada, separando em cinco etapas.

Figura 12 - Diagrama detalhado das etapas do programa



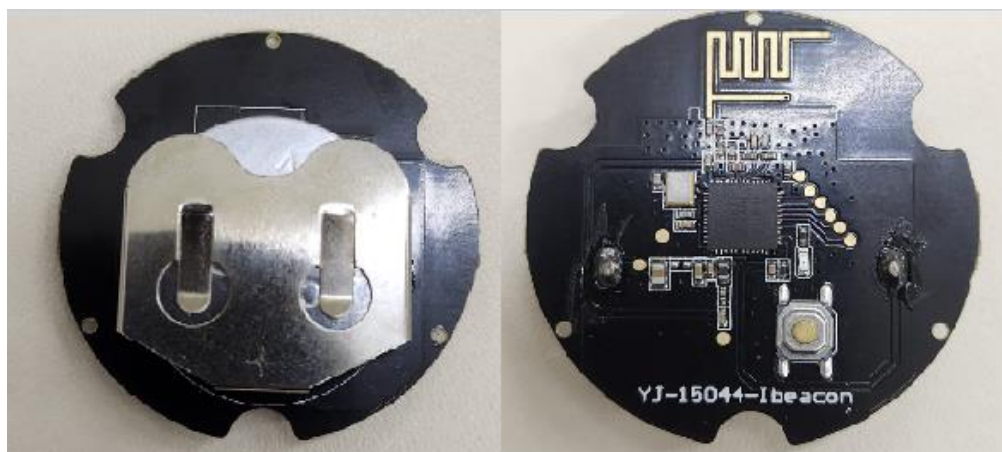
Fonte: o autor

Cada uma destas etapas foram planejadas e desenvolvidas para realizar uma única operação, mas todas elas visando chegar a um resultado. Essas etapas serão melhor explicadas a seguir.

3.1 Etiquetas BLE

As etiquetas BLE utilizadas, conforme mostra a foto 1 são similares as iBeacons, porém de fabricação chinesa. Apesar da qualidade ser inferior, possuem um ótimo custo-benefício. Para o projeto foram utilizadas duas, onde ambas emitem um sinal BLE que contém o MAC e o RSSI, sendo o MAC a identificação do dispositivo e cada um possui o seu e o RSSI a potência do sinal em dB. A potência desse sinal, varia de -20dB, para uma situação de distância inicial igual a zero, até -100dB para sua distância máxima, convertendo isto para metros, tem-se aproximadamente 40 metros (DANTAS, 2021).

Foto 1 - Etiquetas BLE



Fonte: o autor

No projeto, esses beacons emitem o sinal BLE que é captado pelas ESP32. A figura 14 demonstra as *tags* utilizadas, lembrando que ambas utilizam pilhas CR2477 de 3V, as mesmas utilizadas em termômetros. Após a consolidação do método, essas etiquetas serão incorporadas a equipamentos, de modo que não comprometa a estrutura (DANTAS, 2021).

3.2 ESP32

Como mencionado anteriormente, o projeto faz o uso da IDE do Arduino para a compilação no ESP32 do programa desenvolvido.

3.2.1 Configuração ESP32

Para realizar esta compilação é necessário configurar a IDE, assim, a placa que utilizada será identificada e compilada corretamente pelo programa. O passo a passo de como configurar a interface do Arduino é demonstrado no anexo A.

Após ser feita a configuração e a placa “ESP32 Dev Module” selecionada, é necessário instalar duas bibliotecas, sendo elas a PubSubClient e a PubSubClientTools. Elas serão responsáveis pela comunicação dos ESP com o servidor MQTT, o passo a passo dessa instalação é demonstrado no anexo A.

3.2.2 Programa ESP32

Com as configurações iniciais concluídas, é necessário se conectar com o wifi e habilitar o scanner de bluetooth das ESP32. Pelo bluetooth é realizada a comunicação das placas com as etiquetas, onde são recebidos o MAC e o RSSI de todos os dispositivos que emitem um sinal BLE que estão no raio de cobertura. Com isso, faz-se necessário realizar uma filtragem para não captar os dispositivos indesejados, assim não sobrecarregando o buffer. Ao ser feito este filtro, é fundamental também aplicar um filtro para não receber ruídos nos sinais e captar apenas o MAC e o RSSI das *tags*.

Após ser feito o tratamento destes sinais e separado apenas o essencial, utiliza-se o Wi-Fi e as bibliotecas previamente instaladas, PubSubClient e PubSubClientTools, para se conectar com um servidor de MQTT. Essa conexão é feita, utilizando de códigos e funções contidas nas bibliotecas, onde é conectado a um servidor e se inscrevendo em um tópico, como publicador. No caso do projeto foi utilizado o servidor do EMQX. Para se conectar a este servidor foi necessário se conectar a um hospedeiro e em uma porta pré-determinada, no caso é o endereço "broker.emqx.io" e a porta 1883.

Figura 13 - Exemplo de inscrição em tópico

```
Serial.println(client.publish("esp1", message_char_buffer, payloadString.length(), false)); //topico 1
Serial.println(client.publish("esp2", message_char_buffer, payloadString.length(), false)); //topico 2
Serial.println(client.publish("esp3", message_char_buffer, payloadString.length(), false)); //topico 3
```

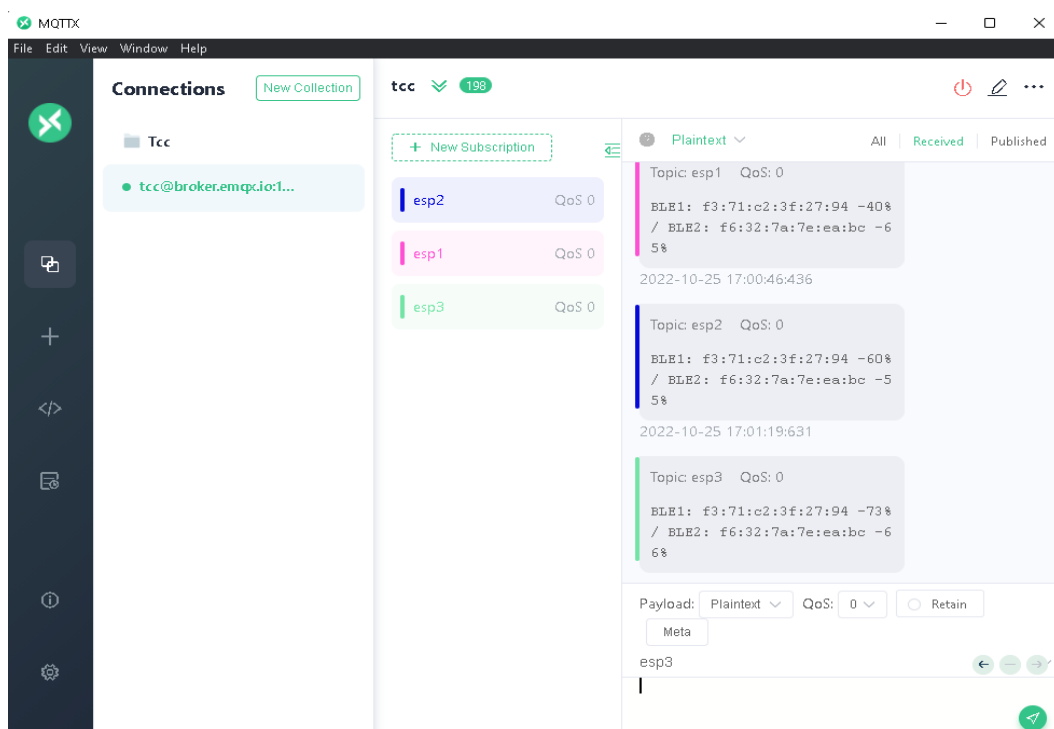
Fonte: o autor

Como mostra na figura 13, após o se conectar ao servidor, cada ESP32 é inscrito em um tópico específico, sendo eles “esp1”, “esp2” e “esp3”, onde o dispositivo irá publicar uma mensagem contendo o endereço MAC e a potência do sinal RSSI, de cada um dos BLE previamente cadastrados. Este processo de publicação é realizado cada vez que os ESP32 terminam de realizar o escaneamento da área, onde foi limitado para a aquisição de no máximo 50 dispositivos, ou a cada cinco segundos, o que for mais rápido. Porém, raras vezes, a aquisição demora mais de três segundos.

3.3 Server MQTT

Como assinalado anteriormente, o servidor MQTT escolhido para ser o Broker do projeto foi o EMQX e o motivo desta escolha é o fato de ser um servidor estável, confiável, gratuito e de fácil manuseio. Possui interface na nuvem e uma interface chamada de MQTTX disponível para download. Esta última, como mostra a figura 14, foi a utilizada.

Figura 14 - Interface MQTTX



Fonte: o autor

Esta interface serve para criar os tópicos e como auxílio, para verificar se as mensagens entre publicador e subscrito estão sendo enviadas.

3.4 Programa em Python - Parte 1

Foi desenvolvido um programa, no qual foi dividido em três partes com o objetivo de realizar a conexão com MQTT, realizar os cálculos de geolocalização e gravar em um arquivo TXT.

3.4.1 Conexão MQTT

A conexão com o servidor é feita de forma semelhante as dos ESP32, através da conexão via rede ou Wi-Fi. Entretanto, ao invés de publicador, o programa foi inscrito nos três tópicos como subscrito. A mensagem armazenada no Broker é recebida e separada através do endereço MAC. De acordo com essa separação é possível determinar de qual BLE é a informação da potência do sinal.

Com a inscrição feita em todos os tópicos, o programa coloca junto a informação de cada tópico referente ao mesmo BLE. No final da triagem, o programa obteve três RSSI de cada etiqueta que são convertidos de *strings* para números inteiros, para poder ser realizados os cálculos e comparativos na próxima etapa. Esse processo de aquisição dos dados é realizado cada vez que uma mensagem é publicada no servidor, de acordo com a taxa de aquisição nas ESP32.

3.4.2 Cálculo

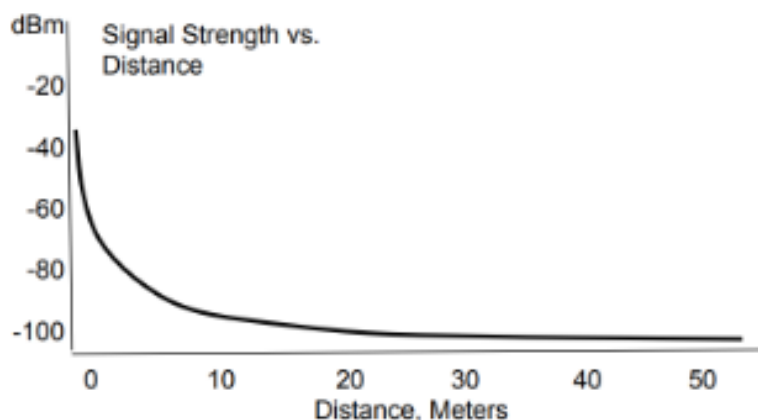
Após obter os três RSSI de cada etiqueta, é possível determinar a localização por alguns métodos. Neste programa foram utilizados dois métodos que direcionam o projeto para dois resultados. Ambos os métodos foram desenvolvidos simultaneamente ao longo do programa, para fins também de validação, visto que ambos devem apresentar o mesmo resultado, sendo assim, utilizado como “prova real”.

O primeiro método é mais simples, baseia-se em comparativos da potência entre os sensores, onde a potência maior reflete ao equipamento estar mais próximo. Levando em conta que o sinal emitido do beacon varia de -20dB até -100dB, o que emitir o sinal mais próximo de -20dB, é considerado mais próximo. Por exemplo, se

os três ESP32 captarem -55dB, -80dB e -43dB, o equipamento considerado mais próximo da placa será o que captou uma potência de -43dB, reduzindo consideravelmente o campo de busca para a localização deste equipamento.

Para o segundo método, primeiramente é necessário converter a potência do sinal em metros. Contudo, esse sinal não é linear, pelo contrário, o sinal se comporta, conforme o gráfico 1. Este sinal depende também do ambiente em que se propaga: se no ambiente houver muitas superfícies planas e reflexivas, o sinal tende a se propagar com ruídos, assim como há perda de sinal caso atravesse objetos ou paredes. Conhecendo essas características, há duas possibilidades para converter a potência em distância: utilizar uma fórmula genérica, que é mais simples e se adapta com uma acurácia aceitável para todos os ambientes ou fazer coletas das intensidades dos sinais de acordo com distâncias pré-definidas. Essa técnica tem o potencial de alcançar uma acurácia superior a anterior, entretanto, para cada ambiente e localização, terá que ser refeita a amostragem, senão sua acurácia é inferior ao da fórmula genérica, tornando essa técnica menos viável. Por este motivo foi utilizado a fórmula genérica.

Gráfico 1 Potência x Distância



Fonte: <http://www.davidgyoungtech.com/2020/05/15/how-far-can-you-go>

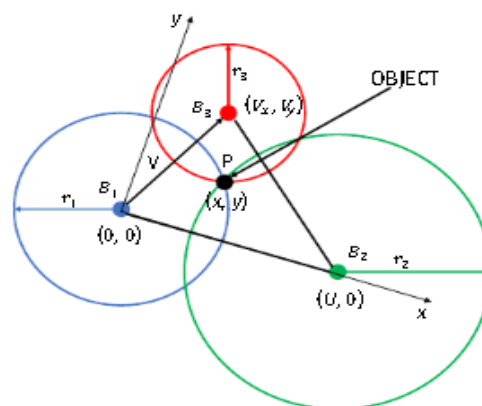
O cálculo da conversão da do sinal em metros é simples e pode ser expresso pela equação (3.1):

$$d = 10^{\frac{p-s}{10.n}} \quad (3.1)$$

Onde d é a distância, p é a potência medida na distância de um metro, s é a potência do sinal e n , a constante que demonstra a transferência deste sinal pelo ar. Essa constante varia de 20 a 40 geralmente, sendo utilizado 30. Após converter todos os RSSI em metros, o programa encaminha estes valores para o segundo método.

Para o segundo método, o de trilateração, cujo processo faz uso de três pontos para determinar a localização do objeto, como mostra a figura 18, são necessárias algumas informações, como conhecer a localização de no mínimo três ESP32 e as distâncias delas até o alvo.

Figura 15 - Exemplo de Trilateração



Fonte: https://www.researchgate.net/publication/338241733_A_Comparison_Analysis_of_BLE-Based_Algorithms_for_Localization_in_Industrial_Environments

Usando a figura 15 como exemplo prático, o objetivo dos cálculos a seguir é descobrir o valor das coordenadas, x e y , do ponto P . É de conhecimento os valores vetoriais dos pontos B_1 , B_2 e B_3 , que são pré-definidos como as posições dos ESP32, e os valores dos raios r_1 , r_2 e r_3 , são os valores do RSSI após serem aplicadas na fórmula 3.1. Tendo estes valores, é necessário descobrir a distância entre B_1 e B_3 . Essa distância é dada pela equação (3.2):

$$V = \sqrt{V_x^2 + V_y^2} \quad (3.2)$$

Onde V é a distância entre os pontos B_1 e B_3 , V_x e V_y são os valores das coordenadas x e y do ponto B_3 . Com o cálculo anterior e como a distância de B_1 e B_2 já é conhecida, visto que é igual ao valor do eixo x (U), todos os valores necessários para calcular as coordenadas do ponto P , podem ser determinados pelas equações (3.3) e (3.4):

$$x = \frac{r_1^2 - r_2^2 + U^2}{2 \cdot U} \quad (3.3)$$

$$y = \frac{r_1^2 - r_3^2 + V^2 - 2 \cdot V_x \cdot x}{2 \cdot V_y} \quad (3.4)$$

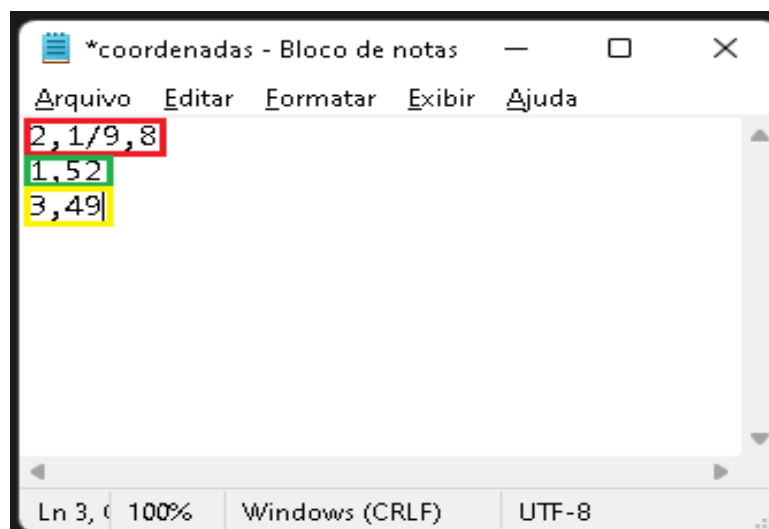
Aplicando os valores conhecidos nas fórmulas 3.3 e 3.4, obtém-se os valores das coordenadas x e y do ponto P .

Esta sequência de procedimentos é aplicada para cada um dos dispositivos cadastrados. Como relatado anteriormente, foram cadastrados dois dispositivos, sendo assim, esse processo será feito duas vezes, gerando P_1 e P_2 , assim como assinalando de qual ESP32 cada um deles está mais próximo. Com esta etapa concluída, este programa passa para a última etapa, a de gravação em um arquivo de txt.

3.4.3 Gravação em TXT

Como última etapa deste programa, é realizada a criação de um arquivo .txt com o nome "coordenadas". Após isto, é feita a gravação de qual ESP32 e o valor de seu RSSI, no qual representa a menor distância, assim como as coordenadas dos pontos P de cada um dos dispositivos cadastrados. Cada vez que o valor é atualizado no servidor, as informações deste arquivo são apagadas e criadas novamente, assim não ocupando muita memória.

Figura 16 - Exemplo de arquivo txt criado



Fonte: o autor

Conforme a figura 16 demonstra, na primeira linha, destacado em vermelho, estão salvas as coordenadas dos dois pontos, P_1 e P_2 , dos dois dispositivos, separados por uma barra (/). Na linha dois e linha três, estão as informações do dispositivo 1 e do dispositivo 2, destacados de verde e amarelo, respectivamente. O primeiro valor representa de qual ESP32 o BLE está mais próximo e o segundo valor é seu respectivo RSSI.

3.5 Programa em Python – Parte 2

Foi desenvolvido um segundo programa, no qual foi dividido em duas partes, cujo objetivo é de realizar a leitura do arquivo txt, realizar a plotagem gráfica do segundo método criando uma interface para apresentar os resultados.

3.5.1 Leitor TXT

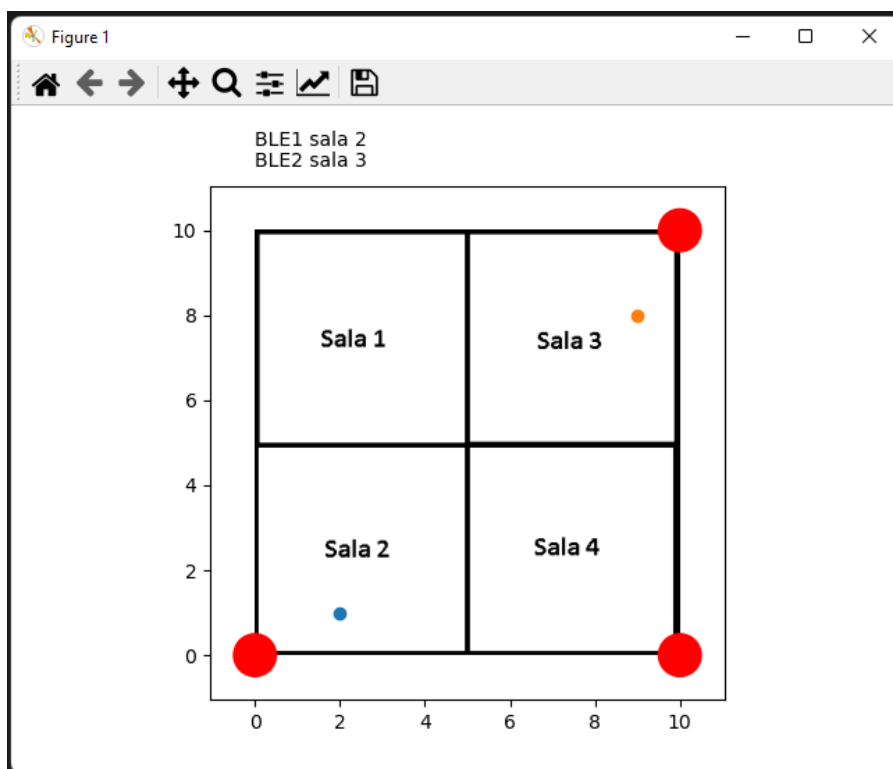
Esta etapa é semelhante ao processo de gravação do programa anterior, entretanto, tem como objetivo ler os dados armazenados no arquivo de texto e repassar para variáveis, transformando de *strings* para números inteiros. É um procedimento inverso ao de gravar, ou seja, ao invés de juntar os dados é necessário separá-los de acordo com seus significados.

Utilizando a figura 16 como exemplo, primeiramente é separado em linhas, onde a destacada de vermelha será a coordenada das etiquetas que deverá ser plotada, e as linhas verde e amarela, os resultados de qual ESP32 e a potência que as BLE₁ e BLE₂ estão mais próximos, respectivamente. Já a primeira linha contém duas coordenadas, “x” e “y”, de cada um dos dispositivos, separadas por uma barra. Após a leitura deste arquivo, suas conversões para números inteiros e separação dos valores para suas devidas variáveis, está pronto para ser realizado a plotagem gráfica.

3.5.2 Interface

Após a leitura, conversão e separação dos valores, é utilizada a biblioteca Matplotlib para a criação de uma interface inicial. Nesta interface são apresentados os resultados dos dois métodos. Os pontos vermelhos, que são maiores, representam a localização dos ESP32. O desenho de fundo é um esboço do laboratório de Engenharia Clínica e manutenção do Hospital de Pronto Socorro de Canoas - RS, onde foi dividido em quadrantes, para simular salas no decorrer dos testes. O ponto azul e o laranja representam os dispositivos BLE₁ e BLE₂ respectivamente. Foi utilizada uma escala de 0 a 10 pontos, tanto no eixo “x” como no “y”. Para esses valores representarem uma escala real, foi atribuído a V_x e a V_y o valor de 10, e U o valor de 10 também.

Figura 17 – Interface



Fonte: o autor

O primeiro método é demonstrado através de duas mensagens no topo da plotagem, onde demonstram de qual ESP32 o equipamento está mais próximo. Como demonstra a figura 17, o BLE₁, ponto azul, por estar mais próximo da placa localizada na sala 1, demonstra que sua localização é nesta mesma sala, já o BLE₂, ponto laranja, por estar mais próximo da placa localizada na sala 4, demonstra que está nesta mesma sala. O segundo método é demonstrado através de uma plotagem gráfica que simula a geolocalização através da trilateralização. Essa geolocalização é atualizada através dos valores do documento de texto. É utilizado a função “*Animation*”, da biblioteca matplotlib, para realizar a atualização automática deste gráfico. Essa função possibilita uma interface mais harmônica, em que é possível escolher uma taxa de atualização que deve ser igual ou maior que a atualização dos dados no arquivo.

4 ANÁLISE DOS RESULTADOS

Os resultados obtidos nos dois métodos foram satisfatórios, apresentando resultados dentro do esperado.

4.1 Método 1

Para a comprovação deste método, foram realizados diversos testes, entre eles, foram coletados 30 para amostra. Estes testes foram feitos de maneira simples. Os BLEs foram colocados de forma manual mais próximo de um ESP32 específico que estavam distribuídos conforme a figura 17, o programa realizava a leitura em paralelo com as anotações de onde estava mais próximo.

Quadro 3 - Resultados método um

Testes	Método Um					
	BLE 1			BLE 2		
	RSSI mais próximo	ESP32 esperado	ESP32 programa	RSSI mais próximo	ESP32 esperado	ESP32 programa
1	46	1	1	67	1	1
2	47	3	3	65	3	3
3	54	2	2	88	3	3
4	79	2	2	71	3	3
5	50	2	2	42	3	3
6	84	3	3	61	3	3
7	56	2	2	73	1	1
8	43	1	1	55	2	2
9	87	2	2	54	2	2
10	56	1	1	66	2	2
11	70	3	3	74	3	3
12	55	3	3	81	2	2
13	64	2	2	79	2	2
14	71	2	2	77	1	1
15	87	2	3	65	3	3
16	50	2	2	47	1	1
17	51	3	3	46	2	2
18	51	3	3	50	2	2
19	83	2	2	68	2	2
20	88	2	2	66	2	2
21	85	1	1	80	1	2
22	41	2	2	54	1	1
23	79	3	3	44	3	3
24	43	1	1	43	2	2
25	41	3	3	47	1	1
26	84	3	3	50	2	2
27	63	3	3	84	2	2
28	52	1	1	80	1	1
29	66	1	1	50	3	3
30	85	1	1	71	3	3

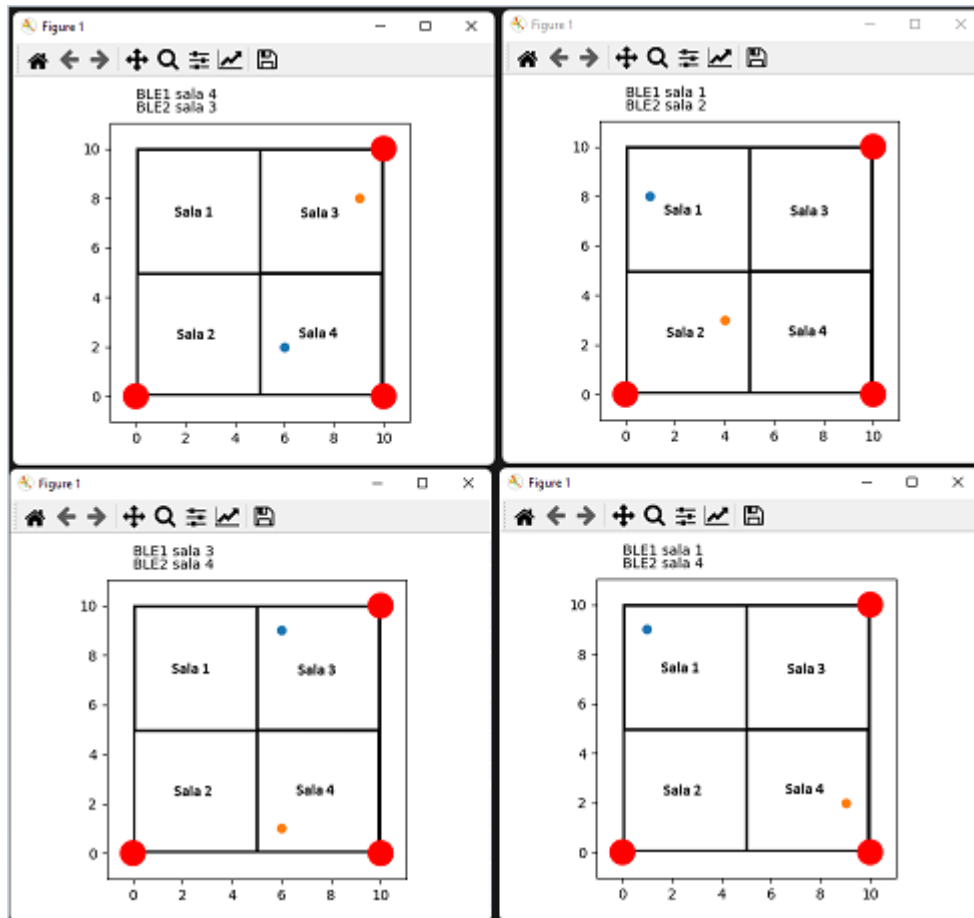
Fonte: o autor

Os resultados obtidos foram os do quadro 3, onde pode-se verificar que praticamente todos os resultados obtidos através do programa, coincidiram com o esperado, com exceção da medição do teste 15 do BLE₁ e do teste 21 do BLE₂, onde há uma divergência de resultados. Esta divergência deve-se ao fato de os dispositivos, nestes testes, estarem próximos ao centro, onde qualquer leve mudança da intensidade do RSSI, seria o suficiente para tender para qualquer um dos três resultados possíveis. Com apenas 2 erros, entre 60 medições, temos uma acurácia de aproximadamente 96%, porém não mostra a localização em tempo real como o próximo método que será apresentado.

4.2 Método 2

A comprovação do segundo método foi semelhante ao anterior, onde foram feitos diversos testes e simulações, colocando o dispositivo em lugares aleatórios pelo laboratório. Os resultados obtidos são demonstrados na figura 18.

Figura 18 - Exemplo de resultados método 2



Fonte: o autor

Através da coleta destas simulações, foi possível criar o quadro 4 para melhor entendimento e para comparação dos resultados obtidos com os esperados.

Quadro 4 - Resultados método dois

Testes	Método Dois										Erro em x	Erro em Y	Erro em x	Erro em Y
	BLE1					BLE2								
	Valor medido		Valor esperado			Erro em x	Erro em Y	Valor medido		Valor esperado				
x	y	x	y	x	y			x	y	x	y			
1	3	6	4	5	1	-1	9	4	9	3	0	-1		
2	7	5	7	6	0	1	5	4	5	5	0	1		
3	7	4	8	5	1	1	2	7	1	7	-1	0		
4	3	7	4	8	1	1	4	6	5	5	1	-1		
5	3	6	4	6	1	0	9	6	10	6	1	0		
6	1	8	2	9	1	1	1	2	1	3	0	1		
7	1	3	1	4	0	1	2	3	2	3	0	0		
8	1	8	0	9	-1	1	6	1	7	2	1	1		
9	7	4	7	4	0	0	4	2	3	2	-1	0		
10	5	7	4	8	-1	1	8	3	9	4	1	1		
11	2	4	3	5	1	1	8	4	7	4	-1	0		
12	9	7	8	7	-1	0	5	7	5	7	0	0		
13	7	2	7	3	0	1	6	2	6	1	0	-1		
14	4	5	4	5	0	0	4	2	5	3	1	1		
15	5	2	5	3	0	1	8	6	8	6	0	0		
16	8	1	8	0	0	-1	1	1	1	2	0	1		
17	5	4	4	5	-1	1	9	1	9	0	0	-1		
18	5	8	6	8	1	0	9	2	10	2	1	0		
19	5	1	6	2	1	1	1	5	1	5	0	0		
20	2	8	1	9	-1	1	1	7	0	7	-1	0		
21	7	8	7	7	0	-1	5	6	6	5	1	-1		
22	7	7	7	8	0	1	7	7	8	7	1	0		
23	3	4	3	4	0	0	3	9	2	9	-1	0		
24	1	8	0	8	-1	0	8	2	9	1	1	-1		
25	4	5	3	5	-1	0	1	3	2	3	1	0		
26	1	8	2	9	1	1	9	2	8	2	-1	0		
27	8	4	7	4	-1	0	1	3	0	3	-1	0		
28	3	9	3	10	0	1	5	8	4	8	-1	0		
29	7	5	7	5	0	0	1	8	0	8	-1	0		
30	4	9	4	9	0	0	4	3	4	3	0	0		

Fonte: o autor

Após a realização deste quadro, observa-se que há, em muitas das medições, um erro de no máximo 1 ponto de diferença entre o calculado/esperado e o medido através do programa. Com os resultados do quadro anterior, foi desenvolvida o quadro 5 com a quantidade de cada erro

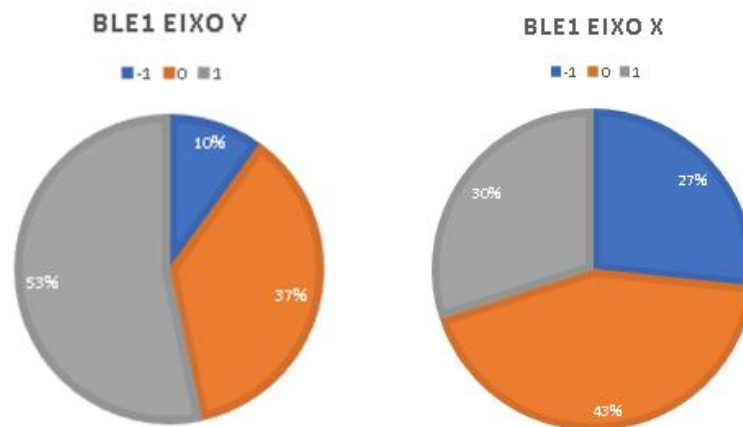
Quadro 5 - Quantidade de erros

Valor	BLE1		BLE2	
	x	y	x	y
-1	8	3	9	6
0	13	11	11	18
1	9	16	10	6
Total	30	30	30	30

Fonte: o autor

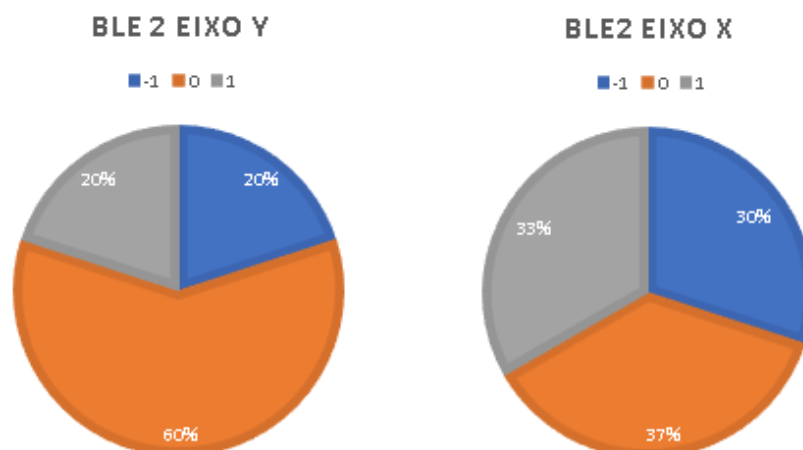
Com este quadro montado, é possível saber a taxa de acerto, sendo considerado acerto quando a diferença entre o medido e o calculado é igual a zero. Foram desenvolvidos gráficos das porcentagens de erros e acertos de cada eixo para os dois BLEs. Como se pode observar nos gráficos 2 e 3, a taxa de erro não passa de 1 ponto, tanto para mais como para menos, pode-se notar que em diversas medições não houve diferença entre a medição e o calculado.

Gráfico 2 - Erros BLE1



Fonte: o autor

Gráfico 3 - Erros BLE2



Fonte: o autor

O local utilizado para os testes possui aproximadamente 25 metros quadrados, que foram divididos em 10 pontos, ou seja, a distância de cada ponto possui 50 centímetros. Cada vez que o valor obtido no programa era decimal, sempre era arredondado. Então, a variação real em metros do erro é de 0,5, tanto para mais como para menos. O que, em um ambiente de 25 metros quadrados, caracteriza-se como uma precisão de 90%

5 CONSIDERAÇÕES FINAIS

O resultado do projeto foi muito satisfatório para o tempo proposto a desenvolvê-lo. Muito foi desenvolvido e aprendido. Foi comprovado ser possível implementar um sistema de localização de equipamentos em um ambiente hospitalar. Há também o fato de ser uma tecnologia de baixo custo, facilitando o acesso a hospitais, clínicas e postos de saúde mais humildes, que não tem muita disponibilidade de orçamento. Além é claro, de grandes hospitais que queiram adicionar o sistema a seu arsenal de gerenciamento.

Sempre há o que melhorar nos projetos, neste não é diferente. Ainda há coisas para desenvolver, como desenvolver na prática outros métodos teóricos de geolocalização, como o multilateralismo e o *fingerprinting*, melhorar os algoritmos e convertê-los para programas executáveis independentes, não sendo necessário uma IDE de Python para reproduzi-los, criar uma interface nova para cadastro e pesquisa, para não ser necessário o operador entender de programação para cadastrar novos equipamentos.

Os dois tipos de geolocalização que podem ser adicionados em trabalhos futuros, agem de maneira diferente um do outro, onde o multilateralismo é uma evolução da trilateração, utilizando mais de três receptores de sinal, melhorando assim a eficácia das medições. Já o *fingerprinting* utiliza, por exemplo, uma rede neural que irá aprender a localização de cada ponto de localização, então quando o objeto estiver em certa localização, a rede neural irá identificar.

Este projeto será levado adiante para fins comerciais. Para tal, é necessário implementar os itens anteriormente citados, além de criar um servidor próprio de MQTT via TCP/IP, para não ser necessário acesso à internet e não depender de um servidor de terceiros, adequar as normas hospitalares, realizar gráficos de *fitting* dos valores do BLE para cada localização e conseguir autorização e aplicar simulações dentro do ambiente hospitalar.

Ao ser aplicado, este projeto diminuirá o tempo necessário para um equipamento ser encontrado, otimizando os recursos e diminuindo os gastos, com isto a entidade poderá ter um uso mais eficaz dos recursos, como adquirir novos equipamentos. Há a possibilidade de ser agregado um sistema de gestão completo, não necessário haver mais de um sistema utilizado simultaneamente. Poderá ser

utilizado para outros fins sem ser o médico-hospitalar, pode ser utilizado por completo ou como base para um sistema de controle de estoque ou de máquinas, nos meios industriais, agricultura, entre outros, é possível a implementação desta ideia em praticamente todo meio que seja necessário um método de controle, bastando realizar alguns ajustes para este meio.

REFERÊNCIAS

ALVES, Paulo. O que é NFC no celular: como funciona e para que serve a tecnologia. **Techtudo**, 2019. Disponível em: <<https://www.techtudo.com.br/noticias/2019/08/o-que-e-nfc-no-celular-como-funciona-e-para-que-serve-a-tecnologia.ghtml>>. Acesso em 18 de maio de 2022.

AUFRANC, Jean-Luc. ESP8266 and ESP32 Differences in One Single Table. **CNX SOFTWARE**, 2016. Disponível em: <<https://www.cnx-software.com/2016/03/25/esp8266-and-esp32-differences-in-one-single-table/>>. Acesso em 5 de setembro de 2022.

BAJUR, Thiago; LUNARDI, Laura. Engenharia Clínica – Inovações tecnológicas como apoio na gestão hospitalar. **Biominas Brasil**, 2016. Disponível em: <<https://biominas.org.br/blog/engenharia-clinica-inovacoes-tecnologicas-como-apoio-na-gestao-hospitalar/#:~:text=Dessa%20forma%2C%20seu%20trabalho%20pode,em%20gest%C3%A3o%20e%20melhoria%20cont%C3%ADnua.%3E.>>. Acesso em 20 de abril de 2022.

CANNIZZARO, Davide; ZAFIRI, Marina; PAGLIARI, Daniele Jahier; PATTI, Edoardo. A Comparison Analysis of BLE-Based Algorithms for Localization in Industrial Environments. **ResearchGate**, 2019. Disponível em: <https://www.researchgate.net/publication/338241733_A_Comparison_Analysis_of_BLE-Based_Algorithms_for_Localization_in_Industrial_Environments>. Acesso em 25 de outubro de 2022.

CARNEIRO, Conrado. Beacon: o que é e quais suas utilizações? **USE**, 2016. Disponível em: <<https://usemobile.com.br/conheca-beacon/>>. Acesso em 05 de maio de 2022.

CURVELLO, André. ESP32 – Um grande aliado para o Maker IoT. **FILIFELOP**, 2018. Disponível em: <<https://www.filipeflop.com/blog/esp32-um-grande-aliado-para-o-maker-iot/>>. Acesso em 5 de setembro de 2022.

DANTAS, Janaina. O que são Beacons? **Tecnoblog**, 2022. Disponível em: <<https://tecnoblog.net/responde/o-que-sao-beacons/>>. Acesso em 05 de maio de 2022.

DELTECNOLOGIA. **Deltecnologia**, 2018. Qual a importância do engenheiro clínico no ambiente hospitalar? Disponível em: <<http://deltecnologia.com.br/qual-a-importancia-do-engenheiro-clinico-no-ambiente->>. Acesso em 20 de abril de 2022.

DIAS, Vinícius. C++: Vale apenas aprender em 2021? **Alura**, 2021. Disponível em: <<https://www.alura.com.br/artigos/formacao-linguagem-c-plus-plus>>. Acesso em 05 de maio de 2022.

FERNANDES, Natalia. **HI Tecnologia**, 2021. Disponível em: <<https://materiais.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/>>. Acesso em 21 de setembro de 2022.

GRANDI, Everton Luis; MARCOS, Pedro de Botelho; PARZIANELLO, Lucas; COSTA, Thiago Manuel Fortunato da. iTracking - Um Framework para Rastreamento Através de RFID. **ResearchGate**, 2015. Disponível em: <https://www.researchgate.net/figure/Figura-2-Esquema-para-o-calculo-de-posicao-por-trilateracao_fig2_283516019>. Acesso em 12 de outubro de 2022.

HANASHIRO, Akira. VS Code - O que é e por que você deve usar? **TREINAWEB**, 2021. Disponível em: <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>>. Acesso em 10 de setembro de 2022.

HOW FAR CAN YOU GO? **Davidyoungtech**, 2020. Disponível em: <<http://www.davidyoungtech.com/2020/05/15/how-far-can-you-go>>. Acesso em 21 de setembro de 2022.

JAVA. Java, 2022. O que é a tecnologia Java e por que preciso dela? Disponível em: <https://www.java.com/pt-BR/download/help/whatis_java.html>. Acesso em 13 de maio de 2022.

JAVA: Como funciona a linguagem e principais características. **Remessa Online**, 2021. Disponível em: <<https://www.remessaonline.com.br/blog/java-como-funciona-a-linguagem-e-principais-caracteristicas/>>. Acesso em 13 de maio de 2022.

MELLO, Heloisa C. Quando deve ser feita a manutenção de equipamentos hospitalares? **Medicalway**, 2021. Disponível em: <<https://blog.medicalway.com.br/quando-deve-ser-feita-a-manutencao-de-equipamentos-hospitalares/>>. Acesso em 21 de abril de 2022.

MICROMED. **Micromed**, 2019. Engenharia Clínica: a importância e os desafios da Medicina atual. Disponível em: <<https://micromed.ind.br/site/engenharia-clinica/>>. Acesso em 20 de abril de 2022.

MORSCH, José Aldair. O que é e como funciona o regime de comodato de equipamentos médicos. **Morsch Telemedicina**, 2020. Disponível em: <<https://telemedicinamorsch.com.br/blog/regime-de-comodato#:~:text=no%20comodato%2c%20os%20bens%20devem,recebimento%20de%20qualquer%20valor%20financeiro.>>. Acesso em 21 de abril de 2022.

MQTT. MQTT: The starting for IoT Messaging, 2022. Página inicial. Disponível em: <<https://mqtt.org/>>. Acesso em 10 de setembro de 2022.

O QUE É ESP32? PRA QUE SERVE? QUANDO USAR? **Lobodarobotica**, 2020. disponível em: <<https://lobodarobotica.com/blog/o-que-e-esp32-pra-que-serve-quando-usar/>>. Acesso em 5 de setembro de 2022.

O que é RFID? **Quebeck**, 2022. Disponível em: <<https://www.quebeckautomacao.com.br/o-que-e-rfid/>>. Acesso em 05 de maio de 2022.

PEREIRA, Vinicius. Python: O que é, quais as vantagens e onde aplicar? **KingBlog**, 2021. Disponível em: <<https://king.host/blog/2021/03/python/>>. Acesso em 02 de maio de 2022.

POLÍCIA prende técnica de enfermagem suspeita de revender equipamentos hospitalares para tratar pacientes com covid-19. **Gauchazh**, 2022. Disponível em: <<https://gauchazh.clicrbs.com.br/seguranca/noticia/2022/05/policia-prende-tecnica-de-enfermagem-suspeita-de-revender-equipamentos-hospitalares-para-tratar-pacientes-com-covid-19-cl33pg2ds007n019iosteabjw.html>>. Acesso em 01 de junho de 2022.

QUINTINO, Eduardo. O que é IDE Arduino? **FILIFELOP**, 2021. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-ide-arduino/>>. Acesso em 5 de setembro de 2022.

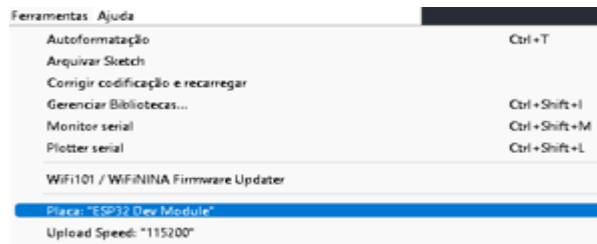
SANTOS, Guilherme. Protocolo MQTT: O Que é, Como Funciona e Vantagens. **AutomaçãoIndustrial**, 2022. Disponível em: <<https://www.automacaoindustrial.info/mqtt/>>. Acesso em 10 de setembro de 2022.
SOARES, Fernando. Sistemas MV e Tasy – O que são e para que servem? **Cmtecnologia**, 2019. Disponível em: <<https://blog.cmtecnologia.com.br/software-gestao-mv-tasy/>>. Acesso em 20 de junho de 2022.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. RFID, 2022. Página inicial. Disponível em: <https://www.gta.ufrj.br/grad/rfid/RFID_>. Acesso em 05 de maio de 2022.

ANEXO A - CONFIGURAÇÃO IDE ARDUINO PARA ESP32

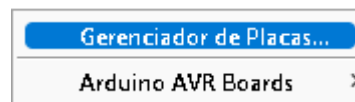
Para a realização da configuração da IDE é fundamental abrir o menu “ferramentas”, abrir a opção “Placa”, selecionar “Gerenciamento de Placas”. E dentro do novo menu que irá abrir, deve-se pesquisar por esp32 e instalar a opção que aparecer. Feita a instalação deve-se selecionar qual modelo de placa irá ser utilizado. Para isto, basta seguir o mesmo procedimento anterior, porém ao invés de entrar no gerenciamento de placas, deve-se selecionar “Esp32 Arduino” e selecionar o modelo da placa. No caso do projeto, será a “ESP32 Dev Module”. Passo a passo é demonstrado conforme as figuras 19 a 24.

Figura 19 – Etapa 1.1 IDE Arduino



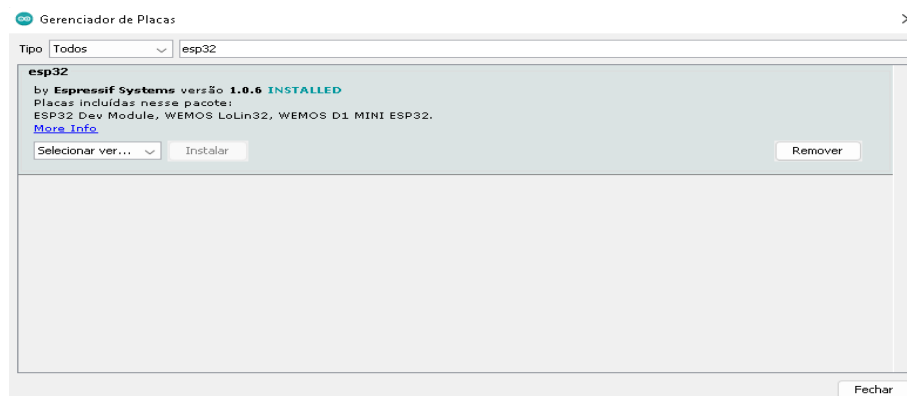
Fonte: o autor

Figura 20 - Etapa 1.2 IDE Arduino



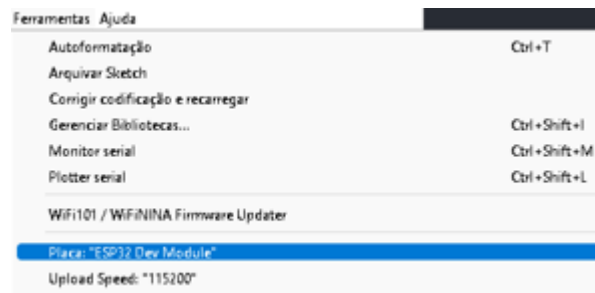
Fonte: o autor

Figura 21 - Etapa 1.3 IDE Arduino



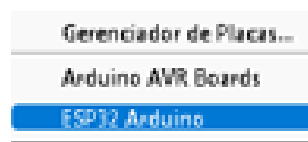
Fonte: o autor

Figura 22 - Etapa 1.4 IDE Arduino



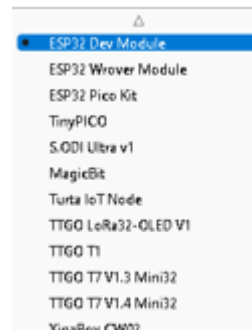
Fonte: o autor

Figura 23 - Etapa 1.5 IDE Arduino



Fonte: o autor

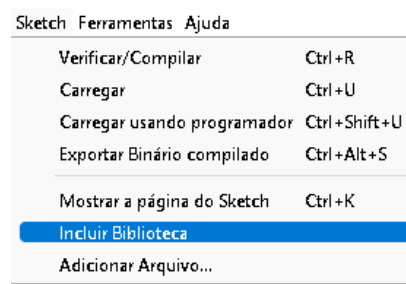
Figura 24 - Etapa 1.6 IDE Arduino



Fonte: o autor

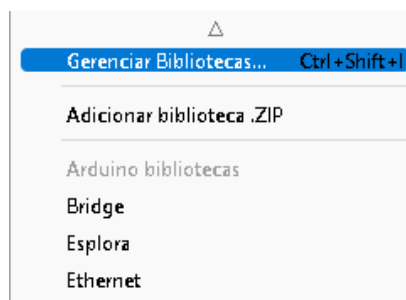
Para a implementação do projeto, foi necessária a instalação de algumas bibliotecas, como a PubSubClient e a PubSubClientTools. Para isto foi necessário entrar no menu “Sketch”, abrir a opção “Incluir Biblioteca” e selecionar “Gerenciar Bibliotecas”. Após esta etapa abre-se uma nova janela e nela, pesquisou-se por PubSubClient e PubSubClientTools e instalá-las. Passo a passo demonstrado conforme as figuras 25 a 27.

Figura 25 - Etapa 2.1 IDE Arduino



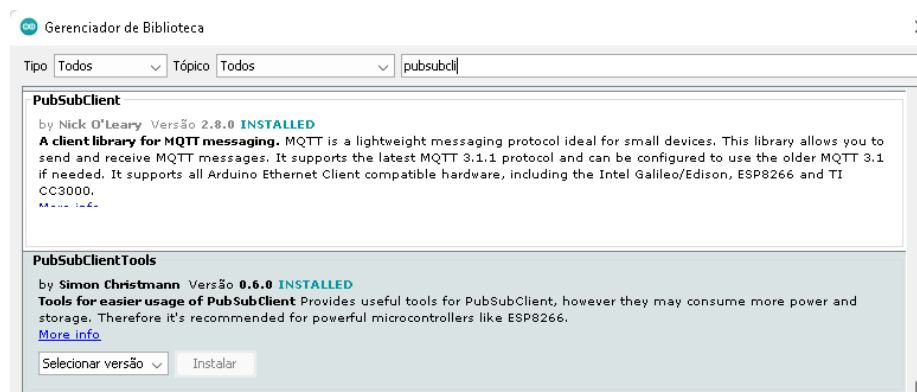
Fonte: o autor

Figura 26 - Etapa 2.2 IDE Arduino



Fonte: o autor

Figura 27 - Etapa 2.3 IDE Arduino



Fonte: o autor