66

**Controle do Documento**

**Histórico de revisões**

| **Data** | **Autor** | **Versão** | **Resumo da atividade** |
| --- | --- | --- | --- |
| 18/11/22 | Alysson Cordeiro. | 1.0 | * Descrição da solução[1.1]. * Implementação da arquitetura da solução[1.2]. * Implementação dos componentes e recursos[2.1, 2 e 3] |
| 30/11/22 | Alysson Cordeiro. | 2.0 | * Implantação dos créditos [8]. * Guia de Montagem[4]. |
| 04/12/22 | Alysson Cordeiro | 3.0 | * Guia de instalação[4]. * Guia de Configuração[5]. |

**Índice**

[**1. Introdução**](#_3p4k6d3g6219) **3**

[1.1. Solução](#_rlngioqecbyk) 3

[1.2. Arquitetura da Solução](#_61uhcal2j77f) 3

[**2. Componentes e Recursos**](#_uvfjwzlomuzy) **4**

[2.1. Componentes de hardware](#_jafy6yk85z5g) 4

[2.2. Componentes externos](#_dq0hfd7wcjor) 4

[2.3. Requisitos de conectividade](#_yxhdlhc9u11x) 4

[**3. Guia de Montagem**](#_v51amp5m28ia) **5**

[**4. Guia de Instalação**](#_ns4i2ee2va9l) **6**

[**5. Guia de Configuração**](#_mjz06zt366c7) **7**

[**6. Guia de Operação**](#_vcwsg1gripyk) **8**

[**7. Troubleshooting**](#_omvzmwr1fxwv) **9**

[**8. Créditos**](#_t6okuol326z9) **10**

# 1. Introdução

**Informação inicial:**

* Por favor leia este manual cuidadosamente antes de utilizar seu aparelho de forma a garantir um uso seguro e adequado.
* Este manual contém as informações básicas necessárias de especificações, instalação e operação do produto.

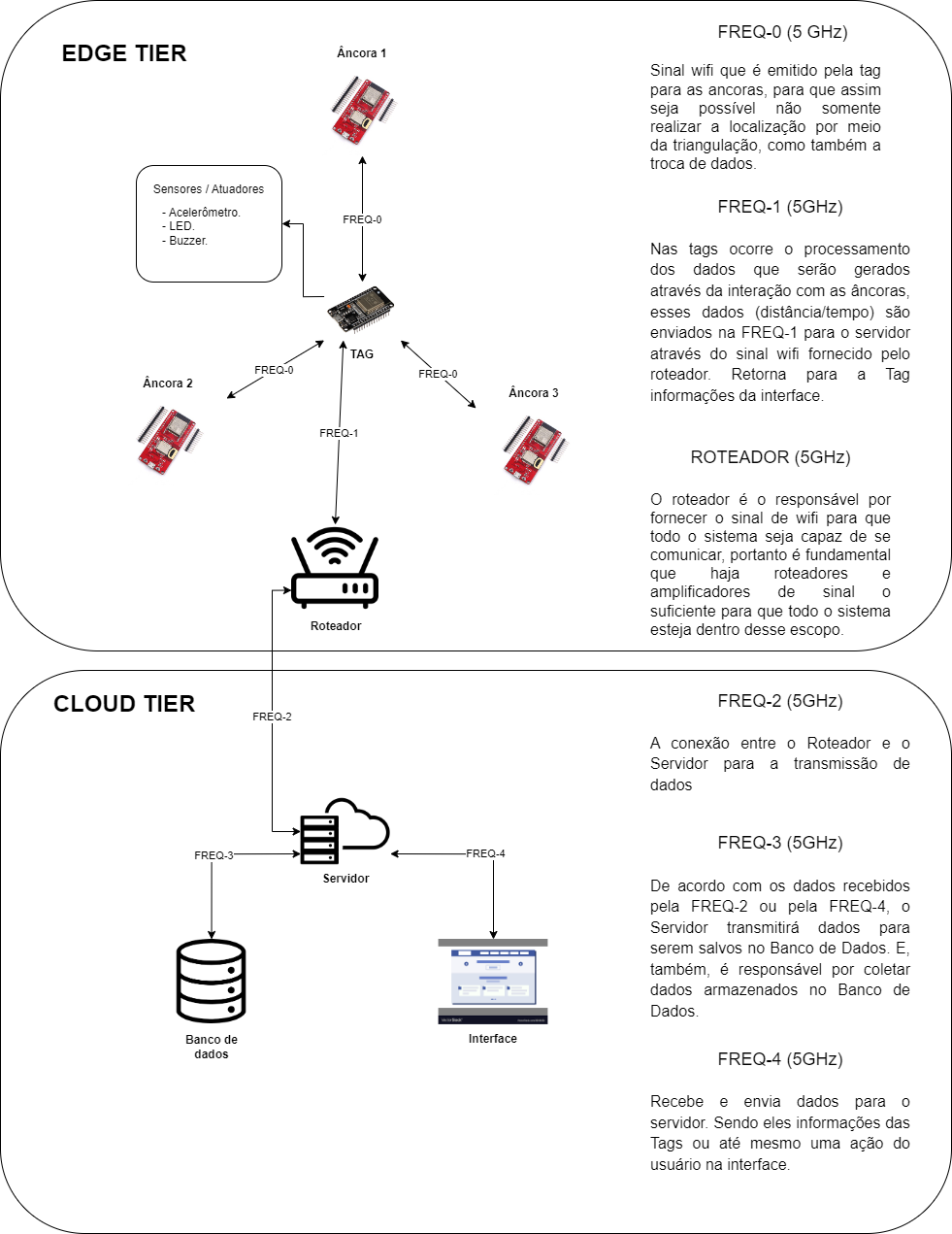
## 1.1. Solução.

Trata-se de uma solução baseada em IOT responsável por encontrar objetos em um ambiente controlado, com o apoio de sensores e tags. Por meio de uma aplicação web com interface gráfica capaz de se comunicar com o hardware utilizando um software embarcado (*firmware*) será possível a visualização da posição de um objeto ou de uma classe de objetos em relação ao espaço da instalação. Além do mais, a identificação é para objetos em ambientes fechados, por meio de técnicas de triangulação; e, também, utiliza a rede interna para transmitir informações para um aplicativo de tela grande pelo qual será possível configurar algumas especificações do dispositivo e acompanhar sua localização.

Vale ressaltar que, na utilização da solução, o produto funciona através de controle de sensores por radiofrequência que identificam e rastreiam de modo automático as tags em ambientes como galpões. Para mais, a comunicação dos beacons com a tag será realizada por meio de sinais wi-fi.

Em suma, o protótipo aumenta a capacidade de encontrar objetos nas instalações da Atech, economizando tempo e melhorando a organização dos ativos físicos da empresa.

## 1.2. Arquitetura da Solução.



# 2. Componentes e Recursos

## 2.1. Componentes de hardware

Não se esqueça também dos consumíveis, como pilhas e baterias.

1. **MICROCONTROLADOR:** componente responsável por propagar os sinais do Wi-Fi de modo a fazer a trilateração da posição do objeto. O microcontrolador responsável pela atuação é o ESP32-S3-WROOM-1-N8 DEV BRD do tipo Transceptor; 802.11 b/g/n (Wi-Fi, WiFi, WLAN), Bluetooth® 5, cuja frequência é de 2,4 GHz da fabricante Espressif Systems.
2. **TAGS:** as informações são armazenadas em um chip denominado de etiqueta, mais conhecido como Tag. Para que haja uma conexão entre a antena e a etiqueta, é necessário apenas que ambas estejam posicionadas dentro de uma certa distância e com um alinhamento adequado aos padrões para que foram projetados. No projeto, também será utilizado um ESP32-S3 como Tag.
3. **ROTEADOR Wi-Fi:** transmissor com alcance suficiente para criar a ligação entre todos os transmissores e a tag. Será usado o ESP32-S3.
4. **ACELERÔMETRO:** mede a aceleração (ou taxa de mudança de velocidade em relação ao tempo) de um corpo em seu próprio quadro de repouso instantâneo, não é o mesmo que aceleração de coordenadas, sendo a aceleração em um sistema de coordenadas fixas.
5. **BUZZER:** emissor de som de baixo custo, utilizado para reproduzir tanto efeitos sonoros simples, como também a capacidade de emitir sons mais complexos como músicas. Funciona com tensão entre 3,5 e 5V.
6. **LED:** após ser acionado por meio da interface, o LED ficará piscando para ajudar na visualização do objeto desejado em meio aos outros objetos.
7. **PROTOBOARD:** a matriz de contato será usada como suporte principal para as execuções dos circuitos elétricos do protótipo, como na montagem do Roteador Wi-Fi, do acelerômetro, do LED e do Buzzer, por exemplo. O Protoboard usado na solução é da Minipa; modelo MP-1680A, cuja tensão máxima é de 300V RMS e corrente de 3A RMS.
8. **RESISTOR:** componente que irá limitar o fluxo da corrente elétrica do circuito. Por meio do chamado efeito joule (w/s), ele é capaz de transformar a energia elétrica em energia térmica. Em suma, o dispositivo faz oposição à passagem da corrente elétrica, oferecendo resistência. Vale lembrar que o resistor usado para a montagem foi de 10kΩ (Ohms) filme de Carbono CR25 1/4W.
9. **JUMPER:** componente para a passagem de corrente elétrica do protoboard para outros materiais que estão no próprio protoboard. Foram usadas fiações (jumpers) dos três tipos: macho-macho; macho-fêmea e fêmea-fêmea.
10. **CABO USB:** usado cabo de Universal Serial Bus (USB) tipo C para conectar o Microcontrolador ao computador.

## 2.2. Componentes externos

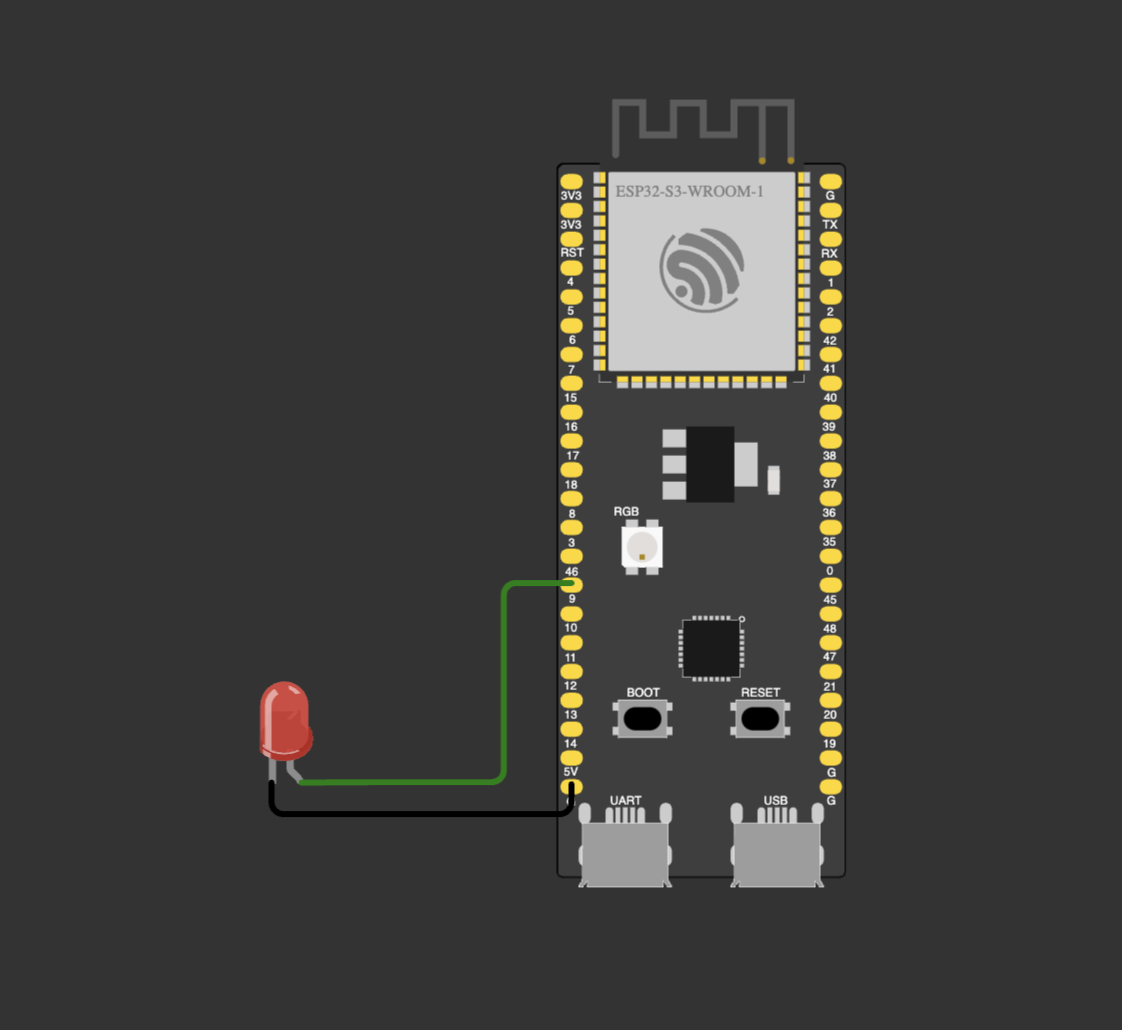
1. **COMPUTADOR:** será usado um computador para exploração dos dados que saíra pela interface, a qual foi desenvolvida usando React, HTML, CSS, Bootstrap e bibliotecas diversas.
2. **ROTEADOR WI-FI:** transmissor com alcance suficiente para criar a ligação entre todos os transmissores e a tag.
3. **SERVIDOR:** servidor em Node.js funcional no localhost e configurável para nuvem.
4. **BANCO DE DADOS:** usado o Schema noSQL com MongoDB, recomendado para aplicações escaláveis.
5. **ARDUINO IDE:** é usado para escrever e fazer upload de programas em placas compatíveis com Arduino, mas também, com a ajuda de núcleos de terceiros, outras placas de desenvolvimento de fornecedores.

## 2.3. Requisitos de conectividade

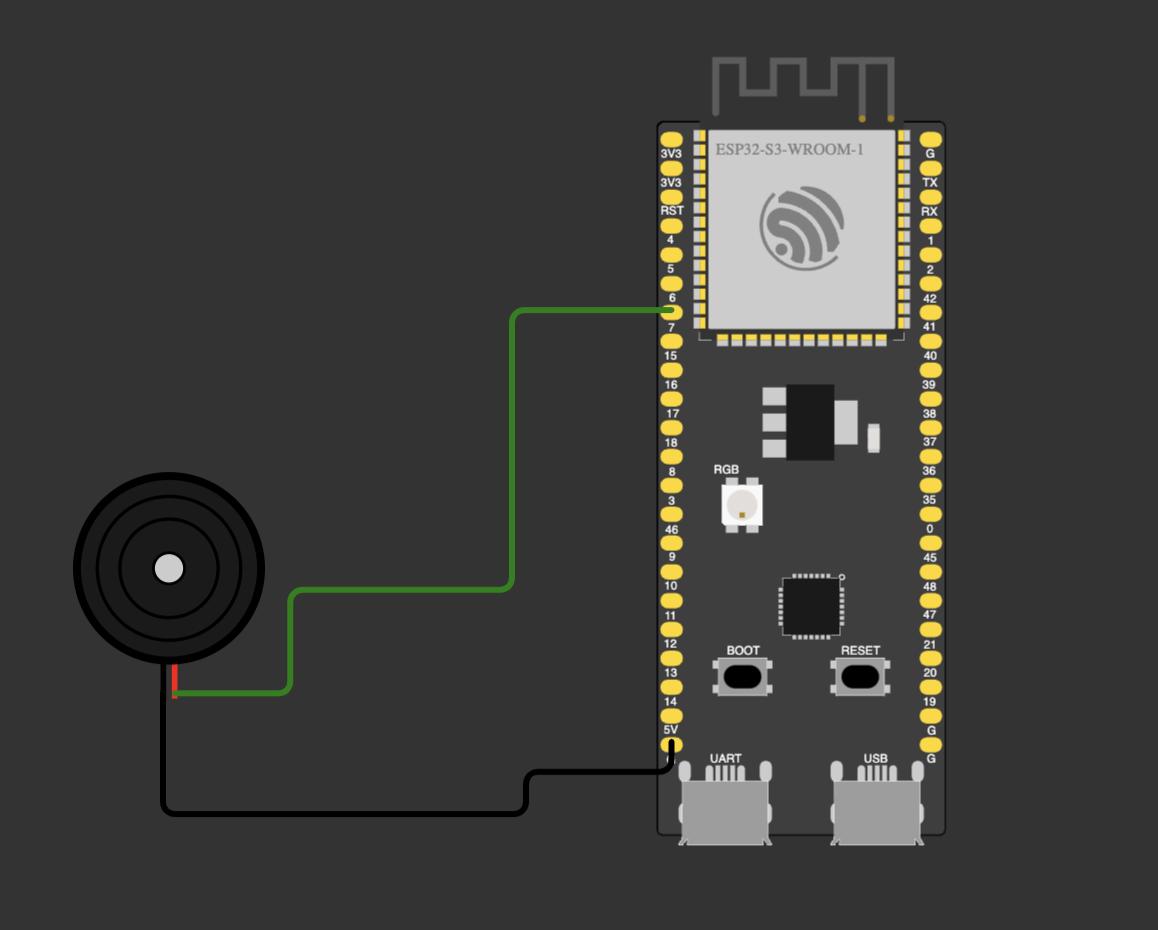
Um Roteador Wi-Fi padrão conectado à rede de internet, para a conexão entre as tags e o servidor. Já a conexão das tags com os beacons é realizada através de uma conexão direta (initiator - receiver). Os protocolos utilizados para a conexão com o servidor através da rede é o TCP/IP, assim como HTTP para os requests da aplicação e do sistema embarcado.

# 3. Guia de Montagem

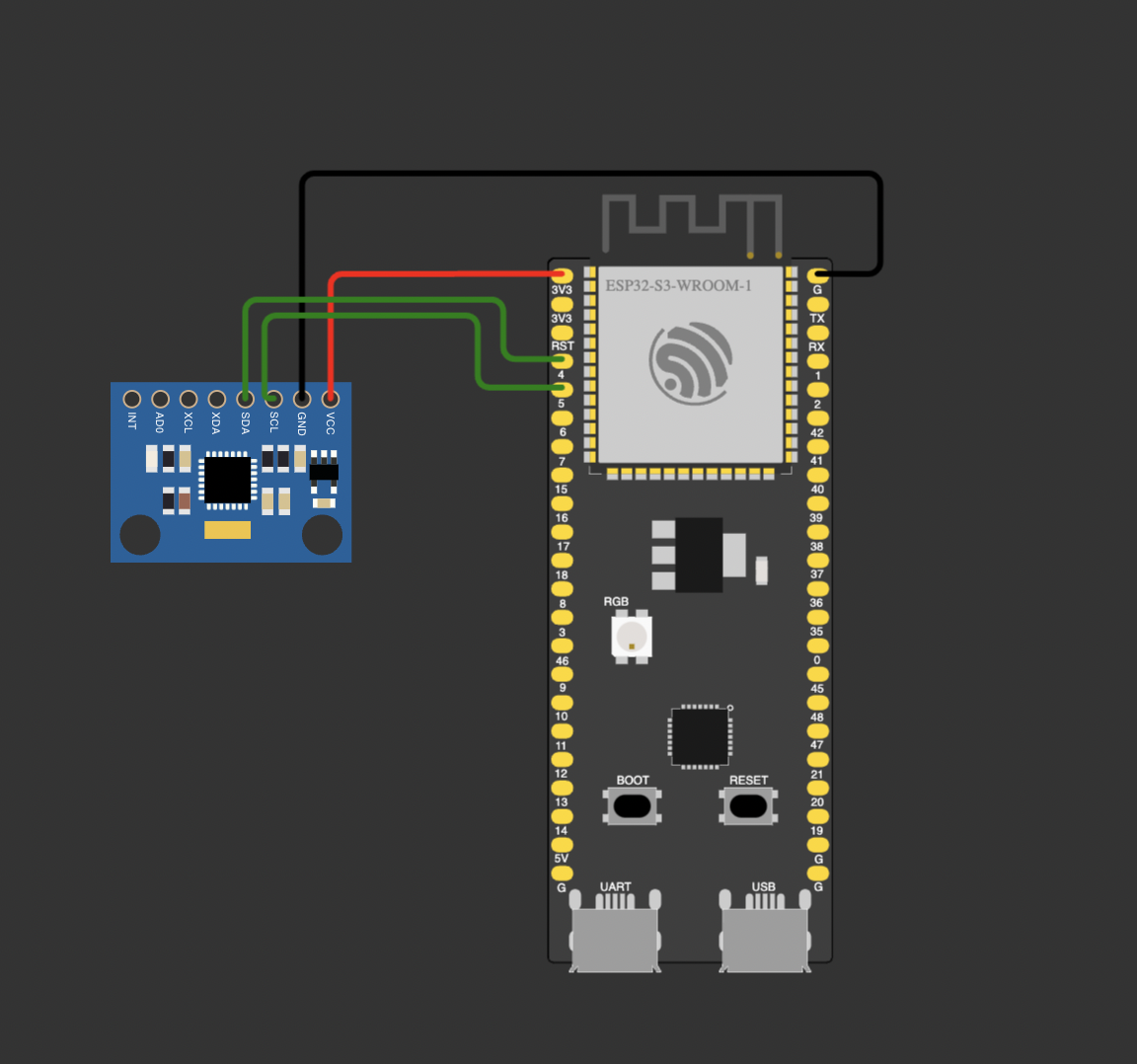
1. Será necessário um led. Para instalá-lo colocaremos um lado do jumper macho-macho na porta 9 do ESP32-S3 e o outro em um resistor de 220K ohms que será conectado no polo positivo("perna" maior) do led. Já o polo negativo("perna menor") será conectado no ground. (figura 1)
2. Será necessário um buzzer. Para instalá-lo colocaremos um lado do jumper macho-macho na porta 7 do ESP32-S3 e o outro que será conectado no polo positivo(representado por um símbolo de "+") do buzzer. Já o polo negativo(lado oposto do "+") será conectado no ground. (figura 2)
3. Será necessário um acelerômetro. Para instalá-lo colocaremos um lado do macho do jumper macho-fêmea na porta 3V3 do ESP32-S3 e o Fêmea no pino VCC do acelerômetro, depois conectamos outro jumper macho-fêmea, colocando o macho na porta 5 e o fêmea no pino SCL do acelerômetro, após, conectamos mais um jumper macho-fêmea, sendo que o macho irá na porta 4 e o fêmea no pino SDA, por fim 2usaremos um último jumper macho-fêmea, será conectado o macho no ground do ESP32-S3 e o fêmea no pino ground do acelerômetro.(figura 3)



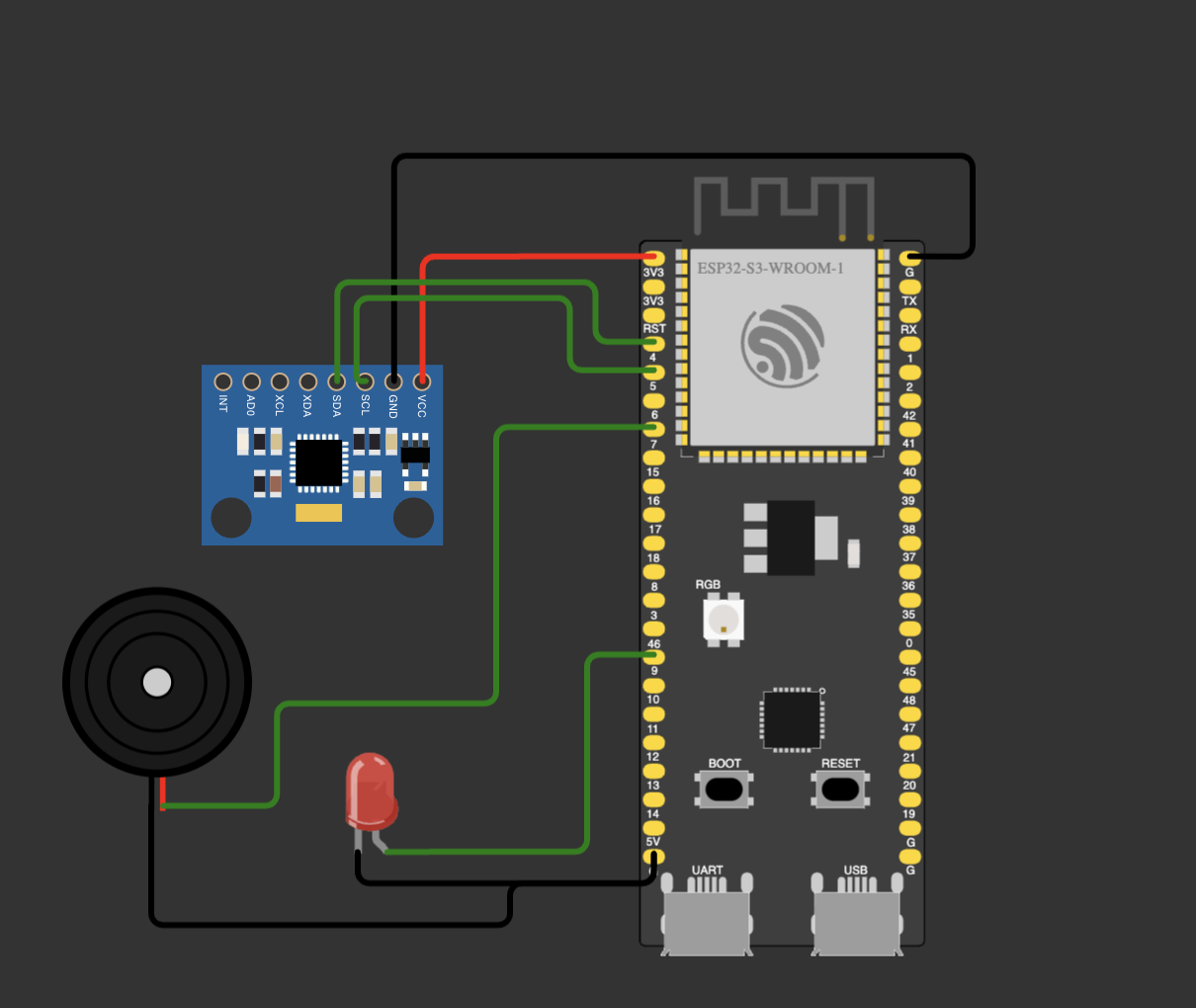














# 4. Guia de Instalação

1. Após completar a montagem do ESP32-S3 (seção 3), meça, dentro do espaço indoor, um ponto exato onde os Beacons deverão ser posicionados - vale lembrar que será usado o método da triangulação. Essa medida é calculada, em metros, do Beacon 1 para o Beacon 2 - medida representada pela incógnita Y - e do Beacon 1 para o Beacon 3 - medida representada pela incógnita X. Obs: o ponto de partida foi configurado para ser Beacon 1, cuja posição é de (0,0). Aqui, usaremos as medidas de 8 metros para Y e de 13 metros para X.

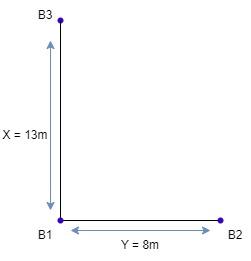


Imagem 4.1: B1 = Beacons 1; B2 = Beacons 2; B3= Beacons 3. Obs; escala de medidas não proporcional à medida real do ambiente.

1. Em seguida, coloque nos pontos medidos os microcontroladores os quais devem estar conectados em uma fonte de energia. Obs: pode ser uma conexão com uma tomada ou uma bateria portátil. Ambas devem ter uma fonte de energia de no máximo de 5V.
2. Vá no ícone de bibliotecas à direita da tela Instale e Inclua as bibliotecas que já vem no próprio software do Arduino IDE como, por exemplo, “*Wifi.h*”, *“HTTPClient.h*”, “*SparkFun\_MMA8452Q.h*”, “Wire.h” e “ *ArduinoJson.h*”.
3. Após configurar e elaborar a codificação, conecte a Tag no computador. Abra a plataforma do Arduino IDE com o código configurado (detalhes de configuração na seção 5).

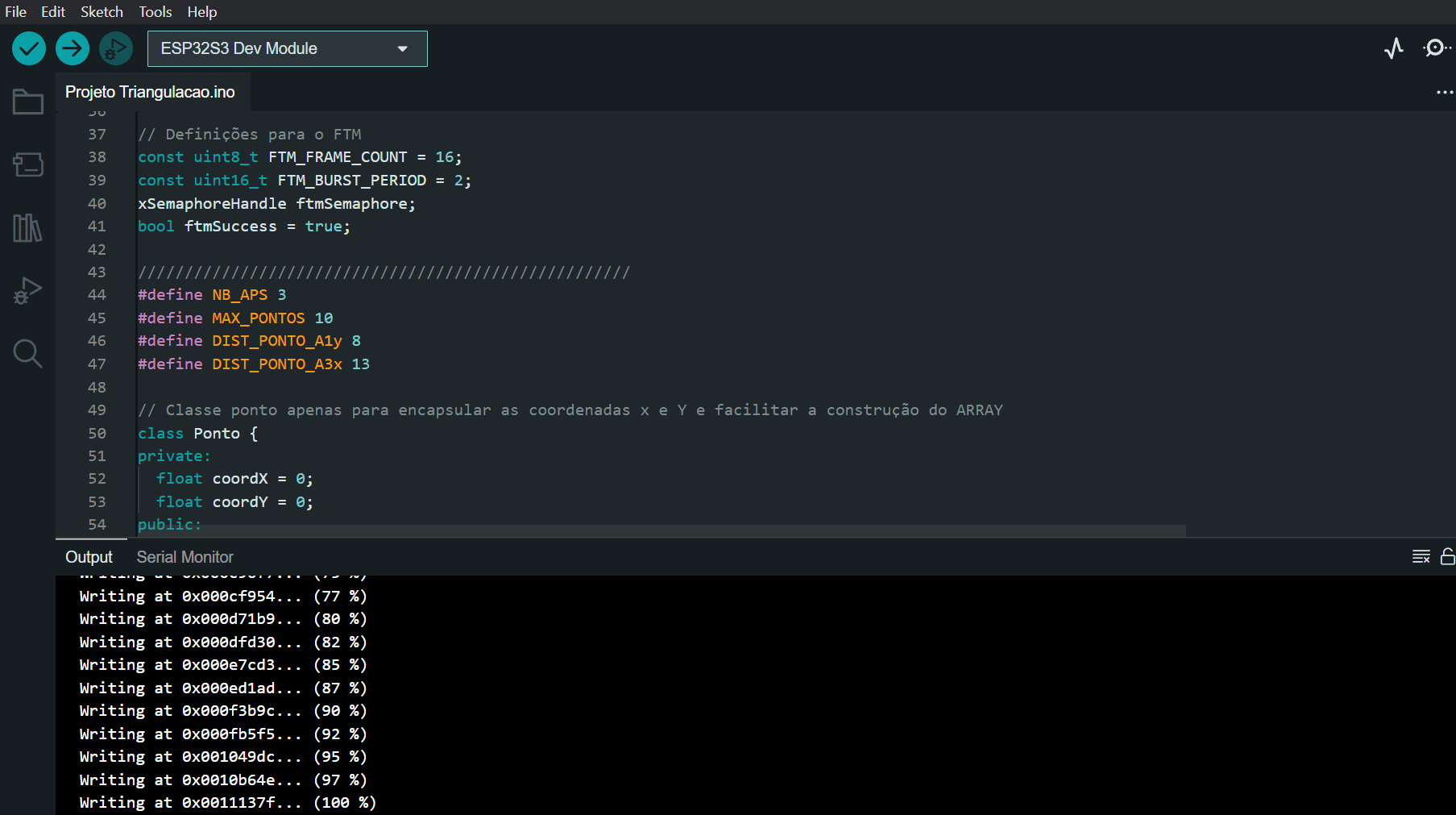


Imagem 4.2: código configurado e compilado com as respectivas funções das medidas das distâncias do Beacons.

1. Após a compilação, vá ao Monitor Serial no canto da tela em cima à esquerda (ícone de uma lupa) para ver o resultado da compilação.
2. Posteriormente, observa-se que o resultado da posição da Tag para cada Beacon é diferente.

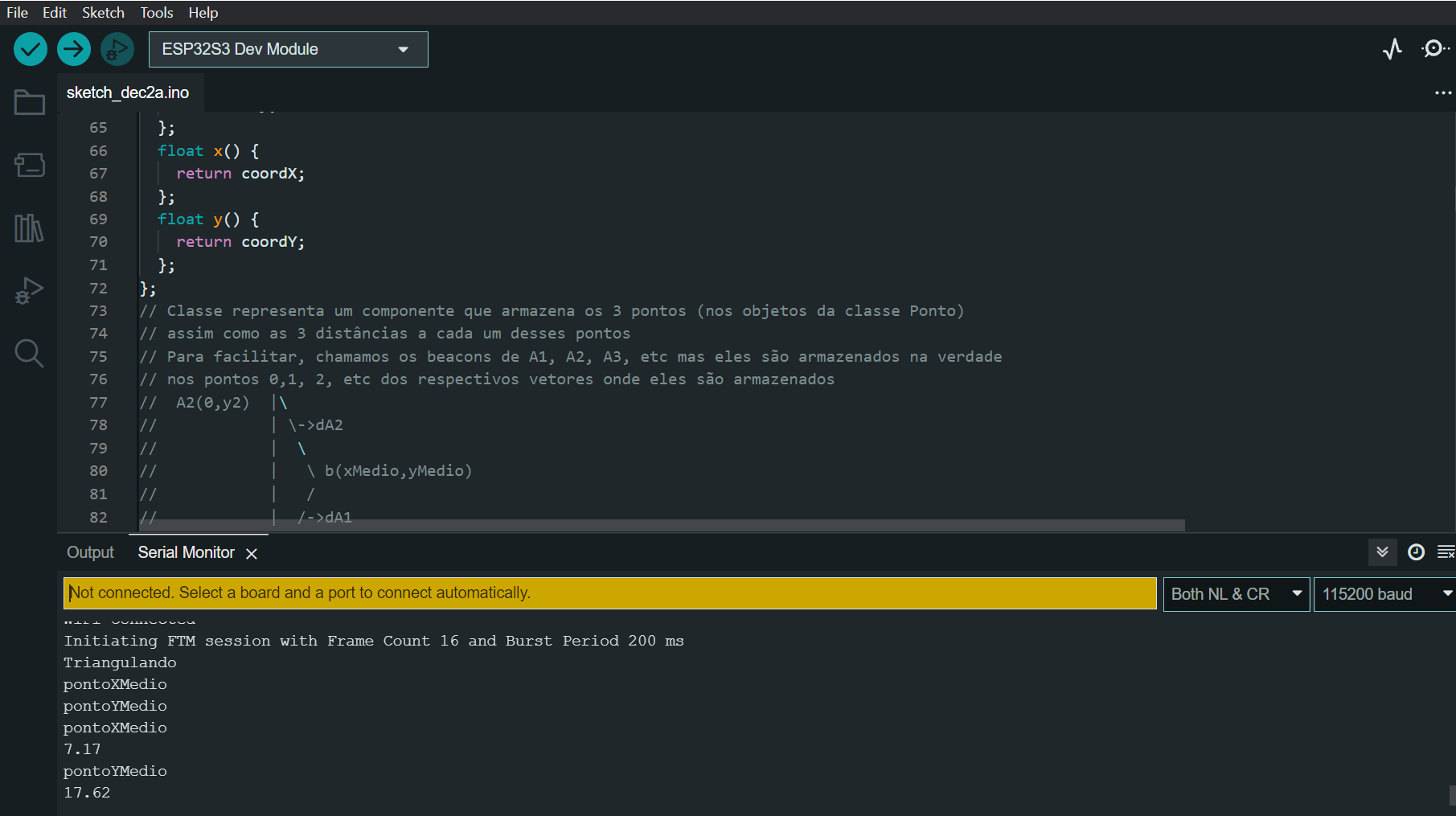


Imagem 4.3: localização da Tag sendo exibida pela distância dela em relação aos Beacons.

E na medida que a Tag se move, os pontos médios vão mudando proporcionalmente às distâncias da localização dos Beacons. Além do mais, a triangulação é dada pelo método do teorema de Pitágoras ***a² + b² = c²***  (imagem 4.5).

1. Obs: no PontoYMedio, observa-se uma variância brusca no valor, pois ao testar o dispositivo, o ambiente apresentava interferência de conexão na rede; e isso prejudicou relativamente no resultado esperado da localização.
2. Obs: considere as coordenadas da Tag como b(xMédio, yMédio) no ponto vermelho em destaque.

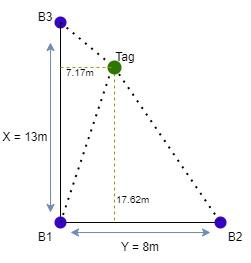


Imagem 4.4: ponto médio da Tag em relação às incógnitas.

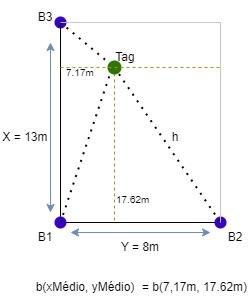


Imagem 4.5: considere h como a hipotenusa.

# 5. Guia de Configuração.

**INFORMAÇÃO INICIAL:**

* Nesse protótipo, não existe uma tela separada que faça a configuração do dispositivo.

1. Primeiramente, de antemão, para configurar a Tag, é necessário definir a rede do roteador e as redes de cada um dos Beacons. Por exemplo, aqui usamos como roteador e a sua senha, respectivamente: “Inteli-COLLEGE” e “QazWsx@123”; e os Beacons como Beacon 1, Beacon 2 e Beacons, respectivamente, como: ”beacon1” para a senha “beacon123”; ”beacon2” para a senha “beacon123; ”beacon3” para a senha “beacon123.

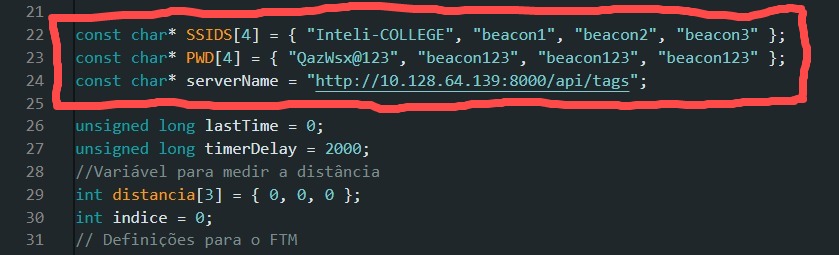


Imagem 5.1: Definindo o roteador, os Beacons e suas senhas

1. Na imagem 5.2, também é necessário definir o ServerName, que é o caminho do servidor, no qual está hospedado o Back-End, adicionando também o api/tag, o qual usaremos como endpoint para mandar os dados para o servidor.
2. Após configurar as redes da Tag é necessário subir o código para a Tag.
3. Configure individualmente os Beacons para conectá-los à rede.

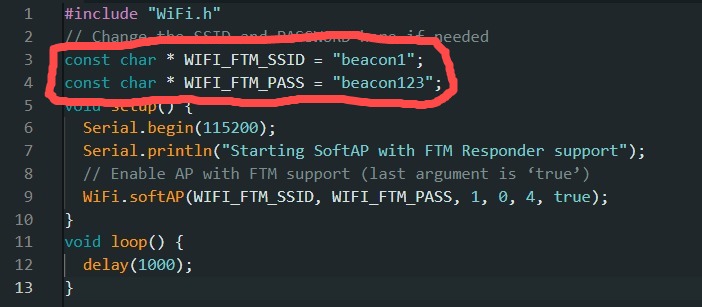
****

Imagem 5.2: configurando os 3 Beacons. Para cada Beacon, um código, pois cada um deverá ter na sua configuração um SSID e uma senha distintas.

1. Configure a rede individual de cada Beacon. Dever-se-á subir o código individualmente para cada um.
2. Conectou? Agora instale os Beacons no espaço indoor (mais detalhes de instalação na seção 4 - Guia de Instalações).

# 6. Guia de Operação

### (sprint 5)

Descreva os fluxos de operação entre interface e dispositivos IoT. Indique o funcionamento das telas, como fazer leituras dos dados dos sensores, como disparar ações através dos atuadores, como reconhecer estados do sistema.

Indique também informações relacionadas à imprecisão das eventuais localizações, e como o usuário deve contornar tais situações.

Utilize fotografias, prints de tela e/ou desenhos técnicos para ilustrar os processos de operação.

# 7. Troubleshooting

### (sprint 5)

Liste as situações de falha mais comuns da sua solução (tais como falta de conectividade, falta de bateria, componente inoperante etc.) e indique ações para solução desses problemas.

| **#** | **Problema** | **Possível solução** |
| --- | --- | --- |
| 1 | Variância brusca na medição das distâncias entre a Tag e os Beacons em repouso. |  |
| 2 | Interferências |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

# 8. Créditos

O Grupo Sauron é composto por 7 desenvolvedores: Alysson Cordeiro; Bruno Leão; Iago Tavares; Felipe Saadi; Luiz Ferreira; Luiz Carlos Jr e Marcos Florêncio.

Este manual está protegido pelas leis internacionais dos direitos autorais, conforme a Lei N° 9.610. de fevereiro de 1998.

Copyright © 2022 Instituto de Tecnologia e Liderança (INTELI) - Ateliê 3/Grupo Sauron.

