



Manual de Instruções

Wall-Y
ATech

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
20/11/2022	Stefano Butori	1.0 1.1 1.2	Atualização das sessões 1,2 e 3
1/12/2022	Eric Tachdjian Rafael Moritz	1.2	Atualização das sessões 4 e 5
13/12/2022	Rodrigo Martins Mateus Almeida Stefano Butori	1.3	Atualizações das sessões 6,7 e 8
16/12/2022	Eric Tachdjian Rafael Moritz	1.4	Correção da sessão 4

Índice

1. Introdução	3
1.1. Solução	3
1.2. Arquitetura da Solução	3
2. Componentes e Recursos	4
2.1. Componentes de hardware	4
2.2. Componentes externos	4
2.3. Requisitos de conectividade	4
3. Guia de Montagem	5
4. Guia de Instalação	6
5. Guia de Configuração	7
6. Guia de Operação	8
7. Troubleshooting	9
8. Créditos	10

1. Introdução

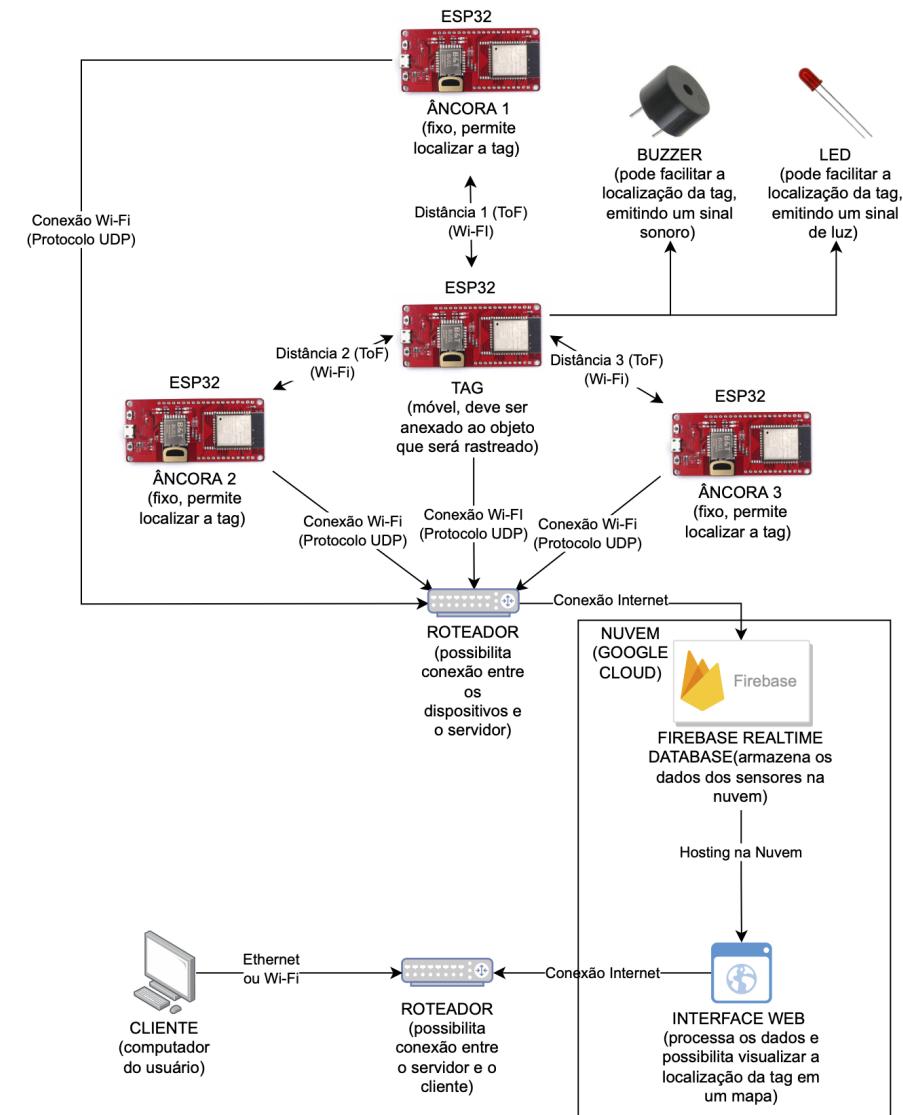
1.1. Solução

Vamos elaborar um sistema capaz de localizar em um ambiente interno (“indoor”) qualquer objeto que possua uma tag acoplada, através de um algoritmo de trilateração

Essa solução pode eventualmente proporcionar uma forma de rastrear objetos e pessoas em tempo real dentro de depósitos e galpões, através de um software simples com design intuitivo.

Primeiramente, o usuário deve colocar tag no objeto que deseja rastrear. Após isso, ele deve acessar a aplicação que será usada para o rastreamento. Ela mostrará um mapa do local, juntamente com um ponto referente à cada objeto que possua uma tag. Dessa forma, o usuário poderá encontrar facilmente o que quiser.

1.2. Arquitetura da Solução



Código (s)	Componente / Conexão	Descrição da função	Tipo: entrada / saída
ÂNCORA 1 ÂNCORA 2 ÂNCORA 3	3 x Microcontrolador ESP32 com módulo Wi-Fi - Modo Âncora	Dispositivos que são fixados em um ponto específico e não se movimentam. São usados para monitorar o ambiente e determinar a posição de um ou mais tags, através da tecnologia Wi-Fi.	entrada, responde a tag com um sinal Wi-Fi para consiga estimar a localização usando a tecnologia Fine Time Measurement (FTM Responder)
TAG	1 x Microcontrolador ESP32 com módulo Wi-Fi - Modo Tag	Dispositivo móvel, que pode transitar pelo ambiente. Recebe sinais e envia respostas para os dispositivos âncora via Wi-Fi. Através da comparação entre o período de tempo que leva para um pacote de dados ir e voltar (Time of Flight - "ToF") é possível calcular a distância entre a Tag e cada âncora e descobrir a posição exata do dispositivo por triangulação.	entrada, emite um sinal Wi-Fi para uma âncora a fim de estimar a localização usando a tecnologia Fine Time Measurement (FTM Initiator)
BUZZER	1 x Buzzer Ativo 5v 12mm	Buzzer que emite som quando ativado, para ajudar na localização do dispositivo "tag", caso a interface não seja suficiente.	saída, emite um som

LED	1 x Led Difuso 5mm	Led que pisca quando ativado, para ajudar na localização do dispositivo "tag", caso a interface não seja suficiente.	saída, emite uma luz piscando
FIREBASE RTDB	1 x Banco de Dados no Google Firebase (Nuvem)	Banco de dados que recebe os dados das âncoras via Wi-Fi	entrada (backend), ao receber os dados da tag e das âncoras e saída (frontend), ao retornar uma página HTML para o cliente
FRONTEND	1 x Página Web Hospedada no Google Firebase (Nuvem)	Frontend na nuvem que irá processar o sinal dos sensores, executando as operações necessárias para calcular a localização do tag (através de um algoritmo de triangularização) e mostrar para o usuário (através de uma interface web em JavaScript / HTML)	saída, fornece uma página HTML para o cliente
ROTEADOR	1 x Dispositivo de Rede do Roteador	Roteador Wi-Fi, utilizado para enviar os dados do servidor para uma rede local ou para a internet	entrada, ao receber dados via Wi-Fi e saída, ao responder com dados para outros dispositivos

2. Componentes e Recursos

2.1. Componentes de hardware

Componentes necessários:

4 x microcontrolador modelo ESP32S3 (3 âncoras, 1 tag)

1 x Protoboard

1 x Buzzer Ativo 5v 12mm

1 x Led Difuso 5mm

1 x cabo USB para USB-C (para configurar os microcontroladores a partir do computador)

1 x Power Bank USB (para a tag)

3 x Fonte USB (para as âncoras)

2.2. Componentes externos

1 x Servidor Web na nuvem para o frontend, usamos o Firebase Hosting do Google

1 x Banco de Dados em Tempo Real na nuvem, usamos o Firebase RTDB do Google

1 x Cliente, em um Computador Desktop ou Notebook (para acessar o servidor Web), com sistema operacional Linux ou Windows.

Esse computador do Cliente também pode ser usado para programar os microcontroladores, com a instalação da IDE Arduino versão 1.8 ou superior.

2.3. Requisitos de conectividade

1 x Roteador Wi-Fi (para possibilitar a conexão entre a nuvem e os microcontroladores, bem como entre cliente e servidor), compatível com os protocolos TCP/IP e HTTP e com acesso à Internet.

3. Guia de Montagem

1. Conecte um dos microcontroladores ESP32S3 através do cabo USB para USB-C e baixe o código da “tag” disponível na pasta “src” do GitHub do projeto. Utilize a IDE Arduino instalada no computador para fazer o upload desse código para o dispositivo ESP32 “tag”.
2. Conecte os demais microcontroladores ESP32S3 através do cabo USB para USB-C e baixe o código da “beacon” disponível na pasta “src” do GitHub do projeto. Utilize a IDE Arduino instalada no computador para fazer o upload desse código para os dispositivos ESP32 “beacon”.
3. O dispositivo “tag” deve ser montado em um protoboard com o LED e o Buzzer.
4. O dispositivo tag deve ser conectado em um Power Bank USB, para que possa ser deslocado no ambiente e ter uma autonomia de bateria.

5. Os dispositivos “âncora” devem ser encaixados em uma tomada (ou computador) com uma fonte de alimentação e dispostos de forma espalhada em pontos pré-determinados no ambiente.
6. Uma vez que essa configuração tenha sido realizada, suba o Servidor Web no Firebase e realize uma requisição através da “tag”, que retornará a distância. Para mais informações, veja as seções a seguir.

4. Guia de Instalação

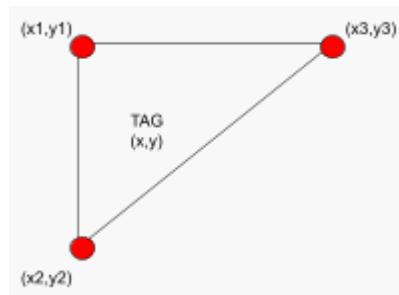
1. Primeiramente é necessário conectar os beacons e a tag na rede Wi-Fi. Para isso, basta conectar cada microcontrolador colocando o nome e a senha da rede em cada código individualmente.

```
// ----- Wi-Fi -----  
  
#define routerSSID "PLACE_ROUTER_SSID_HERE" // Definir essa constante como o SSID da rede do roteador local  
#define routerPSW "PLACE_ROUTER_PASSWORD_HERE" // Definir essa constante como a senha da rede do roteador local
```

2. Também é necessário conectar ambos (beacons e tags) no firebase:

```
// ----- Firebase -----  
  
#define API_KEY "FIREBASE_API_KEY" // Definir essa constante como a chave da API do RTDB do Firebase  
#define DATABASE_URL "FIREBASE_DATABASE_URL" // Definir essa constante como a URL do RTDB do Firebase
```

3. Uma vez montado e configurado, é necessário posicionar as “âncoras”. Para posicioná-las, é necessário montar uma espécie de triângulo para executar a trilateração. O posicionamento exato pode ser definido pelo usuário, assim como mostrado abaixo. A localização do beacon não pode ter mais de 40m de distância do centro da sala/armazém. A distância entre os beacons não pode ser maior que 80m.



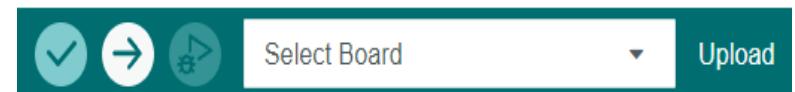
4. Agora é necessário posicionar a tag no ativo que será rastreado, conectando-a à uma fonte de energia, seja ela um powerbank ou uma tomada, utilizando o cabo USB da tag. Uma vez configurado e conectado corretamente, a localização do ativo encontrada pelos beacons vai ser enviada ao firebase.

5. O buzzer, pode ser instalado utilizando a porta informada no código (Porta 2) e é necessária a instalação para o funcionamento dessa função no Front-End.
6. O alcance efetivo do equipamento é de até 45 metros (O alcance pode ser menor caso existam muitos objetos no caminho e nem todos os beacons precisam estar dentro desse alcance para que a localização funcione).

5. Guia de Configuração

Como configurar os beacons e tags:

1. Abrir o código fonte na aplicação Arduino IDE
2. Conectar o ESP32 ao computador usando um cabo USB ligado à porta de comunicação (“COM”)
3. Clicar na seta “upload” no canto superior esquerdo da tela e esperar a instalação terminar.



Como cadastrar um beacon:

1. Abrir o servidor Web

2. Abrir a sidebar e selecionar “Beacons”



Local	Posição X (m)	Posição Y (m)	
ateli2	3	7	  
home	5	2.5	 
ateli3	7	6	 
ateli3	2	1	 
ateli3	8	1	 



3. Preencher o formulário com a posição (x,y) e as informações do Wi-Fi (SSID e senha) e clicar em “salvar”

Como cadastrar um local:

1. Abrir a sidebar e selecionar “Local”





2. Preencher o formulário com o ID, descrição e o endereço MAC dos três beacons que estarão no local e clicar em “salvar”

Ativo 2	Beacon 3	Tamanho (X)	Tamanho (Y)		
A:9P	Z3:43:8L	15	9	Salvar	
FA:E2:S2:	F4:12:FA:E2:05: 94	10	8		
B1-35-	D0-5F-9D-38-EA-3B	5	5		

Como cadastrar uma tag:

1. Abrir a sidebar e selecionar “Tags”



2. As tags são criadas automaticamente, então basta clicar em editar e trocar o nome para o do ativo em questão



1. Quando se cadastra um beacon, é preciso cadastrar também o nome e a senha do Wi-Fi do dispositivo.

Lista de Beacons					
ID	SSID	Senha	Cor	Local	Posição X (m)
A1:B2:C3:D4	beacon3_at	senha	Cor	Local	X (m)
D0-5F-9D-3B-EA-3B	esp32-beacon3	12345678ab	#00F	home	5
F4:12:FA:E2:05:94	beacon3_ateli	12345678ab	#000	atelier3	7
F4:12:FA:E2:12:B0	beacon1_ateli	12345678ab	#000	atelier3	2
F4:12:FA:E2:52:8C	beacon2_ateli	12345678ab	#000	atelier3	8

6. Guia de Operação

Operação dos ESP32s (Arduino IDE)

Após ter os códigos que serão usados no ESP32 (beacon / tag) em sua máquina, o usuário deve registrar os dados da rede local e do Firebase nos arquivos, antes de fazer upload para os dispositivos.

Como cadastrar o Wi-Fi:

```

tag.ho
1 // ----- BIBLIOTECAS -----
2
3 #include <String.h>
4 #include <string.h>
5 #include <bits/stdc++.h>
6 #include <WiFi.h>
7 #include <Firebase_ESP_Client.h>
8 #include "addons/tokenhelper.h"
9 #include "addons/RTDBHelper.h"
10 #include <Adafruit_NeoPixel.h>
11
12 // ----- CONST -----
13
14 // ----- ESP32 -----
15
16 #define BUZZER 2
17 #define PIN 48
18 #define NUMPIXELS 1
19
20 // ----- Wi-Fi -----
21
22 #define routerSSID "ROUTER_SSID" // Definir essa constante como o SSID da rede do roteador local
23 #define routerPSW "ROUTER_PASSWORD" // Definir essa constante como a senha da rede do roteador local
24
25 // ----- Firebase -----
26
27 #define API_KEY "FIREBASE_API_KEY" // Definir essa constante como a chave da API do RTDB do Firebase
28 #define DATABASE_URL "FIREBASE_RTDB_URL" // Definir essa constante como a URL do RTDB do Firebase
29
30 // ----- FTM -----
31
32 // Configurações
33 const uint8_t FTM_FRAME_COUNT = 16;

```



Nas constantes “routerSSID” e “routerPSW”, o usuário deve substituir, respectivamente, os placeholders “ROUTER_SSID” e “ROUTER_PASSWORD” pelo SSID e pela senha do roteador local, o qual proverá conexão à internet para os ESPs.

```

tag.ho
1 // ----- BIBLIOTECAS -----
2
3 #include <String.h>
4 #include <string.h>
5 #include <bits/stdc++.h>
6 #include <WiFi.h>
7 #include <Firebase_ESP_Client.h>
8 #include "addons/tokenhelper.h"
9 #include "addons/RTDBHelper.h"
10 #include <Adafruit_NeoPixel.h>
11
12 // ----- CONST -----
13
14 // ----- ESP32 -----
15
16 #define BUZZER 2
17 #define PIN 48
18 #define NUMPIXELS 1
19
20 // ----- Wi-Fi -----
21
22 #define routerSSID "ROUTER_SSID" // Definir essa constante como o SSID da rede do roteador local
23 #define routerPSW "ROUTER_PASSWORD" // Definir essa constante como a senha da rede do roteador local
24
25 // ----- Firebase -----
26
27 #define API_KEY "FIREBASE_API_KEY" // Definir essa constante como a chave da API do RTDB do Firebase
28 #define DATABASE_URL "FIREBASE_RTDB_URL" // Definir essa constante como a URL do RTDB do Firebase
29
30 // ----- FTM -----
31
32 // Configurações
33 const uint8_t FTM_FRAME_COUNT = 16;

```



Já nas constantes “API_KEY” e “DATABASE_URL”, o usuário deve substituir, respectivamente, os placeholders “FIREBASE_API_KEY” e “FIREBASE_RTDB_URL” pela chave da API e pela URL do Real Time

Database, ambos do Firebase, para que cada dispositivo seja capaz de ler e enviar informações para o banco de dados na nuvem.

Operação no Front-end (Browser)

Para acessar a aplicação web, o usuário deve acessar o link de deploy do front-end definido no Firebase. No caso do exemplo, o endereço é <https://atech-wally.web.app/>. Assim que acessar esse link, o navegador mostrará a tela de localização de tags no mapa.



Na tela “Localizar”, o usuário pode visualizar as tags (pontos vermelhos) e os beacons (pontos pretos) em determinado local, registrados no banco de dados (Firebase Real Time Database).

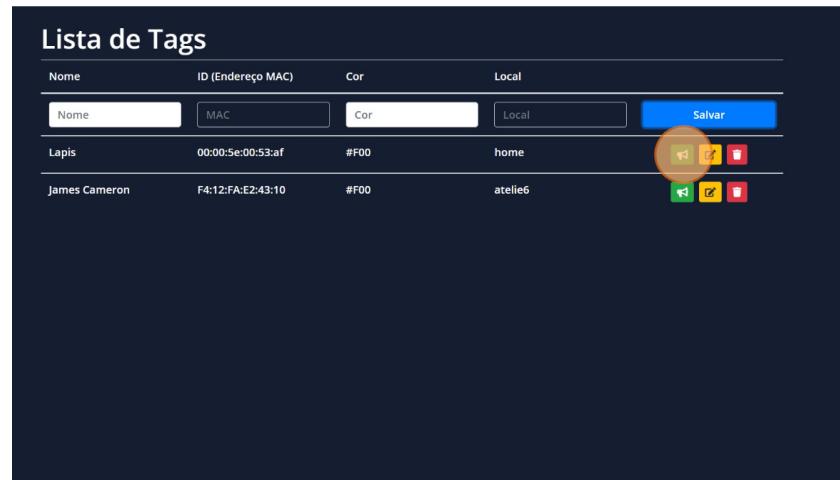


Pode-se mudar o local de visualização clicando no dropdown acima do mapa e selecionando o nome da sala.



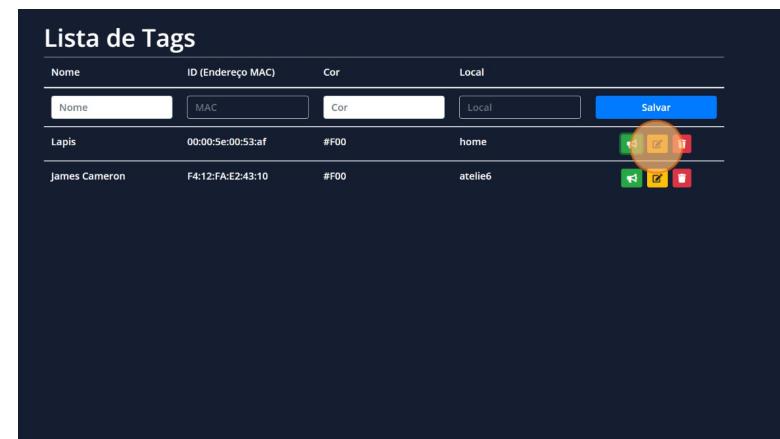
Para acessar a barra lateral e, assim, mudar para outra tela (beacons, tags e locais), deve-se clicar no menu, no canto superior esquerdo. No menu, o usuário escolhe uma das opções.

Na tela “Tags” o usuário tem várias opções de operação, que podem ser feitas em cada tag. Em suma, o usuário usa os botões localizados no canto esquerdo de cada linha (cada tag) da lista.



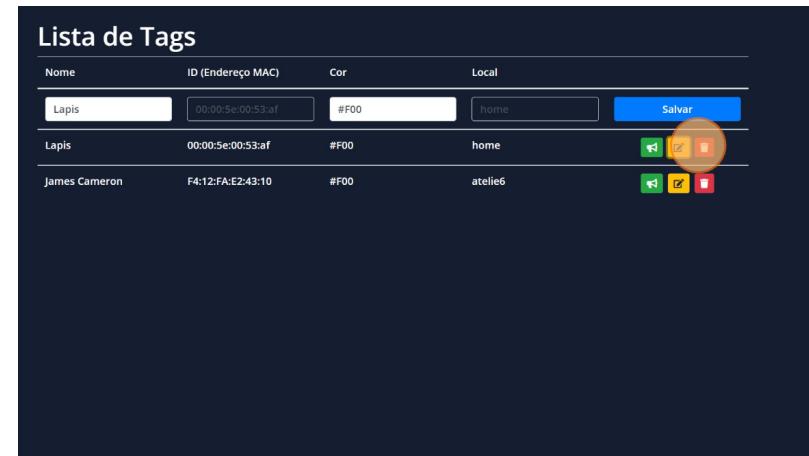
Nome	ID (Endereço MAC)	Cor	Local	
Nome	MAC	Cor	Local	Salvar
Lapis	00:00:5e:00:53:af	#F00	home	 
James Cameron	F4:12:FA:E2:43:10	#F00	atelie6	  

Para acionar os atuadores (BUZZER e LED) de uma determinada tag, o usuário precisa clicar no botão verde com ícone de megafone (indicado acima). Essa ação altera um valor no firebase, o qual será lido pela tag.



Nome	ID (Endereço MAC)	Cor	Local	
Nome	MAC	Cor	Local	Salvar
Lapis	00:00:5e:00:53:af	#F00	home	  
James Cameron	F4:12:FA:E2:43:10	#F00	atelie6	  

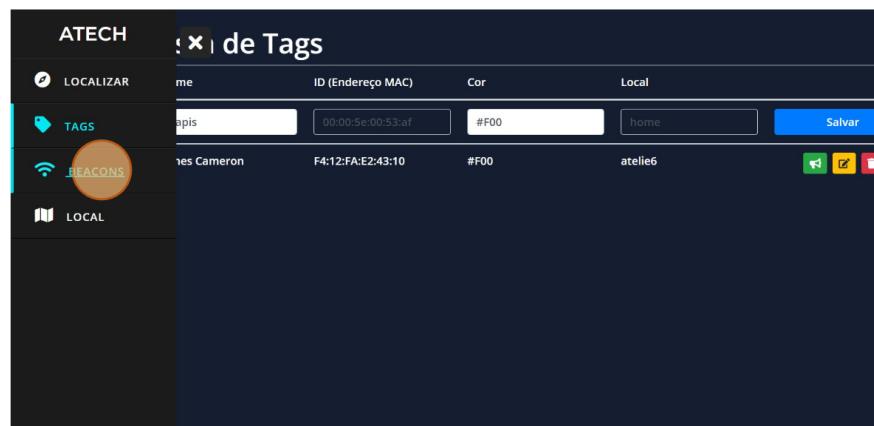
Para editar as informações de uma tag, o usuário seleciona o botão amarelo com ícone de lápis (indicado acima). As informações de endereço MAC e local de cada tag são definidas automaticamente. Dessa forma, o usuário altera somente o “nome” e a “cor” de exibição de cada um desses dispositivos.



Nome	ID (Endereço MAC)	Cor	Local	
Nome	MAC	Cor	Local	Salvar
Lapis	00:00:5e:00:53:af	#F00	home	  
James Cameron	F4:12:FA:E2:43:10	#F00	atelie6	  

Para excluir uma tag, o usuário deve clicar no botão vermelho com ícone de lixeira. Se um ESP com o mesmo endereço MAC estiver sendo usado como tag ele, mesmo já tendo sido excluído, enviará informações para o banco de dados e aparecerá na lista com nome e cor indefinidos .

Para acessar a tela de beacons, o usuário deve abrir o menu lateral e selecionar a opção “beacons”.



Nessa tela, pode-se adicionar, editar e excluir beacons.

Lista de Beacons							
ID	SSID	Senha	Cor	Local	Posição X (m)	Posição Y (m)	
F4:12:FA:E2:50:94	beacon3_ateli_e3	12345678ab	#000	atelie6	9	2.5	 
F4:12:FA:E2:12:B0	beacon1_ateli_e3	12345678ab	#000	atelie6	2.5	6.25	 
F4:12:FA:E2:52:8C	beacon2_ateli_e3	12345678ab	#000	atelie6	2.5	1.25	 

Para adicionar uma âncora, o usuário precisa fornecer todas as informações necessárias: ID (endereço MAC), SSID e senha da rede que será gerada pelo beacon, cor de exibição, local onde o beacon está e as posições X e Y (ponto onde o dispositivo está localizado na sala).

As operações de editar e excluir ocorrem de maneira similar com as tags, com a diferença de que, no processo de edição, o usuário pode mudar qualquer informação.

É importante que as informações de posição dos beacons sejam as mais precisas possíveis, para evitar erros no cálculo de localização da tag.

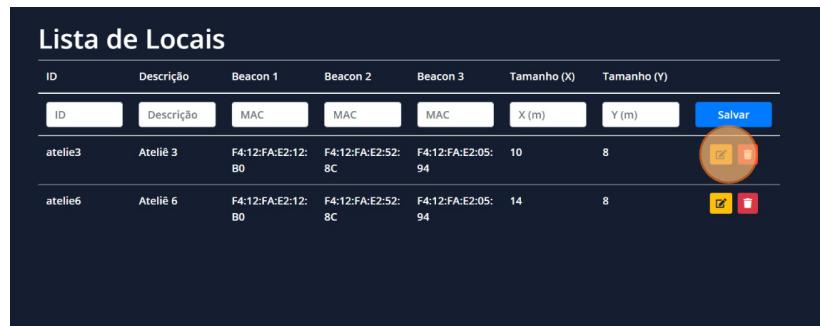
Para acessar a tela de locais, o usuário deve abrir o menu lateral e selecionar a opção “Local”.

As operações de editar e excluir ocorrem de maneira similar com a tela de beacon.

É importante que as informações dimensionais de cada local seja o mais preciso possível, para que possíveis erros de posicionamento dos beacons e, consequentemente, de localização das tags sejam evitados.



Nessa tela, pode-se adicionar, editar e excluir locais.



Para adicionar um local, o usuário precisa fornecer todas as informações necessárias: ID (único para cada local), descrição, endereços MAC de cada um dos três beacons que estarão posicionados na sala e as dimensões da área.

7. Troubleshooting

#	Problema	Possível solução
1	Falta de Conectividade na Tag com o Wi-Fi	Verifique se o nome da rede (SSID) e a senha no código da Tag são os mesmos da rede Wi-Fi que deseja conectar
2	Falta de Conectividade na tag com a Internet	Verifique se o roteador Wi-Fi está com o acesso funcionando na Internet
3	Falta de conectividade das Tags com os Beacons	Verifique se os beacons estão em uma distância acessível e cadastrados no banco de dados
4	Falta de energia nos Beacons ou Tags	Verifique se os beacons e tags estão carregados em uma fonte

		de energia (tomada ou Powerbank)
5	A posição da Tag não está correta no mapa	Verifique se as tags estão em uma distância adequada dos beacons e não há obstáculos no caminho

8. Créditos

Beatriz Hirasaki Leite - Frontend e Documentação

Eric Tachdjian - Frontend e Documentação

Mateus Soares de Almeida - Backend (ESP32) e Documentação

Rafael Lupovici Moritz - Backend (ESP32) e Documentação

Rodrigo Moraes Martins - Frontend e Documentação

Stefano Tosi Butori - Frontend e Documentação