

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
16/11/2022	João Pedro Carazzato	3.1	Preenchimento da seção 1, 2 e 3.
01/12/2022	João Pedro Carazzato	4.1	Preenchimento da seção 4 e 5.
14/12/2022	João Pedro Carazzato	5.1	Preenchimento da seção 6, 7 e 8. Revisão do documento.

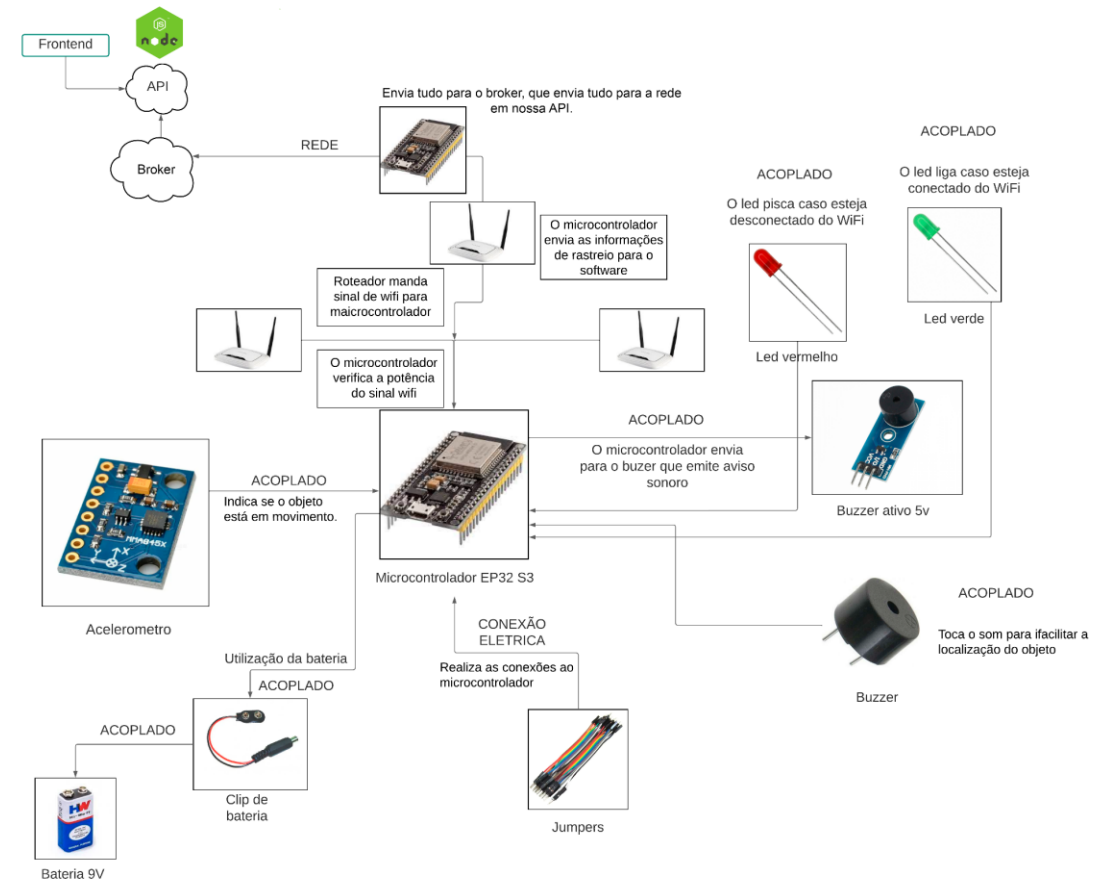
Índice

1. Introdução	3
1.1. Solução	3
1.2. Arquitetura da Solução	4
2. Componentes e Recursos	5
2.1. Componentes de hardware	5
2.2. Componentes externos	5
2.3. Requisitos de conectividade	5
3. Guia de Montagem	6
4. Guia de Instalação	7
5. Guia de Configuração	8
6. Guia de Operação	9
7. Troubleshooting	11
8. Créditos	12

1. Introdução

1.1. Solução (sprint 3)

A solução busca resolver o problema do nosso stakeholder IPT com a perda de seus aparelhos, para fazermos isso, criamos um localizador de equipamentos através do Wi-Fi local, funcionando através do acoplamento de nosso localizador no



equipamento enviando e obtendo informações através dos roteadores de Wi-Fi locais.

1.2. Arquitetura da Solução (sprint 3)

Coloque aqui o diagrama da arquitetura final da sua solução, o mesmo da seção 2.3 do IoTDoc (para imagens grandes, utilize o esquema da figura 2 abaixo). Utilize legendas e descrições para explicar sua imagem, seguindo a tabela produzida no IoTDoc.

Apresentamos todas as conexões necessárias para o funcionamento do nosso ESP32 e como cada uma funciona além de demonstrar a conexão da rede e sua função diante do backend, do frontend e do broker.

2. Componentes e Recursos (sprint 3)

2.1. Componentes de hardware

Componente / Conexão	Descrição da função
ESP 32 S3	Microcontrolador que irá gerenciar todas as entradas e saídas.
Buzzer ativo 5V	Emitir um som para localizar o objeto.
4 resistores: 2 de 5V e 2 de 12V	Limitar as tensões dos componentes.
10 jumpers	Conectar a tensão dos componentes.
Clips de bateria 9V	Conectar a bateria ao microcontrolador.
Bateria 9V	Fornecer energia ao sistema.
Giroscópio	Indicar se o objeto está em movimento.
1 botão 3.3V	Desligar o Buzzer
2 LEDs 5V	Indicar se o Wi-Fi está conectado ou não

2.2. Componentes externos

Componente / Conexão	Descrição da função
Dispositivo com acesso a internet	Onde todas as opções serão vistas pela parte do usuário
HiveMQ	Servidor de nuvem para conectar os dispositivos de IoT
Arduino IDE	Software de edição de código

2.3. Requisitos de conectividade

Para o funcionamento dos dispositivos, precisamos possuir acesso a internet no dispositivo e utilizar o processo de comunicação MQTT, após isso precisamos registrar o dispositivo no nosso banco de dados através do site para assim fazermos seu rastreamento.

3. Guia de Montagem

(sprint 3)

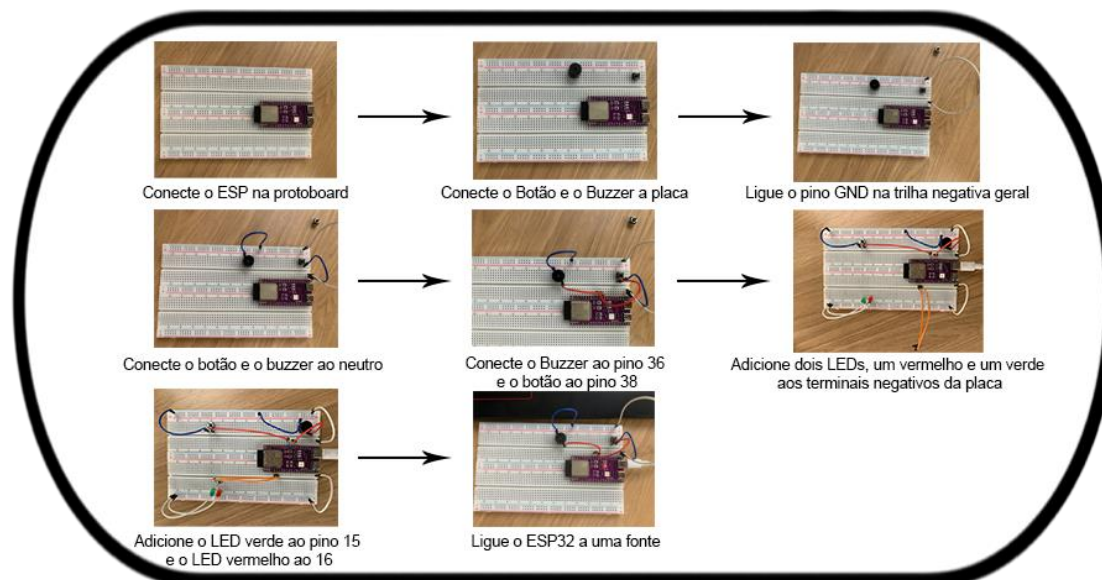


Figura 2: Processo passo a passo de como montar o sistema e deixá-lo funcionando

4. Guia de Instalação

(sprint 4)

1. Baixar a IDE do Arduino diretamente do site:

<https://www.arduino.cc/en/software>

Downloads



Arduino IDE 2.0.2

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

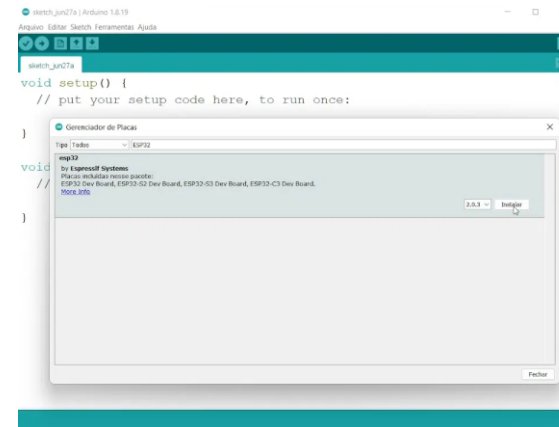
DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (386-64)
Linux ZIP file 64 bits (386-64)

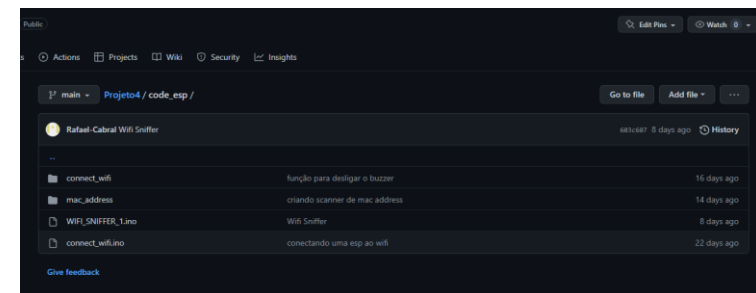
macOS 10.14 "Mojave" or newer, 64 bits

2. Baixar as dependências do ESP32 na IDE do Arduino: <https://embarcados.com.br/como-programar-o-esp32-na-arduino-ide/>

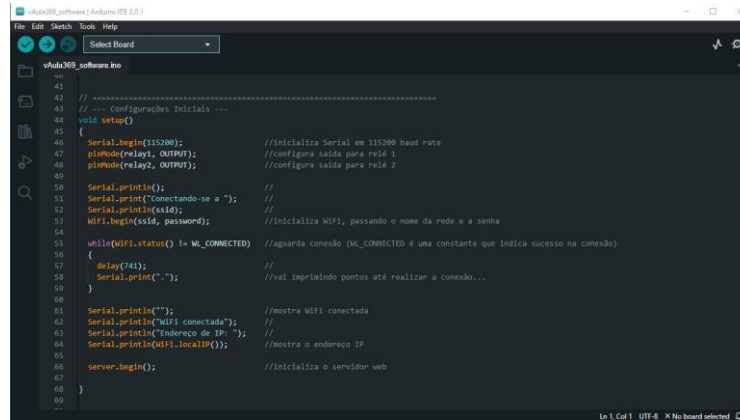


3. Baixar source do projeto através do github:

<https://github.com/2022M4T1-Inteli/Projeto4>



4. Abrir o código da source na IDE do Arduino:



```

41 // -----
42 // --- Configurações Iniciais ---
43 void setup()
44 {
45   Serial.begin(115200); //Inicializa Serial em 115200 baud rate
46   pinMode(relay1, OUTPUT); //configura saída para relé 1
47   pinMode(relay2, OUTPUT); //configura saída para relé 2
48
49   Serial.println(); //
50   Serial.print("Conectando-se a "); //
51   Serial.println(ssid); //
52   WiFi.begin(ssid, password); //Inicializa WiFi, passando o nome da rede e a senha
53
54   while(WiFi.status() != WL_CONNECTED) //aguarda conexão (WL_CONNECTED é uma constante que indica sucesso na conexão)
55   {
56     delay(750); //
57     Serial.print("."); //vai imprimindo pontos até realizar a conexão...
58   }
59
60   Serial.println(""); //mostra WiFi conectada
61   Serial.println("WiFi conectada"); //
62   Serial.println("Endereço de IP: "); //
63   Serial.println(WiFi.localIP()); //mostra o endereço IP
64
65   server.begin(); //inicializa o servidor web
66
67 }

```

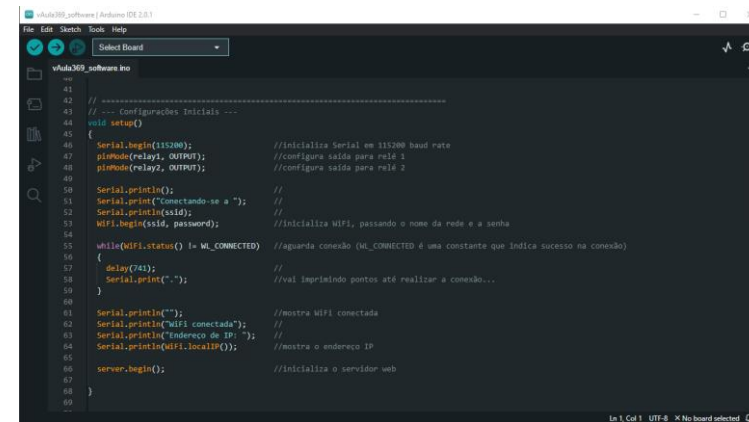
5. Efetuar o deploy para o ESP32 através do ícone de seta no campo superior da IDE:



5. Guia de Configuração

(sprint 4)

1. Abra o código do projeto na IDE do Arduino:



```

41 // -----
42 // --- Configurações Iniciais ---
43 void setup()
44 {
45   Serial.begin(115200); //Inicializa Serial em 115200 baud rate
46   pinMode(relay1, OUTPUT); //configura saída para relé 1
47   pinMode(relay2, OUTPUT); //configura saída para relé 2
48
49   Serial.println(); //
50   Serial.print("Conectando-se a "); //
51   Serial.println(ssid); //
52   WiFi.begin(ssid, password); //Inicializa WiFi, passando o nome da rede e a senha
53
54   while(WiFi.status() != WL_CONNECTED) //aguarda conexão (WL_CONNECTED é uma constante que indica sucesso na conexão)
55   {
56     delay(750); //
57     Serial.print("."); //vai imprimindo pontos até realizar a conexão...
58   }
59
60   Serial.println(""); //mostra WiFi conectada
61   Serial.println("WiFi conectada"); //
62   Serial.println("Endereço de IP: "); //
63   Serial.println(WiFi.localIP()); //mostra o endereço IP
64
65   server.begin(); //inicializa o servidor web
66
67 }

```

2. Alterar as informações das linhas 8, 9 e 10 remetentes ao nome do WiFi, sua senha e o servidor do Broker para as de uso próprio:

```

7 PubSubClient mqttClient(wifiClient);
8 const char* ssid = "Inteli-COLLEGE";
9 const char* password = "QazWsx@123";
10 char* mqttServer = "31632e1a776b4e068513a22883b3537b.s1.eu.hivemq.cloud";
11 int mqttPort = 8883;

```

3. Altere também o login do Broker para o de uso próprio localizado na linha 23 em ordem de nome e senha:

```
if (mqttClient.connect(clientId.c_str(), "Inteli-iot-teste", "V.5!RHZ4a@3UGct")) {
  Serial.println("Connected.");
  // subscribe to topic
  mqttClient.subscribe("esp32/message");
}
```

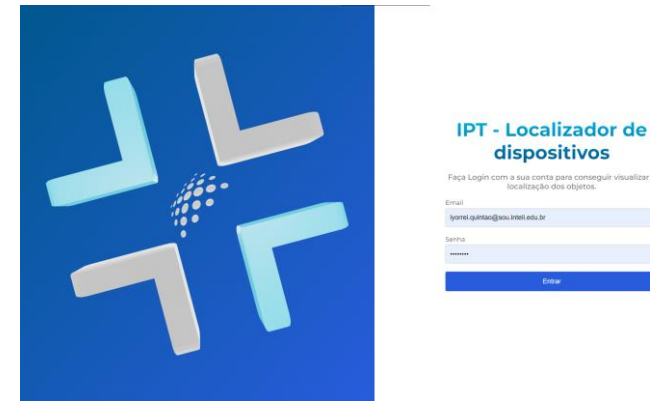
4. Efetuar o redeploy para o ESP32 através do ícone de seta no campo superior da IDE:



6. Guia de Operação

(sprint 5)

1. Login - A página de login é onde você utilizará as informações que lhe foram passadas para entrar em nosso sistema.

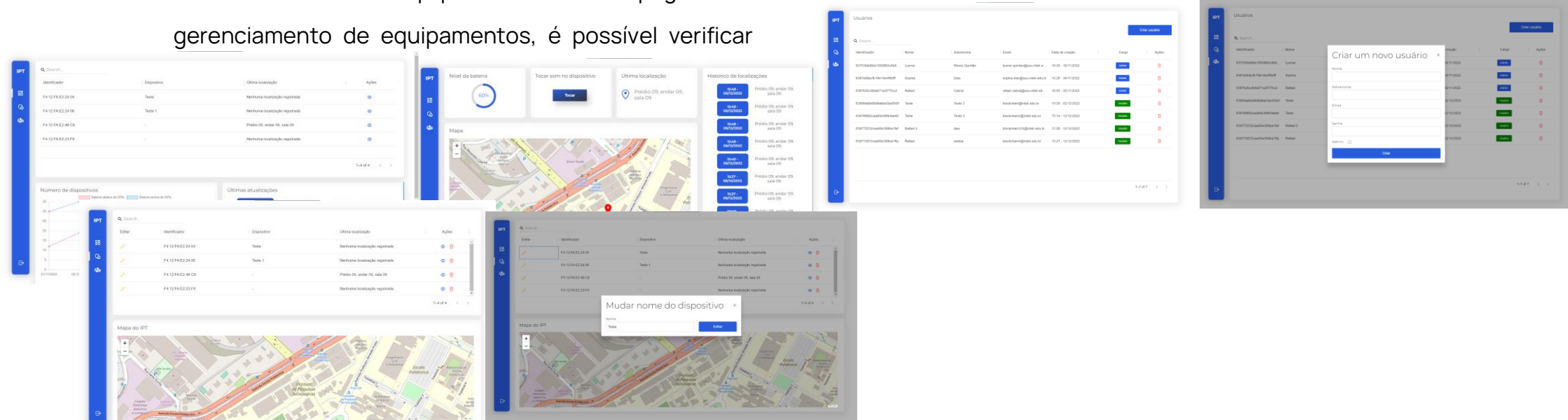


2. Informações - Na página de informações podemos verificar onde nosso equipamento foi visto a última vez,

podendo identificá-lo no mapa e a data da sua última atualização.

3. Gerenciamento de Equipamentos - Na página de gerenciamento de equipamentos, é possível verificar

também deletá-las e defini-las como uma conta de administrador ou de usuário.



quais itens estão registrados, ver suas atualizações, alterar seu nome de identificação e até mesmo excluir um equipamento que não é mais usado.

4. Gerenciamento de Usuários - Na página de gerenciamento de usuários é possível criar novas contas para novos integrantes do sistema, além disso é possível

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	Código não conseguir dar deploy no ESP.	Reiniciar e tentar novamente.
2		

8. Créditos

(sprint 5)

Agradecimentos especiais a toda equipe da iPoint pelo seu trabalho e dedicação de todo o tempo para esse projeto:

- Cristiane Andrade Coutinho
- João Pedro Gonçalves Carazzato
- Lyorrei Shono Quintão
- Rafael Alves Cabral
- Sophia Mello Dias
- Yves Levi Paixão Lapa

Agradecimentos especiais também a todos os colaboradores e professores que contribuíram para o nosso projeto:

- Instituto de Pesquisa e Tecnologia (IPT)
- Inteli
- Daniel Paz de Araujo
- André Luiz Braga
- Fabiana Martins de Oliveira
- Fatima Toledo
- Filipe Gonçalves
- Geraldo Magela Vasconcelos
- Tiago Sanches da Silva
- Victor Bruno Alexander Rosetti de Quiroz

Obrigado!