

Desenvolvimento de um Localizador de Objetos utilizando Bluetooth Low Energy e MQTT para Monitoramento IoT

Sumário

- Introdução e Motivação
- Trabalhos Relacionados
- Abordagem Proposta
- Resultados Finais
- Conclusões

Resumo

- A evolução das tecnologias sem fio, tem possibilitado o desenvolvimento de dispositivos cada vez mais inteligentes e interconectados
- Uma dessas formas, é através da Internet das coisas (IoT) que, mesmo sendo recente, vem crescendo ano após ano, em pesquisa e no uso
- Podemos citar exemplos na agricultura (Santos, 2019), saúde (Yuehong, 2016) e na indústria (Perera, 2014)
- Este trabalho, apresenta um projeto que propõe o desenvolvimento de um localizador de objetos utilizando a tecnologia Bluetooth Low Energy (BLE) em conjunto com o protocolo MQTT para monitoramento IoT

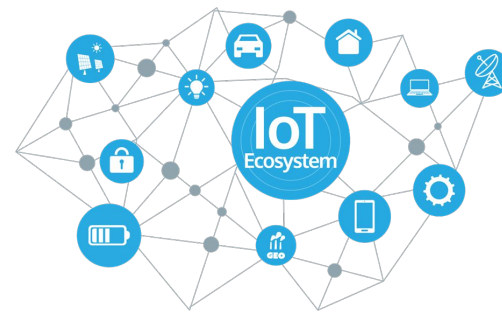


Figura 1: Fonte:
https://deinfo.uepg.br/~alunoso/2019/AEP/Internet_of_Things/A%20INTERNET%20DAS%20COISAS_arquivos/image002.png

Introdução

- Para que todos esses dispositivos tenham efetividade, independente do objetivo e contexto em que são aplicados, existem muitos desafios complexos que envolvem esses sistemas
- (Ma, 2011) cita os principais:
 - (i) heterogeneidade
 - (ii) disponibilidade de dispositivos
 - (iii) grande quantidade de dados
 - (iv) segurança e privacidade entre outros



Figura 2: Fonte:
<https://www.each.usp.br/petsi/jornal/wp-content/uploads/2015/06/internet-of-things-loT.png>

Motivação

- O projeto oferece uma solução prática e eficiente para a automação e monitoramento de objetos
- A implementação desse localizador proporcionará uma solução versátil e de baixo custo para o monitoramento de objetos em tempo real
- Pode ser aplicado em diversas situações
 - Encontrar objetos perdidos em residências
 - Monitorar a movimentação de ativos em empresas
 - Rastrear produtos em um armazém



Figura 3: Fonte:
https://deinfo.uepg.br/~alunos/2019/AEP/Internet_of_Things/A%20INTERNET%20DAS%20COISAS_arquivos/image002.png

Trabalhos Relacionados

- Lira et al.(2019);
 - ESP 32, Bluetooth Low Energy, MQTT, Arduino IDE
- Badihi et al. (2019);
 - ESP 32, Bluetooth Low Energy, MQTT, local de construção
- Riesebois et al. (2022);
 - Bluetooth Low Energy, smartphones, trilateração

Ficha Técnica: Software

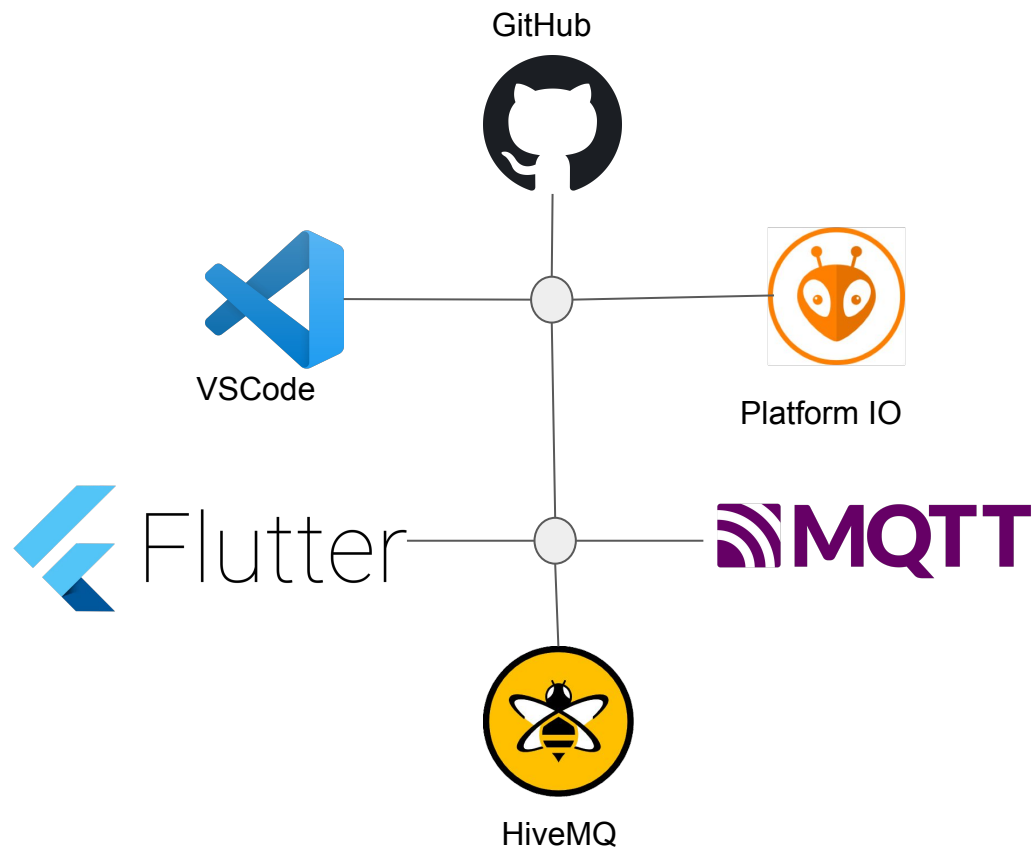


Figura 4:Figura 1. Ficha técnica.
Fonte: Próprios autores.

Arquitetura

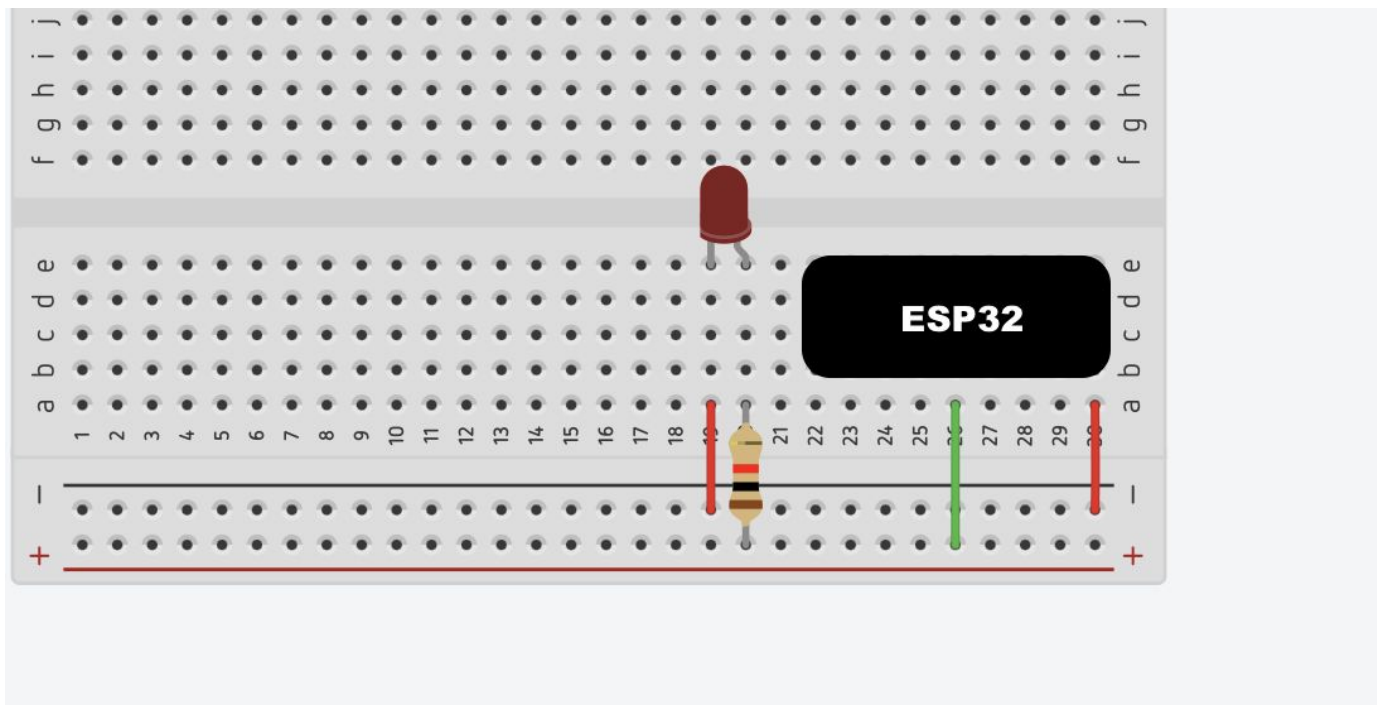


Figura 5. Arquitetura ESP 32.
Fonte: Próprios autores.

Abordagem Proposta

Arquitetura

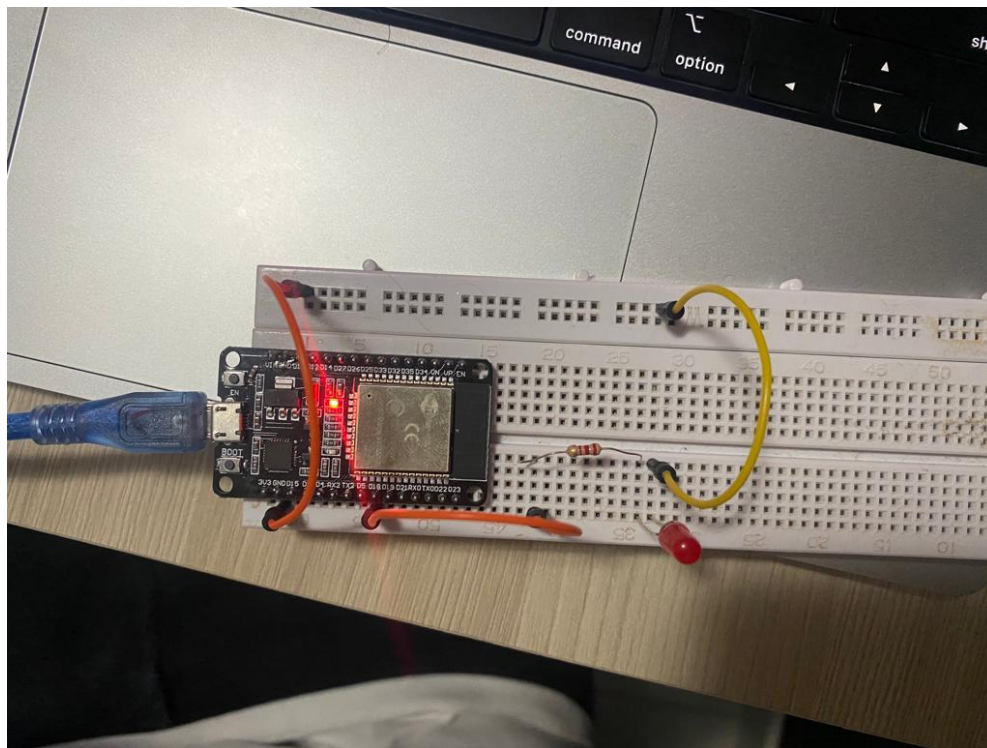


Figura 6. Montagem e arquitetura do ESP32. Fonte: Próprios autores.

Protótipo: Cálculo da distância

$$\text{Distância} = 10 \wedge ((\text{rssi_ref} - \text{RSSI}) / (10 * \text{N}))$$

rssi_ref = Frequência RSSI medida de um objeto a 1M no nosso ambiente;

RSSI = Frequência do sinal BLE do objeto atual;

N = Variável de ambiente (2 .. 4);

Protótipo: Cálculo da distância

- O resultado obtido pela fórmula é dado em uma escala logarítmica.
- Ao elevar 10 ao resultado da fração, estamos convertendo a escala logarítmica em uma escala linear,
- Resultando em uma estimativa da distância em metros.



Figura 7. Rssi LED. Fonte:
https://www.digi.com/resources/documentation/Digidocs/90001456-13/resources/images/rf_kits/img_rssi_led_smt_242x223.png

Abordagem Proposta

Variáveis

```
const char *SSID = "WIFI";S
const char *PWD = "SENHA_WIFI";
int scanTime = 120; //In seconds
BLEScan* pBLEScan;
WiFiClient wifiClient;S
PubSubClient mqttClient(wifiClient);
char *mqttServer = "broker.hivemq.com";
int mqttPort = 1883;
```

Figura 9. Código fonte variáveis. Fonte: autores.

Abordagem Proposta

Setup

```
void setup() {  
    pinMode(SCK, OUTPUT);  
    Serial.begin(9600);  
  
    connectToWiFi();  
    setupMQTT();  
    Serial.println("Scanning...");  
  
    BLEDevice::init("felipe");  
    pBLEScan = BLEDevice::getScan();  
    pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());  
    pBLEScan->setActiveScan(true);  
    pBLEScan->setInterval(100);  
    pBLEScan->setWindow(99);  
}
```

Figura 10. Código fonte setup. Fonte: autores.

Abordagem Proposta

Broker MQTT

```
// Wireless Connection
void connectToWiFi() {
    Serial.print("Connecting to ");

    WiFi.begin(SSID, PWD);
    Serial.println(SSID);
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
    }
    Serial.println(" Connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
```

Figura 11. Código fonte conexão wireless. Fonte: autores.

Abordagem Proposta

Broker MQTT

```
void setupMQTT() {  
    mqttClient.setServer(mqttServer, mqttPort);  
}  
  
void reconnect() {  
    Serial.println("Connecting to MQTT Broker...");  
    while (!mqttClient.connected()) {  
        Serial.println("Reconnecting to MQTT Broker..");  
        String clientId = "CLIENT ESP32 - ";  
        clientId += String(random(0xffff), HEX);  
  
        if (mqttClient.connect(clientId.c_str())) {  
            Serial.println("Connected.");  
            // subscribe to topic  
            mqttClient.subscribe("/teste");  
        }  
    }  
}
```

Figura 12. Código fonte setup MQTT. Fonte: autores.

Abordagem Proposta

Loop

```
void loop() {  
  if (!mqttClient.connected())  
    reconnect();  
  mqttClient.loop();  
  
  BLEScanResults foundDevices = pBLEScan->start(scanTime, false);  
  Serial.print("Devices found: ");  
  Serial.println(foundDevices.getCount());  
  Serial.println("Scan done!");  
  pBLEScan->clearResults();  
  delay(2000);  
}
```

Figura 13. Código fonte loop. Fonte: autores.

Abordagem Proposta

Broker MQTT

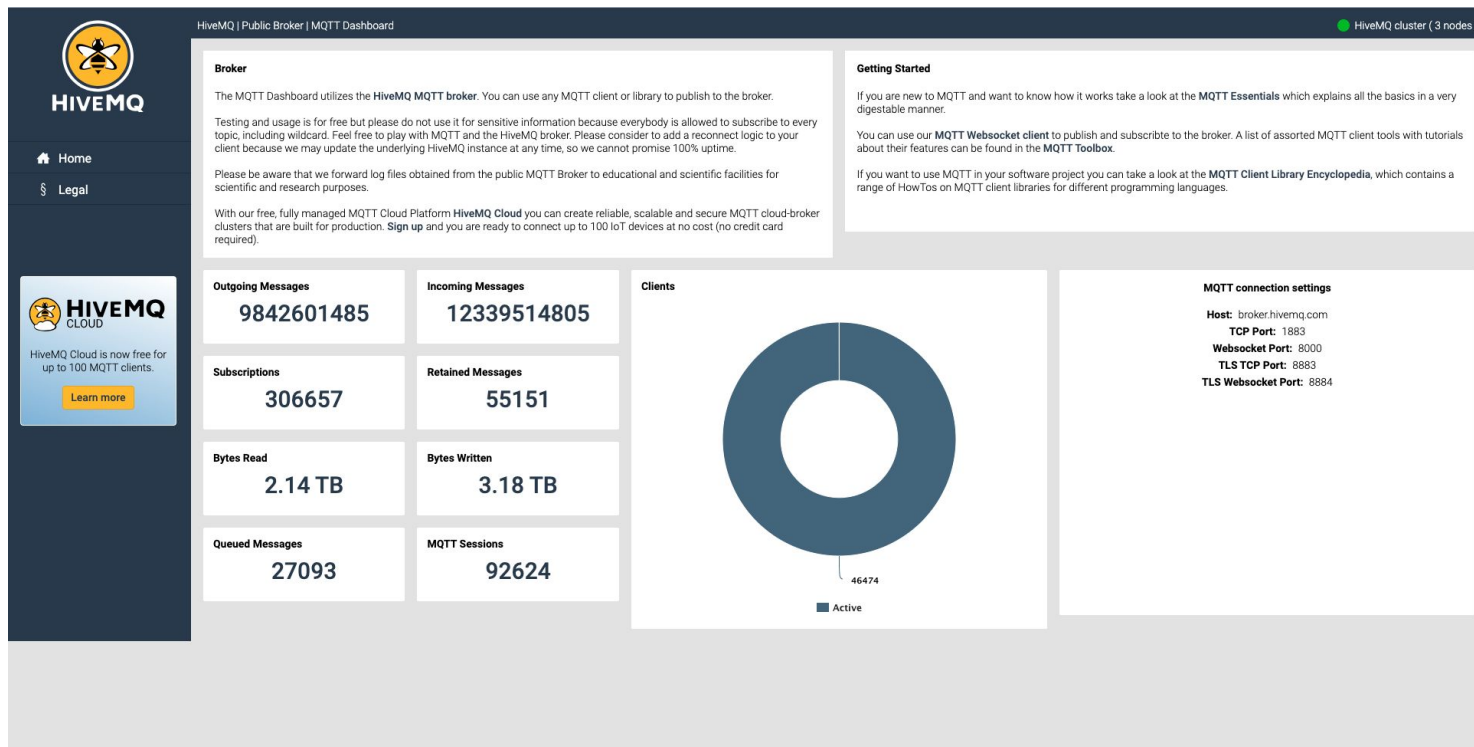
```
class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {  
    void onResult(BLEAdvertisedDevice advertisedDevice) {  
        Serial.printf("Publicando MAC= %s e RSSI= %d \n", advertisedDevice.getAddress().toString().c_str(), advertisedDevice.getRSSI());  
        const char* data = advertisedDevice.getAddress().toString().c_str() + '#' + advertisedDevice.getRSSI();  
        digitalWrite(SCK,HIGH);  
        delay(500);  
        digitalWrite(SCK,LOW);  
        delay(500);  
        mqttClient.publish("/teste/trabalhoOIRC", data);  
    }  
};
```

Figura 14. Código fonte resultados e operações. Fonte: autores.

Abordagem Proposta

Broker MQTT

Figura 12. Broker público da empresa HiveMQ. Fonte: HiveMQ (2023).



Abordagem Proposta

Protótipo

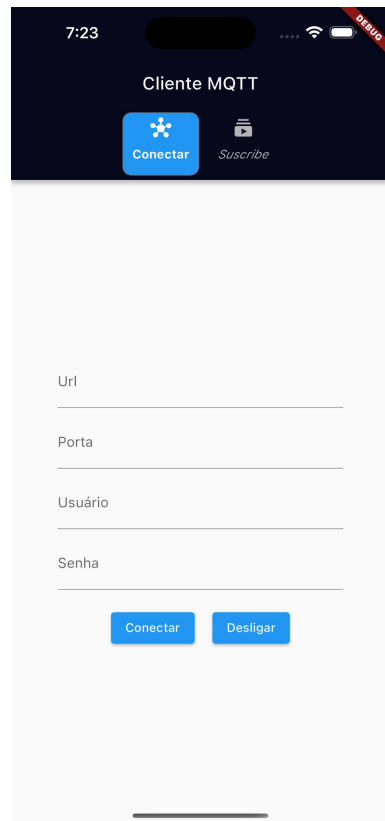


Figura 16. Landing page do aplicativo criado. Fonte: Próprios autores.

Abordagem Proposta

Protótipo

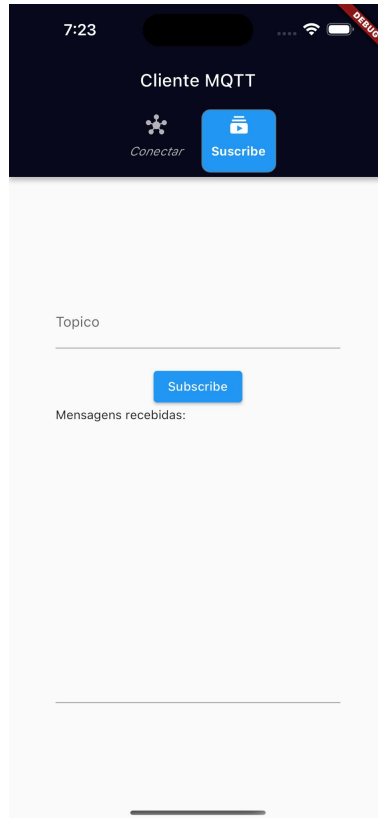


Figura 17. Tela de suscribe do aplicativo. Fonte: Próprios autores.

Abordagem Proposta

Protótipo

```
flutter: 1-2023-07-03 07:24:34.418019 -- MqttServerConnection._onData -- message available event fired
flutter: 1-2023-07-03 07:24:34.418837 -- MqttConnectionHandlerBase::messageAvailable - message type is MqttMessageType.publish
flutter: 1-2023-07-03 07:24:34.419016 -- PublishingManager::handlePublish - publish received from broker with topic /teste/trabalho0IRC
flutter: Received message: topic is /teste/trabalho0IRC, payload is e5:39:fb:b0:62:43#-71
```

Figura 18. Recebendo informações do tópico ao qual fomos suscritos. Fonte: Próprios autores.

Abordagem Proposta

Protótipo

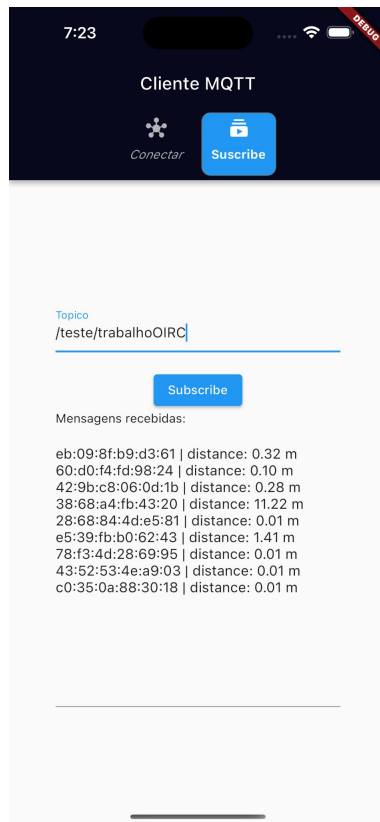


Figura 19. Recebendo informações via MQTT. Fonte: Próprios autores.

Resultados Finais

Para esta segunda etapa do trabalho, pretendíamos concluir:

- Otimizar o cálculo da distância com o RSSI;
- Enviar os dados para um broker MQTT;
- Produzir ou utilizar uma interface amigável para monitoramento;
- Diminuir o processamento do ESP 32.

Resultados Finais

Para esta segunda etapa do trabalho, conseguimos:

- Foram feitos testes para otimizar as variáveis da fórmula da distância;
- Foram enviados dados para um Broker MQTT público da HIVEMQ;
- Foi desenvolvida uma interface amigável para monitoramento das mensagens;
- O processamento da fórmula da distância foi passada para o aplicativo, tirando processamento do ESP 32.

Conclusão

- O localizador de objetos desenvolvido para esse trabalho, tem o potencial de trazer benefícios significativos em termos de automatização e eficiência em diferentes setores.
- A qualidade de sinal do Bluetooth BLE, tanto neste trabalho quanto nos trabalhos relacionados, mostrou-se confiável e com uma mínima taxa de erro considerada aceitável.

Conclusão

- O erro se deve ao fato de os testes terem sido feitos principalmente em ambientes com a presença de estruturas ou objetos que causam interferência, como paredes e outros dispositivos, respectivamente.
- Outro exemplo de aplicação é o citado por Lira et al.(2019), que menciona como trabalhos futuros, a utilização de um sistema de destravamento de portas através do Bluetooth BLE, sem necessidade de chave, apenas por reconhecimento de que um dispositivo cadastrado anteriormente no localizador está próximo.

Referências

Khalid, A., Memon, I. (2021). Bluetooth-Based Traffic Tracking System Using ESP32 Microcontroller. In: Patnaik, S., Yang, X.S., Sethi, I. (eds) *Advances in Machine Learning and Computational Intelligence. Algorithms for Intelligent Systems*. Springer, Singapore.

https://doi.org/10.1007/978-981-15-5243-4_70

LIRA, Filipe Almeida; C. JUNIOR, Francisco L.; DO NASCIMENTO, Erik J. F. ; JUCA, Sandro C. S. ; M. JÚNIOR, Jose N. . Localizador de objetos em curtas distâncias baseado em Bluetooth BLE com monitoramento IoT via MQTT. In: ESCOLA REGIONAL DE COMPUTAÇÃO APLICADA À SAÚDE (ERCAS), 7. , 2019, Teresina. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. 109-114.

R, Venkatesh & Mittal, Vikas & Tammana, Hrudaya. (2021). Indoor Localization in BLE using Mean and Median Filtered RSSI Values. 227-234. 10.1109/ICOEI51242.2021.9453000.

Random Nerd Tutorials. (n.d.). Getting Started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE. Retirado de: <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>

BADIHI, Behnam et al. Intelligent construction site: on low cost automated indoor localization using bluetooth low energy beacons. In: 2019 IEEE Conference on Wireless Sensors (ICWiSe). IEEE, 2019. p. 29-35.

Rácz-Szabó A, Ruppert T, Bántay L, Löcklin A, Jakab L, Abonyi J. Real-Time Locating System in Production Management. *Sensors*. 2020; 20(23):6766. <https://doi.org/10.3390/s20236766>

Referências

RIESEBOS, Robert; DEGELER, Viktoriya; TELLO, Andrés. Smartphone-based real-time indoor positioning using BLE beacons. In: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). IEEE, 2022. p. 1281-1288.

Santos, I. B. dos, Sandmann, A., Souza, B. E. de, Schmidt, C. A. P., Filho, P. L. de P., Melges, A. I., & Marcolin, J. F. (2019). Internet das coisas (IoT) aplicada ao agronegócio: Projeto e implementação de um gateway de IoT sobre a plataforma Arduino para simplificar a automatização da aquicultura

YUEHONG, Y. I. N. et al. The internet of things in healthcare: An overview. Journal of Industrial Information Integration, v. 1, p. 3-13, 2016.

PERERA, Charith et al. A survey on internet of things from industrial market perspective. IEEE Access, v. 2, p. 1660-1679, 2014.

Créditos

Ayran - ayranduarte5@gmail.com

Matias Gutierrez- matiguti17@gmail.com

Victor Eduardo Requia - victorrequia@gmail.com

Universidade do Estado de Santa Catarina - Joinville - SC

Professora Janine Kniess

03/07/2023