CENTRO UNIVERSITÁRIO FEI

ERICH RAMOS BORGES

GABRIEL MARQUES SILVA

OTAVIO MEVES RIBEIRO

ROBERT RIVERA M DA SILVA

VINÍCIUS HENRIQUE SOUZA DE MELO

WESLLEY SILVA SANTOS

**ESTUDO SOBRE REDE SEM FIO APLICADA NA GEOLOCALIZAÇÃO DE AGENTES CIRCULANTES**

São Bernardo do Campo

2025

ERICH RAMOS BORGES

GABRIEL MARQUES SILVA

OTAVIO MEVES RIBEIRO

ROBERT RIVERA M DA SILVA

VINÍCIUS HENRIQUE SOUZA DE MELO

WESLLEY SILVA SANTOS

**ESTUDO SOBRE REDE SEM FIO APLICADA NA GEOLOCALIZAÇÃO DE AGENTES CIRCULANTES**

Trabalho de Conclusão de Curso apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia Elétrica. Orientado pelo Prof. Marco Antônio Assis de Melo.

São Bernardo do Campo

2025

ERICH RAMOS BORGES

GABRIEL MARQUES SILVA

OTAVIO MEVES RIBEIRO

ROBERT RIVERA M DA SILVA

VINÍCIUS HENRIQUE SOUZA DE MELO

WESLLEY SILVA SANTOS

**ESTUDO SOBRE REDE SEM FIO APLICADA NA GEOLOCALIZAÇÃO DE AGENTES CIRCULANTES**

Trabalho de Conclusão de Curso, apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia Elétrica

Comissão julgadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Orientador e presidente

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examinador (1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examinador (2)

São Bernardo do Campo

2025

**AGRADECIMENTOS**

Dedicamos este trabalho a todos os membros de nossa família, que nos acompanharam ao longo desta jornada, seja em presença física ou espiritual. Nossa gratidão também se estende aos amigos que, por diferentes motivos, não puderam concluir o curso conosco. A cada um de vocês, nosso reconhecimento e apreço por terem sido parte fundamental de nossa trajetória.

"Os verdadeiros amigos são como diamantes, pedras preciosas e raras. Falsos amigos são como folhas de outono, encontradas em toda parte."  
*Bruce Lee*

**RESUMO**

A geolocalização desempenha um papel crucial em setores que demandam monitoramento em tempo real, especialmente em ambientes internos, onde tecnologias como o *GPS* enfrentam limitações significativas devido a barreiras físicas e interferências. Este trabalho propõe o desenvolvimento de um sistema de geolocalização baseado em *Internet of Things* (*IoT*) para o Metrô de São Paulo, utilizando o microcontrolador ESP32-C6 em conjunto com redes Wi-Fi6. Diferentemente de abordagens tradicionais que buscam coordenadas geográficas precisas, o sistema identifica zonas de proximidade em relação aos pontos de acesso Wi-Fi, com base em medições da intensidade de sinal (*RSSI*). Essa abordagem privilegia simplicidade, baixo custo e eficiência, sendo adequada às necessidades práticas de monitoramento interno.

A metodologia adotada incluiu etapas de mapeamento ambiental, coleta de dados, definição de zonas de teste e implementação de um sistema *Real-Time Location System* (*RTLS*), com visualização em tempo real por meio de bibliotecas em *Python*. A aplicação considerou desafios técnicos, como interferências de sinal e escalabilidade em redes densas, além de aspectos éticos relacionados à privacidade de dados dos funcionários, em conformidade com as diretrizes da Lei Geral de Proteção de Dados (*LGPD*).

**Palavras-chave**: geolocalização. *IoT*. privacidade. *RSSI*. transporte público. Wi-Fi6.

**ABSTRACT**

Geolocation plays a crucial role in sectors that require real-time monitoring, especially in indoor environments, were technologies such as GPS face significant limitations due to physical barriers and interference. This work proposes the development of a geolocation system based on Internet of Things (IoT) for the São Paulo Metro, using the ESP32-C6 microcontroller in conjunction with Wi-Fi 6 networks. Unlike traditional approaches that aim for precise geographic coordinates, the system identifies proximity zones relative to Wi-Fi access points, based on signal strength measurements (RSSI). This approach prioritizes simplicity, low cost, and efficiency, making it suitable for practical indoor monitoring applications.

The methodology included environmental mapping, data collection, definition of test zones, and the implementation of a Real-Time Location System (RTLS), with real-time visualization using Python libraries. The application considered technical challenges such as signal interference and scalability in dense networks, as well as ethical aspects related to employee data privacy, in compliance with Brazil's General Data Protection Law (LGPD).

**Keywords**: geolocation. IoT. privacy. RSSI. public transportation. Wi-Fi 6.

**LISTA DE ILUSTRAÇÕES**

[Figura 1 – Exemplo de dispositivos conectados simultaneamente via BLE 25](#_Toc200321424)

[Figura 2 – Proposta inicial de Dashboard elaborado 28](#_Toc200321425)

[Figura 3 - Fluxograma do Funcionamento do Projeto 34](#_Toc200321426)

[Figura 4 - Algoritmo do Microcontrolador 35](#_Toc200321427)

[Figura 5 - Imagem Placa DevKitC-1 ESP-C6 N8 38](#_Toc200321428)

[Figura 6 - Imagem Bateria Li-Ion 18650 3,7V 2500mAh 2C 38](#_Toc200321429)

[Figura 7 - Carregador Duplo para Bateria Li-Ion 18650 39](#_Toc200321430)

[Figura 8 - Carregador Bateria Lithium 1A com Proteção - USB C -TP4056 39](#_Toc200321431)

[Figura 9 - Conversor step-up (boost) DC-DC – XL6009 40](#_Toc200321432)

[Figura 10 - Interface desenvolvida através do Elipse E3. 41](#_Toc200321433)

[Figura 11 - Interface do software desenvolvido. 42](#_Toc200321434)

[Figura 12 - Tela de agentes mais próximos a incidente. 42](#_Toc200321435)

[Figura 13 - Interface gráfica em Dash, com o local de reportar um novo incidente, incidentes registrados, e mapa da linha do Metrô com a localização de dois dispositivos. 44](#_Toc200321436)

[Figura 14 - Localização dos dispositivos sendo recebida pela aplicação Python através do MQTT 45](#_Toc200321437)

[Figura 15 - Interface com dados de localização recebidos 46](#_Toc200321438)

[Figura 16 - Hardware Projetado 46](#_Toc200321439)

[Figura 17 - Protótipo do teste de carga 47](#_Toc200321440)

[Figura 18 - Protótipo em Protoboard Finalizado 47](#_Toc200321441)

[Figura 19 - Placa de Circuito Impresso 49](#_Toc200321442)

[Figura 20 - Placa de Circuito Impresso e componentes 50](#_Toc200321443)

[Figura 21 - Projeto da Case Através do Fusion360 50](#_Toc200321444)

**LISTA DE TABELAS**

[Tabela 1 - Grau de automação e responsabilidades operacionais no transporte metroferroviário 21](#_Toc200321445)

[Tabela 2 - Lista de Materiais 37](#_Toc200321446)

**LISTA DE ABREVIATURAS E SIGLAS**

**AMQP** – Advanced Message Queuing Protocol (Protocolo Avançado de Fila de Mensagens)

**AOA** – Angle of Arrival (Ângulo de Chegada)

**AOD** – Angle of Departure (Ângulo de Partida)

**API** – Application Programming Interface (Interface de Programação de Aplicações)

**APs** – Access Points (Pontos de Acesso)

**BLE** – Bluetooth Low Energy (Bluetooth de Baixo Consumo)

**BLE Mesh** – Bluetooth Low Energy Mesh Networking (Rede Mesh com Bluetooth de Baixo Consumo)

**BSSID** – Basic Service Set Identifier (Identificador de Conjunto de Serviço Básico)

**CCO** – Centro de Controle Operacional

**CSS** – Cascading Style Sheets (Folhas de Estilo em Cascata)

**DF** – Direction-Finding (Determinação de Direção)

**ESP-IDF** – Espressif IoT Development Framework (Framework de Desenvolvimento IoT da Espressif)

**FEI** – Fundação Educacional Inaciana Padre Saboia de Medeiros

**GPIO** – General-Purpose Input/Output (Entrada/Saída de Uso Geral)

**GPS** – Global Positioning System (Sistema de Posicionamento Global)

**GUI** – Graphical User Interface (Interface Gráfica de Usuário)

**HTML** – HyperText Markup Language (Linguagem de Marcação de Hipertexto)

**IEEE** – Institute of Electrical and Electronics Engineers (Instituto de Engenheiros Eletricistas e Eletrônicos)

**IoT** – Internet of Things (Internet das Coisas)

**KNN** – K-Nearest Neighbors (K-Vizinhos Mais Próximos)

**LGPD** – Lei Geral de Proteção de Dados

**MAC** – Media Access Control (Controle de Acesso ao Meio)

**MIMO** – Multiple Input Multiple Output (Múltiplas Entradas e Múltiplas Saídas)

**MISO** – Multiple Input Single Output (Múltiplas Entradas e Uma Saída)

**MQTT** – Message Queuing Telemetry Transport (Transporte de Telemetria com Enfileiramento de Mensagens)

**MQTT Broker** – Message Queuing Telemetry Transport Broker (Servidor Distribuidor MQTT)

**MUSIC** – Multiple Signal Classification (Classificação de Sinais Múltiplos)

**NLOS** – Non-Line of Sight (Sem Linha de Visada Direta)

**OLED** – Organic Light-Emitting Diode (Diodo Emissor de Luz Orgânico)

**OPEX** – Operational Expenditure (Despesas Operacionais)

**PCB** – Printed Circuit Board (Placa de Circuito Impresso)

**PHY** – Physical Layer (Camada Física)

**PIMS** – Plant Information Management System (Sistema de Gerenciamento de Informações da Planta)

**QoS** – Quality of Service (Qualidade de Serviço)

**RISC-V** – Reduced Instruction Set Computing – V (Computação com Conjunto Reduzido de Instruções – Versão V)

**RSSI** – Received Signal Strength Indicator (Indicador de Intensidade do Sinal Recebido)

**RTLS** – Real-Time Location System (Sistema de Localização em Tempo Real)

**SCADA** – Supervisory Control and Data Acquisition (Supervisão e Aquisição de Dados)

**SDK** – Software Development Kit (Kit de Desenvolvimento de Software)

**SRAM** – Static Random Access Memory (Memória RAM Estática)

**SSID** – Service Set Identifier (Identificador de Conjunto de Serviço)

**STA** – Station Mode (Modo Estação)

**TCC** – Trabalho de Conclusão de Curso

**TCP/IP** – Transmission Control Protocol/Internet Protocol (Protocolo de Controle de Transmissão/Protocolo de Internet)

**TDOA** – Time Difference of Arrival (Diferença de Tempo de Chegada)

**TLS** – Transport Layer Security (Segurança da Camada de Transporte)

**TOA** – Time of Arrival (Tempo de Chegada)

**USB-C** – Universal Serial Bus Type-C (Barramento Universal do Tipo C)

**UTO** – Unattended Train Operation (Operação de Trem Não Tripulado)

**VPNs** – Virtual Private Network (Rede Virtual Privada)

**WPA** – Wi-Fi Protected Access (Acesso Wi-Fi Protegido)

**WSN** – Wireless Sensor Network (Rede de Sensores sem Fio)

**Wi-Fi** – Wireless Fidelity (Fidelidade Sem Fio)

**PCB** – Printed Circuit Board (Placa de Circuito Impresso)

**LISTA DE SÍMBOLOS**

**dBm –** Decibéis por miliwatt

**µA​ –** Microampère

**MHz –** Mega-hertz

**MB –** Megabyte

**–** Distância

**–** Expoente de percurso, variável conforme o ambiente

**–** Constante que representa perdas adicionais, como atenuação por barreiras

**–** Potência transmitida pelo ponto de acesso

**–** Potência recebida no dispositivo

**–** Coordenadas cartesianas

**–** Estado no instante k (posição e velocidade)

**–** Matriz de transição de estado, que relaciona estados consecutivo **–** É o intervalo de tempo entre medições

**–** Matriz de controle, que relaciona entradas externas (uk) ao estado

**–** Entrada de controle, como aceleração

**C –** Velocidade da luz

**mAh** **–** Miliampere-hora

**V –** Volts

**C –** Coulombs

**GHz –** Gigahertz

**SUMÁRIO**

1 INTRODUÇÃO 11

2 REVISÃO DA LITERATURA 13

2.1 Conceitos De Geolocalização 13

2.2 REDES WI-FI E GEOLOCALIZAÇÃO 14

2.2.1 Estrutura Básica de uma Rede Wi-Fi 14

2.2.2 Identificação de Acess Point 15

2.2.3 Influência do Ambiente na Localização 16

2.3 MODELOS MATEMÁTICOS E ESTATÍSTICOS EM GEOLOCALIZAÇÃO 17

2.3.1 Mapeamento do Ambiente 17

2.3.2 Coleta de Dados RSSI 18

2.4 PRIVACIDADE DO USUÁRIO DO SISTEMA 18

2.5 AUTOMAÇÃO NÍVEL GOA4 E SUA RELAÇÃO COM SISTEMAS DE GEOLOCALIZAÇÃO NO METRÔ 19

2.6 IOT E GEOLOCALIZAÇÃO 21

2.6.1 O Papel da Geolocalização em Sistemas IoT 22

2.6.1.2 Tecnologias para Geolocalização em Ambientes Internos 22

2.6.1.3 Aplicações de Geolocalização em Emergências 23

2.6.1.4 Desafios na Implementação de Geolocalização 23

2.7 ESP32 – C6: HARDWARE E FUNCIONALIDADES 24

2.7.1 Características Técnicas 24

2.7.2 Funções de Geolocalização do ESP32 24

2.7.3 Protocolos Suportados 25

2.8 SOFTWARES E TECNOLOGIAS AUXILIARES 26

2.8.1 Ferramentas e Bibliotecas do ESP32-C6 para WI-FI 28

2.8.2 Biblioteca Streamlit do Python 29

2.8.3 Biblioteca Dash do Python 29

2.8.4 Broker Mosquitto 30

2.8.4 Elipse Software 31

2.8.5 Sistema SCADA vs. PIMS 32

3 METODOLOGIA 33

3.1 ALGORITMO DO DISPOSITIVO 34

3.2 mqtt broker 35

3.3 ABORDAGEM UTILIZADA 35

3.4 Ambientes de Teste e Redes IoT 36

3.5 Ambientes de Teste e Rede 36

3.6 LISTA DE MATERIAIS 36

3.7 TESTES DO SISTEMA DE LOCALIZAÇÃO DO SISTEMA IOT 40

3.7.1 Sistema de Monitoramento da Geolocalização 40

3.7.2 Testes do sistema para verificação de erros 44

3.7.3 Protótipo do Dispositivo de Geolocalização 46

**3.8** HARDWARE 48

3.8.1.1 ESTIMATIVA DE AUTONOMIA ENERGÉTICA 48

3.8.2 Montagem do hardware 49

CONCLUSÃO 52

REFERÊNCIAS 53

# 1 INTRODUÇÃO

A geolocalização tem se consolidado como uma tecnologia essencial em diversos setores da sociedade moderna, oferecendo soluções para rastreamento de ativos, monitoramento de pessoas, logística, segurança e automação de processos. A capacidade de identificar a localização de um dispositivo em tempo real possibilita uma série de aplicações críticas, tanto em ambientes externos quanto internos. Com o avanço das tecnologias de comunicação sem fio e a disseminação de dispositivos conectados, especialmente no contexto da Internet das Coisas (IoT), os sistemas de localização têm evoluído significativamente em termos de precisão, escalabilidade e viabilidade operacional (YANG et al., 2021).

Entretanto, a aplicação de geolocalização em ambientes internos – como edifícios, túneis e estações de metrô – apresenta desafios técnicos relevantes. A presença de obstáculos físicos, a multiplicidade de reflexões de sinal, a interferência entre dispositivos e a ausência de visada direta com satélites comprometem a eficácia de soluções tradicionais, como o Global Positioning System (GPS). Diante dessas limitações, surgem abordagens alternativas baseadas em redes sem fio, como Wi-Fi e Bluetooth Low Energy (BLE), combinadas com algoritmos matemáticos de estimação de posição. Tais sistemas frequentemente utilizam métricas como o Received Signal Strength Indicator (RSSI) para estimar a proximidade entre dispositivos e pontos de referência, podendo ser aprimorados com o uso de técnicas de aprendizado de máquina (PAULO, 2018; MANIKANDAN, 2023).

Neste contexto, este trabalho propõe o desenvolvimento de um sistema de localização indoor com foco na operação do Metrô de São Paulo, utilizando o microcontrolador ESP32-C6 como núcleo computacional. O objetivo é permitir o monitoramento da posição de funcionários dentro das instalações metroviárias, fornecendo informações de localização com base na zona de proximidade dos dispositivos em relação aos Access Points (APs) da rede Wi-Fi. Ao invés de estimar coordenadas exatas, a proposta se concentra em identificar em qual região ou zona o agente se encontra, o que é suficiente para fins operacionais e oferece uma abordagem mais simples, eficiente e economicamente viável.

A escolha do ESP32-C6 justifica-se por suas características técnicas avançadas, incluindo suporte a Wi-Fi 6, arquitetura RISC-V, baixo consumo energético e compatibilidade com protocolos modernos de comunicação, como MQTT. Além disso, sua capacidade de varredura ativa de redes permite a coleta periódica de dados de RSSI e BSSID, que são utilizados para inferir a localização aproximada. A infraestrutura existente da rede Wi-Fi do metrô pode ser aproveitada para o funcionamento do sistema, eliminando a necessidade de grandes investimentos em novas tecnologias.

Este trabalho está estruturado para apresentar, inicialmente, os fundamentos teóricos da geolocalização indoor e as limitações dos métodos convencionais. Em seguida, são descritas as tecnologias utilizadas, o projeto de hardware, a modelagem do sistema e os procedimentos de implementação. Por fim, são discutidos os resultados obtidos por meio de testes práticos, os desafios enfrentados durante o desenvolvimento e as possibilidades de aprimoramento e expansão da solução para outras aplicações em mobilidade urbana e logística operacional.

# 2 REVISÃO DA LITERATURA

## 2.1 Conceitos De Geolocalização

Os serviços baseados em localização desempenham um papel relevante nas pesquisas voltadas ao desenvolvimento de ferramentas para o controle de processos que exigem informações espaciais em tempo real. Segundo a revisão de literatura realizada por Yang et al. (2021), os estudos mais recentes têm proposto diversos esquemas e sistemas de posicionamento com o objetivo de viabilizar soluções eficientes em serviços baseados em localização.

Os sistemas de geolocalização podem ser classificados em duas grandes categorias: internos e externos. Essa distinção está associada às diferentes necessidades operacionais, sendo que suas aplicações se estendem desde ambientes comerciais e industriais até sistemas de transporte, como o metrô. No entanto, os sistemas internos apresentam maior complexidade técnica, especialmente no que se refere à obtenção de precisão adequada para ambientes fechados. De acordo com (YANG et al., 2021), essa limitação exige abordagens específicas para contornar os obstáculos típicos desses cenários.

O estudo conduzido por Sophia et al. (2021) reforça essa questão ao destacar que o *Global Positioning System* (GPS), embora altamente eficiente em ambientes externos, apresenta baixa eficácia em locais fechados, onde obstáculos físicos comprometem a propagação do sinal. Para mitigar essas deficiências, o estudo propôs a utilização da tecnologia *Bluetooth Low Energy* (BLE), implementando um sistema de localização com o microcontrolador ESP32-NodeMCU. O objetivo foi superar as restrições impostas à penetração do sinal GPS em estruturas sólidas, característica comum em espaços internos.

Atualmente, não há um consenso sobre qual tecnologia oferece o melhor desempenho para localização indoor. Contudo, os sistemas baseados em Wi-Fi apresentam ampla difusão e têm sido incentivados por sua capacidade de aproveitamento da infraestrutura já existente em ambientes públicos (MANIKANDAN, 2023). Nesse contexto, Manikandan (2023) avaliou a viabilidade de um protótipo baseado no módulo ESP8266, um dos mais utilizados em aplicações de *IoT*, aliado à técnica de *Received Signal Strength Indicator* (RSSI), para realizar leituras de sinal em ambientes fechados com o objetivo de determinar a posição de dispositivos.

Para que um sistema de localização indoor seja funcional e aplicável, Paulo (2018) destaca a importância de que ele seja preciso, de baixo custo, modular e facilmente replicável em diferentes ambientes. O autor ainda descreve diversas abordagens utilizadas nesse tipo de sistema, como *Time of Arrival* (TOA), *Time Difference of Arrival* (TDOA), *RSSI* e *Dead Reckoning*, além de métodos baseados em *Motion Fingerprint* e *Signal Fingerprint*, frequentemente associados a algoritmos de *machine learning*. Essas técnicas utilizam dados de Wi-Fi, *Bluetooth* e sensores embarcados, como acelerômetros, giroscópios e magnetômetros, para melhorar a precisão do rastreamento.

As soluções de localização interna podem ser divididas em dois grupos principais: com dispositivos e sem dispositivos. Nas soluções baseadas em dispositivos, o objeto ou indivíduo monitorado carrega um equipamento de comunicação, como um *smartphone* ou sensor dedicado, que se comunica com os *Access Points* (APs) distribuídos no ambiente. Essa abordagem é especialmente eficaz em cenários controlados, nos quais o dispositivo pode ser previamente fornecido ao usuário, garantindo maior precisão (PAULO, 2018).

Em contrapartida, soluções sem dispositivos são ideais para casos em que não é possível acoplar nenhum equipamento à pessoa ou objeto monitorado. Exemplos comuns incluem o rastreamento de idosos, proteção da fauna silvestre, detecção de intrusos e resposta a emergências. Nesses casos, a localização é realizada por meio da análise das alterações no padrão de propagação do sinal causadas pelo movimento no ambiente, dispensando a presença de sensores junto ao alvo monitorado (PAULO, 2018).

## 2.2 REDES WI-FI E GEOLOCALIZAÇÃO

### 2.2.1 Estrutura Básica de uma Rede Wi-Fi

As redes Wi-Fi operam com base na especificação do *Institute of Electrical and Electronics Engineers* (IEEE) 802.11, que define os parâmetros e protocolos necessários para uma comunicação sem fio eficiente. Em uma rede Wi-Fi estruturada, dispositivos como *laptops, smartphones* e *tablets* conectam-se a *Access Points* (APs), que funcionam como intermediários, transmitindo dados por meio de sinais de rádio para a rede cabeada. Os APs constituem elementos centrais da infraestrutura, permitindo que múltiplos dispositivos acessem simultaneamente os serviços da rede (MOURA, 2007).

A troca de informações ocorre por meio da transmissão de pacotes de dados que seguem os protocolos definidos nas camadas *Physical Layer* (PHY) e *Media Access Control* (MAC), assegurando uma comunicação bidirecional eficiente e confiável entre os dispositivos e os APs. Essa comunicação utiliza dois identificadores fundamentais: o *Service Set* Identifier (SSID), responsável por nomear a rede de forma visível ao usuário, e o *Basic Service Set* Identifier (BSSID), que identifica unicamente cada AP com base em seu endereço MAC. Ambos são essenciais para a configuração, manutenção e autenticação da conexão (MOURA, 2007).

### 2.2.2 Identificação de Acess Point

O *Basic Service Set Identifier* (BSSID) é um identificador único utilizado em redes Wi-Fi para representar um *Access Point* (AP). Derivado do endereço *Media Access Control* (MAC) do AP, esse identificador desempenha papel fundamental na comunicação em redes sem fio. De acordo com o padrão IEEE 802.11, o BSSID é essencial para a estruturação e manutenção da rede, pois permite que os dispositivos conectados reconheçam e interajam com um ponto de acesso específico, mesmo quando há múltiplos APs operando na mesma rede (MOURA, 2007).

A principal função do BSSID é garantir que um dispositivo se conecte ao AP apropriado, o qual é responsável pela distribuição dos sinais de rádio que viabilizam a comunicação entre os dispositivos móveis e a infraestrutura cabeada da rede. Cada AP transmite um sinal único, sendo o BSSID associado a esse sinal o elemento que permite localizar precisamente o ponto de acesso dentro de um determinado ambiente (MOURA, 2007).

Nos sistemas de localização em tempo real baseados em redes Wi-Fi, o BSSID assume um papel ainda mais relevante. Quando utilizado em conjunto com o *Received Signal Strength Indicator* (RSSI), torna-se possível estimar a posição de um dispositivo móvel por meio de técnicas como a triangulação.

O RSSI indica a intensidade do sinal recebido de cada AP e, ao ser correlacionado com a posição conhecida desses pontos de acesso, possibilita o cálculo da localização aproximada do dispositivo (MOURA, 2007). Essa abordagem é amplamente utilizada em ambientes internos, onde a aplicação de tecnologias como o GPS é inviável devido à presença de barreiras físicas, como estruturas metálicas ou túneis realidade comum em sistemas de transporte público e instalações industriais.

A combinação entre BSSID e RSSI é explorada em diversas técnicas de geolocalização indoor, permitindo rastreamento de dispositivos com elevado grau de precisão. Os dados obtidos por esses parâmetros são utilizados em modelos de mapeamento espacial, viabilizando a navegação em ambientes com infraestrutura Wi-Fi densa e a localização eficiente de objetos ou pessoas. Assim, o BSSID passa a ser um elemento central não apenas para a comunicação de rede, mas também para sistemas de localização em tempo real (MOURA, 2007).

### 2.2.3 Influência do Ambiente na Localização

A RSSI (*Received Signal Strength Indicator*) é uma das principais métricas utilizadas em sistemas de localização com base em Wi-Fi, sendo altamente sensível ao ambiente em que opera. Ambientes internos apresentam desafios relevantes para essa tecnologia, sobretudo pela presença de barreiras físicas que afetam a propagação do sinal. Segundo Moura (2007), diversos fatores impactam diretamente a precisão da medição da RSSI, como:

* **Paredes e divisórias:** Materiais como concreto e metal podem atenuar significativamente o sinal, criando incertezas nas leituras. Essa atenuação varia conforme a espessura e a composição do material; por exemplo, paredes de concreto ou metal podem reduzir o sinal em até 20 dBm, enquanto estruturas de vidro ou madeira geram perdas menores, mas ainda relevantes;
* **Mobiliário e presença humana:** Objetos diversos e corpos humanos absorvem ou dispersam os sinais de rádio, ocasionando variações de até 10 dBm na intensidade do sinal, comprometendo a confiabilidade da leitura;
* **Condições ambientais:** Umidade, temperatura e interferência eletromagnética de outros equipamentos eletrônicos também alteram a propagação do sinal. Experimentos demonstram que, mesmo em dispositivos estacionários, a RSSI pode variar até 15 dBm devido a essas influências externas.

Um fator crítico associado à propagação do sinal em ambientes fechados é a ocorrência de *Non-Line of Sight* (NLOS). Nessa condição, o sinal não percorre um caminho direto entre o ponto de acesso (AP) e o dispositivo receptor, sendo desviado por obstáculos físicos, o que gera múltiplos percursos e reflexões. Essa situação dificulta a correlação entre a intensidade do sinal e a distância real ao AP, podendo ocasionar erros de localização de até 7 metros, dependendo da metodologia e da acurácia do sistema utilizado (MOURA, 2007).

## 2.3 MODELOS MATEMÁTICOS E ESTATÍSTICOS EM GEOLOCALIZAÇÃO

A geolocalização em ambientes internos impõe desafios adicionais devido à presença de barreiras físicas, superfícies reflexivas e interferências causadas por dispositivos eletrônicos. Nessas condições, a propagação do sinal geralmente ocorre em regime NLOS, em que o caminho direto entre o transmissor e o receptor é obstruído, resultando em múltiplos percursos devido à reflexão, difração e atenuação do sinal. Esse fenômeno compromete a acurácia das estimativas de localização e exige o uso de modelos matemáticos e estatísticos que considerem tais distorções para melhorar a confiabilidade dos sistemas de posicionamento (PAHLAVAN; LI; MAKELA, 2002).

### 2.3.1 Mapeamento do Ambiente

A etapa inicial do projeto consiste no mapeamento detalhado do ambiente de testes, conforme as diretrizes metodológicas propostas por Moura (2007). Esta fase é fundamental para garantir a confiabilidade das medições de sinal e a consistência dos modelos de localização aplicados.

O mapeamento envolve as seguintes atividades:

* Identificação dos pontos de acesso Wi-Fi disponíveis no ambiente;
* Análise das barreiras físicas, como paredes, divisórias, mobiliário e estruturas metálicas, que possam provocar atenuação, reflexão ou difração do sinal;
* Delimitação das zonas de teste, com a definição de coordenadas georreferenciadas específicas para a coleta de dados de intensidade de sinal.

Essas informações permitem configurar modelos de propagação mais aderentes à realidade do ambiente, ajustando variáveis como o expoente de perda de percurso (η) e as perdas adicionais (C), que representam as atenuações associadas a obstáculos físicos, conforme identificado durante a análise do espaço.

### 2.3.2 Coleta de Dados RSSI

A coleta de dados será realizada utilizando o ESP32-C6, que registra o RSSI nos pontos de teste delimitados. A distância () entre o ponto de acesso e o dispositivo é calculada utilizando a equação de perda de percurso em espaço livre:

Onde:

Exemplo teórico:

## 2.4 PRIVACIDADE DO USUÁRIO DO SISTEMA

A adoção de sistemas de rastreamento no ambiente metroviário suscita discussões importantes sobre o equilíbrio entre segurança operacional e privacidade individual. Em especial, o rastreamento de funcionários configura-se como uma medida sensível, devendo ser avaliada com cautela, sobretudo em sistemas tradicionais de transporte, nos quais há operadores embarcados. Nesses contextos, o monitoramento constante pode ser interpretado como uma violação da privacidade dos trabalhadores, o que dificulta sua aceitação ética e legal.

Entretanto, em linhas totalmente automatizadas, como aquelas classificadas no nível *Unattended Train Operation* (*UTO*), a ausência de operadores a bordo transforma o rastreamento de pessoal em uma necessidade operacional. Nesse cenário, a geolocalização de agentes assume um papel estratégico para garantir a segurança dos usuários e a eficiência na resposta a situações emergenciais.

A segurança, portanto, prevalece sobre a privacidade em circunstâncias críticas, como falhas técnicas, eventos clínicos ou riscos à integridade física de passageiros e funcionários. O monitoramento em tempo real possibilita a rápida mobilização de equipes treinadas, reduzindo o tempo de resposta e contribuindo para o gerenciamento de ocorrências de maneira eficaz e coordenada.

Ainda assim, é fundamental que o rastreamento seja restrito aos profissionais cuja função demanda atuação direta em emergências. A implementação dessa medida deve seguir critérios bem definidos, com base em justificativas técnicas e operacionais que garantam o uso proporcional e legítimo da informação.

O sistema proposto, portanto, não deve ser interpretado como uma ferramenta de vigilância constante, mas sim como um recurso técnico voltado exclusivamente à segurança em contextos específicos. O respeito à privacidade dos funcionários deve ser assegurado mediante transparência nos procedimentos, consentimento quando aplicável e conformidade com a *Lei Geral de Proteção de Dados Pessoais* (Lei n.º 13.709/2018). Dessa forma, é possível promover a segurança sem comprometer os direitos individuais dos profissionais envolvidos.

## 2.5 AUTOMAÇÃO NÍVEL GOA4 E SUA RELAÇÃO COM SISTEMAS DE GEOLOCALIZAÇÃO NO METRÔ

A operação no modo *Unattended Train Operation* (*UTO*), correspondente ao Grau de Automação 4 (*GoA4*), representa o estágio mais avançado da automação metroferroviária, caracterizado pela completa ausência de operadores a bordo. Essa configuração visa garantir máxima eficiência, redução de custos operacionais e elevação dos padrões de segurança, permitindo que todas as funções críticas — como partida, parada, aceleração, frenagem e resposta a falhas leves — sejam executadas automaticamente por sistemas computacionais centralizados (BORLONI; COPCHE; SANTOS, 2019).

A adoção do *GoA4* tem se mostrado especialmente vantajosa em linhas de alta demanda, como as existentes em grandes metrópoles, pois possibilita o aumento da frequência dos trens, maior flexibilidade operacional e redução de falhas humanas. Contudo, a ausência de operadores torna indispensável a implantação de soluções tecnológicas complementares, capazes de assegurar a segurança tanto dos passageiros quanto dos trabalhadores. Nesse cenário, os sistemas de geolocalização em tempo real surgem como ferramentas fundamentais para manter uma supervisão eficaz das atividades operacionais.

Em sistemas *GoA4*, os trens são monitorados por um Centro de Controle Operacional (*CCO*), responsável por gerenciar remotamente todas as funções do sistema. Tecnologias de geolocalização são integradas a essa estrutura para permitir a identificação instantânea da posição de funcionários e ativos em áreas operacionais como túneis, plataformas, salas técnicas e via permanente assegurando que a equipe mais próxima e capacitada seja acionada em situações emergenciais.

De acordo com Borloni, Copche e Santos (2019), os sistemas *GoA4* requerem soluções de rastreamento que atendam a níveis elevados de integridade e disponibilidade, como os exigidos pela certificação de segurança *SIL 4*. A localização precisa dos colaboradores é um fator determinante para reduzir o tempo de resposta a falhas, otimizar rotinas de manutenção e permitir uma atuação imediata em ocorrências.

Além disso, conforme demonstrado por Kyriakidis, Happee e De Winter (2015), mesmo em sistemas altamente automatizados, as interações entre operadores humanos e a infraestrutura técnica continuam sendo elementos críticos. A capacidade de coordenar remotamente as ações de campo, com base em dados de geolocalização transmitidos ao *CCO*, contribui para manter os níveis exigidos de segurança e operacionalidade em contextos urbanos complexos como o sistema metroviário.

Em termos de redundância e segurança, os sistemas no nível *GoA4* são projetados para cumprir os mais altos padrões de confiabilidade, incorporando redundância em componentes críticos e tecnologias como portas de plataforma para a proteção de passageiros e funcionários. Além disso, apresentam menor *Operational Expenditure* (*OPEX*) aproximadamente 10% inferior devido à eliminação de operadores embarcados, à otimização do uso de energia e à adoção de estratégias de manutenção preditiva. A flexibilidade operacional inerente a esse modelo permite ajustes ágeis frente a variações na demanda, como aumento de frequência em horários de pico ou resposta a interrupções inesperadas.

A automação em *GoA4* depende fortemente da integração de tecnologias avançadas, como sensores, redes de *IoT* e algoritmos de geolocalização. Esses recursos desempenham papel central no monitoramento e na gestão da localização não apenas dos trens, mas também de agentes de campo e ativos críticos, sobretudo em emergências, manutenção e atendimento ao cliente (BORLONI; COPCHE; SANTOS, 2019).

Conforme relatado pela AEAMESP (2016), o Metrô de São Paulo já avalia a implementação de sistemas destinados à detecção de interferências na via, o que reforça a relevância da localização precisa de equipes de manutenção e operação. Ocorrências como falhas elétricas, presença de objetos na via ou emergências médicas exigem uma resposta ágil, e a alocação eficiente da equipe mais próxima pode ser decisiva para a contenção de incidentes e a preservação da segurança operacional.

Tabela 1 - Grau de automação e responsabilidades operacionais no transporte metroferroviário

Fonte: Elaborado pelos autores (2025)

## 2.6 IOT E GEOLOCALIZAÇÃO

A integração entre dispositivos *IoT* e sistemas de geolocalização no Metrô de São Paulo representa uma solução estratégica para aprimorar o desempenho das operações, aumentar a segurança e melhorar o atendimento ao público. Com o auxílio da geolocalização em tempo real, é possível estabelecer uma comunicação ágil entre os dispositivos conectados, facilitando a identificação rápida da localização de funcionários e permitindo sua mobilização para situações emergenciais.  
 Diante da dinâmica dos ambientes metroviários, onde tempo e precisão são fatores críticos, a adoção de sistemas que identifiquem com exatidão a posição dos agentes operacionais contribui significativamente para o atendimento de ocorrências, otimização de recursos e aprimoramento da experiência dos usuários.

### 2.6.1 O Papel da Geolocalização em Sistemas IoT

Em sistemas IoT, a geolocalização vai além da simples obtenção de coordenadas geográficas. Ela permite monitorar dinamicamente a posição relativa de dispositivos e agentes, o que é fundamental para a resposta a ocorrências em tempo real e para a otimização de recursos em ambientes de alta complexidade.

A tecnologia Wi-Fi 6, amplamente difundida e compatível com a infraestrutura já existente no metrô, possibilita a identificação de zonas de cobertura com base na intensidade do sinal. Essa abordagem, empregada neste projeto, elimina a necessidade de sensores adicionais ou métodos de trilateração, reduzindo os custos e a complexidade de implantação.

Embora outras tecnologias, como o Bluetooth Low Energy (BLE), sejam aplicáveis a sistemas de localização indoor, sua utilização foi aqui limitada a fins comparativos. A escolha pela rede Wi-Fi se justifica pela robustez da cobertura, compatibilidade e simplicidade operacional, fatores essenciais em um ambiente de transporte público.

#### 2.6.1.2 Tecnologias para Geolocalização em Ambientes Internos

A geolocalização indoor enfrenta obstáculos técnicos, como a atenuação de sinal e a presença de múltiplos caminhos de propagação, que dificultam a aplicação de sistemas baseados em GPS. Como alternativa, o uso de redes Wi-Fi com análise de RSSI tem se mostrado eficaz.

Neste trabalho, adotou-se um modelo baseado na comparação da intensidade de sinal (RSSI) de múltiplos pontos de acesso, utilizando o microcontrolador ESP32-C6. A lógica de associação ao ponto com sinal predominante e envio do BSSID do ponto de acesso de rede conectado demonstrou ser funcional para determinar a zona de atuação do agente em tempo real, sem a necessidade de algoritmos complexos.

Soluções híbridas que integram Wi-Fi e BLE são comuns em outros contextos, mas os resultados obtidos neste projeto indicam que a estrutura baseada exclusivamente em Wi-Fi 6 é suficiente para garantir precisão e estabilidade necessárias, mantendo baixo consumo energético dos dispositivos e maior economia, uma vez que utiliza a infraestrutura já existente.

#### 2.6.1.3 Aplicações de Geolocalização em Emergências

A identificação da localização de agentes em tempo real é fundamental para respostas rápidas a eventos operacionais. O sistema desenvolvido neste trabalho permite, por meio da integração entre ESP32-C6, Wi-Fi 6 e o protocolo MQTT, identificar automaticamente os dispositivos mais próximos de uma ocorrência reportada.

Essa funcionalidade está implementada em dashboards desenvolvidos com as bibliotecas Dash e Streamlit, que exibem visualmente a posição dos agentes e incidentes no mapa da linha em operação. Essa solução proporciona decisões mais ágeis e assertivas, aumentando a eficiência operacional do centro de controle.

#### 2.6.1.4 Desafios na Implementação de Geolocalização

A principal dificuldade na aplicação de geolocalização no ambiente metroviário está relacionada às barreiras físicas e à interferência do sinal em túneis e estações subterrâneas. No entanto, a adoção do Wi-Fi 6 e a escolha de um sistema baseado em RSSI permitiram uma comunicação estável e eficiente mesmo nesses cenários adversos.

Além disso, a questão da privacidade foi considerada desde a concepção do projeto. O sistema limita o rastreamento a emergências ou manutenção, e os dados transmitidos são protegidos por autenticação e criptografia, em conformidade com a Lei Geral de Proteção de Dados (LGPD). Assim, a segurança da informação e o respeito aos direitos dos funcionários são garantidos.

A solução proposta apresenta, portanto, uma arquitetura escalável, de baixo custo e compatível com a realidade técnica do transporte público. Sua aplicação prática permite ganhos reais em agilidade, segurança e eficiência operacional, sendo uma alternativa viável para modernização da gestão em tempo real de agentes de campo.

## 2.7 ESP32 – C6: HARDWARE E FUNCIONALIDADES

### 2.7.1 Características Técnicas

O ESP32-C6 é um microcontrolador de baixo consumo desenvolvido pela Espressif Systems, voltado para aplicações em *Internet of Things* (*IoT*), com foco em conectividade sem fio de última geração e eficiência energética. Suas características o tornam especialmente adequado para sistemas de localização em tempo real, como o proposto neste trabalho.

O chip integra conectividade Wi-Fi 6 (802.11ax) operando na faixa de 2,4 GHz, com suporte aos modos de operação anteriores (Wi-Fi 4 – 802.11n, e Wi-Fi 3 – 802.11g). A tecnologia Wi-Fi 6 garante melhor desempenho em ambientes densamente povoados, maior confiabilidade de comunicação e otimização do uso de energia, fatores essenciais em cenários como estações de metrô, onde há múltiplos dispositivos conectados simultaneamente (ESPRESSIF SYSTEMS, 2023).

O núcleo de processamento do ESP32-C6 é baseado na arquitetura *RISC-V*, com largura de dados de 32 bits e frequência de até 160 MHz. Ele inclui 320 KB de ROM, 512 KB de SRAM e suporte a memória *flash* externa. O chip também disponibiliza múltiplas interfaces de comunicação, incluindo GPIOs programáveis (até 30, a depender do encapsulamento), ADCs de 12 bits, UART, SPI, I2C, I2S, PWM e sensores integrados, como medidor de temperatura (ESPRESSIF SYSTEMS, 2023).

Outro destaque é o suporte a modos de economia de energia, como *Light Sleep* e *Deep Sleep*, permitindo reduzir o consumo de corrente para valores extremamente baixos durante períodos de inatividade. No modo *Deep Sleep*, o ESP32-C6 pode operar com corrente inferior a 10 µA, o que o torna ideal para aplicações alimentadas por bateria ou que exijam operação contínua com baixa manutenção (ESPRESSIF SYSTEMS, 2023).

Esses recursos tornam o ESP32-C6 uma base tecnológica robusta para sistemas embarcados de localização, possibilitando coleta contínua de dados com confiabilidade e autonomia energética.

### 2.7.2 Funções de Geolocalização do ESP32

O ESP32-C6 possui um Software Development Kit (SDK), o Espressif IoT Development Framework (ESP-IDF), que possui funcionalidades para realizar as tarefas de geolocalização. Esses recursos permitem que o dispositivo obtenha informações úteis em cenários de localização indoor, como:

Identificação dos APs: O ESP32-C6 pode escanear redes Wi-Fi próximas para identificar APs disponíveis e registrar informações como SSID e BSSID. A força do sinal RSSI de cada AP é utilizada para determinar a proximidade do dispositivo ao ponto de acesso, essencial para triangulação em geolocalização sem GPS.

Sincronização e coleta de dados via ESP-NOW: O protocolo ESP-NOW, integrado ao ESP32-C6, permite comunicação eficiente entre dispositivos, sincronizando sensores e obtendo dados relevantes para geolocalização. Esse recurso é particularmente útil para aplicações em que a coleta de dados de múltiplos sensores é essencial.

Modos de operação otimizados para geolocalização: Utiliza modos de baixo consumo, como o Light Sleep, para operar sensores e realizar coletas periódicas de dados, equilibrando eficiência energética e desempenho.

Integração com outras tecnologias de sensores: O ESP32-C6 pode trabalhar com sensores adicionais, como acelerômetros e giroscópios para complementar dados de posição e melhorar a precisão da localização.

Figura 1 – Exemplo de dispositivos conectados simultaneamente via BLE

A black cell phone with bluetooth symbol on screen

Description automatically generated

*Fonte: Os autores (2025)*

### 2.7.3 Protocolos Suportados

O ESP32-C6 oferece suporte a múltiplos protocolos de comunicação sem fio, o que o torna uma plataforma altamente versátil para aplicações em *Internet of Things* (*IoT*), incluindo sistemas de localização em tempo real, redes de sensores e automação residencial.

O protocolo IEEE 802.11ax (Wi-Fi 6), operando na faixa de 2,4 GHz, proporciona maior eficiência na comunicação sem fio em ambientes com alta densidade de dispositivos. A tecnologia Wi-Fi 6 oferece vantagens como menor latência, maiorrendimento e economia de energia por meio de recursos como *Target Wake Time* (*TWT*), que contribuem para aplicações que exigem atualização contínua de dados com baixo consumo energético (ESPRESSIF SYSTEMS, 2023).

Além do Wi-Fi, o chip suporta o protocolo proprietário *ESP-NOW*, que permite comunicação sem conexãoentre dispositivos ESP, eliminando a necessidade de associação à rede. O *ESP-NOW* é ideal para aplicações com envio de pacotes curtos, com até 250 bytes, e baixa latência, sendo amplamente utilizado em cenários de transmissão de dados entre sensores e atuadores (ESPRESSIF SYSTEMS, 2023).

O ESP32-C6 também integra suporte ao padrão IEEE 802.15.4, permitindo a implementação de tecnologias como *Zigbee 3.0* e *Thread*. Esse suporte o torna compatível com redes *mesh* robustas e de baixo consumo energético, amplamente adotadas em ambientes domésticos e industriais, além de viabilizar a integração com o ecossistema *Matter* para automação interoperável (ESPRESSIF SYSTEMS, 2023).

Esses protocolos garantem que o ESP32-C6 possa ser utilizado em uma ampla variedade de cenários de conectividade, incluindo ambientes com infraestrutura Wi-Fi existente ou que demandem comunicação direta entre dispositivos, mesmo em condições de rede limitada.

## 2.8 SOFTWARES E TECNOLOGIAS AUXILIARES

A implementação do sistema de geolocalização proposto neste trabalho contou com o suporte de plataformas e bibliotecas especializadas, utilizadas para o desenvolvimento embarcado, a comunicação entre dispositivos e a visualização em tempo real dos dados. Essas ferramentas foram escolhidas com base na compatibilidade com o microcontrolador ESP32-C6, no licenciamento de código aberto e na capacidade de operar em ambientes de alta densidade de dispositivos, como o metrô de São Paulo.

O *Espressif IoT Development Framework* (ESP-IDF) foi a principal plataforma utilizada no desenvolvimento embarcado. Trata-se do *framework* oficial daEspressif Systems, voltado para programação de microcontroladores da série ESP32. Ele permite controle direto das interfaces de comunicação sem fio, como Wi-Fi, e fornece suporte nativo para leitura de parâmetros essenciais à geolocalização, como o *Received Signal Strength Indicator* (RSSI) e o *Basic Service Set Identifier* (BSSID). Essas funcionalidades viabilizaram a coleta precisa dos dados de sinal utilizados no sistema.

A distribuição do ESP-IDF sob licença *Apache 2.0* permite uso irrestrito em ambientes acadêmicos e industriais, favorecendo a construção de soluções personalizadas sem custos de licenciamento. No contexto da Fundação Educacional Inaciana Padre Sabóia de Medeiros (FEI), essa característica possibilitou maior autonomia técnica no desenvolvimento do protótipo, além de reforçar a viabilidade da proposta em aplicações reais.

A comunicação entre os dispositivos ESP32-C6 e o sistema de supervisão foi realizada por meio do protocolo *Message Queuing Telemetry Transport* (MQTT), amplamente utilizado em aplicações de *Internet IoT* devido à sua leveza, escalabilidade e suporte a atualizações frequentes. O envio de dados como RSSI, BSSID e carga da bateria foi feito por meio de tópicos específicos em um *broker*, permitindo o tratamento e a visualização dessas informações em tempo real.

Para a construção da interface visual de acompanhamento da localização dos agentes, foram utilizadas as bibliotecas *Streamlit* e *Dash*, ambas baseadas em Python. A primeira foi utilizada na prototipação da aplicação, dada sua simplicidade na criação de interfaces web a partir de scripts lineares. Já a biblioteca *Dash* foi escolhida para a versão final, devido ao seu desempenho superior na atualização dinâmica de dados e à integração estável com o protocolo MQTT.

A junção dessas tecnologias resultou em uma solução robusta, modular e responsiva, capaz de operar com confiabilidade em ambientes internos complexos, como estações e túneis metroviários. Além disso, a escolha por ferramentas de código aberto e amplamente utilizadas no setor reforça a possibilidade de expansão futura do sistema, tanto em escala quanto em funcionalidades.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Figura 2 – Proposta inicial de Dashboard elaborado

Fonte: Elaborado pelos autores (2025)

### 2.8.1 Ferramentas e Bibliotecas do ESP32-C6 para WI-FI

Para as aplicações com Wi-Fi, o ESP32-C oferece uma *Application Programming Interface* (API) (APIs fornecem interfaces para que funções ou recursos específicos de um software possam ser acessados sem que o programador precise entender os detalhes internos de como esses recursos são implementados) permitindo, por exemplo, ser configurado nos modos estação do Sistema de Transferência de Arquivos (STA) e o AP.

Além disso, a plataforma oferece funções para escanear redes Wi-Fi, conectar-se a essas redes e obter informações detalhadas sobre as redes disponíveis. Para este estudo, duas funcionalidades cruciais são a leitura de RSSI e a identificação de BSSIDs, que são frequentemente usadas em aplicações de localização interna.

A função que mais será utilizada ao decorrer deste trabalho será a esp\_wifi\_scan, utilizada para iniciar a varredura de redes Wi-Fi. Após a varredura da realizada nas redes Wi-Fi, ao utilizar a função esp\_wifi\_scan\_get\_ap\_records é possível obter os registros das redes escaneadas, incluindo o BSSID e RSSI, para serem utilizados posteriormente no código.

Essas informações são essenciais para determinar a intensidade do sinal de cada ponto de acesso e identificar as redes à medida que os dispositivos se movem pelo ambiente.

### 2.8.2 Biblioteca Streamlit do Python

O Streamlit é um framework open source desenvolvido em Python, projetado para a criação rápida e eficiente de aplicações web interativas, com foco em ciência de dados, aprendizado de máquina e visualização de informações. Diferentemente de frameworks tradicionais como Flask ou Django, que exigem uma separação clara entre as camadas de frontend, backend e roteamento, o Streamlit permite que interfaces gráficas sejam construídas diretamente a partir de scripts Python, dispensando o uso de HTML, CSS ou JavaScript.

No contexto do desenvolvimento do dashboard de monitoramento da linha de metrô incluindo a exibição do mapa, a localização de funcionários, registros de incidentes e outros pontos críticos o Streamlit se mostrou uma ferramenta fundamental. Sua capacidade de simplificar o desenvolvimento de páginas web e a posterior publicação em ambientes de cloud facilitou significativamente a integração dos dados e a criação de uma interface acessível e responsiva.

A escolha do Streamlit como base para o sistema se justifica, principalmente, por sua facilidade de uso e flexibilidade. Com poucos comandos, é possível construir dashboards, exibir gráficos, tabelas, mapas e adicionar componentes interativos como botões, sliders, caixas de seleção, entre outros. O fluxo de trabalho é intuitivo: o desenvolvedor escreve o código normalmente em Python, e o Streamlit renderiza automaticamente os elementos visuais no navegador. Essa abordagem contribuiu para uma prototipação ágil e eficaz da aplicação proposta neste trabalho.

### 2.8.3 Biblioteca Dash do Python

O Dash é uma biblioteca do Python desenvolvida pela empresa Plotly que permite a criação de aplicações web interativas e altamente customizáveis. Sendo uma alternativa robusta aos frameworks mais complexos como Django e Flask. O frontend e o backend são programados diretamente com a linguagem Python, não sendo necessário conhecimentos em HTML, CSS e JavaScript.

No contexto do trabalho, se mostrou uma alternativa viável para a construção do sistema de monitoramento da localização dos dispositivos na linha do Metrô, pois possibilita a integração com bases de dados e atualizações em tempo real, além da criação de layouts personalizados e adição de botões, dropdowns, sliders para interação com o usuário.

### 2.8.4 Broker Mosquitto

O *broker* Mosquitto foi utilizado neste trabalho como elemento central na comunicação entre os dispositivos ESP32-C6 e os sistemas de supervisão e visualização. Trata-se de uma implementação leve e eficiente do protocolo MQTT, desenvolvida pela Eclipse Foundation, amplamente empregada em aplicações de IoT por sua escalabilidade, baixa latência e compatibilidade com diferentes plataformas (ECLIPSE, 2023).

No contexto do sistema proposto, o Mosquitto atua como intermediador na troca de mensagens entre os dispositivos de campo e os painéis desenvolvidos com as bibliotecas Streamlit e Dash. Ele recebe e distribui dados como RSSI, BSSID e o nível da bateria dos dispositivos, por meio de tópicos MQTT publicados em tempo real. Essa estrutura possibilitou uma comunicação assíncrona e responsiva, essencial para aplicações que exigem monitoramento contínuo, como ambientes metroviários (INPIXON, 2023).

Durante os testes realizados, o *broker* foi configurado em rede local, o que garantiu uma operação segura e independente de conexão com a internet. Essa configuração reduz a latência, elimina a dependência de infraestrutura externa e contribui para o aumento da segurança dos dados trafegados. Além disso, o Mosquitto suporta autenticação e criptografia por meio do protocolo TLS, assegurando conformidade com a Lei Geral de Proteção de Dados (LGPD), conforme orientações de VIEIRA (2020) e PINHEIRO JUNIOR; CAVALHEIRO (2020).

A robustez da solução foi evidenciada pela estabilidade na recepção e publicação de mensagens MQTT, sem perda de pacotes significativos mesmo em cenários simulados de alta densidade de dispositivos. Sua integração com os *dashboards* desenvolvidos também demonstrou facilidade de uso e compatibilidade com bibliotecas modernas em Python (STREAMLIT, 2023; DASH, 2023), consolidando o Mosquitto como um componente estratégico na arquitetura geral do sistema.

### 2.8.4 Elipse Software

O *Elipse EPM* é uma plataforma classificada como *Plant Information Management System* (*PIMS*), amplamente utilizada para análise, gerenciamento e visualização de dados industriais em tempo real. Reconhecida por sua eficiência, flexibilidade e escalabilidade, a solução é aplicada em diversos setores — como energia, transporte, manufatura e automação industrial — e já integra o ecossistema tecnológico do Metrô de São Paulo, sendo empregada em suas operações para coleta e análise de dados operacionais (ELIPSE SOFTWARE, 2023).

No contexto deste Trabalho de Conclusão de Curso, o *Elipse EPM* é considerado como uma plataforma estratégica para futura integração com o sistema de geolocalização desenvolvido com o *ESP32-C6*. A plataforma oferece uma interface robusta e personalizável para monitoramento em tempo real, sendo ideal para a análise dos dados de *Received Signal Strength Indicator* (*RSSI*) coletados pelos dispositivos de localização instalados no ambiente ferroviário.

Sua utilização potencial permitirá centralizar e visualizar os dados de geolocalização com precisão, gerar relatórios gráficos, detectar padrões de movimentação dos funcionários e implementar análises preditivas voltadas à segurança operacional. Com a integração ao *Elipse EPM*, o sistema poderá evoluir de uma solução de localização local para um recurso amplamente gerenciável em ambientes SCADA e PIMS já utilizados institucionalmente.

Além disso, o *Elipse EPM* permite integração com bancos de dados relacionais, sistemas externos e plataformas de inteligência artificial. Essas funcionalidades proporcionam uma infraestrutura ideal para o crescimento do projeto, permitindo que os dados capturados pelos dispositivos *IoT* sejam transformados em insights operacionais relevantes. Alarmes automatizados, painéis personalizados e manutenção preditiva baseada em dados históricos são algumas das aplicações possíveis com essa integração futura.

Assim, o uso do *Elipse EPM* representa uma possibilidade concreta e coerente com os sistemas já adotados pelo Metrô de São Paulo, agregando valor ao projeto pela compatibilidade técnica e pela capacidade de escalar a solução para níveis corporativos de análise e controle.

### 2.8.5 Sistema SCADA vs. PIMS

A escolha do PIMS, como o Elipse EPM, para o monitoramento do sistema de geolocalização em tempo real dos funcionários no metrô, em vez de um Supervisory Control and Data Acquisition (SCADA), como o Elipse E3, é justificada pelas necessidades analíticas e preditivas do projeto. Enquanto um sistema SCADA é voltado para controle e monitoramento de missão crítica, como em subestações elétricas ou sistemas de bombeamento, ele não é a solução mais adequada para ambientes onde a análise avançada e a manipulação de dados são essenciais.

O PIMS, como o Elipse EPM, permite não apenas visualizar a localização dos funcionários em tempo real, mas também realizar análises, cálculos, personalizar relatórios e adaptar a visualização dos dados às necessidades específicas do projeto. Isso faz com que seja a ferramenta ideal para um analista de operação ou manutenção, que precisa extrair insights valiosos a partir dos dados coletados. Além disso, a capacidade de integrar informações de diversos sensores e fontes torna o PIMS indispensável para uma abordagem mais preditiva e estratégica.

Enquanto o SCADA, como o Elipse E3, é ideal para funções de controle imediato, ele limita a manipulação de dados em tempo real e se restringe às telas e relatórios predefinidos. Por outro lado, o PIMS oferece flexibilidade para análises avançadas e adaptação contínua, sendo mais adequado para otimizar processos operacionais e de manutenção no ambiente do metrô.

Portanto, o PIMS, com sua capacidade analítica superior e integração avançada, é a solução mais apropriada para o projeto proposto, possibilitando não apenas a segurança e eficiência operacional em tempo real, mas também análises preditivas e otimização estratégica, aspectos essenciais para o sucesso do sistema de geolocalização no metrô.

# [3 METODOLOGIA](#Indicador42)

A metodologia adotada neste trabalho baseia-se no desenvolvimento e validação de um sistema de localização de agentes operacionais por meio da infraestrutura de rede Wi-Fi 6 já existente no ambiente do Metrô de São Paulo. O projeto foi estruturado em três frentes principais: desenvolvimento embarcado, comunicação entre dispositivos e interface supervisória.

Para a coleta de dados, foram utilizados dispositivos embarcados baseados no microcontrolador ESP32-C6, configurados para realizar varreduras periódicas das redes Wi-Fi disponíveis. Esses dispositivos identificam o ponto de acesso (*Access Point* – AP) com o sinal mais forte, registram o valor de intensidade do sinal (*Received Signal Strength Indicator* – RSSI) e o identificador do AP (*Basic Service Set Identifier* – BSSID). Essas informações, acompanhadas de um identificador único do dispositivo, são publicadas em um tópico MQTT hospedado em um *broker* Mosquitto instalado em uma máquina com sistema operacional Linux.

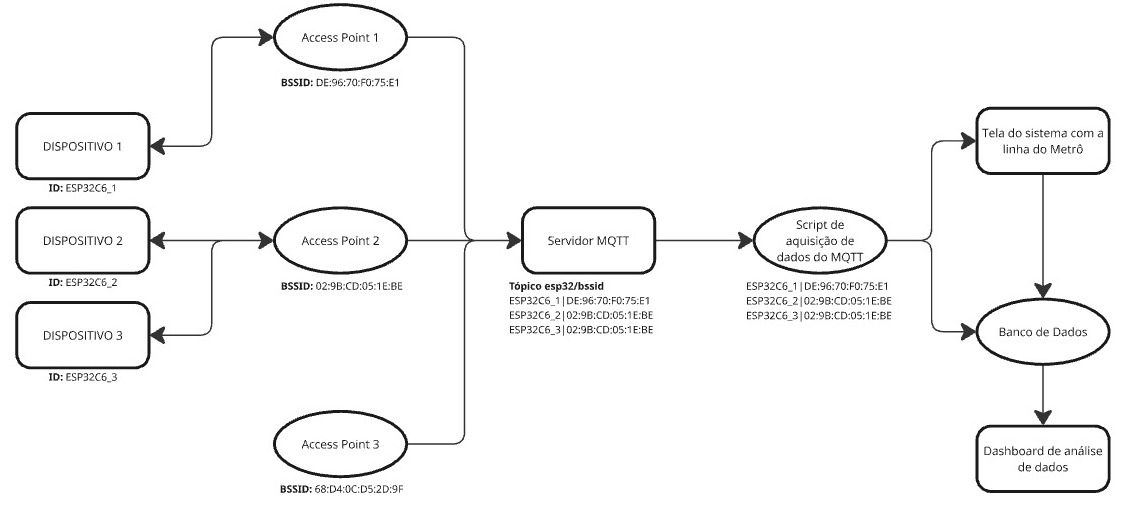
A aplicação supervisória, desenvolvida em Python com as bibliotecas Streamlit e Dash, é executada na mesma máquina que hospeda o *broker*. Ela se inscreve no tópico “esp32/bssid” para receber em tempo real os dados dos dispositivos. Com base em um mapeamento prévio das localizações associadas aos BSSIDs, o sistema estima a posição de cada dispositivo e exibe essas informações visualmente em um mapa representativo da linha do metrô.

Além da visualização contínua, a aplicação permite o registro de incidentes por parte de usuários e a seleção de ocorrências para tratativa. Quando um incidente é selecionado, o sistema realiza a análise das localizações dos dispositivos conectados e destaca, no mapa, o agente mais próximo da ocorrência. Esse fluxo visa apoiar a tomada de decisão operacional de forma ágil e fundamentada.

Os dados gerados durante a operação do sistema — incluindo o histórico de incidentes e de localizações são armazenados em um banco de dados SQL. Esse banco alimenta um *dashboard* analítico voltado à avaliação posterior de desempenho, cobertura e padrões operacionais, contribuindo para futuras melhorias nos processos da organização.

Todas as etapas de desenvolvimento foram conduzidas pelos autores, desde a prototipação até os testes em bancada, que incluíram simulações de falha de conexão Wi-Fi, variações de intensidade de sinal e envio contínuo de dados via MQTT. O sistema apresentou estabilidade de comunicação, baixa latência e autonomia energética adequada para operação prolongada, validando sua viabilidade para aplicação em ambientes operacionais críticos, como o metrô.

Figura 3 - Fluxograma do Funcionamento do Projeto

Fonte: Os autores (2025)

## 3.1 ALGORITMO DO DISPOSITIVO

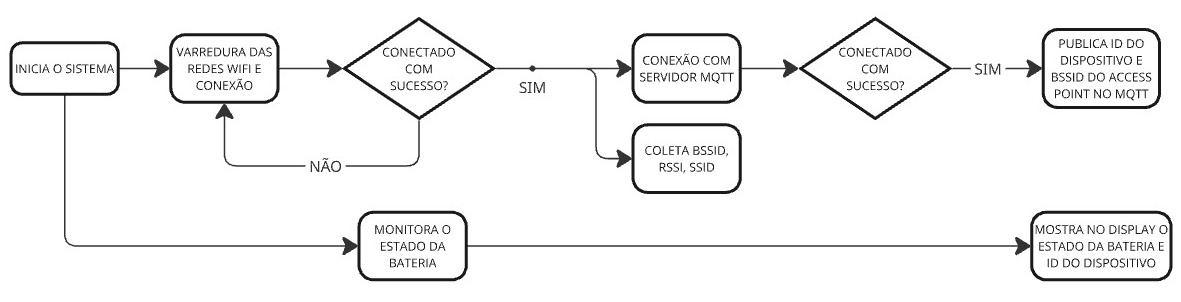
O funcionamento do sistema inicia-se com a ativação dos módulos de hardware e software necessários, incluindo a interface Wi-Fi e o suporte ao protocolo MQTT. O dispositivo então tenta estabelecer conexão com a rede Wifi e ao servidor MQTT. Caso a conexão não seja bem-sucedida, o algoritmo implementa um mecanismo de tentativas, assegurando a resiliência do sistema (SOPHIA et al., 2021).

Uma vez conectado à rede Wi-Fi do Metrô, com o intuito de coletar parâmetros fundamentais, como o identificador BSSID, que é atrelado ao ID configurado no dispositivo para envio ao servidor MQTT.

Além da coleta de sinal, o sistema realiza monitoramento contínuo do estado da bateria do dispositivo, função essencial para aplicações *IoT* que operam em cenários sem fonte de alimentação constante. Esse monitoramento possibilita ações de alerta ao usuário quando a carga atinge níveis críticos.

O algoritmo foi projetado para operar em ciclos de 10 segundos, equilibrando a frequência de atualização com o consumo energético. Essa periodicidade assegura a atualização contínua das informações sem comprometer a autonomia do equipamento.

O fluxograma do funcionamento da programação do dispositivo se encontra abaixo:

Figura 4 - Algoritmo do Microcontrolador

Fonte: Elaborado pelos autores (2025)

## [3.2 mqtt broker](#Indicador43)

O Mosquitto MQTT foi instalado e configurado de forma a corresponder aos parâmetros necessários no ESP32-C6, como IP e porta utilizados. Dessa forma a comunicação entre o servidor MQTT, dispositivos e aplicações podem ocorrer sem problemas, desde que estejam todos na mesma rede.

## 3.3 ABORDAGEM UTILIZADA

Neste projeto, o objetivo principal é identificar a localização do dispositivo utilizando os pontos de acesso Wi-Fi. A proposta não visa determinar a localização de forma precisa, mas sim fornecer uma localização relativa funcional, adequada às operações internas do sistema metroviário. Assim, foram adotadas as seguintes diretrizes:

* Trilateração não será utilizada: não há necessidade de estimar posições cartesianas com base em múltiplas distâncias. O sistema identifica apenas ponto de acesso Wi-Fi no qual o dispositivo do funcionário está conectado;
* SCADA não será utilizado na presente fase: apesar de sua ampla aplicação industrial, o sistema supervisório SCADA foi substituído por uma abordagem mais leve, baseado em código aberto. No entanto, sua implementação futura é considerada viável e foi levada em consideração.

## 3.4 Ambientes de Teste e Redes IoT

A necessidade de uma rede confiável torna-se ainda mais crítica em ambientes urbanos complexos como o metrô, onde obstáculos físicos (como estruturas metálicas, paredes e variações de profundidade) afetam a propagação do sinal. Esses desafios exigem soluções técnicas que garantam cobertura estável e desempenho consistente da rede, de forma que o sistema *IoT* opere com precisão e baixa latência, mesmo diante das limitações impostas pela infraestrutura existente (MOURA, 2007; GONÇALVES, 2021).

Para validar a efetividade da solução proposta, foram conduzidos testes preliminares em ambiente controlado, utilizando roteadores e repetidores que simulam os pontos de acesso de rede Wi-Fi 6 do Metrô. Os roteadores e repetidores foram configurados de forma que funcionem como AP da mesma rede, dessa forma a máquina Linux e os dispositivos conseguem se conectar a ela e se comunicarem.

## 3.5 Ambientes de Teste e Rede

A primeira etapa da montagem de uma rede de teste envolve a escolha dos equipamentos necessários. Para o ambiente de testes foram utilizados roteadores e repetidores comuns residenciais disponíveis para os testes. Foram configurados de tal forma que fosse possível simular a utilização real no ambiente final.

## 3.6 LISTA DE MATERIAIS

Para o desenvolvimento do dispositivo de geolocalização IoT que será utilizado pelos agentes circulantes do metrô de São Paulo, foram utilizados os materiais descritos na tabela abaixo.

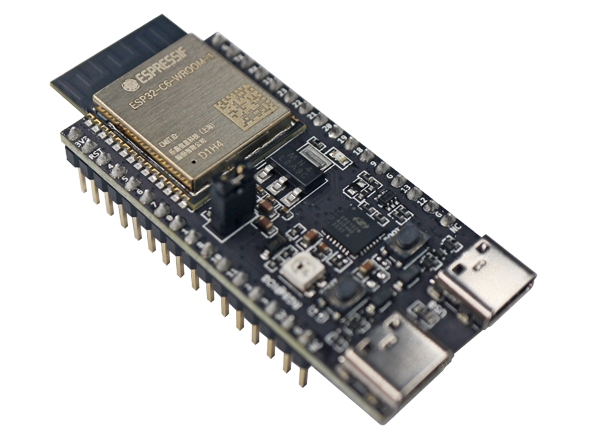
Tabela 2 - Lista de Materiais

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | **Quantidade** | **Valor Unitário (R$)** | **Valor Total (R$)** |
| Bateria Li-Ion 18650 3,7V 2600mAh | 4 | 39,90 | 159,6 |
| Carregador Duplo para Bateria Li-Ion 18650 | 2 | 35,00 | 70,00 |
| Suporte para 2 Baterias Li-Ion 18650 | 2 | 10,50 | 21,00 |
| Placa DevKitC-1 ESP32-C6 N8 | 2 | 78,00 | 156,00 |
| Display OLED 128x32 Px - 0.91" - 4 Pin - Azul | 2 | 19,90 | 39,80 |
| Protoboard 830 Pontos | 1 | 13,30 | 13,30 |
| Kit Jumper Macho Macho - 65 pçs | 1 | 8,90 | 8,90 |
| Kit Jumper Fêmea Fêmea - 40 pçs - 20cm | 1 | 8,40 | 8,40 |
| Kit Jumper Macho Fêmea - 40 pçs - 20cm | 1 | 8,40 | 8,40 |
| Carregador Bateria Lithium 1A com Proteção - USB C -TP4056 | 4 | 4,50 | 18,00 |
| Placa de Circuito Impresso | 2 | 50,00 | 100,00 |
| Conversor step-up (boost) DC-DC – XL6009 | 2 | 13,50 | 27,00 |
| Preço Total | R$ 627,70 | | |

Fonte: Elaborado pelos autores (2025)

**DevKitC-1 ESP2-C6 N8:** A placa é baseada no microcontrolador ESP32-C6, sendo projetada para aplicações em sistemas embarcados com conectividade sem fio. Possui suporte a Wi-Fi 6 (802.11ax), o que possibilita a comunicação com alta eficiência e baixo consumo de energia, fatores essenciais para sistemas de geolocalização em tempo real. Conta com 8 MB de memória Flash, SRAM integrada e porta USB Type-C, sendo ideal para projetos compactos e ambientes com alta densidade de dispositivos conectados (ESPRESSIF SYSTEMS, 2023).

Figura 5 - Imagem Placa DevKitC-1 ESP-C6 N8



Fonte: ESPRESSIF.2023

**Bateria Li-Ion 18650 3,7V 2600mAh:** A bateria recarregável do tipo 18650 é amplamente utilizada por sua alta densidade energética, confiabilidade e capacidade de fornecer corrente contínua por longos períodos. Com tensão nominal de 3,7 V e capacidade de 2600mAh, ela garante autonomia para dispositivos embarcados que operam em campo sem fonte de energia externa.

Figura 6 - Imagem Bateria Li-Ion 18650 3,7V 2500mAh 2C

Foto em preto e branco

Descrição gerada automaticamente com confiança média

Fonte: BATERIA Li-Ion 18650 3,7V 2500mAh 2C. 2024.

**Carregador Duplo para Bateria Li-Ion 18650:** Este carregador permite o reabastecimento simultâneo de até duas baterias 18650, sendo um componente essencial para garantir a disponibilidade de energia nos ciclos de operação contínua do dispositivo. A recarga eficiente e segura das baterias é vital para garantir o desempenho consistente do sistema.

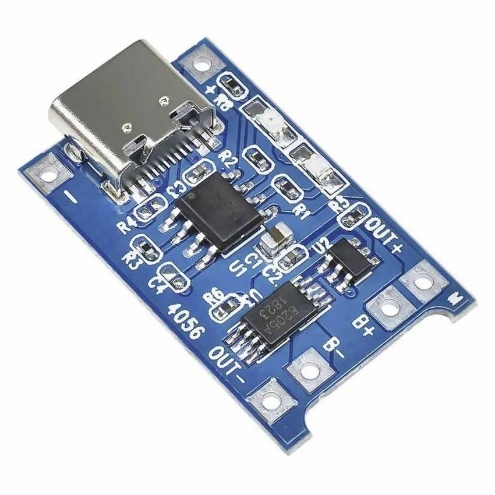
Figura 7 - Carregador Duplo para Bateria Li-Ion 18650



Fonte: Carregador duplo para bateria Li-Ion 18650.

**Carregador Bateria Lithium 1A com Proteção - USB C - TP4056:** Módulo eletrônico para recarga de baterias Li-Ion via porta USB-C. Conta com circuito de proteção contra sobrecarga, sobrecorrente e curto-circuito, sendo responsável por garantir a segurança do sistema de alimentação do dispositivo de geolocalização.

Figura 8 - Carregador Bateria Lithium 1A com Proteção - USB C -TP4056

  
Fonte: Carregador Bateria Lithium 1A com Proteção – USB C – TP4056

**Módulo Regulador de Tensão Step Up XL6009:** Conversor step-up (boost) DC-DC capaz de elevar tensões de entrada entre 3,5 V e 32 V para saídas ajustáveis de até 35 V. Ele oferece alta eficiência e corrente de saída de até 2 A, ideal para alimentar microcontroladores e periféricos a partir de baterias Li-ion. Possui ajuste por trimpot e proteção contra sobreaquecimento.

Figura 9 - Conversor step-up (boost) DC-DC – XL6009

Uma imagem contendo circuito

O conteúdo gerado por IA pode estar incorreto.  
Fonte: step-up-xl6009

## 3.7 TESTES DO SISTEMA DE LOCALIZAÇÃO DO SISTEMA IOT

### 3.7.1 Sistema de Monitoramento da Geolocalização

A imagem apresentada na Figura 9 refere-se à primeira tentativa de desenvolvimento de uma interface de acompanhamento do sistema de geolocalização, baseada na visualização da Linha 17–Ouro do Metrô de São Paulo. Essa interface foi projetada para funcionar em ambiente SCADA, com o objetivo de integrar o sistema de localização ao painel institucional já utilizado pelo Metrô. No entanto, após reuniões com os responsáveis técnicos da companhia, optou-se por não seguir com essa abordagem na versão final do projeto, uma vez que foi acordado o investimento em ferramentas mais flexíveis e desvinculadas da arquitetura SCADA tradicional. Apesar de não ter sido adotada, essa alternativa permitiu testar a potencialidade do ambiente SCADA e demonstrou que há viabilidade para implementar variáveis e recursos capazes de suportar um sistema de localização em tempo real, caso futuramente se deseje expandir a aplicação para as plataformas oficiais de controle operacional. Essa etapa de experimentação contribuiu para a validação técnica do conceito e fundamentou a escolha por soluções mais abertas e ágeis para o protótipo final.

Figura 10 - Interface desenvolvida através do Elipse E3.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

Com o objetivo de desenvolver um sistema simplificado para acompanhamento do mapa de localizações do metrô, utilizou-se a interface fornecida pela biblioteca Streamlit, da linguagem Python. Através da integração com a API do OpenStreetMap e utilizando as coordenadas conhecidas da Linha Ouro do metrô, foi testada a justaposição de diferentes ícones informativos: um ícone verde representa os trens em deslocamento; ícones vermelhos indicam as estações; e ícones azuis sinalizam os Access Points presentes em cada estação.

Além disso, foi implementada uma interface lateral à esquerda, permitindo ao usuário filtrar categorias de incidentes reportáveis pela aplicação, como ocorrências de segurança ou de manutenção. A Figura 10, apresenta a interface web de acompanhamento desenvolvida por meio da codificação:

Figura 11 - Interface do software desenvolvido.

Mapa

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

Para complementação da solução, foi adicionado ao final da rolagem da interface, a opção de partir da seleção de uma estação específica e da categoria de incidente (segurança ou manutenção), a identificação dos usuários conectados aos Access Points (APs) mais próximos e as ordenações com base na distância até o local selecionado. A lista resultante (Figura 11) exibe a identificação dos APs e a distância em metros, facilitando a alocação eficiente de recursos para resposta rápida. Além disso, a interface inclui um painel lateral de filtros e apresenta atualizações periódicas automáticas, promovendo maior dinamismo e eficiência no acompanhamento das ocorrências reportadas.

Figura 12 - Tela de agentes mais próximos a incidente.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

Toda a interface em Streamlit foi desenvolvida de forma simulada, com os dados de localização dos dispositivos de forma fixa. Ao realizar a integração com o servidor MQTT e os dispositivos reais realizando a subida dos dados, o ambiente desenvolvido apresentou instabilidades e não se comportou da forma desejada. Não realizou as atualizações das posições dos dispositivos em tempo real e não calculou as distâncias.

Para contornar essa limitação de atualização e coleta de dados, foi usado a biblioteca Dash, da linguagem Python, com o objetivo de exibir em tempo real a localização dos dispositivos ESP32 conectados via protocolo MQTT. A aplicação permite ao usuário reportar incidentes por estação, como exemplificado na estação Morumbi, além de registrar automaticamente o horário da ocorrência e destacar o dispositivo mais próximo do ponto reportado ao ser selecionado na tela um dos incidentes em aberto.

O sistema utiliza um mapa interativo para representar graficamente os dispositivos, estações e tipos de evento com diferentes ícones. O trem foi simulado no sistema e se movimenta periodicamente entre as estações, sendo também uma das opções de local para ser escolhida como opção para se reportar um novo incidente.

Na Figura 11, é exemplificada a aplicação do sistema por parte do operador do Metrô em que ao ser selecionado o incidente na estação “Morumbi”, foi demonstrado que mesmo havendo dois dispositivos conectados e próximos ao local do incidente, o dispositivo ESP32C6\_2 era o mais próximo entre eles. Representando que o agente circulante com esse dispositivo poderia ser acionado e indicado para se mover ao local. Dessa forma, reduzindo o tempo de resposta para o incidente.

Figura 13 - Interface gráfica em Dash, com o local de reportar um novo incidente, incidentes registrados, e mapa da linha do Metrô com a localização de dois dispositivos.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

### 3.7.2 Testes do sistema para verificação de erros

Durante a etapa de testes, foi realizada a verificação da comunicação entre os dispositivos ESP32-C6 e os pontos de acesso (Access Points) por meio do protocolo MQTT. A imagem abaixo mostra os dados recebidos pela aplicação Python, confirmando que os pacotes de dados enviados pelos dispositivos foram corretamente processados e reconhecidos pelo broker. Os testes evidenciaram que o ESP32-C6 foi capaz de identificar e se conectar aos Access Points disponíveis, retornando as informações de endereço MAC do ponto de acesso ao qual o dispositivo estava conectado no momento da transmissão. A comunicação ocorreu de forma estável e contínua, com múltiplos dispositivos (identificados como ESP32C6\_1 e ESP32C6\_2) alternando a conexão entre diferentes Access Points, o que comprova a eficácia da implementação e a correta integração do hardware com a rede sem fio configurada para o ambiente de testes.

Figura 14 - Localização dos dispositivos sendo recebida pela aplicação Python através do MQTT

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

Na etapa de testes de comunicação e integração com a interface desenvolvida em Dash, foi possível validar em tempo real a troca de dados entre os dispositivos ESP32-C6 e o sistema de monitoramento via protocolo MQTT. A Figura 14 demonstra dois pontos principais: à esquerda, a interface gráfica exibe a localização dos Access Points no mapa, com atualização contínua da lista de dispositivos conectados (ESP32C6\_1 e ESP32C6\_2), bem como a última hora de recepção de dados.

À direita, o terminal registra as mensagens recebidas do broker, identificando os dispositivos e os respectivos endereços MAC dos Access Points aos quais estavam conectados. Essa integração permitiu visualizar de forma sincronizada os dados enviados pelos dispositivos e recebidos pela interface, garantindo a confiabilidade da comunicação entre o hardware e o sistema de visualização. O teste comprova que a aplicação é capaz de monitorar múltiplos dispositivos simultaneamente e de exibir as informações com precisão em ambiente dinâmico e instantaneamente.

Figura 15 - Interface com dados de localização recebidos

Interface gráfica do usuário, Texto

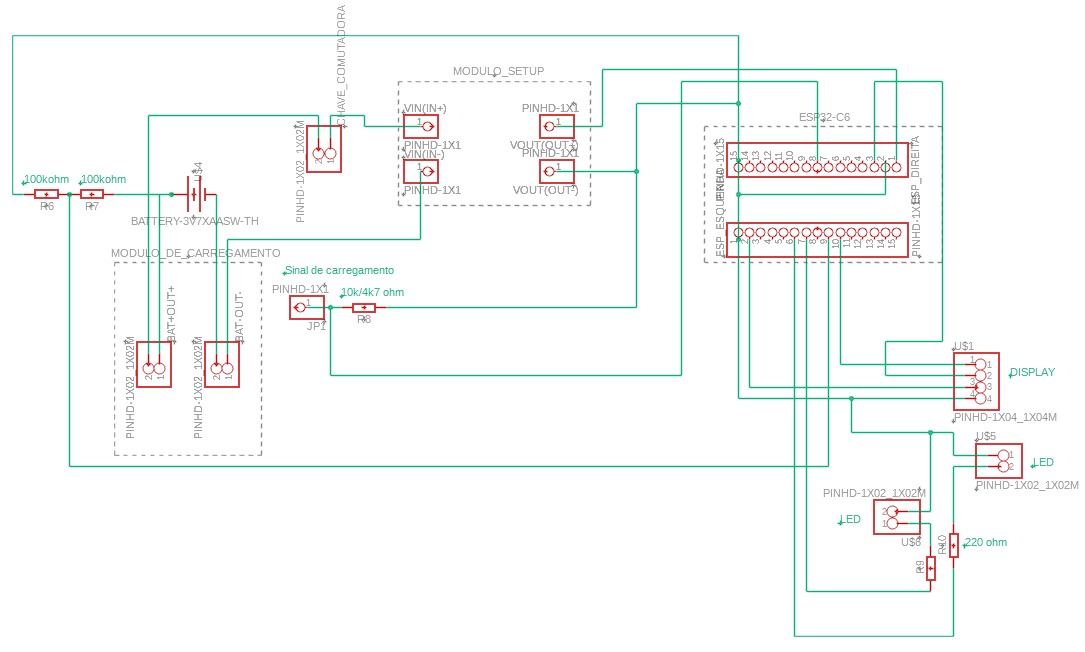
O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

### 3.7.3 Protótipo do Dispositivo de Geolocalização

Desenvolveu-se o diagrama elétrico apresentado na Figura 15 por meio do AutoDesk Fusion 360 para representar o sistema elétrico do hardware e realizar os testes do protótipo.

Figura 16 - Hardware Projetado



Fonte: Elaborado pelos autores (2025)

Após a definição e obtenção dos componentes necessários para a solução, foi elaborado o teste de alimentação do ESP32-C6 (Figura 17), conectado ao módulo de carga e acoplado ao módulo de Setup, com o objetivo de determinar o nível da bateria e viabilizar a indicação dessa informação durante o uso do dispositivo e proteção dele no uso

Figura 17 - Protótipo do teste de carga

Uma imagem contendo Diagrama

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

No circuito montado na protoboard, foi implementada a prototipação do sistema completo, que servirá como solução para a problemática levantada. Esse sistema inclui o esquema de montagem do protótipo e teste de carga, conforme discutido anteriormente, e, adicionalmente, componentes de interatividade externa, como o display OLED e dois LEDs, destinados à comunicação visual com o agente circulante.

Figura 18 - Protótipo em Protoboard Finalizado

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

## **3.8** HARDWARE

O hardware desenvolvido neste projeto foi concebido para garantir operação contínua em ambientes operacionais como o metrô, com ênfase em autonomia energética, confiabilidade na comunicação e sinalização visual ao usuário. O sistema é baseado no microcontrolador ESP32-C6, compatível com o padrão Wi-Fi 6, conforme descrito nos tópicos anteriores.

A alimentação do dispositivo é fornecida por duas baterias recarregáveis do tipo Li-Ion 18650, com tensão nominal de 3,7 V, acopladas a um suporte com chave liga/desliga. O circuito de recarga utiliza o módulo TP4056 com porta USB-C, que inclui proteção contra sobrecarga e sobrecorrente. Um sistema de monitoramento de tensão foi implementado, permitindo a leitura da carga da bateria por meio de pino analógico, com envio periódico dos dados ao painel supervisório via protocolo MQTT.

Para permitir o funcionamento do sistema tanto com baterias quanto com alimentação direta via USB, foi integrado um conversor DC-DC Step-Up com saída variável ajustável entre 4,75 V e 5,4 V, garantindo alimentação estável ao conjunto.

O dispositivo inclui também dois LEDs indicadores e um display OLED de 0,91” (128x32 px), utilizados para sinalizar o status de operação, conectividade e nível de carga. O encapsulamento foi projetado com aberturas específicas para os LEDs, display e porta USB-C, além de possibilitar a fixação do equipamento ao uniforme dos agentes.

O protótipo foi submetido a testes em bancada, contemplando simulações de perda e restabelecimento de conexão Wi-Fi, bem como o envio contínuo de dados via MQTT. Os resultados demonstraram estabilidade na comunicação, operação ininterrupta e autonomia energética estimada em aproximadamente 11 horas com uma única carga, confirmando a viabilidade da solução em cenários operacionais reais, como o ambiente metroviário.

#### 3.8.1.1 ESTIMATIVA DE AUTONOMIA ENERGÉTICA

A autonomia do sistema foi estimada com base na capacidade nominal da bateria Li-Ion 18650 utilizada (2600 mAh), no consumo médio do circuito durante operação contínua (aproximadamente 200mA) e na eficiência do sistema, considerada em torno de 85% devido às perdas nos conversores e periféricos.

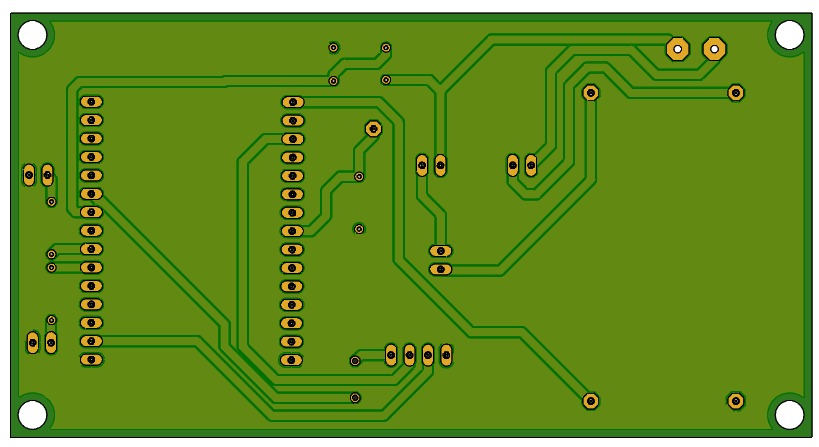
Aplicando a fórmula:

O valor obtido representa uma estimativa teórica da autonomia do dispositivo, considerando seu funcionamento com Wi-Fi ativo, envio de dados via MQTT, display OLED e LEDs indicadores operando em ciclos intermitentes. Embora não tenha sido realizada a medição prática contínua da duração da bateria, os parâmetros adotados refletem condições de uso próximas à operação prevista no ambiente metroviário.

#### 3.8.2 Montagem do hardware

Após a validação dos testes realizados em protoboard, foi desenvolvida a concepção da Placa de Circuito Impresso (PCB) utilizando o software Autodesk Fusion 360. A figura a seguir apresenta a versão projetada do circuito, consolidando a organização física dos componentes e as conexões elétricas do sistema.

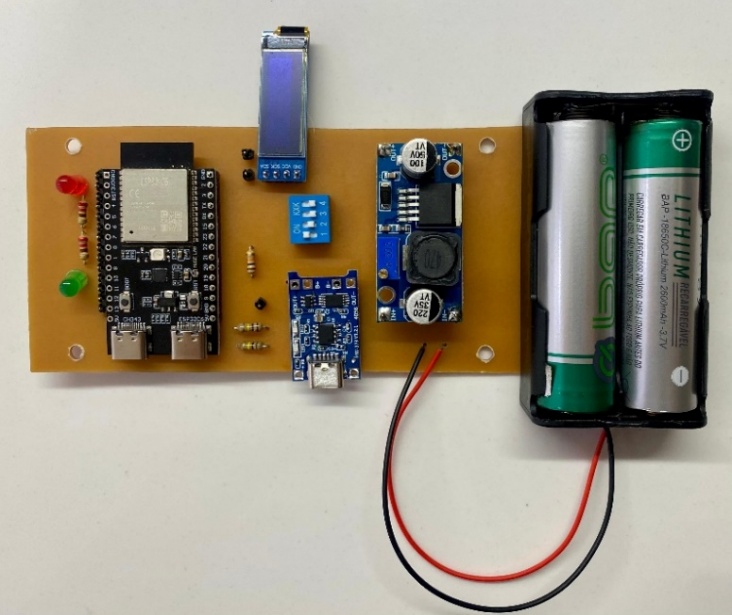
Figura 19 - Placa de Circuito Impresso



Fonte: Elaborado pelos autores (2025)

Com o projeto da PCB finalizada através do software Fusion 360, procedeu-se à fabricação da placa. Em seguida, foi realizada a soldagem dos componentes eletrônicos diretamente sobre a superfície da placa, conforme apresentado na imagem a seguir.

Figura 20 - Placa de Circuito Impresso e componentes



Fonte: Elaborado pelos autores (2025)

Com a PCB montada e os componentes devidamente soldados, foi realizada a modelagem de superfície do suporte físico da placa utilizando o software Autodesk Fusion 360. A imagem a seguir ilustra o modelo tridimensional desenvolvido para acomodar e proteger a estrutura do circuito.

Figura 21 - Projeto da Case Através do Fusion360

Desenho técnico

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelos autores (2025)

Posteriormente, efetuou-se a impressão do suporte modelado por meio de uma impressora 3D. Em seguida, realizou-se a montagem do protótipo final, integrando a Placa de Circuito Impresso (PCB), o suporte impresso e o módulo de carregamento duplo.

# CONCLUSÃO

O desenvolvimento deste trabalho resultou em uma solução funcional e eficiente de geolocalização indoor voltada ao ambiente metroviário, explorando tecnologias emergentes da Internet das Coisas (IoT). O uso do microcontrolador ESP32-C6, aliado a pontos de acesso Wi-Fi, demonstrou ser uma abordagem tecnicamente viável para a identificação de zonas de proximidade em tempo real, com baixo custo e fácil implementação características essenciais em ambientes como o Metrô de São Paulo.

Foram conduzidas etapas completas de mapeamento, testes e implementação de dashboards interativos. Os resultados comprovaram a eficácia do sistema, com destaque para a biblioteca Dash, que apresentou melhor desempenho na integração com o protocolo MQTT, garantindo estabilidade e responsividade na visualização dos dados.

Este trabalho diferencia-se por apresentar uma abordagem simplificada e replicável, que aproveita infraestrutura já existente, tornando-se especialmente relevante para sistemas públicos que demandam soluções acessíveis, escaláveis e de rápida implantação. Além de ter sido considerado e testado a utilização dos sistemas reais do Metrô, como o Elipse E3, sendo possível a futura implementação nesse sistema.

A aplicação prática mostrou-se promissora em situações que exigem resposta imediata a emergências ou falhas operacionais, demonstrando aos operadores qual é o funcionário mais próximo do incidente reportado, sendo esse o problema principal levantado inicialmente. Além disso, o uso do MQTT e a conformidade com a LGPD reforçam o compromisso com segurança e privacidade dos dados.

Conclui-se que a arquitetura desenvolvida atende aos requisitos técnicos e operacionais propostos e encontra-se em estágio apto para validação em ambiente real. Para consolidar seus benefícios, recomenda-se a integração com plataformas industriais como SCADA/PIMS e a realização de testes em escala piloto no sistema metroviário com os softwares oficiais.

Com isso, este trabalho não apenas propõe uma solução funcional, mas contribui para a transformação digital da operação metroviária, com foco em segurança, agilidade e inovação sustentável.

# [REFERÊNCIAS](#Indicador27)

**SOPHIA, S. et al.** Bluetooth Low Energy based Indoor Positioning System using ESP32. In: *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2021.

**MANIKANDAN et al.** ESP-8266-based indoor localization system. In: *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*. Anais [...]. IEEE, 2023.

**PAULO, S.** *Indoor location system: Human tracking application in a ship maneuvering simulator*. Disponível em: <https://bdta.abcd.usp.br/directbitstream/3d75e98c-4fb6-4fa3-86cd-1bba0d874014/Humberto%20Sassaki%20e%20Rodrigo%20Suguiura%20-monografia.pdf>. Acesso em: 24 out. 2024.

**MOURA, André Iasi.** *WBSL: um sistema de localização de dispositivos móveis em redes* Wi-fi*.* São Paulo: A. I. Moura, 2007.

**ABREU, Rafael Fernando Pereira.** *Localização Indoor em Ambientes Inteligentes.* Universidade do Minho, 2014. Disponível em: <https://repositorium.sdum.uminho.pt/bitstream/1822/37255/1/eeum_di_dissertacao_pg20978.pdf>. Acesso em: 24 nov. 2024.

**BAI, Lu; KHORSHED, Maurice D.; AL-MAHMOOD, Rafiqul I.; AL-SUMAITI, Ahmed S.** Indoor Localization Using Trilateration and Location Fingerprinting Methods. In: *Machine Learning for Indoor Localization and Navigation*, 2023. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-031-26712-3_4>. Acesso em: 24 nov. 2024.

**YANG, T.; CABANI, A.; CHAFOUK, H.** A survey of recent indoor localization scenarios and methodologies. *Sensors*, Basel, v. 21, n. 23, p. 8086, 2021. Disponível em: <https://www.mdpi.com/1424-8220/21/23/8086>. Acesso em: 24 nov. 2024.

**MANIKANDAN, P.** Improved Indoor Localization with Machine Learning Techniques for IoT Applications. *arXiv preprint* arXiv:2402.11433, 2024. Disponível em: <https://arxiv.org/abs/2402.11433>. Acesso em: 24 nov. 2024.

**RECK, Marcelo Sala.** *Beacons BLE – Bluetooth Low Energy: Design e Análise de um Sistema de Localização Indoor.* Universidade de Caxias do Sul, 2016. Disponível em: <https://repositorio.ucs.br/xmlui/bitstream/handle/11338/2455/TCC%20Marcelo%20Sala%20Reck.pdf>. Acesso em: 24 nov. 2024.

**GUL, F. A. I.; TUDOSE, D.; URCANU, T.** A versatile IoT development board for environmental sensing and biometric applications. In: *ROEDUNET CONFERENCE: NETWORKING IN EDUCATION AND RESEARCH (ROEDUNET)*, 23., 2024, Bucharest. Proceedings [...]. Bucharest: IEEE, 2024. p. 1–6.

**BARRAL VALES, V. et al.** Fine time measurement for the internet of things: A practical approach using ESP32. *IEEE Internet of Things Journal*, v. 9, n. 19, 2022, p. 18305–18318.

**SAAD, E.; ELHOSSEINI, M.; HAIKAL, A. Y.** Recent achievements in sensor localization algorithms. *Alexandria Engineering Journal*, v. 57, n. 4, p. 4219–4228, 2018. Disponível em: <https://doi.org/10.1016/j.aej.2018.11.008>. Acesso em: 24 nov. 2024.

**GUNHARDSON, E.** *Indoor positioning using angle of departure information.* Linköping University, Department of Science and Technology, 2015.

**O’KEEFE, B.** *Finding location with time of arrival and time difference of arrival techniques.* ECE Senior Capstone Project, 2017. Disponível em: <https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/FireBrick_OKeefe_F1.pdf>. Acesso em: 24 nov. 2024.

**CAIAFA, J.** *Automação, comunicação e vigilância no metrô de São Paulo.* 2020. Disponível em: <https://www.scielo.br/j/gal/a/8w8rFKtBPqQg4d6fk6Zmndy/>. Acesso em: 24 nov. 2024.

**METRÔ DE SÃO PAULO.** *Indicadores de mobilidade urbana na RMSP a partir da pesquisa OD-Metrô.* 2020. Disponível em: <https://www.scielo.br/j/cm/a/CQZs4RBzJkRWDrXXb9SBg3H/>. Acesso em: 24 nov. 2024.

**PINHEIRO JUNIOR, J. A.; CAVALHEIRO, A. L.** Cidades inteligentes: insights e contribuições das pesquisas brasileiras. *Urbe - Revista Brasileira de Gestão Urbana*, v. 13, n. 2, p. 201–217, 2020. Disponível em: <https://www.scielo.br/j/urbe/a/3LscvBK8vN86Q3fyFvzx7Fw/>. Acesso em: 24 nov. 2024.

**GONÇALVES, L.** *Arquitetura da infraestrutura e mobilidade urbana: uma análise sobre projeto, espaço urbano e metrô de São Paulo.* Tese (Doutorado) – Universidade de São Paulo, 2021. Disponível em: <https://www.academia.edu/43816310/>. Acesso em: 24 nov. 2024.

**VIEIRA, M. M.** *Internet das Coisas: da Teoria à Prática.* 2020. Disponível em: <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>. Acesso em: 24 nov. 2024.

**GARCIA, P. S. R.; KLEINSCHMIDT, J. H.** *Tecnologias Emergentes de Conectividade na IoT: Estudo de Redes LPWAN.* Sociedade Brasileira de Telecomunicações, 2017. Disponível em: <https://www.sbrt.org.br/sbrt2017/anais/1570361881.pdf>. Acesso em: 24 nov. 2024.

**ISODA, M. K. de T.; MORI, K. K.** *O metrô nas periferias da Região Metropolitana de São Paulo.* 2020. Disponível em: <https://repositorio.usp.br/item/003089416>. Acesso em: 24 nov. 2024.

**BORLONI, R. N.; COPCHE, F.; SANTOS, M. A. P.** Inovação, integração e automação de uma linha de metrô utilizando cenários operacionais. In: *SEMANA DE TECNOLOGIA METROFERROVIÁRIA*, 25., 2019, São Paulo. Anais [...]. São Paulo: AEAMESP, 2019. Disponível em: <https://www.aeamesp.org.br/biblioteca/stm/17smtf110916pl0901.pdf>. Acesso em: 24 nov. 2024.

**ESPRESSIF SYSTEMS.** *ESP32-C6-DevKitC-1 Development Board.* 2023. Disponível em: <https://br.mouser.com/new/espressif/espressif-esp32-c6-devkitc-1-board/>. Acesso em: 1 dez. 2024.

**BATERIA Li-Ion 18650 3,7V 2500mAh 2C.** 2024. Disponível em: <http://surl.li/medvxd>. Acesso em: 1 dez. 2024.

**IP67 CASE FOR PYSENSE/PYTRACK.** Disponível em: <http://surl.li/qyuwjw>. Acesso em: 8 maio 2025.

**ESPRESSIF SYSTEMS.** *ESP32-C6 Datasheet.* 2023. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf>. Acesso em: 18 maio 2025.

**IEEE.** *Smart city integration: how IoT is reducing emergency response times and saving lives.* IEEE Public Safety, 2023. Disponível em: <https://publicsafety.ieee.org/topics/smart-city-integration-how-iot-is-reducing-emergency-response-times-and-saving-lives/>. Acesso em: 18 maio 2025.

**INPIXON.** *Bluetooth Low Energy (BLE) for indoor positioning.* 2023. Disponível em: <https://www.inpixon.com/technology/standards/bluetooth-low-energy>. Acesso em: 18 maio 2025.

**KYRIAKIDIS, M.; HAPPEE, R.; DE WINTER, J. C. F.** Public opinion on automated driving: results of an international questionnaire among 5000 respondents. *Transportation Research Part F: Traffic Psychology and Behaviour*, v. 32, p. 127–140, 2015. Disponível em: <https://doi.org/10.1016/j.trf.2015.04.014>. Acesso em: 18 maio 2025.

**OASIS.** *MQTT Version 5.0.* OASIS Standard, 2019. Disponível em: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>. Acesso em: 18 maio 2025.

**ELIPSE SOFTWARE.** *Elipse E3 supervisiona Linha 4-Amarela do Metrô de São Paulo.* 2023. Disponível em: <https://www.elipse.com.br/elipse-e3-supervisiona-linha-4-amarela-do-metro-de-sao-paulo/>. Acesso em: 18 maio 2025.

**STREAMLIT.** *Streamlit Docs.* Disponível em: <https://docs.streamlit.io>. Acesso em: 18 maio 2025.

**CURTO-CIRCUITO.** *Carregador Bateria Lithium 1A com Proteção – USB C – TP4056.* Disponível em: <https://curtocircuito.com.br/carregador-bateria-lithium-1a-com-protecao-usb-c-tp4056.html>. Acesso em: 19 maio 2025.

**ELETROGATE.** *Suporte para 2 baterias 18650 com chave on/off.* Disponível em: <https://www.eletrogate.com/suporte-para-2-baterias-18650-com-chave-onoff>. Acesso em: 19 maio 2025.

**DASH**. *Dash user guide*. 2023. Disponível em: https://dash.plotly.com/introduction. Acesso em: 5 jun. 2025.

**ECLIPSE.** *Eclipse Mosquitto - An open source MQTT broker*. 2023. Disponível em: <https://mosquitto.org>. Acesso em: 5 jun. 2025.  
INPIXON. *Indoor Intelligence for Real-Time Location Systems*. 2023. Disponível em: <https://www.inpixon.com>. Acesso em: 5 jun. 2025.

**PINHEIRO JUNIOR, E. P.; CAVALHEIRO, G. D.** *Lei Geral de Proteção de Dados Pessoais: Comentada artigo por artigo*. São Paulo: Revista dos Tribunais, 2020.  
STREAMLIT. *Streamlit documentation*. 2023. Disponível em: <https://docs.streamlit.io>. Acesso em: 5 jun. 2025.

**VIEIRA, D. M.** *Privacidade e proteção de dados pessoais na era digital*. Rio de Janeiro: Elsevier, 2020.  
  
**STEP-UP-XL6009** Disponível em: <https://www.easytronics.com.br/step-up-xl6009>. Acesso em: 8 de jun. 2025.  
Rio de Janeiro: Elsevier, 2020.