

Protocolo MQTT em Rede de Sensores Sem Fio Para Controle De Pessoas

Lucas Alves Lira¹, Raphael Melo Guedes¹

¹Instituto de Matemática e Estatística – Universidade do Estado do Rio de Janeiro (UERJ)

²Departamento de Ciência da Computação
Universidade do Estado do Rio de Janeiro (UERJ) – Rio de Janeiro, RJ – Brazil

lucaslira@gmail.com, raphael.guedes@ime.uerj.br

Abstract. *This work aims to monitor areas with pedestrian circulation, such as hospital environments. To this end, sensor devices are employed that communicate through an Interest-Centered Network. These sensors enable dynamic data collection and deployment without the need for wiring between nodes, allowing the implementation of a large number of geographically distributed devices. The proposed network operates in a centralized manner, adopting a communication model oriented toward the content or interest of the transmitted packets. In order to develop a functional sensor-based monitoring system, a communication protocol, supporting libraries, and a graphical application were implemented to demonstrate the proposed solution. Furthermore, an experiment involving three sensor nodes was conducted to validate and illustrate the operation of the system.*

Resumo. *Este trabalho tem como objetivo o monitoramento de áreas com circulação de pessoas, como ambientes hospitalares. Para isso, são empregados dispositivos sensores que se comunicam por meio de uma Rede Centrada em Interesses. Esses sensores possibilitam a implementação e a coleta dinâmica de dados sem a necessidade de cabeamento entre os nós, viabilizando a implantação de um grande número de dispositivos distribuídos geograficamente. A rede proposta opera de forma centralizada, adotando um modelo de comunicação orientado ao conteúdo ou interesse dos pacotes transmitidos. Com o propósito de desenvolver um sistema funcional de monitoramento baseado em sensores, foram implementados um protocolo de comunicação, bibliotecas de suporte e uma aplicação gráfica para demonstração da solução. Além disso, realizou-se um experimento com três nós sensores, a fim de validar e exemplificar o funcionamento do sistema proposto.*

1. Introdução

A necessidade de compreender e, eventualmente, prever fenômenos está diretamente associada à capacidade de sensoriá-los. O sensoriamento desempenha papel fundamental na observação, no monitoramento e no controle de processos naturais e artificiais. Seja no estudo de mudanças climáticas, no acompanhamento da saúde humana ou na automação de sistemas industriais, a capacidade de coletar, interpretar e reagir a dados provenientes do ambiente é elemento essencial para a tomada de decisões. À medida que a sociedade se torna cada vez mais orientada a dados, a precisão e a eficiência dos sistemas de sensoriamento tornam-se fatores críticos, reforçando a importância de soluções tecnológicas inovadoras, confiáveis e adaptáveis.

Dentre as diversas aplicações possíveis, destaca-se o monitoramento e a localização de pessoas em ambientes específicos. O desenvolvimento deste trabalho foi motivado por uma demanda apresentada pelo setor de inovação da UERJ, inicialmente no contexto de um *Hackathon*, relacionada à dificuldade de consulta à localização de pacientes no Hospital Universitário Pedro Ernesto (HUPE). Relatos indicaram a inexistência de um sistema eficiente de acompanhamento de pacientes em trânsito dentro da unidade hospitalar, inclusive daqueles que ainda não haviam recebido alta médica. Tal cenário evidencia a necessidade de mecanismos que permitam a rápida localização de indivíduos, contribuindo para maior fluidez operacional, redução de falhas administrativas e melhor resposta a eventuais emergências médicas.

Em diálogo com os responsáveis pelo HUPE, constatou-se não apenas a ausência de um sistema de localização adequado, mas também restrições significativas de infraestrutura para implantação de soluções tradicionais. Por se tratar de um ambiente hospitalar, intervenções estruturais tendem a ser complexas, além de envolverem custos elevados. Dessa forma, qualquer proposta deveria atender simultaneamente aos requisitos de viabilidade técnica, simplicidade de implantação e baixo custo.

Nesse contexto, as Redes de Sensores Sem Fio (RSSFs) apresentam-se como alternativa adequada, pois possibilitam a coleta dinâmica de dados por meio de comunicação sem fio, reduzindo a necessidade de cabeamento e facilitando a implantação em ambientes já consolidados. A solução proposta baseia-se na utilização de dispositivos distribuídos aos pacientes para identificação individual, os quais podem ser rastreados por sensores instalados estrategicamente no ambiente hospitalar. A escolha dos dispositivos levou em consideração, principalmente, seu baixo custo e ampla disponibilidade. Os dados coletados são então encaminhados a um servidor central, permitindo a realização de consultas e o acompanhamento em tempo real da localização dos pacientes na área monitorada.

Para a operação da rede, optou-se pela adoção de uma arquitetura centrada em conteúdo, mais especificamente o conceito de Redes Centradas em Interesse (ICN – *Information-Centric Networks*) [Ahlgren et al. 2012] e uso do protocolo MQTT (*Message Queuing Telemetry Transport*). Essa abordagem foi selecionada por oferecer vantagens em termos de escalabilidade, eficiência no consumo de recursos, robustez e segurança, características relevantes para aplicações de monitoramento distribuído.

Assim, neste trabalho, as informações capturadas pelos sensores — incluindo suas medições e identificações — são mapeadas em interesses dentro da arquitetura proposta, viabilizando a construção de um sistema de monitoramento funcional. Para tanto, foi

necessário o desenvolvimento de bibliotecas específicas e a implementação de um protocolo de comunicação entre os dispositivos. Como contribuições do trabalho podemos destacar: implementação de arquitetura RSSF baseada em MQTT para rastreamento *in-door*, customização de *firmware* para suporte a protocolo *publish/subscribe*, proposta de padronização de interesses via tópicos MQTT, desenvolvimento de API modular extensível e validação experimental com *hardware* de custo reduzido.

A organização deste artigo é a seguinte: na Seção 2, são apresentados trabalhos correlatos; na Seção 3, são discutidos os fundamentos das Redes Centradas em Interesse e o protocolo MQTT adotado na solução; na Seção 4, descreve-se a arquitetura proposta e o sistema de controle de pessoas; na Seção 5, detalha-se o funcionamento dos nós sensores; na Seção 6, apresenta-se o experimento realizado; na Seção 7 apresentamos a interface gráfica criada para o controle e acompanhamento no rastreamento de pessoas, e por fim, a Seção 8 apresenta as conclusões e direções futuras.

2. Soluções disponíveis no mercado

Como primeiro passo foi realizada uma pesquisa para encontrarmos soluções que atendessem ao problema de localização de pessoas, justamente que utilizassem redes sem fio aliada a um preço satisfatório.

Na literatura podemos encontrar uma série de propostas de redes de sensores sem fio (RSSF) para diversas aplicações, tais como agricultura [Marinho et al. 2017], controle de temperatura [da Costa et al. 2021], combate a incêndios [Silva et al. 2024], entre outras [Cao-hoang and Duy 2017].

Para a aplicação de controle de pessoas foram encontradas algumas soluções, porém todas apresentaram algum aspecto negativo em principal referente ao preço das assinaturas e/ou a falta de liberdade de customização da operação da rede. Na Tabela 1 temos um comparativo das soluções encontradas e algumas características de cada solução.

Tabela 1. Comparativo de soluções encontradas

Solução	Localização	Plataforma	Customização	Preço
Navigine	Sim	Android, IOS, web	Alta	Assinatura mensal
Pointtr	Sim	Android, IOS, web	Média	Preços sob entrevista
Estimote	Sim	Android, IOS, web	Baixa	Preços sob entrevista
IndoorAtlas	Limitada	Android, IOS, web	Baixa	Preços sob entrevista

Dentre as características comparadas as mais importantes para o problema são a liberdade de customização e o preço da solução. A liberdade de customização está relacionada a facilidade de modificação da solução, seja na visualização quanto no tipo de dado coletado pelo sensor. Quanto ao preço, em todas as soluções os valores cobrados só são disponibilizados a clientes com CNPJ. Desta forma conseguimos apenas a maneira de cobrança e não o valor em si. Estas duas características motivaram a adoção de uma solução própria que aliasse facilidade de customização a um preço reduzido.

Ao fim de diversas pesquisas encontramos um conjunto de dispositivos que atendiam aos nossos requisitos. Foram utilizados dispositivos BLE Beacon e um Gateway BLE de baixo custo, fabricados por fornecedor comercial, compatíveis com comunicação sem fio em 2.4 GHz e integração via rede Ethernet.

Na Figura 1 temos os dispositivos escolhidos. Destacamos que uma das considerações exigidas era que a solução empregada fosse barata, sendo assim, o objetivo do trabalho foi desenvolver uma solução com redução do custo na implantação do projeto. O equipamento escolhido consiste de aparelhos que podem ser importados com facilidade, com custo aproximado de importação de R\$ 10,00 por nó sensor e R\$ 0,50 centavos por nó emissor (cotações do período de 2024).



Figura 1. À esquerda temos nó o emissor, no centro o seu interior e à direita o nó sensor

Outro ponto de destaque é que o equipamento, no caso, os sensores podem ser customizáveis uma vez que o código fonte do *firmware* do dispositivo foi disponibilizado pelos desenvolvedores. Assim, diferente das outras soluções, torna-se possível a modificação de qual dado deve ser coletado e o que deve ser realizado na coleta. Esta facilidade de modificação deve-se ao fato do dispositivo vir com o mínimo para ligar, ou seja, a configuração de fábrica do dispositivo vem praticamente em branco, daí a justificativa do preço ser bem reduzido. Desta forma, o equipamento adotado segue o paradigma *white box*, nos quais o equipamento foi fornecido com configuração mínima, cabendo ao usuário a definição e implantação da pilha de *softwares* conforme os requisitos da aplicação.

Definido o equipamento a ser utilizado, passou-se à análise da forma de comunicação mais adequada para a solução proposta. Diversos trabalhos na literatura destacam o protocolo MQTT (*Message Queuing Telemetry Transport*) como uma alternativa amplamente empregada e eficiente em aplicações de Internet das Coisas e Redes de Sensores Sem Fio [Martins and Zem 2015], sobretudo em razão de sua leveza, baixo consumo de banda e modelo de comunicação baseado em publicador/assinante. Diante dessas características e de sua ampla adoção em soluções similares [Quincozes et al. 2019, Carneiro da Cunha and Macêdo Batista 2022, Rocha 2022], optou-se, neste trabalho, pela utilização do protocolo MQTT como mecanismo de comunicação da solução desenvolvida.

3. Fundamentação Teórica

Antes de detalhar os componentes que compõem a rede proposta, bem como as interações estabelecidas entre eles, faz-se necessária a apresentação de alguns conceitos fundamentais que servem de base para a compreensão da arquitetura adotada. Esses conceitos fornecem o embasamento teórico necessário para contextualizar as escolhas tecnológicas realizadas ao longo deste trabalho e facilitar o entendimento do funcionamento do sistema como um todo.

3.1. Redes Centradas em Interesses

Redes Centradas em Interesses, também conhecidas como Redes Centradas em Informação (*Information-Centric Networking – ICN*), constituem um paradigma de comunicação no qual o foco da rede deixa de ser o endereçamento dos dispositivos e passa a ser o conteúdo das informações trocadas. Nesse modelo, aplicações expressam interesses por dados nomeados, e a infraestrutura de comunicação é responsável por localizar e entregar o conteúdo correspondente, independentemente da localização dos produtores e/ou consumidores. No contexto deste trabalho, esse paradigma foi materializado por meio do protocolo MQTT, no qual os tópicos de comunicação desempenham o papel de interesses da rede.

Vale ressaltar que nas redes tradicionais a comunicação é orientada a endereços IPs, por exemplo: *enviar de um endereço IP de origem para um endereço IP de destino*, e o conteúdo vem, em geral, de uma fonte específica. Já nas redes centradas em interesse a comunicação é orientada ao conteúdo/interesse. Um nó expressa um interesse, por exemplo: */hospital/paciente/123/localizacao* e a “rede” entrega o conteúdo de onde ele estiver disponível.

O interesse (*interest*) é uma solicitação por um conteúdo nomeado, ou seja, os dados possuem nomes semânticos, não endereços. Produtor e consumidor não precisam estar ativos ao mesmo tempo e não precisam se conhecer, além de poderem mudar de lugar.

3.2. Protocolo MQTT

O MQTT é um protocolo de comunicação leve amplamente utilizado em aplicações de IoT. Baseado no paradigma publicador/assinante [Hunkeler et al. 2008], que utiliza um *broker* para intermediar a comunicação entre os nós (elementos) da rede, Figura 2. A escolha do MQTT neste trabalho se justifica por sua leveza, ampla adoção, facilidade de implementação e aderência natural ao paradigma de redes centradas em interesses.

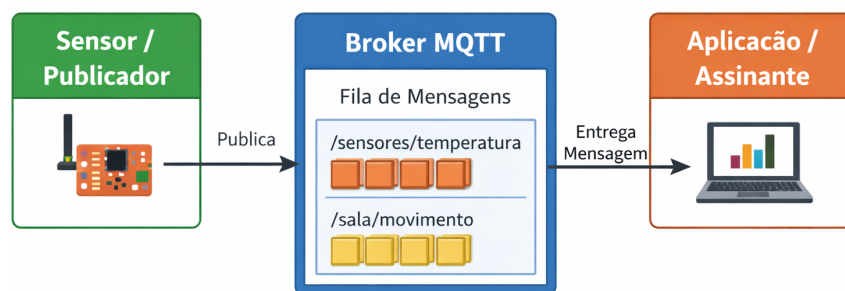


Figura 2. Componentes do protocolo MQTT

A publicação de mensagens em tópicos específicos equivale à disponibilização de conteúdos, enquanto a subscrição a esses tópicos representa a expressão de interesse por tais informações. Os nós sensores publicam informações de localização associadas a identificadores de interesse, enquanto que algum nó subscrito pode consumir conteúdos sem a necessidade de conhecimento prévio sobre a origem dos dados. Essa associação permite implementar uma arquitetura centrada em interesses sobre uma infraestrutura IP tradicional, mantendo o desacoplamento entre produtores e consumidores, favorecendo

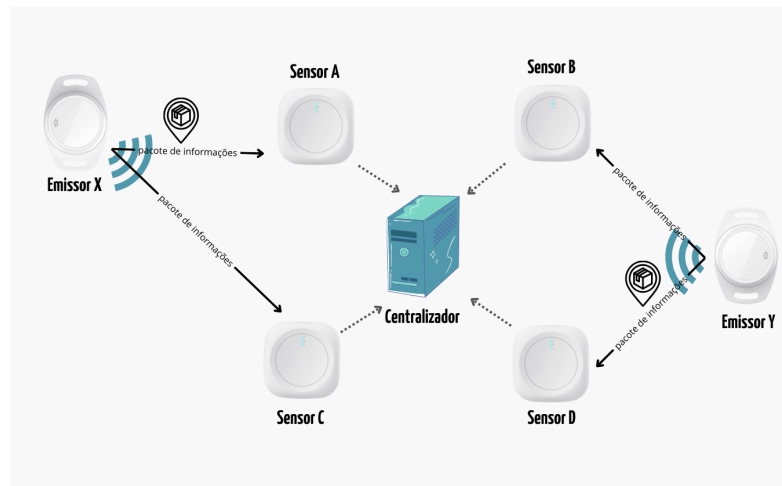


Figura 3. Topologia geral da arquitetura proposta

escalabilidade, flexibilidade e mobilidade, características essenciais para aplicações baseadas em Redes de Sensores Sem Fio.

4. Arquitetura Proposta

A arquitetura proposta neste trabalho foi concebida de forma modular, visando flexibilidade e facilidade de expansão. O sistema é estruturado em três tipos principais de nós: **emissores**, **sensores** e nó **centralizador**, interconectados por uma infraestrutura de comunicação baseada no protocolo MQTT.

Os **emissores** têm como única função a emissão periódica de sinais contendo identificadores únicos, neste caso a associados a pessoas monitoradas. Esses dispositivos operam de forma passiva e com baixo consumo energético.

Os **sensores** são responsáveis pela varredura do ambiente em busca dos sinais emitidos pelos emissores. Cada nó sensor realiza a leitura da intensidade do sinal recebido (RSSI - *Received Signal Strength Indicator*), aplica filtros básicos para redução de ruído e encapsula as informações coletadas em mensagens MQTT. Essas mensagens são publicadas em tópicos hierárquicos que representam os interesses da rede.

O **nó centralizador** atua como consumidor das informações publicadas. Ele se inscreve nos tópicos de interesse, processa os dados recebidos, realiza o armazenamento persistente e disponibiliza mecanismos de visualização para o usuário final.

Um aspecto central da arquitetura é o mapeamento dos interesses da rede diretamente nos tópicos MQTT. Essa abordagem permite que novos nós sensores ou aplicações consumidoras sejam adicionados sem necessidade de reconfiguração dos elementos já existentes, reforçando o desacoplamento entre produtores e consumidores de dados.

4.1. Protocolo de Comunicação e Gerenciamento de Informações nos Nós

A comunicação no sistema foi realizada utilizando a rede, e o nosso objetivo foi comunicar as medidas dos nós sensores, isto é os sinais *beacons* recebidos ao nó central. Para isto deve-se prover um protocolo para obter tais medidas e promover o funcionamento da rede. A primeira análise feita, para a solução, foi permitir a pluralidade dos sensores

RSSFs e, por consequência, múltiplas formas de dados de medidas que deveriam ser trocadas entre os nós. A segunda análise foi relativa ao encaminhamento que deve ser feito usando o interesse no nó centralizador, ou seja, como usar o interesse para trafegar os dados dos sensores ao nó central. A partir das análises anteriores, propusemos um esquema de base de dados distribuída simplificada e *online*, onde os dados são as medidas e outras informações dos nós atrelados aos endereços MAC de cada dispositivo.

4.1.1. Interesse

Antes de começarmos a detalhar a base de dados distribuída e o protocolo de comunicação, é necessário destacar a importância do interesse na rede a ser usada para trafegar as informações do sistema. O interesse é uma cadeia de caracteres que pode ser escolhida livremente pela aplicação para assinar um conteúdo ou publicá-lo na rede. O sistema propõe um formato padrão de interesse para ambas as ações de assinar e publicar, como parte da especificação do protocolo, permitindo e facilitando o uso de aplicações na rede.

4.1.2. Identificação dos Nós (Campo identificação do nó)

O sistema prevê que cada nó emissor e sensor seja identificado para troca de informações através do protocolo. Esta identificação usa o endereço MAC da interface de rede disponível. Quando se deseja enviar uma mensagem para todos os nós (1 para N) usa-se o caractere “*” no campo identificação do nó do interesse, caso contrário pode-se definir um nó específico para enviar a mensagem (1 para 1) usando a identificação dele. Do ponto de vista de assinatura (registro) do interesse, um nó pode optar por receber requisições que foram feitas a todos os nós da rede usando o caractere “*” no campo identificação do nó e/ou receber apenas requisições feitas a tópicos específicos do interesse.

4.1.3. Base de Dados

Bases de dados consistem em conjuntos organizados de registros relacionados entre si, estruturados de modo a possibilitar armazenamento, consulta e recuperação eficiente das informações. No contexto deste trabalho, os dados são armazenados no formato de documentos compostos por chaves (ou campos), nos quais cada registro contém pares chave–valor que representam atributos específicos. Um mesmo nó sensor pode estar associado a múltiplos documentos, dependendo das informações coletadas ou do histórico de medições. Cada registro é identificado de forma única por meio de um identificador (ID) e do endereço MAC do dispositivo, garantindo integridade e rastreabilidade dos dados. Um exemplo da estrutura adotada é apresentado na Figura 4.

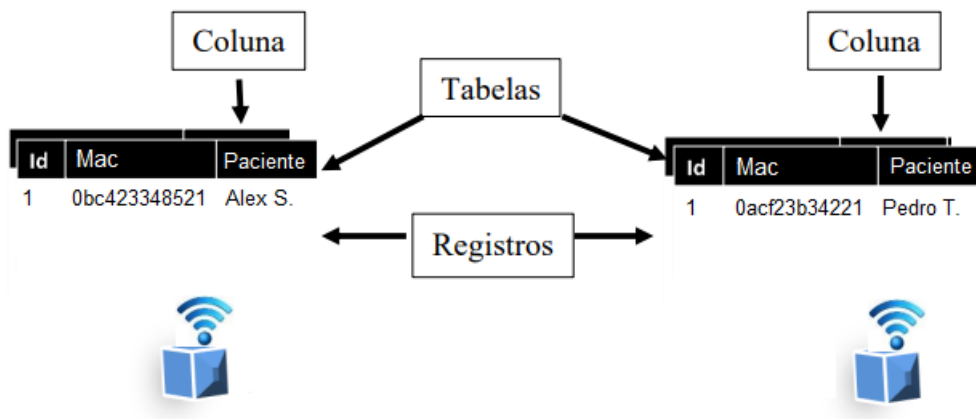


Figura 4. Estrutura da base de dados distribuída com valores de exemplo

5. Aplicação, Serviço, Biblioteca e API

Python e C foram usados para complementar o *software* do nó sensor e dos nós emissores, incluindo suas APIs. Cada parte de código foi criado para os sistemas operacionais Windows ou Linux. O serviço do nó sensor mantém os dados de código de envio e recebimento, a API dos Módulos Sensor e o Aplicativo Demo de configuração foram os únicos *softwares* atrelados ao equipamento de fábrica.

5.1. Serviço do nó sensor

O serviço do nó sensor é responsável por controlar os sensores que estão vinculados a ele (podem envolver proximidade, temperatura, umidade e outros) e coordenar o envio e recebimento de mensagens. Em circunstâncias típicas, o serviço continuará indefinidamente e primeiro executará as atividades listadas abaixo na ordem:

1. **Encapsulamento dos dados recebidos:** As informações recebidas pelos emissores, em alcance, são recebidas com intervalos regulares e sintetizadas de acordo com os módulos sensores envolvidos para serem retransmitidas. Módulos, sensores e valores relativos são parte desses dados.
2. **Carregamento das configurações do arquivo:** As configurações operacionais do serviço do nó sensor podem ser definidas previamente usando o arquivo de configuração. A estrutura do arquivo é comparável à do formato *JavaScript Object Notation* (JSON). A configuração que carrega a biblioteca de um módulo para ser usado como sensor de proximidade e, se necessário, alerta para quando determinadas condições são atingidas.
3. **Carregamento das bibliotecas dos módulos/sensores:** No nível da configuração interna dos equipamentos, a comunicação com os módulos e seus sensores é realizada por meio de bibliotecas construídas com chamadas predefinidas pelo fabricante. Nesta fase de carregamento, o serviço carrega essas bibliotecas do arquivo de configuração e as verifica para tornar a informação disponível.
4. **Inicia tarefa infinita de recebimento das mensagens:** O serviço tratará as consultas feitas à rede e a transmissão de mensagens aos tópicos relativos.
5. **Inicia tarefa infinita para testar condições para enviar pacotes ao centralizador:** Periodicamente, os valores do sensor são verificados para ver se atendem aos requisitos de ativação especificados no arquivo de configuração do equipamento. O serviço então criará a mensagem e transmitirá pela rede.

5.2. API Módulo dos Sensores

Vale mencionar que, a API do Sensor Module permite a criação de novas interfaces entre os mais diversos dispositivos de detecção e o serviço do nó sensor. O sensor contém uma série de entradas e saídas com vários protocolos, permitindo que os fabricantes criem novos sensores e, consequentemente, novos modos de comunicação, formatos para dados do sensor e soluções que exijam a criação de códigos para ler as medições desses sensores. Este estudo sugere que uma API com chamadas genéricas poderia ser criada para vários sensores que podem ser compiladas e carregadas a partir do arquivo de configuração para suportar um desenvolvimento rápido, amplo e diversificado, garantindo a customização da aplicação. A seguir estão as chamadas necessárias de um módulo para cada sensor:

- **Connect:** Conecta o sensor ao sistema, agregando-o à rede;
- **Open:** Abre a comunicação com um sensor para obter medições;
- **Read:** Lê a medida;
- **Close:** Fecha a comunicação com um sensor;
- **Disconnect:** Desconecta fisicamente o sensor, removendo-o da rede.

A biblioteca para o nó sensor foi criada de forma a possibilitar a construção de vários programas para o nó sensor. Pode ser usada com as linguagens de programação C, Python e JS, bem como com os sistemas operacionais Linux ou Windows. A principal responsabilidade dessa biblioteca é fornecer proativamente uma lista de nós, sensores e dados que estão presentes na rede; consequentemente, a API se encarrega de consultar periodicamente os sensores para coletar suas medições e preencher sua lista de nós e sensores usando o protocolo de comunicação MQTT.

Para adicionar um novo nó emissor com seus dados, a API executa um algoritmo. Todo o sistema é construído em torno da mensagem de *keep alive*, que os sensores enviam periodicamente ao nó centralizador, facilitando a localização de novos nós sensores na rede e/ou determinação se eles ainda estão operacionais. Quando a API recebe uma mensagem de *keep alive* de um nó emissor que antes era desconhecido, ela passa uma chamada para saber as informações e o registro completo do sensor que contém seu valor. Após a conclusão do algoritmo, a API realizará consultas repetidamente na lista destes endereços, procurando os dados de cada sensor para manter a precisão de suas informações. O nó é removido da lista se sua mensagem de *keep alive* não for vista por um período de tempo predeterminado que pode ser facilmente configurável usando a API.

Todas as ideias e soluções sugeridas neste trabalho foram testadas usando um aplicativo de exemplo que foi criado utilizando a API do nó sensor. Assim como os outros *softwares* desenvolvidos neste trabalho, o aplicativo foi criado como serviço executável na linguagem de programação Python e uma interface visual que fornece recursos gráficos. Devido ao fato de que este programa é uma aplicação executável, o dispositivo no qual a aplicação de demonstração é executada pode ser diferente do dispositivo que realiza a configuração dos dispositivos.

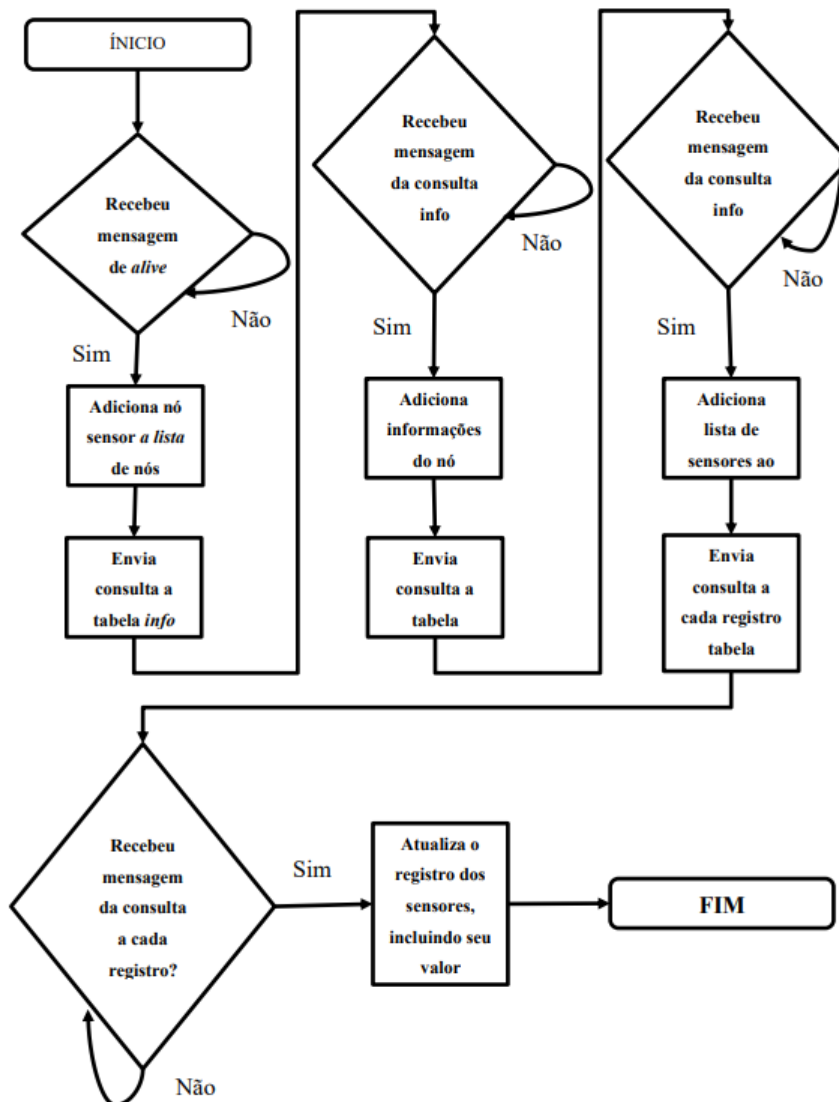


Figura 5. Fluxograma da API nó coletor

6. Implementação da Solução Proposta

A implementação do sistema foi realizada considerando as limitações de *hardware* típicas de ambientes de RSSF, bem como a necessidade de interoperabilidade entre diferentes plataformas. Os nós sensores escolhidos utilizam microcontroladores com suporte a *Bluetooth Low Energy* e conectividade *Wi-Fi*, permitindo a integração direta com o *broker*.

No *software* embarcado dos nós sensores, o funcionamento segue um ciclo contínuo composto por varredura de sinais emitidos (*beacons*) pelos nós emissores, processamento local e publicação dos dados. Durante a varredura, os identificadores dos emissores e os valores de RSSI são capturados. Em seguida, esses dados passam por uma etapa de filtragem simples, com o objetivo de reduzir flutuações abruptas causadas por interferências do ambiente.

Após o processamento local, as informações são publicadas no *broker* MQTT por meio de mensagens estruturadas, contendo identificador do sensor, intensidade do sinal e

carimbo temporal. A organização hierárquica dos tópicos permite distinguir facilmente diferentes sensores e dispositivos monitorados.

O *broker* MQTT utilizado foi o Mosquitto [Eclipse Foundation 2024a], configurado para operar com mecanismos de segurança baseados em TLS (*Transport Layer Security*), garantindo confidencialidade e integridade das mensagens transmitidas. A autenticação dos clientes é realizada por meio de certificados digitais, reduzindo o risco de acesso não autorizado.

O nó centralizador foi implementado em Python, integrando bibliotecas para comunicação MQTT, persistência dos dados em banco remoto e uma interface gráfica para visualização das informações coletadas. Essa aplicação é responsável por consolidar os dados provenientes dos múltiplos sensores e apresentar ao usuário uma visão unificada do estado do sistema.

Foram desenvolvidas bibliotecas de *software* para os nós sensores, responsáveis pela varredura de dispositivos emissores, filtragem dos sinais e publicação dos dados via MQTT.

6.1. Experimento e Implementação

Um experimento de prova de conceito foi conduzido com três nós sensores distribuídos em um ambiente. A disposição dos sensores permitiu avaliar a sobreposição dos sinais emitidos e a capacidade do sistema em inferir proximidade relativa entre dispositivos monitorados.

Foi desenvolvida uma aplicação servidora em Python para interagir com a interface visual *web* embarcada no próprio equipamento, a qual oferece os recursos gráficos necessários para a configuração de cada dispositivo, Figura 6.

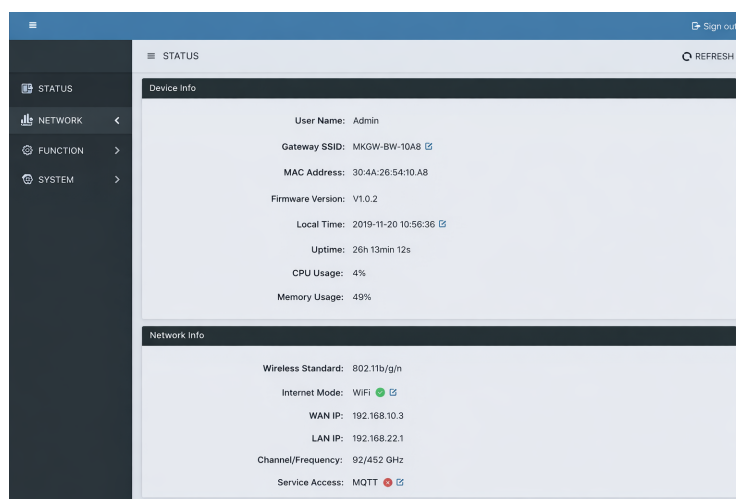


Figura 6. Interface *web* com informações de um dos dispositivos

A interface pode ser acessada a partir de qualquer ponto da rede, uma vez que a aplicação de configuração do dispositivo opera como um servidor *web*, disponibilizando páginas por meio de um endereço específico. Para sua utilização, é necessário apenas um navegador compatível. A aplicação oferece páginas destinadas à configuração do dispositivo e à consulta de registros (*logs*) de dados.

As páginas permitem a interação com os nós sensores de maneira a ter acesso às suas informações básicas de configuração. Nas páginas de controle é possível ver o estado atual de saúde do equipamento e da conexão de rede, além de ser possível ativar ou desativar seu funcionamento e outras configurações específicas, Figura 7.

Figura 7. Página com configurações específicas do dispositivo

A execução bem-sucedida do experimento demandou não apenas a modificação do código-base do dispositivo designado como sensor, com o objetivo de viabilizar a utilização efetiva do protocolo MQTT, mas também a implementação de um mecanismo de interceptação das mensagens transmitidas. Dessa forma, o processo envolveu tanto a adequação funcional do dispositivo quanto a criação de uma infraestrutura complementar para monitoramento das comunicações.

A primeira etapa consistiu na adaptação do código-fonte do equipamento, incorporando suporte completo ao protocolo MQTT. Essa atividade exigiu a compreensão detalhada da arquitetura interna do dispositivo, bem como a análise das especificidades do protocolo — tais como modelo publicador/assinante, níveis de QoS e gerenciamento de sessões — a fim de garantir uma integração eficiente e compatível com as limitações computacionais do hardware embarcado.

Contudo, a simples modificação do código não era suficiente para atender aos objetivos experimentais. Tornou-se necessária a criação de um ambiente capaz de realizar a interceptação controlada das mensagens MQTT, permitindo o monitoramento em tempo real e a captura dos dados trafegados. Para isso, desenvolveu-se uma camada adicional de funcionalidades no ambiente operacional do sistema, possibilitando a coleta estruturada das comunicações sem alterar o comportamento esperado da aplicação principal.

O ambiente projetado foi concebido de modo a não interferir no fluxo normal de operação do dispositivo, assegurando transparência no processo de interceptação. Durante o desenvolvimento, foram considerados aspectos como eficiência computacional, segurança na transmissão dos dados e escalabilidade da solução, garantindo que a captura das mensagens ocorresse de forma confiável e sem comprometer a estabilidade do sistema.

Para viabilizar a interceptação das mensagens, adotou-se o Mosquitto como *bro-*

ker MQTT. A escolha fundamentou-se em sua implementação consolidada do protocolo, ampla compatibilidade com dispositivos IoT e suporte nativo a mecanismos de segurança, especialmente por meio da utilização de *Transport Layer Security* (TLS). Além disso, o Mosquitto apresenta baixo consumo de recursos e elevada estabilidade, características adequadas ao cenário experimental proposto.

6.1.1. Experimento de Teste

Para testar a aplicação e disposição de equipamentos adotamos uma abordagem estratégica para atender às exigências da topologia BSS (*Basic Service Set*). A fim de estabelecer uma comunicação eficiente, implementamos um arranjo com três nós sensores, um emissor e um computador pessoal como centralizador de dados, executando o *software* da aplicação.

Os três nós sensores foram configurados de acordo com as especificações do protocolo MQTT para serem capazes de coletar dados dos emissores em suas proximidades e encaminhá-los para o nó centralizador. Esses nós sensores funcionaram como os elementos principais da infraestrutura da rede, executando a comunicação intermediária entre os dispositivos e permitindo o acesso à rede sem fio.

O computador pessoal foi utilizado como nó centralizador, por sua vez, desempenhou o papel de um cliente na rede, recebendo os dados coletados pelos nós sensores e realizando transmissões adicionais ao banco de dados conforme necessário. O computador pessoal estava conectado à mesma rede sem fio, agindo como um ponto central para receber, processar e analisar os dados provenientes dos nós sensores permitindo que fosse possível monitorar e gerenciar todas as operações da rede de forma conveniente e centralizada.

Em resumo, a aplicação desenvolvida utilizou esta combinação de três nós sensores, um emissor e um computador pessoal como parte integrante de uma topologia BSS. O equipamento foi posicionado triangularmente a uma distância relativa de aproximadamente 10 metros entre si, para que pudéssemos estudar casos onde o sinal poderia estar sobreposto. A configuração foi projetada para também garantir uma coleta e análise de dados de maneira organizada e centralizada para os propósitos da aplicação desenvolvida.

7. Interface Gráfica para Controle de Pessoas

Como a ideia é o uso de controle de pacientes em um hospital, desenvolveu-se uma interface que permitiu o cadastro e visualização do rastreamento de pacientes. Assim, foi desenvolvida uma aplicação em Python batizada de “Where is it”, com intuito de exemplificar o ponto principal deste trabalho que é demonstrar a aplicabilidade prática do equipamento utilizando o protocolo MQTT mencionado. A aplicação conta com uma tela principal (*dashboard*) para visualização ampla dos registros no sistema e demais informações filtráveis.

Tkinter, uma biblioteca padrão em Python, foi escolhida para criar a interface gráfica interativa (GUI). A GUI apresenta um campo de entrada para inserção de informações de cadastro como também a inserção de endereços MAC, endereços que vem no próprio emissor e são únicos para cada emissor, sendo assim a informação que

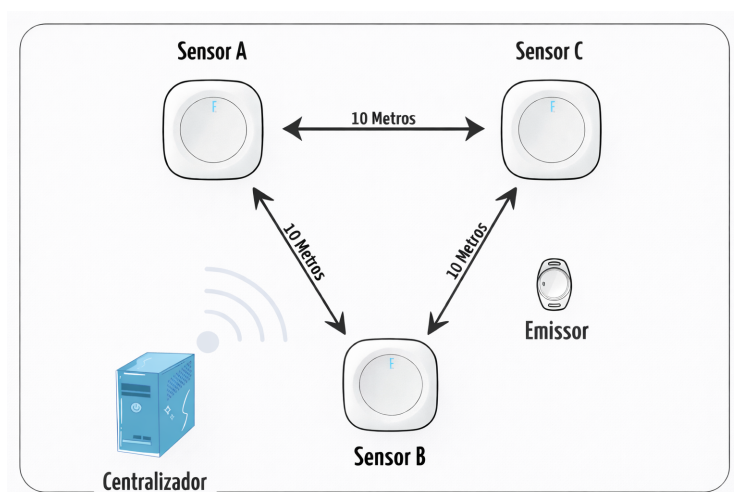


Figura 8. Disposição dos equipamentos no teste realizado

distingue cada paciente. A interatividade em tempo real foi alcançada através do uso de *callbacks* que respondem dinamicamente às ações do usuário, neste caso seu rastreo de posicionamento.

A interface Tkinter utiliza a API Firebase para realizar consultas em tempo real ao banco de dados. Quando um endereço MAC é inserido, o sistema consulta a coleção para obter a localização associada. Essa informação é então exibida instantaneamente na interface gráfica. Além disso, a biblioteca paho-mqtt [Eclipse Foundation 2024b] foi integrada para comunicação assíncrona o que permite a inscrição em um tópico MQTT correspondente, configurando *callbacks* para processar mensagens recebidas. A combinação entre MQTT e Firebase [Google LLC 2024] possibilita uma visão ampla e em tempo real sobre os dispositivos.

A interação com o Firebase foi implementada principalmente nos eventos associados aos botões, opções de busca e preenchimento de tabelas.

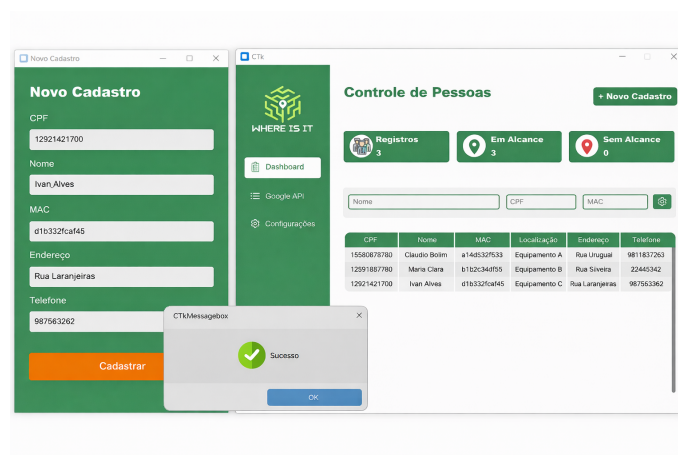


Figura 9. Exemplo de cadastro de pessoas

A abordagem para integrar Tkinter com Firebase proporcionou uma fundação sólida e a medida que avançamos, foi percebido que esta integração oferecia flexibilidade

para aprimorar a interface, implementar consultas mais complexas e explorar recursos adicionais disponíveis em ambas as bibliotecas. Então resolvemos atender a alguns requisitos de aplicações para demonstrar esse processo de integração para o desenvolvimento de trabalhos futuros ou de comercialização.

Como o monitoramento dos equipamentos é o foco da aplicação, foi aproveitado os componentes do Tkinter para fazer uma integração com um sistema de mapas e coordenadas do Google API de modo a exemplificar uma possível integração completa entre as aplicações implementadas em diferentes localidades. A integração com um sistema de ERP (Planejamento de Recursos Empresariais) agregaria valor ao sentido da implementação de modo a termos uma visão mais ampla do controle de dispositivos.



Figura 10. Parte integrada à API do Google

Levando em consideração a sensibilidade das informações de pessoas atreladas ao sistema, utilizei componentes do Tkinter para demonstrar a implementação de um sistema de *login* e autenticação controlado por dados de *token* do Firebase ou validação com o Google. A flexibilidade da aplicação forneceria bloqueio às funcionalidades relacionadas de acordo com a necessidade da implementação desejada.

7.1. Serviço de interceptação de mensagens MQTT e atualização de proximidade

Para poder extrair do equipamento os dados que estavam sendo publicados no *broker* do Mosquitto foi desenvolvido um *windows service* em Python, que contempla a lista de dispositivos coletores cadastrados no sistema e itera todos os tópicos referentes com objetivo de obter o arquivo json com os dados. Este serviço aciona a sua função com um temporizador customizável a fim de não congestionar a aplicação.

O processo de atualização de proximidade depende da mensagem publicada no tópico tendo no corpo de sua mensagem um valor de endereço MAC e sua posição relativa do equipamento em RSSI (*Received Signal Strength Indication*). O RSSI, ou Indicação de Intensidade de Sinal Recebido, é a medida da potência do sinal de rádio que está sendo recebido pelo dispositivo, como no caso um receptor Bluetooth. A qualidade do sinal RSSI é uma métrica fundamental em redes sem fio para fornecer as informações importantes sobre a potência e a estabilidade da conexão, e foi utilizado como indicador no mapa de localização.

Em ambiente de teste foi desenvolvida uma função para decodificar o corpo da mensagem e encontrar somente os tópicos relativos aos endereços MAC cadastrados.

Levando em consideração tais publicações a função ordena os valores relacionados em uma ordem de importância, concluindo que o dispositivo registrado no endereço MAC **FD2763402044** se encontra em proximidade maior do dispositivo coletor **sensor1**. Logo, uma requisição de *update* é feita para o registro no Firebase para que possa ser atualizado o campo '*location*' que será usado pela interface para mostrar o mapa com a planta relativa.

Ao clicar nos registros da tabela na aplicação o mapa de proximidade é aberto com a planta da localidade onde a solução foi implantada e, com os indicadores que a função de importância definiu com a imediação do dispositivo registrado à algum sensor.



Figura 13. Planta demonstrando que o MAC consultado se encontra ao alcance entre -40 dBm e -30 dBm do sensor



Figura 14. Planta demonstrando que o MAC consultado se encontra ao alcance entre -80 dBm e -60 dBm do sensor. Indicando afastamento do dispositivo

Existe a possibilidade de haver interposição de sinais frequente, o que poderia levar a uma incerteza da aplicação em relação a localização do endereço MAC cadastrado. Neste caso, três soluções são cabíveis. A aplicação pode tratar a incerteza indicando sinais simultâneos na planta, o que significaria possível localização do MAC entre os dois sinais.

Podemos também, tratar toda a área longe dos dispositivos coletores como “zona cinza”, onde não se pode ter certeza da localização do endereço buscado. Neste ambiente o sistema trataria a última informação recebida pelo MAC como um *trace back* e poderia

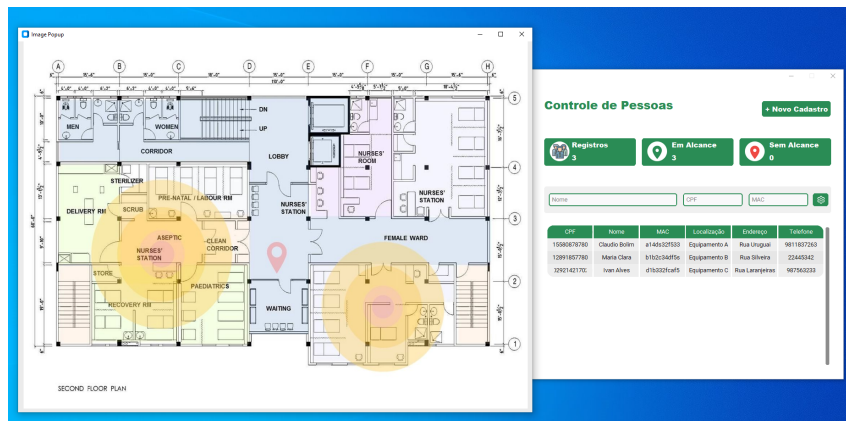


Figura 15. Planta demonstrando que o MAC consultado se encontra entre os dois sinais

indicar a localização do último registro coletado já que a função de *update* atualiza o banco com um *timestamp*.

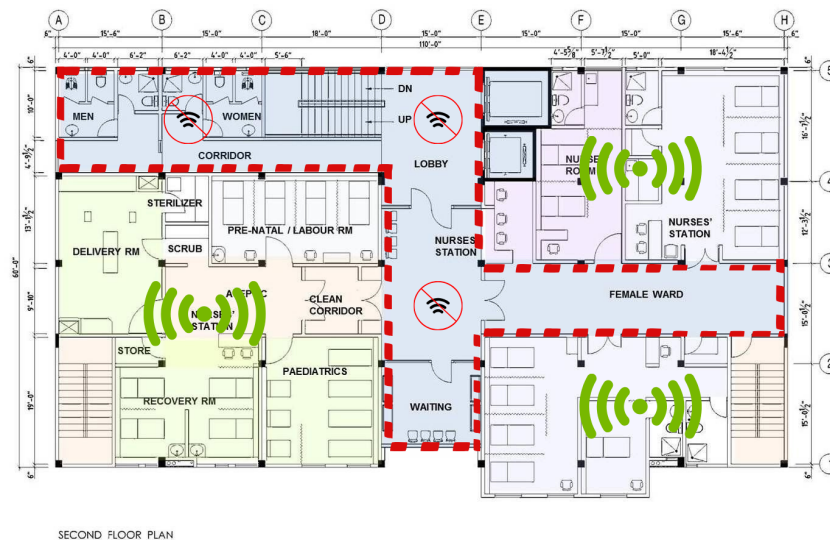


Figura 16. Planta demonstrando zonas cinzas onde o sinal fraco e a incerteza da aplicação implica uma área de falta de localização do MAC a ser tratada como desejado

Uma terceira solução definitiva seria aumentar o número relativo de equipamentos, e estudar a melhor forma de cobrir todas as áreas com um sinal de propagação forte e sem “zonas cinzas”.

8. Conclusão

Este trabalho teve como objetivo o desenvolvimento de um sistema de monitoramento baseado em Rede de Sensores Sem Fio (RSSF), contemplando a criação de aplicativos de monitoramento, integração entre nós sensores, *softwares* associados e nós de monitoramento, além da definição de um protocolo de consulta e resposta capaz de demonstrar

a interoperabilidade da solução. A fundamentação teórica evidenciou que o paradigma publicador/assinante, orientado por interesses, apresenta-se como abordagem adequada para RSSFs, oferecendo baixo custo e flexibilidade para diferentes cenários de aplicação. A plataforma computacional e os sensores selecionados demonstraram ampla disponibilidade e interoperabilidade do ponto de vista de *hardware*.

A aplicação desenvolvida mostrou-se eficaz no monitoramento em tempo real, destacando-se a utilização do protocolo MQTT como elemento central da comunicação. Sua arquitetura leve e orientada a mensagens possibilitou transmissão rápida e confiável de dados entre os dispositivos e a central de controle, contribuindo diretamente para a precisão das informações e para a tomada de decisões em tempo hábil. Aliado a isso, o uso de equipamentos de baixo custo ampliou a acessibilidade e a viabilidade da solução, permitindo sua reprodução e implantação em diferentes contextos sem comprometer desempenho ou qualidade. Assim, a solução proposta cumpre o objetivo de rastreamento básico, economicamente viável, extensível e pode ser implantada gradualmente.

Os resultados experimentais confirmaram que o sistema atende aos requisitos propostos para utilização de RSSFs, configurando-se como solução viável para redes de monitoramento internas. Contudo, aspectos inerentes às RSSFs — como consumo de energia, tolerância a falhas, escalabilidade, topologia dinâmica, condições ambientais e segurança — devem ser cuidadosamente considerados no projeto e na implementação, pois impactam diretamente a confiabilidade e a eficiência da rede. Vale ressaltar que o RSSI fornece uma estimativa de proximidade, não sendo um sistema de triangulação preciso e que pode sofrer interferência, porém para o propósito de localização, mesmo que aproximada, cumpre com o esperado.

Como perspectivas futuras, destacam-se o aprimoramento dos mecanismos de segurança (autenticação, criptografia e confiabilidade de entrega), a ampliação do número de nós para análises quantitativas em diferentes topologias, a comparação de estratégias de armazenamento e distribuição de dados, o registro contínuo das medições para geração de indicadores e a validação da solução em cenários reais de aplicação. Dessa forma, conclui-se que a combinação entre MQTT, arquitetura orientada a mensagens e dispositivos de baixo custo resulta em uma solução robusta, escalável e com potencial de implementação em diversos ambientes, como contextos hospitalares, industriais e domésticos.

Referências

- Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36.
- Cao-hoang, T. and Duy, C. N. (2017). Environment monitoring system for agricultural application based on wireless sensor network. In *2017 Seventh International Conference on Information Science and Technology (ICIST)*, pages 99–102.
- Carneiro da Cunha, B. and Macêdo Batista, D. (2022). Avaliação da integração do protocolo mqtt em uma plataforma de cidades inteligentes. *Revista Eletrônica de Iniciação Científica em Computação*, 20(2). Acesso em: 12 fev. 2026.
- da Costa, V., Nery, M. M., Paes, G., de Souza, P., Almentero, E., and Branco, L. (2021). Monitoramento e suporte a controle de estufas climatizadas utilizando nuvem e mqtt.

- In *Anais da VII Escola Regional de Sistemas de Informação do Rio de Janeiro*, pages 80–87, Porto Alegre, RS, Brasil. SBC.
- Eclipse Foundation (2024a). Eclipse Mosquitto: An Open Source MQTT Broker. <https://mosquitto.org/>. Acessado em: 12 Feb. 2026.
- Eclipse Foundation (2024b). Eclipse Paho MQTT Python Client. Acessado em: 12 Feb. 2026.
- Google LLC (2024). Firebase Realtime Database Documentation. Acessado em: 12 Feb. 2026.
- Hunkeler, U., Truong, H. L., and Stanford-Clark, A. (2008). MQTT-S - A publish/subscribe protocol for wireless sensor networks. In *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 791–798.
- Marinho, C. S., Barbosa, G. I., Silva, J. F. M. C., and Menezes, J. W. M. (2017). Rede de sensores sem fio para aplicações em agricultura familiar. In *XXXV SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS – SBrT2017*, pages 995–996.
- Martins, I. R. and Zem, J. L. (2015). Estudo dos protocolos de comunicação mqtt e coap para aplicações machine-to-machine e internet das coisas. *Revista Tecnológica da Fatec Americana*, 3(1):64–87.
- Quincozes, S., Emilio, T., and Kazienko, J. (2019). Mqtt protocol: Fundamentals, tools and future directions. *IEEE Latin America Transactions*, 17(09):1439–1448.
- Rocha, M. C. (2022). Sistema de monitoramento baseado em mqtt. Trabalho de conclusão de curso (bacharelado em engenharia de controle e automação), Universidade Federal De Santa Catarina, Florianópolis.
- Silva, A. C., Aguiar, C. G., Fagundes, G. P., and Vilas Boas, R. R. (2024). Automação em sistemas de detecção e alarme de incêndio. Trabalho de Conclusão de Curso (Tecnólogo em Automação Industrial), Faculdade de Tecnologia "Adib Moisés Dib".