

# Desenvolvimento de um Sistema Ciber Físico de Eficiência Energética e Condicionamento Ambiental Utilizando ThingsBoard

Thiago Maximo Pavão<sup>1</sup>, Rafael Ifanger Ribeiro<sup>1</sup>, Antonio Pestana Neto<sup>1</sup>

<sup>1</sup>Divisão de Sistemas Ciberfísicos - DISCF  
CTI Renato Archer – Campinas/SP

tmpavao@cti.gov.br, riribeiro@cti.gov.br, antonio.pestana@cti.gov.br

**Abstract.** *Thermal comfort is essential for the health and productivity of occupants in any environment. This work presents the development of a system that automatically regulates environmental variables, with a focus on minimizing the energy consumption required to achieve thermal comfort. The article documents the creation of the technological foundation necessary for implementing this system, including sensor modules, actuators, and the use of ThingsBoard for managing devices and storing the generated data. This technological foundation will serve as the basis for future studies and implementations aimed at efficient environmental control.*

**Resumo.** *O conforto térmico é essencial para a saúde e a produtividade dos ocupantes em qualquer ambiente. Este trabalho apresenta o desenvolvimento de um sistema que regula automaticamente as variáveis ambientais, com foco em minimizar o consumo energético necessário para alcançar o conforto térmico. O artigo documenta a criação da base tecnológica necessária para a implementação desse sistema, que inclui módulos sensoriais, atuadores e a utilização do ThingsBoard para gerenciar os dispositivos e armazenar os dados gerados. Essa fundação tecnológica servirá de base para futuros estudos e implementações voltadas ao controle ambiental eficiente.*

## 1. Introdução e Motivação

O conforto térmico em ambientes internos, como escritórios, é crucial para assegurar o bem-estar dos ocupantes. A falta de conforto térmico não só pode comprometer a saúde em situações extremas, mas também afetar a concentração, causando desconfortos como dores de cabeça. Atualmente, esse conforto é geralmente mantido por sistemas de condicionamento ambiental, sendo o ar condicionado o mais comum.

No entanto, o uso inadequado desses equipamentos frequentemente resulta em um consumo de energia maior do que o necessário para alcançar o conforto térmico desejado. Diversos fatores contribuem para isso, incluindo o desconhecimento dos usuários sobre o uso eficiente dos aparelhos, o desejo de aliviar rapidamente o desconforto térmico e a falta de percepção precisa das necessidades térmicas pessoais. Por exemplo, um usuário pode ajustar o termostato para uma temperatura que julga confortável, mas que é significativamente inferior a outra que também atenderia suas necessidades, resultando em um consumo energético desnecessário.

Para mitigar esse problema, propõe-se o desenvolvimento de um sistema automático de controle ambiental, capaz de monitorar variáveis como temperatura, umidade relativa e o consumo de energia dos equipamentos de climatização. Esse sistema atuaria diretamente no controle desses dispositivos e seria capaz de receber feedback dos ocupantes, ajustando-se continuamente às suas necessidades térmicas. O objetivo final do sistema é alcançar o conforto térmico dos ocupantes com o menor consumo energético possível.

Outros sistemas IoT para controle ambiental já foram desenvolvidos, cada um com seus focos e particularidades. Um exemplo pode ser visto em [Figueiredo et al. 2020], que utiliza uma arquitetura diferente para o sistema, realizando medições de variáveis ambientais e atuando em equipamentos de condicionamento ambiental. A principal diferença para o sistema apresentado neste trabalho é o foco em alcançar o conforto térmico, levando em consideração também a eficiência energética. O desenvolvimento foi fortemente guiado pelas lições aprendidas a partir da implementação mostrada em [Neto et al. 2022], com o objetivo de aprimorar o que fosse possível.

## **2. O Sistema**

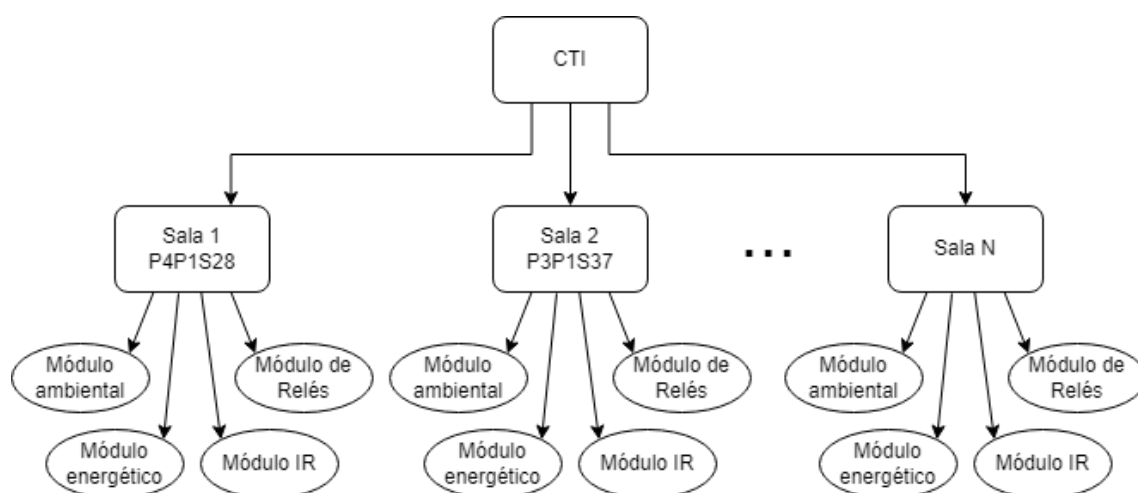
Para gerenciar os módulos físicos do sistema, como sensores e atuadores, foi escolhida a plataforma de código aberto ThingsBoard. Além de gerenciar esses módulos, a plataforma oferece ferramentas de organização hierárquica, gerenciamento automático de banco de dados para armazenamento de dados dos dispositivos e recursos para visualização dos dados, permitindo a criação de dashboards dentro da própria plataforma. A API de comunicação do ThingsBoard com os dispositivos também permite o envio de comandos, possibilitando o controle dos equipamentos de condicionamento ambiental por meio da plataforma.

Além disso, a equipe de desenvolvimento do ThingsBoard disponibiliza ferramentas adicionais, como uma biblioteca para a IDE do Arduino que facilita a interação com a plataforma, simplificando o código de aplicação presente nos módulos. Também há um cliente REST para Python e outras linguagens, que permite a interação com a plataforma e suas funcionalidades por meio de programas externos. Isso é essencial para realizar análises externas e implementar algoritmos de controle, que podem buscar dados dos dispositivos e enviar ações de controle utilizando a API REST fornecida.

A estrutura hierárquica criada na plataforma para uma melhor organização e facilidade de interação programática com os módulos é apresentada na Figura 1. Os retângulos, denominados “Ativos” na nomenclatura do ThingsBoard, representam qualquer entidade desejada, exceto dispositivos. Neste caso, foi criado um ativo raiz denominado “CTI” e, para cada sala monitorada e controlada pelo sistema, um ativo adicional. Cada sala contém os dispositivos presentes nela, incluindo sensores e atuadores, que estão representados por elipses e configurados na plataforma como “Dispositivos”.

## **3. Os Módulos**

A plataforma ThingsBoard foi instalada em um dos servidores do CTI, permitindo que os módulos criados se comuniquem com ela via Wi-Fi. Programas externos que interagem com a plataforma podem se comunicar pela rede ou ser executados no mesmo servidor. Isso será útil, por exemplo, na implementação do controle ambiental, que poderá ser



**Figura 1. Estrutura hierárquica do sistema na plataforma ThingsBoard.**

realizado por um programa rodando no mesmo servidor e interagindo diretamente com o ThingsBoard.

No entanto, para possibilitar essa implementação, é necessário, primeiramente, construir os módulos físicos (sensores e atuadores) e desenvolver seus respectivos softwares. Isso permitirá a coleta de dados ambientais e energéticos, bem como a atuação sobre o ambiente. O módulo de relés controlará equipamentos como umidificadores, desumidificadores, ventiladores, entre outros, enquanto o módulo IR (infravermelho) será responsável pelo controle do ar condicionado. Além disso, foi implementada uma forma de feedback da sensação térmica dos ocupantes da sala por meio de uma página web. A seguir, serão detalhadas as implementações de cada um desses componentes do sistema.

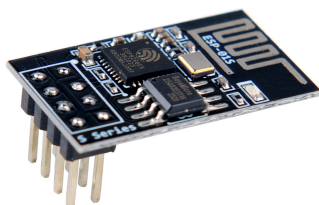
A comunicação entre os módulos e a plataforma ThingsBoard é realizada via MQTT, um protocolo de mensagens leve e eficiente, ideal para dispositivos com recursos limitados e redes instáveis. O MQTT permite uma comunicação assíncrona e de baixa latência entre os módulos e o servidor, garantindo que os dados de telemetria sejam transmitidos de maneira confiável e que os comandos de controle sejam recebidos pelos dispositivos em tempo real.

### 3.1. Módulo Ambiental

O módulo ambiental é responsável por medir temperatura e umidade relativa, com a opção de medir também a pressão atmosférica. Existem diversos sensores disponíveis para essa finalidade, variando em termos de precisão, exatidão, tipo de interface de comunicação e tecnologia utilizada para realizar as medições. O software foi desenvolvido para suportar vários sensores, como DHT11, DHT22, DS18B20 e BME280.

Devido à simplicidade das funcionalidades necessárias—dada apenas pela comunicação com um sensor e via Wi-Fi—foi possível utilizar a placa microcontroladora ESP-01 (Figura 2) como o principal componente do módulo. Embora seja geralmente usada apenas como um módulo Wi-Fi para outras microcontroladoras, é possível carregar um programa na ESP-01 e utilizá-la como um módulo independente, desde que as necessidades de recursos sejam limitadas. A placa, baseada no chip ESP8266, não possui tanta memória e capacidade de processamento quanto outras placas mais modernas, mas

é suficiente para esta aplicação e permite um módulo compacto, reduzindo seu tamanho físico.



**Figura 2. Microcontrolador ESP-01 utilizado no módulo ambiental.**

O software desenvolvido para o módulo segue a seguinte lógica geral: o módulo inicia e se conecta ao ThingsBoard, reportando o status do sensor e realizando a sincronização temporal com o tempo real através de uma requisição NTP. Em seguida, entra em um loop que aguarda o momento correto para realizar a medição, interagindo com o sensor para obter os dados. Se houver perda de conexão com o Wi-Fi ou com o ThingsBoard, o programa tenta reconectar-se antes de enviar os dados medidos como telemetria para o ThingsBoard.

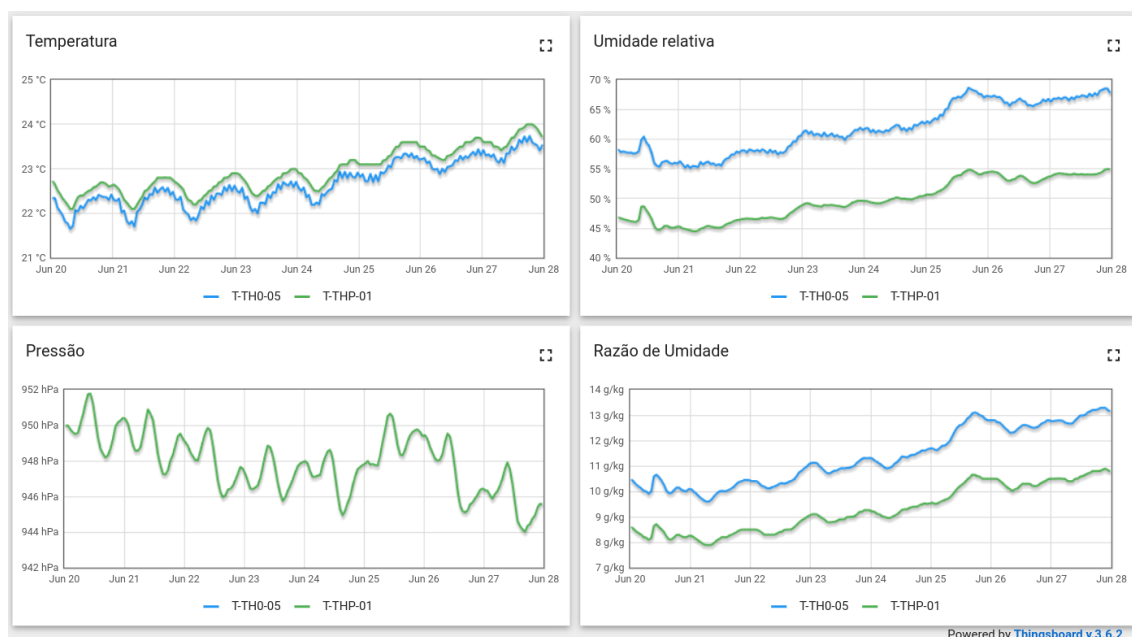
É importante destacar que o timestamp da medição é armazenado no momento em que ela é feita; assim, mesmo que o envio do dado seja atrasado por reconexões, o tempo armazenado no ThingsBoard será o momento exato da medição. Para melhorar a qualidade do sinal medido, a medição do sensor é realizada cinco vezes (valor configurável), e apenas a média é considerada.

Utilizando as medidas de temperatura, umidade relativa e pressão, o sistema pode calcular outras variáveis úteis para análise, como umidade absoluta e razão de umidade, que são então enviadas ao ThingsBoard. Para sensores que não medem a pressão atmosférica, foi configurado um valor padrão de 950 hPa, uma boa aproximação para o local dos experimentos.

Outra lógica importante é a manutenção da sincronização temporal com o tempo real. Como a requisição NTP é feita via UDP, não há garantia de que haverá resposta do servidor, pois tanto a requisição quanto a resposta podem ser perdidas e não serão retransmitidas. Além disso, o servidor NTP configurado pode estar offline. Nesse caso, o módulo continua utilizando o tempo do seu relógio interno até que a conexão seja restabelecida.

O programa foi projetado para tentar sincronizar o tempo a cada 6 horas, enviando uma notificação de erro ao ThingsBoard caso não haja resposta por mais de 3 tentativas consecutivas, garantindo que qualquer problema de sincronização seja detectado antes de se tornar crítico.

Na Figura 3, pode ser visto um dashboard feito na plataforma mostra uma comparação das medidas realizadas e enviadas por dois dos sensores testados, o BME280 e o DHT22. É possível observar que os sensores apresentam pequenas divergências em suas medições, embora com um valor quase fixo de diferença, indicando a necessidade de calibração dos valores medidos, o que ainda precisa ser incorporado ao programa.



**Figura 3. Comparação de sensores: módulo T-THP-01 com o sensor BME280 e T-TH0-05 com o sensor DHT22.**

### 3.2. Módulo de Energia

Este módulo é responsável por monitorar o consumo de energia de um aparelho, obtendo parâmetros básicos como potência ativa, tensão RMS e corrente RMS. Além disso, o programa desenvolvido também calcula parâmetros de qualidade de energia, permitindo uma análise mais abrangente. Entre esses parâmetros estão a potência aparente, potência reativa, fator de potência, distorção harmônica total da corrente e da tensão, e a frequência da rede elétrica.

O processamento principal do sistema é realizado por um módulo ESP32, que utiliza seus dois conversores analógico-digitais (ADC) para amostrar simultaneamente as saídas dos sensores de tensão e corrente, com cada ADC dedicado a um sensor específico. A ESP32 possui um processador de dois núcleos, o que permite ao sistema executar duas tarefas principais em paralelo: a amostragem dos sinais e o processamento dos dados coletados. Essas tarefas ocorrem simultaneamente, com cada núcleo do processador dedicado a uma dessas funções.

Normalmente, o segundo ADC e o segundo núcleo da ESP32 são utilizados para gerenciar a conexão Wi-Fi. No entanto, para otimizar o desempenho do módulo, foi decidido utilizar uma ESP-01 dedicada para essa comunicação. As informações de energia são transmitidas entre a ESP32 e a ESP-01 via uma conexão UART, conforme detalhado a seguir.

Devido à grande quantidade de dados transmitidos entre as placas, foi incorporado um algoritmo de verificação de erros de transmissão conhecido como CRC (Cyclic Redundancy Check). Esse algoritmo gera um valor de 32 bits com base na mensagem completa a ser enviada. O destinatário, então, recalcula o CRC e verifica se o valor calculado corresponde ao transmitido. Caso um erro seja detectado, a placa envia um pedido de retransmissão pela interface UART, garantindo que apenas dados corretos sejam enviados.

ao ThingsBoard.

A lógica do programa que executa na ESP-01 é semelhante à explicada para o módulo ambiental. Porém, em vez de se comunicar com sensores de temperatura e umidade, a ESP-01 se comunica com a ESP32 via UART no momento da medição. Apesar da relativa lentidão da comunicação UART, ela é adequada para o projeto, já que as medidas requisitadas pela ESP-01 correspondem à média dos valores de energia desde a última requisição. Atualmente, uma requisição é feita a cada 30 segundos, um intervalo muito maior do que o necessário para transmitir as informações entre as placas.

Nas Figuras 4 e 5, são apresentados os resultados de um teste realizado para medir o consumo de energia de um ar condicionado. Nas figuras, é possível observar três períodos de funcionamento do ar condicionado: aproximadamente das 16:00 às 16:20, das 16:23 às 16:27, e das 17:53 às 18:06. Na Figura 5, nota-se um aumento progressivo da corrente consumida, acompanhado por uma pequena queda na tensão da rede elétrica nos períodos de funcionamento do equipamento. A frequência da rede permanece muito próxima de 60Hz, conforme o esperado.

Na Figura 4, observa-se um aumento na potência consumida durante estes períodos, com a potência caindo para zero nos intervalos em que o aparelho está desligado. É importante destacar que o valor de energia acumulada é enviado continuamente ao servidor, registrando o consumo de energia ao longo do tempo. Mesmo que o módulo perca a conexão com o ThingsBoard por um período, por exemplo, uma hora, o valor de energia acumulada refletirá o consumo real de energia durante o período offline, garantindo a precisão dos dados.

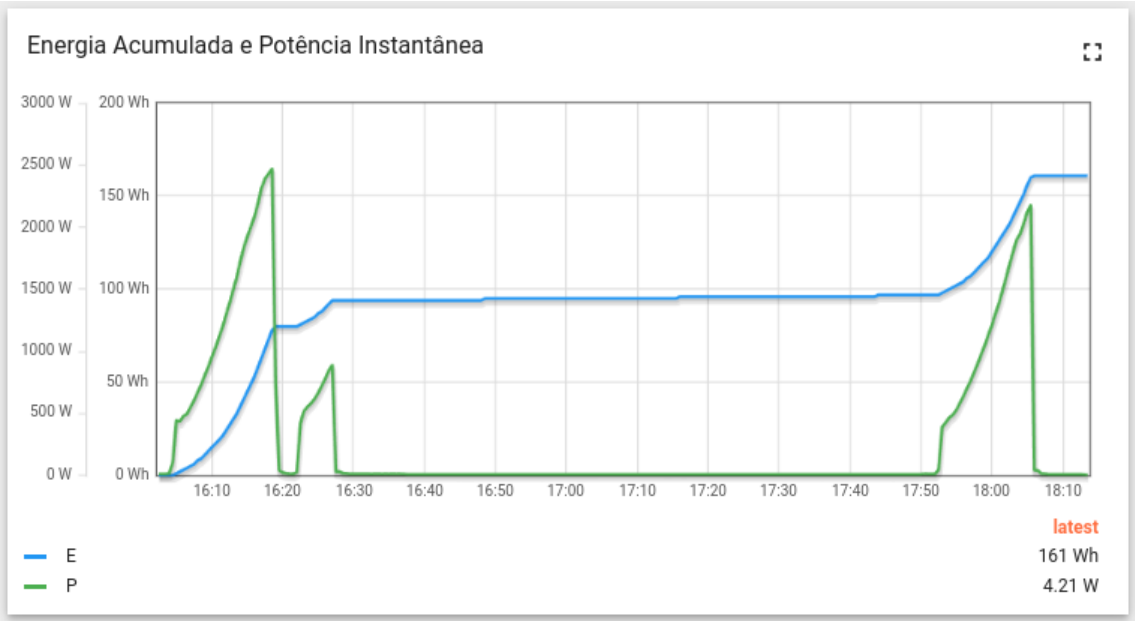
Existem melhorias que podem ser implementadas no módulo, como a transmissão de dados apenas durante os períodos em que o equipamento está ativamente consumindo energia. Além dos dados apresentados nas figuras, o espectro do sinal também é enviado, permitindo uma análise mais detalhada da qualidade da rede elétrica. No entanto, para economizar espaço no banco de dados, esse espectro poderia ser enviado somente quando fosse detectada uma distorção harmônica significativa.

### 3.3. Módulo de Relés

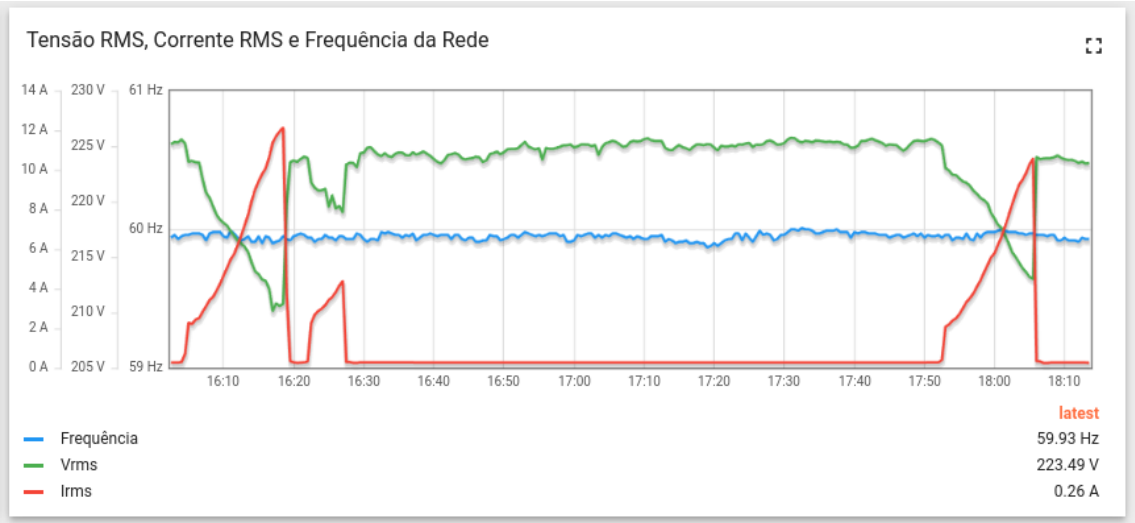
Este módulo é responsável por controlar certos equipamentos, sendo capaz de ligar e desligar a alimentação fornecida a eles. Para isso, é utilizada a API fornecida pelo ThingsBoard chamada RPC (Remote Procedure Calls). Essa API foi projetada para o envio de comandos aos dispositivos, encaixando-se perfeitamente na aplicação necessária.

O software no dispositivo é relativamente simples: não há necessidade de sincronização temporal, visto que basta executar o comando solicitado. O módulo se inscreve nas RPCs, definindo funções que devem ser chamadas quando uma RPC é recebida. Cada RPC é composta por um nome e parâmetros, semelhante a uma chamada de função. No módulo, foram definidas duas RPCs que implementam as seguintes funções:

- *setDeviceState* - Recebe como parâmetro um número de dispositivo e o estado desejado (ON/OFF).
- *getDeviceState* - Retorna o estado atual de cada dispositivo controlado pelo módulo.



**Figura 4. Gráfico gerado pelo Thingsboard com os dados de energia acumulada e potência instantânea medidos pelo módulo de energia em um período de testes com o ar condicionado.**



**Figura 5. Tensão RMS, corrente RMS e Frequência da rede elétrica reportados ao Thingsboard no teste.**

O programa foi desenvolvido de maneira genérica, permitindo o controle de vários dispositivos. Cada tomada elétrica presente no módulo é controlada por um relé e possui um número de dispositivo associado. A relação entre qual dispositivo está vinculado a cada número deve ser configurada manualmente, podendo ser ajustada também no ThingsBoard. Na Figura 6, são mostradas algumas placas prontas, com diferentes números de relés, compatíveis com a ESP-01.



**Figura 6. Exemplos de placas com 1, 2 e 4 relés, compatíveis com a ESP-01.**

### **3.4. Módulo IR**

Este módulo de controle é responsável pelo ar condicionado, permitindo configurar cada parâmetro que normalmente é ajustado pelo controle remoto do equipamento. Esses parâmetros incluem velocidade do vento, temperatura, modo de operação, entre outros. Assim como no módulo de relés, o envio de comandos é realizado através de uma RPC (Remote Procedure Call), onde cada um dos parâmetros configuráveis é definido e enviado ao ThingsBoard.

Para garantir que o comando foi efetivamente recebido pelo sistema e não bloqueado por algum motivo antes de chegar ao ar condicionado, o módulo conta com um microfone que verifica o acoplamento com o ar condicionado, aguardando pela frequência do som emitido pelo aparelho após receber um comando via IR. Para realizar essa verificação, foi utilizado o Algoritmo de Goertzel, que detecta a amplitude da frequência emitida pelo equipamento.

### **3.5. Página de Feedback**

Um componente fundamental do sistema é a capacidade de receber feedback dos ocupantes da sala, permitindo que o sistema se ajuste em tempo real às sensações das pessoas no ambiente controlado. O objetivo é alcançar o ponto de menor consumo energético que satisfaça as necessidades térmicas dos ocupantes. Dessa forma, o sistema pode, por exemplo, consumir menos energia em ambientes com pessoas que não sentem tanto calor.

Para receber esse feedback, foi desenvolvido um servidor web em Python, utilizando o framework Flask. Esse servidor se comunica com o ThingsBoard para recuperar



as salas mostradas na hierarquia da Figura 1 e registrar o feedback enviado como dado de telemetria no ativo correspondente à sala onde o voto foi realizado.

Dessa maneira, qualquer usuário acessa a mesma página, mas as sensações reportadas são salvas separadamente para cada sala. Isso é feito através de argumentos na URL, e, para o ambiente do CTI, essa abordagem é suficiente. Em outras implementações, poderia ser necessário utilizar autenticação para gerenciar de forma controlada os feedbacks enviados.

Na Figura 7, é mostrado um exemplo de voto realizado na página de feedback da sala de manutenção. Ao enviar o feedback, o dado é salvo no banco de dados do ThingsBoard como telemetria do ativo associado à sala correspondente, permitindo futuras análises sobre as condições relatadas pelos ocupantes.

Neste exemplo, o servidor identifica em qual ativo do ThingsBoard registrar o voto através do parâmetro na URL, *room=P3P1S37*, que é o nome do ativo configurado no ThingsBoard. O texto “Sala de Manutenção” também é recuperado do ThingsBoard, sendo a descrição configurada para o ativo. Isso demonstra a facilidade de adicionar novas salas ao sistema: ao criar um novo ativo na hierarquia, dentro do elemento raiz “CTI”, a página de feedback já estará disponível na URL correspondente.

SCF Energético Ambiental

192.168.15.12:5000/?room=P3P1S37

Sistema Ciberfísico Energético Ambiental

Feedback Dashboards

## Feedback de Sensação Térmica

### Sala da Manutenção

☐ Congelante

☐ Muito frio

☐ Frio

☐ Neutro

☒ Calor

☐ Muito calor

☐ Escaldante

Enviar

Sensação enviada com sucesso!

Último feedback registrado na sala: Calor

Um novo feedback poderá ser enviado em 10 minutos.

**Figura 7. Exemplo de voto realizado na página de feedback da Sala de Manutenção.**

O valor enviado ao ThingsBoard para armazenamento varia de -3 (Congelante) a 3 (Escaldante), seguindo a escala do PMV. Esse valor pode ser utilizado para prever a sensação das pessoas no ambiente, considerando diversas variáveis, como temperatura e umidade relativa. Um estudo sobre os parâmetros envolvidos no cálculo do PMV pode ser encontrado em [Ekici 2013]. Entretanto, a página de feedback desenvolvida não substitui o uso do PMV, mas complementa-o, garantindo que o sistema seja flexível e atenda às sensações específicas e particulares das pessoas presentes. O PMV pode ser utilizado em conjunto para reduzir a necessidade de interação do usuário com o sistema.

#### 4. Resultados e Conclusão

Os resultados obtidos com o desenvolvimento de cada componente do sistema são promissores. A implementação sobre a plataforma ThingsBoard tem se mostrado extremamente confiável, e o foco na simplicidade de cada módulo ajudou a minimizar a probabilidade de erros no sistema. Até o momento, os testes realizados demonstraram que o sistema é capaz de operar de forma contínua e estável por longos períodos. Ocasionalmente algum módulo reporta problemas de leitura de sensores, mas é capaz de se recuperar automaticamente e seguir com a coleta de dados.

A escalabilidade oferecida pela plataforma ThingsBoard é um ponto forte, permitindo a incorporação futura de novos módulos e funcionalidades sem comprometer a integridade do sistema existente. Isso é especialmente relevante, dado o número elevado de variáveis que podem ser incorporadas na tomada de decisão do sistema. O estudo de [Fang et al. 2020] demonstra como a integração de dados fisiológicos, como a frequência cardíaca, pode proporcionar uma compreensão mais profunda das necessidades de conforto térmico. No futuro, o sistema poderia monitorar esse tipo de dado, ampliando ainda mais suas capacidades.

O desenvolvimento bem-sucedido destes módulos tecnológicos prepara o terreno para a próxima fase do projeto: a análise de dados e a experimentação com diferentes algoritmos de controle. Com a base tecnológica estabelecida, será possível explorar o uso de técnicas avançadas, como inteligência artificial, para otimizar o controle ambiental e melhorar a eficiência energética.

#### Referências

- Ekici, C. (2013). A review of thermal comfort and method of using fanger's pmv equation. pages 61–64.
- Fang, Y., Lim, Y., Ooi, S. E., Zhou, C., and Tan, Y. (2020). Study of human thermal comfort for cyber-physical human centric system in smart homes. *Sensors*, 20(2).
- Figueiredo, R. E., Alves, A. A., Monteiro, V., Pinto, J. G., Afonso, J. L., and Afonso, J. A. (2020). Development and evaluation of smart home iot systems applied to hvac monitoring and control. *EAI Endorsed Transactions on Energy Web*, 8(34).
- Neto, A. P., Resende, R. A. O., Nicolau, M., Pimentel, M. R., Ribeiro, R. I., Filho, G. B., Fruett, F., and Ramos, J. J. G. (2022). A cyber-physical system for energy efficiency and indoor air conditioning of multiple office rooms. In *2022 Symposium on Internet of Things (SIoT)*, pages 1–4.