

# SISTEMA DE MONITORAMENTO DE TEMPERATURA, UMIDADE E PRESENÇA, BASEADO EM IoT, PARA *DATACENTERS* E AMBIENTES COMPUTACIONAIS CRÍTICOS

Guilherme da Silva Donizetti<sup>1</sup>  
Eugênio Sper de Almeida<sup>2</sup>

<sup>1</sup>Aluno do CST em Análise e Desenvolvimento de Sistemas; e-mail:  
guilherme.donizetti@fatec.sp.gov.br

<sup>2</sup>Professor da Fatec Cruzeiro – Prof. Waldomiro May; e-mail: eugenio.almeida@fatec.sp.gov.br

**Área do Conhecimento:** Sistemas de Computação.

**Palavras-chave:** IoT; *datacenter*; temperatura; umidade; presença.

## INTRODUÇÃO

*Datacenters* e ambientes computacionais críticos oferecem serviços contínuos de Tecnologia da Informação (TI) e contém servidores e demais equipamentos necessários para prover serviços computacionais (YOGENDRA, 2012). De acordo com a norma ANSI/TIA-942 (*Telecommunications Infrastructure Standard for Data Centers*), o controle de temperatura (entre 20°C e 25°C), umidade relativa do ar (40% e 45%) e presença de pessoas são diretrizes para garantir o funcionamento dos recursos e a segurança dos dados (ANSI/TIA, 2012). Atualmente este monitoramento é realizado utilizando equipamentos proprietários que possuem preço elevado. Uma solução baseada em IoT permite a diminuição do custo através da utilização de sensores e microcontroladores de valores mais acessíveis e ferramentas de código aberto. Ciuffoletti (2018) apresenta uma revisão de literatura de IoT de baixo custo. IoT é uma rede aberta que reúne objetos inteligentes e possui a capacidade de se auto-organizar e de compartilhar dados, informações e recursos, e com isso agir e reagir às mudanças do ambiente (Madakam et al., 2015).

## OBJETIVOS

Este projeto tem como objetivo desenvolver um sistema de monitoramento em tempo real de temperatura, umidade e presença para *datacenters* e ambientes computacionais críticos, baseado em Internet das Coisas (IoT).

## METODOLOGIA

O trabalho adotou ferramentas *open source* para o desenvolvimento da solução. O custo está relacionado apenas à aquisição de sensores e microcontroladores. Para a captura dos dados de temperatura e umidade relativa foi usado o sensor *M5StickC ENV HAT II.R*. Suas características incluem faixa de operação entre 0 e 60 °C, precisão de 0.2 °C e umidade entre 0% e 100% com precisão de 2%. O sensor *M5StickC PIR Hat* é um sensor de indução humana que capta o sinal de infravermelho em até 5 metros, com tempo de resposta de 2 segundos e opera em ambientes com temperatura na faixa de -20 a 80 °C. O *M5StickC ESP32-PICO Mini IoT Development Kit 3* é baseado no microcontrolador (MCU) *ESP32-PICO* e possui as seguintes funcionalidades: *WiFi*, *Bluetooth* e *display* de 1,14 polegadas acoplado. Os sensores são conectados diretamente ao MCU. O microcontrolador foi programado utilizando o Arduino IDE, através da biblioteca *Adafruit\_Sensor*. Para que fosse reconhecido pela porta USB, foi usado o *driver* FTDI que realiza esse reconhecimento e o *driver* CP210 para o funcionamento do *firmware*. Um workflow (Fig. 1) foi implementado utilizando o *broker* MQTT Mosquitto (2.0.11), Node-Red (1.2.2), InfluxDB (1.6.4), Grafana (8.1.1), agente SMTP chamado sSMTP e o microframework Python Flask. Todas essas ferramentas foram programadas para garantir que os dados ambientais sejam capturados, transmitidos para armazenamento e visualizados. Os tópicos do *broker* usaram o formato “prédio/sala/posição”. O

prédio e a sala têm valores predefinidos, respectivamente CPT e DC. A posição é formada pela coordenada do eixo X e Y de onde está o sensor no datacenter. Um possível tópico é CPT/DC/AB0102, onde o tópico se refere aos dados do prédio CPT, da sala DC, coluna AE e linhas 01 e 02.

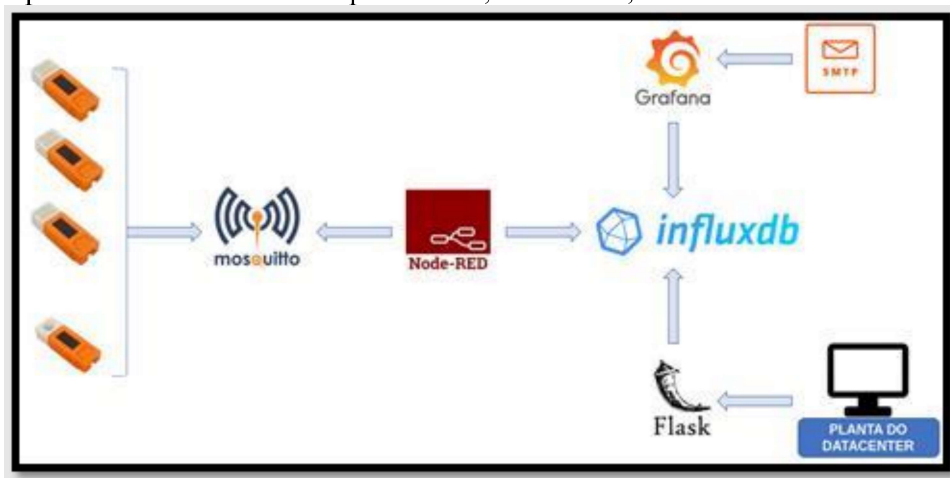


Figura 1: Workflow do projeto

Utilizando Flask, desenvolveu-se uma página Web que apresenta a planta baixa do *datacenter*, segundo o padrão da ANSI/TIA-942. Nela é apresentada a divisão dos pisos, destacando os corredores quentes e frios e os corredores de racks. Os pisos na horizontal formam uma linha, com identificação numérica, e na vertical formam colunas, identificados por letras. Ao longo dos corredores frios foram posicionados os sensores com espaço de duas colunas entre si (120 centímetros). Quando os sensores enviam os dados coletados, enviam junto a identificação do piso onde estão localizados, formado pela junção da identificação da coluna e da linha. Então a aplicação consulta os dados de temperatura, umidade e a identificação do piso, com esses dados o programa localiza dentro da planta qual piso mudará de cor de acordo com os valores consultados.

## RESULTADOS E DISCUSSÃO

Após coletar os dados, o MCU transmite para o *broker* Mosquitto através do protocolo MQTT. Os dados enviados ao Mosquitto são coletados pelo Node-Red, que possui recursos que executam funções específicas e são chamados “nós”. Os nós *MQTT In* capturam os dados publicados nos tópicos do *broker*. O nó *Functions* aceita qualquer programação funcional em JavaScript, então foi programado para adicionar metadados referentes à localização, modelo dos sensores e margem de erro dos dados capturados. Em seguida, os dados são transmitidos para o armazenamento por meio de um nó do Node-Red que é específico do banco de dados. O armazenamento dos dados utiliza o InfluxDB, um sistema de gerenciamento de banco de dados não estruturado, específico para séries temporais e apresenta alto desempenho na gravação e leitura dos registros. A visualização dos dados é feita através dos gráficos de linhas e de *Gauge* que apresentam a média dos valores de temperatura e umidade dos últimos cinco (5) minutos de cada corredor frio e atualizados a cada cinco (5) segundos. Toda vez que o sensor de indução humana registra a presença de pessoas, o armazenamento recebe o valor “1”, caso não haja presença é armazenado “0”. Esses valores também são representados em gráfico de linhas por oferecer uma interpretação muito clara. A cada atualização dos gráficos, as regras de alertas são analisadas, caso detecte algum ponto fora da curva o sistema notifica por e-mail enviando uma imagem do gráfico, esse recurso de alertas foi aplicado utilizando o Grafana e a ferramenta sSMTP, que é um leve e simples agente de correio eletrônico que executa apenas o envio de mensagens usando o protocolo SMTP (*Simple Mail Transfer Protocol*). Outra visualização dos dados é por meio da planta do datacenter, desenvolvida com ferramentas para Web, o desenvolvimento do software seguiu os padrões de arquitetura MVC. A planta consulta a base de dados a cada cinco (5) segundos e seleciona os registros dos últimos dez (10) segundos para garantir leitura de todos os sensores. Os dados retornados de cada consulta são analisados para atribuir Verde (se as variáveis ambientais estiverem dentro dos limiares),

Amarelo (se as variáveis estiverem em cima dos limiares) e Vermelho (caso alguma variável esteja fora dos limiares). As cores são atribuídas no piso a que se referem os dados coletados e analisados e mostra dentro da planta se cada ponto monitorado está seguro ou não; caso a aplicação não encontre dados referentes aos últimos 10 segundos, será exibido uma página informando que os dados não foram encontrados. A ferramenta apresenta dados atuais e individuais de cada sensor em campo.

## CONCLUSÕES

Com base no referencial teórico foi possível criar um workflow que atendesse às características de um sistema de Internet das Coisas e formado por ferramentas *open source* e de baixo custo. O sistema desenvolvido com base nesse workflow opera em tempo real atendendo a criticidade de *datacenters* e outros ambientes de computação, apresenta os dados através de recursos fáceis de serem interpretados e que mostram as condições térmicas dos corredores do datacenter e sobre a presença de pessoas, para tomadas de decisão em menor tempo possível e análise futura dos dados históricos. A planta baixa permite a visão da área do ambiente que está sendo monitorado e apresenta as informações de temperatura/umidade específica de cada ponto de coleta de dados oferecendo maior precisão no monitoramento.

## REFERÊNCIAS

ANSI/TIA-942, American National Standard Institute / Telecommunication Industry Association. **Normas de Infra-estrutura de Telecomunicações para Data Centers**. 2005.

CIUFFOLETTI, A. (2018). **Low-cost IoT: A holistic approach**. *Journal of Sensor and Actuator Networks*, 7(2), 19.

MADAKAM, Somayya; RAMACHANDRASASTRI, Ramaswamy; TRIPATHI, Siddharth. **Internet of Things (IoT): A Literature Review**. 2015. Disponível em: <[https://www.scirp.org/pdf/JCC\\_2015052516013923.pdf](https://www.scirp.org/pdf/JCC_2015052516013923.pdf)> Acesso em: 9 out. 2020.

YOGENDRA, Joshi Kumar, P. **Energy efficient thermal management of data centers**. Springer Science & Business Media. 2012.