

# Automação Predial Inteligente com a Nuvem – Construção do ambiente inteligente com CLP e ferramenta de programação visual *Node-RED*

Thiago C. dos Santos\*, Adolfo Bauchspiess\*

\* *Lab. de Automação e Robótica - LARA, ENE/FT/UnB, Brasília-DF*  
*e-mail: thiagocs\_99@hotmail.com, adolfobs@ene.unb.br*

---

**Abstract:** One of the most relevant services offered by Smart Environments is thermal comfort in buildings, integrating the available information regarding the automated location. Temperature and humidity sensors were used to obtain the values of these quantities in the location under analysis and in their adjacent environments. The cloud was also integrated into the control system, providing information regarding meteorological data at the time of analysis and the weather forecast, which makes it possible to carry out predictive control. For the communication between the automation devices, a technology was adopted that makes use of radiofrequency, allowing a secure connection, due to the encryption used, and of low consumption. The implemented on-off controller, developed by inserting an open source PLC in the context of building automation, proved itself a cheaper and simpler option when compared to other existing solutions.

**Resumo:** Um dos serviços mais relevantes oferecidos por Ambientes Inteligentes (AmI) é o conforto térmico em prédios, integrando as informações disponíveis relativas ao local automatizado. Sensores de temperatura e umidade foram utilizados para a obtenção dos valores destas grandezas no local em análise e em seus ambientes adjacentes. A nuvem também foi integrada ao sistema de controle, disponibilizando informações referentes aos dados meteorológicos no momento de análise e à previsão do tempo, o que torna possível a realização de um controle preditivo. Para a comunicação entre os dispositivos de automação, adotou-se uma tecnologia que faz uso de radiofrequência, possibilitando uma conexão segura, por conta da criptografia utilizada, e de baixo consumo. O controlador liga-desliga implementado, desenvolvido por meio da inserção de um CLP de código aberto no contexto da automação predial, evidenciou uma opção mais barata e simples quando comparado a outras soluções já existentes.

*Keywords:* Smart Buildings, Cloud, Visual Programming, Node-RED, IoT, PLC.

*Palavras-chave:* Automação predial, Nuvem, Programação em Blocos, Node-RED, IoT, CLP.

---

## 1. INTRODUÇÃO

O conforto térmico é algo indispensável para o desenvolvimento de diversas atividades presentes no cotidiano da sociedade contemporânea. Tal fator é tão desejado e, muitas vezes, necessário, que sua importância é destacada por uma norma regulamentadora, a NR-17, na qual se recomenda que, nos locais de trabalho onde são executadas atividades que exijam solicitação intelectual e atenção constantes, a temperatura, a umidade e outras condições de ergonomia sigam a padrões pré-estabelecidos.

Diante desse contexto, a presença de equipamentos de ventilação, aquecimento e ar condicionado torna-se cada vez maior tanto em prédios comerciais quanto residenciais e, consequentemente, a demanda energética também aumenta. Segundo a Agência Internacional de Energia, o crescente uso destes equipamentos em residências e escritórios em todo o mundo será um dos principais impulsores da demanda global de energia nas próximas

três décadas, o que enfatiza a necessidade do aumento da eficiência energética nos equipamentos utilizados para a climatização artificial.

De acordo com Moreno et al. (2014), as organizações de padronização reconhecem que, além da arquitetura física de um edifício, sistemas inteligentes e automatizados são necessários para melhorar o conforto e a eficiência energética. Dessa forma, uma alternativa utilizada para a solução deste problema é a implementação de ambientes inteligentes (AmI) que possibilitam a existência do conforto térmico com um consumo menor de eletricidade em decorrência do uso de sensores, atuadores e informações obtidas através da internet que possibilitam um controle automático do sistema.

Neste trabalho, para estudar ambientes inteligentes, buscou-se o desenvolvimento de um controle efetivo da sala de reuniões de um laboratório, no qual são utilizados microcontroladores, sensores, computadores e *softwares* para

se obter maior eficiência energética do equipamento de ar condicionado. Ademais, utilizou-se um *software* aberto para se analisar o uso de um controlador lógico programável (CLP) no ambiente de automação predial.

## 2. FUNDAMENTAÇÃO TEÓRICA

Para a criação do sistema de controle do ambiente de estudo foram utilizados protocolos de comunicação distintos de acordo com a aplicação necessária, tais protocolos foram o *MQTT* e o *Modbus TCP/IP*, além da tecnologia *LoRa*. Estas ferramentas também foram utilizados em outros trabalhos, tais como Lu et al. (2020), Teizer et al. (2020), Ferreira et al. (2019) e Perković et al. (2020).

### 2.1 MQTT

O *MQTT*, sigla de *Message Queuing Telemetry Transport*, é um protocolo de mensagens baseado no modelo de publicação e assinatura, simples e leve, projetado para dispositivos restritos e redes de baixa largura de banda, alta latência ou não confiáveis, desenvolvido com base na pilha de protocolos TCP/IP.

O sistema de comunicação consiste de clientes que se comunicam com um servidor, conhecido como “*broker*”, que é o responsável por organizar as mensagens em tópicos. Tais clientes podem ser tanto publicadores quanto inscritos das informações. Os publicadores são responsáveis por publicar todas as informações de interesse em um tópico, como por exemplo, um valor de temperatura obtido por um sensor. Já o inscrito por sua vez, ao se inscrever em um tópico passa a receber as informações que são publicadas no mesmo. Algo importante a se destacar é o fato de que um cliente pode ser inscrito de um tópico e publicador de outro.

Além disso, o papel do *broker* para esta comunicação é fundamental, visto que quando um publicador possui uma informação para enviar aos outros clientes, esse estabelece uma conexão com o *broker* que, ao ter a conexão estabelecida, distribui as informações aos clientes que estão inscritos no tópico. Dessa forma, os clientes devem estar conectados ao *broker* antes de se tornarem inscritos ou publicadores.

No caso do *broker* receber uma informação destinada a um tópico que não possui inscritos no momento, a informação é descartada ao menos que o publicador indique que isso não deve ocorrer. Assim, os inscritos recebem os dados mais recentes dos tópicos ao invés de esperar por uma nova atualização do publicador.

Devido a todas as características citadas, o *MQTT* tem se mostrado como um protocolo ideal para comunicações emergentes de “*machine-to-machine*” (*M2M*) e para as aplicações de “*Internet of Things*” (*IoT*) de dispositivos conectados e para aplicativos móveis onde a largura de banda e a energia da bateria são muito importantes.

### 2.2 Modbus TCP/IP

O *Modbus* é um protocolo de requisição-resposta que utiliza um relacionamento do tipo mestre-escravo no qual

a comunicação sempre ocorre em pares, isto é, um dispositivo deve iniciar a requisição e então aguardar por uma resposta, sendo o dispositivo iniciador, responsável por iniciar cada interação. Dessa forma, aquele que solicita a informação é chamado de mestre, já os que fornecem informação são os escravos.

O protocolo *Modbus* possui algumas versões, as que são destinadas para aplicações seriais (*Modbus RTU* e *Modbus ASCII*) e a utilizada em aplicações *Ethernet* (*Modbus TCP*). O *Modbus TCP/IP* faz uso dos protocolos TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*) que são usados como protocolo para transporte para a Internet.

Ao se utilizarem esses protocolos, os dados são passados para o TCP, onde informações adicionais são anexadas e fornecidas ao IP. O IP coloca os dados em um pacote e os transmite. O TCP deve estabelecer uma conexão antes de transferir os dados, pois é um protocolo baseado em conexão. Dessa forma, por se tratar de um protocolo do tipo mestre-escravo, ou no caso do *Modbus TCP*, do tipo cliente-servidor, o servidor aguarda a conexão com o cliente e, ao ser estabelecida, envia as suas informações a este.

Outro ponto importante em relação ao protocolo *Modbus* é a maneira com que os dados são armazenados. Neste protocolo, existem quatro bancos de dados ou faixas de endereço: *coils*, entradas discretas, registradores *holding* e registradores de entrada. Estas faixas de endereço são importantes para a definição do tipo de dado e do direito de acesso a eles por parte do mestre e dos escravos. Na tabela abaixo, encontram-se as características de cada bloco de memória.

Tabela 1. Blocos de memória do protocolo *Modbus*.

Bloco de memória	Tipo de dados	Acesso ao mestre	Acesso ao escravo
Coils	Booleano	Leitura/escrita	Leitura/escrita
Entradas discretas	Booleano	Somente leitura	Leitura/escrita
Registradores holding	Palavra não sinalizada	Leitura/escrita	Leitura/escrita
Registradores de entrada	Palavra não sinalizada	Somente leitura	Leitura/escrita

Por fim, uma característica fundamental do *Modbus* é a sua aplicação. Sendo assim, é importante se destacar que o principal uso deste protocolo se dá no contexto de automação, com destaque ao ramo industrial, no qual é frequentemente atrelado ao uso dos Controladores Lógicos Programáveis (CLPs).

### 2.3 LoRa

*LoRa* (abreviação de *long range*) é uma tecnologia de comunicação sem fio que tem como principais características transmissões de longo alcance, baixo consumo de energia em sistemas de borda e transmissão segura de dados, aspectos que a tornam interessante para o contexto de *IoT*. Além disso, estes fatores inserem a tecnologia no sistema de comunicação *LPWAN* (*Low Power Wide*

Area Network), no qual estão inseridas as redes de longa distância e baixa potência.

O *LoRa* possui um protocolo padrão conhecido com *LoRaWAN*. Este é um protocolo de camada de controle de acesso médio (*MAC*), utilizado para gerenciar a comunicação entre *gateways LPWAN* e dispositivos de nó final como um protocolo de roteamento, mantido pela *LoRa Alliance*, uma associação aberta sem fins lucrativos que tem como objetivo padronizar a *LPWAN* para permitir a implantação em larga escala da *IoT* por meio do desenvolvimento e promoção do padrão aberto.

Por outro lado, a tecnologia *LoRa* também permite a implementação de protocolos distintos, o que também possibilita adequá-la a situação na qual será utilizada.

### 3. METODOLOGIA

Para a criação do ambiente inteligente, foram utilizados quatro microcontroladores ESP32 LoRa que possibilitam conexão *Wi-Fi*, *Bluetooth* e *LoRa*, permitindo que o ambiente criado possua conexão sem fio. Ademais, para conseguir as informações empregadas no controle do ambiente, a temperatura e a umidade, foi feito o uso de sensores DHT22 que enviam os dados de maneira digital aos microcontroladores.

Dessa forma, dois ESP32 foram utilizados em conjunto com os sensores para realizar a medida da temperatura e da umidade nas regiões ao redor do ambiente analisado, isto é, os valores das grandezas no ambiente externo à sala e na sala vizinha. Além disso, outro ESP32 também foi conectado a um sensor, com o intuito de se obter as medidas dentro do ambiente no qual se realizou o controle. Este mesmo microcontrolador foi responsável por implementar o controle, acionando um relé de estado sólido que controlava o atuador.

Um dos objetivos do trabalho também foi a avaliação do comportamento de um CLP no contexto da automação predial. Tendo isto em vista, os microcontroladores que realizaram a aquisição das grandezas foram programados fazendo-se uso do *OpenPLC*, um CLP de código aberto que possibilita a programação utilizando-se as linguagens padronizadas para os CLPs de acordo com a norma IEC 61131-3. Por conta disto, um *Raspberry Pi* também foi integrado ao material utilizado, visto que neste foi executado o servidor do *OpenPLC*.

Além disso, enquanto alguns ESP32 foram utilizados como nós da rede, um deles realizou o papel de *gateway*, isto é, o responsável por receber as informações enviadas pelos demais através da tecnologia *LoRa* e publicar na nuvem, na qual foi criado um supervisão utilizando-se a ferramenta de programação *Node-RED*, permitindo a manipulação dos dados e o controle do sistema. Além do mais, utilizou-se o banco de dados *Cloudant* da IBM para salvar os dados das aquisições. Na figura 1, é possível se observar o diagrama da conexão dos dispositivos utilizados.

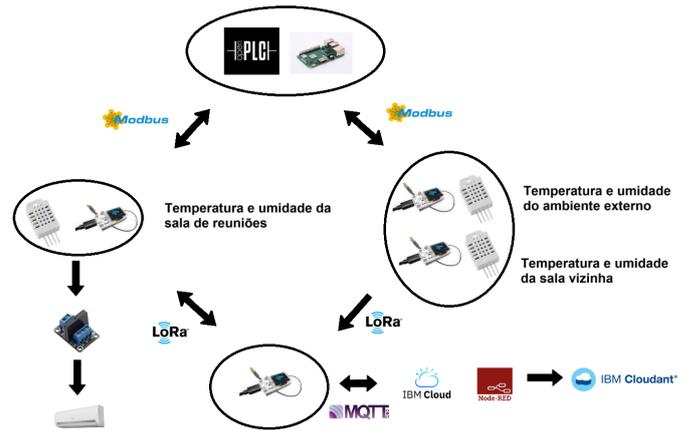


Figura 1. Conexão dos dispositivos.

#### 3.1 Ambiente de Teste

Inicialmente, os testes do sistema projetado foram feitos em uma maquete de madeira com divisórias em seu interior, cujo objetivo era simular a sala de reuniões do Laboratório de Robótica e Automação (LARA) da Universidade de Brasília. Além disso, a maquete também possuía aberturas nas divisórias de forma a representar as portas e janelas do ambiente, contendo um tampo de vidro, com o intuito de se analisar o fluxo de calor e as mudanças de temperatura no ambiente escolhido.

Na maquete, ao invés de se resfriar o ambiente, utilizou-se um secador de cabelo como atuador de forma a aquecê-lo de acordo com a temperatura de referência adotada. Dessa forma, neste ambiente de teste, foram utilizados somente dois ESP32 conectados aos DHT22 e outro microcontrolador responsável por desempenhar o papel de *gateway*.

Este ambiente de teste foi adotado a princípio em decorrência da constante de tempo do processo ser menor, pois o seu tamanho era reduzido. Além disso, o controle da temperatura era mais fácil de ser realizado neste do que na sala de reuniões. A maquete utilizada pode ser observada na figura a seguir.



Figura 2. Maquete do LARA.

Depois de concluída a primeira parte do teste, o objetivo era que fosse feito o uso efetivamente da sala de reuniões do LARA, vista na figura 3, para a construção do ambiente

inteligente, porém esta não pode ser utilizada por conta da pandemia.

A sala mencionada possui divisórias de vidro que a separa dos ambientes vizinhos, o que a torna bastante sensível à temperatura destes. Ademais, há uma janela na sala que possibilita a influência da radiação solar ao longo do dia na temperatura da mesma, sendo sensível termicamente também ao ambiente externo.



Figura 3. Sala de reunião do LARA.

Para que fosse realizada a aquisição de dados e a análise da estrutura desenvolvida para o ambiente inteligente, o experimento final foi realizado na residência de um dos integrantes do grupo de pesquisa. Em decorrência da pandemia, foi utilizada uma maquete constituída de papelão, sem divisórias, na qual foi realizada o controle da temperatura, semelhante ao teste realizado na maquete de madeira.

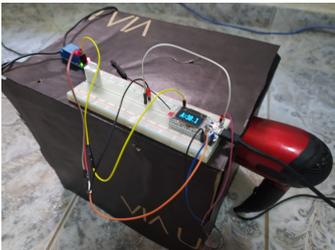


Figura 4. Maquete nova.

Assim, foi utilizado um secador como atuador, de forma a aquecer o ambiente e manter a temperatura constante conforme a referência adotada. Para isso, foram utilizados dois ESP32 conectados aos sensores de temperatura e umidade, de forma a se obter os valores destas grandezas para o ambiente interno e externo à maquete. Ademais, fez-se uso de outro ESP32 para desempenhar o papel de *gateway*.

### 3.2 Controle do ambiente

No caso dos testes realizados tanto na maquete de madeira do LARA quanto na maquete de papelão, foi projetado um controlador liga-desliga para que fosse feito o controle dos atuadores. Para os dois casos, a temperatura de referência adotada foi de 35 °C, assim, utilizou-se um

intervalo simétrico centrado neste valor para controlar o atuador. Dessa forma, utilizou-se o seguinte algoritmo:

- Ligar o atuador se a temperatura for menor que 34,9 °C;
- Desligar o atuador se a temperatura for maior que 35,1 °C.

Além disso, para se avaliar o processo em diferentes valores de temperatura, alterou-se a temperatura de referência para 50 °C ao longo do experimento, mantendo o mesmo intervalo simétrico centrado neste valor. Depois, retornou-se a referência original para que fosse observado o resfriamento.

Para o caso da sala de reuniões, a temperatura de referência adotada foi de 23 °C. Sabendo disso, o algoritmo adotado é dado por:

- Ligar o atuador se a temperatura for maior que 23,3 °C;
- Desligar o atuador se a temperatura for maior que 22,7 °C.

A programação do controlador liga-desliga foi feita utilizando-se as linguagens *Ladder* e *Structured Text*, duas das linguagens padrões de CLPs segundo a norma IEC 61131-3.

### 3.3 Comunicação entre os dispositivos

A comunicação entre os dispositivos é um ponto fundamental para a construção do ambiente inteligente. Sabendo disso, utilizou-se a tecnologia *LoRa* para conectar os microcontroladores que desempenham os papéis de *gateway* e de nós da comunicação, dispensando a necessidade de se utilizar fios para este propósito.

Como *LoRa* utiliza radiofrequência, há uma limitação com relação às frequências que podem ser adotadas para a comunicação. No Brasil, as faixas permitidas pela Anatel são de 902 a 907,5 MHz e de 915 a 928 MHz. Tendo isto em vista, escolheram-se as frequências de 915, 920, 923 e 926 MHz para a aplicação no ambiente de teste.

Apesar de se utilizar a tecnologia *LoRa*, o protocolo *LoRaWAN* não foi utilizado, pois este requer o uso de *gateways* específicos. Sabendo disso, foi criado um protocolo próprio para a aplicação desenvolvida. Neste protocolo, adotou-se a topologia de rede em estrela, de forma que os nós só podem enviar informação ao *gateway*.

Dessa forma, para que se inicie a comunicação, o nó deve solicitar uma permissão para o envio dos dados ao *gateway* (*request to send*). Assim, caso este não esteja se comunicando com outro dispositivo, envia ao nó a autorização (*clear to send*) e é iniciada a troca de informações.

Em seguida, o nó encaminha o pacote contendo os dados (*payload*) e, caso o envio tenha sido bem sucedido, o *gateway* responde com uma confirmação de que os recebeu (*acknowledge*). Além disso, como o protocolo foi estruturado para que os dados sejam enviados somente dos nós para o *gateway*, aproveitou-se o *acknowledge* para que o *gateway* pudesse realizar o envio dos dados obtidos na nuvem aos demais. No diagrama a seguir, podem ser observados os detalhes da comunicação:



Figura 5. Diagrama da comunicação.

Outro ponto importante em relação ao protocolo é o fato de que, para impedir interferências no momento em que a troca de informações está ocorrendo, é realizada uma mudança na frequência de comunicação.

O *request to send* e o *clear to send* são enviados pelos dispositivos na frequência de 915 MHz, porém para o encaminhamento do *payload*, o *gateway* e o nó que recebeu a permissão de envio trocam a sua frequência de comunicação para 920, 923 ou 926 MHz, a depender do nó que está se comunicando. Após o envio e recebimento do *acknowledge*, ambos retornam a frequência original.

No protocolo estabelecido, estão presentes dois tipos de pacote de dados (*payload*), um que é enviado do nó que controla o ambiente analisado ao *gateway* e o outro que é enviado pelos demais nós. Tais pacotes podem ser observados nas figuras a seguir.

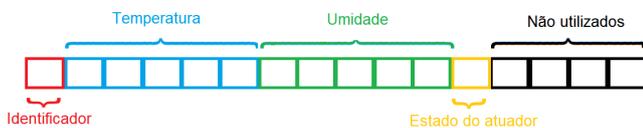


Figura 6. Pacote de dados enviado pelo nó interno.

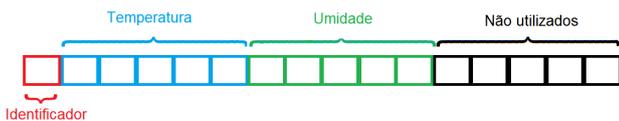


Figura 7. Pacote de dados enviado pelos nós externos.

Como pode ser visto nas imagens, os pacotes que compõem o protocolo criado contém os seguintes tipos de dados:

- **Identificador:** indica de qual dispositivo o dado foi enviado;
- **Temperatura e umidade:** dados adquiridos dos ambientes;
- **Estado do atuador:** indica se o atuador está ligado ou desligado;

Estes pacotes possuem 16 bytes de tamanho. Mesmo que nem todos estes bytes sejam utilizados, tal extensão foi adotada em decorrência do tipo de criptografia escolhida

para garantir a segurança dos dados. Dessa forma, para certificar que os dados enviados estariam seguros, adotou-se a AES-128, uma criptografia de blocos com chave simétrica que faz o uso de uma chave privada para realizar tanto a encriptação quanto a descriptação dos dados.

### 3.4 Nuvem

Na implementação da nuvem no sistema construído, utilizou-se como base a ferramenta de programação *Node-RED* hospedada no servidor da *IBM Cloud*. Tal ferramenta permitiu a criação de um supervisorio para o ambiente com a utilização de interfaces gráficas que possibilitam ao usuário observar tanto as informações das grandezas que estão sendo controladas, com a presença de gráficos que registram o comportamento do sistema, quanto o estado dos atuadores, se estão ligados ou desligados. Além do mais, possibilita a escolha manual da temperatura de referência do ambiente.

Todas estas funções foram elaboradas tendo como base o protocolo *MQTT* que possibilita ao microcontrolador que desempenha o papel de *gateway* a comunicação com o servidor do *Node-RED* através da inscrição e publicação em tópicos pré-definidos.

Além disso, o *Node-RED* também possibilitou o envio dos dados recebidos pelo *MQTT* e mostrados em sua interface gráfica diretamente para o *Cloudant*, um banco de dados disponibilizado pela IBM, além de permitir a conexão com outros serviços como, por exemplo, o *OpenWeatherMap*, um serviço que por meio de uma API disponibiliza informações meteorológicas, tais como, velocidade do vento, pressão, umidade, previsão do tempo, dentre outras.

Dessa forma, foi possível se anexar ao supervisorio desenvolvido uma interface que disponibiliza as informações meteorológicas, um recurso muito importante, visto que tais dados podem ser utilizados para realizar um controle ainda mais efetivo no ambiente, pois cria a possibilidade de se realizar um controle preditivo tendo como referência a previsão do tempo.

A interface gráfica do supervisorio desenvolvida com o *Node-RED* pode ser vista nas figuras a seguir:



Figura 8. Interface do *Node-RED*.

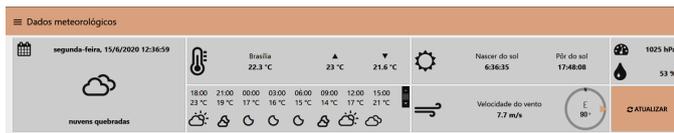


Figura 9. Dados meteorológicos do *Node-RED*.

#### 4. RESULTADOS

Com o intuito de se avaliar o ambiente inteligente projetado, foram registrados os dados de temperatura do local durante um certo período. Estes podem ser observados no gráfico a seguir:

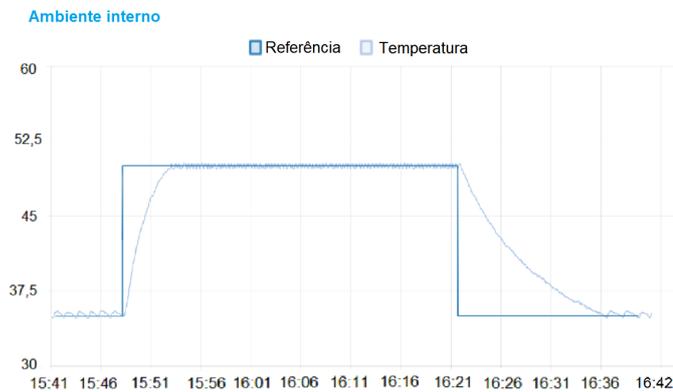


Figura 10. Curva de temperatura obtida no experimento.

Ao se observar o gráfico, nota-se que inicialmente a curva de temperatura fica oscilando em torno da referência inicial de 35 °C. Essa oscilação é decorrente do fato de que a temperatura aumenta para chegar à referência, porém ela ultrapassa um pouco, visto que mesmo o secador desligando, o ambiente tende a esquentar um pouco mais até que a temperatura volte a cair. Depois, quando a temperatura cai e fica um pouco abaixo do valor de referência, o atuador é ligado e a curva volta a subir. Isso se mantém de forma que a temperatura média seja igual a temperatura de referência desejada.

Ademais, depois de um tempo, a referência de temperatura é alterada para 50 °C e a curva cresce até atingir tal valor. Ao alcançar 50 °C, observa-se que a curva de temperatura volta a oscilar pelo mesmo motivo citado anteriormente, porém a frequência de oscilação aumenta. Isso se deve ao fato de que, em temperaturas mais elevadas, o ambiente tende a resfriar mais rápido e o atuador necessita de ser ligado mais vezes para manter a temperatura média próxima da referência.

Por fim, a referência é alterada mais uma vez para 35 °C, sendo observado um decaimento exponencial até que a curva atingisse o novo valor de referência.

Além do controle implementado, também é importante se analisar o uso do controlador lógico programável de código aberto no projeto. O CLP é um dispositivo que se destaca no âmbito da automação industrial por ser um dispositivo robusto, possuindo imunidade a ruídos eletromagnéticos, e por possibilitar uma programação facilitada.

No desenvolvimento do ambiente inteligente foi utilizado um CLP de código aberto de forma a se avaliar a inserção de um CLP no contexto da automação predial. Dessa forma, é perdida a robustez deste dispositivo, tendo em vista que é utilizado um microcontrolador para desempenhar tal função. Por outro lado, com a utilização do *OpenPLC* é possível utilizar microcontroladores que são mais baratos do que os controladores lógico programáveis disponíveis no mercado.

Para exemplificar, na sala de reuniões do LARA, é utilizado o CLP *MT-531 Ri Plus* da *Full Gauge* para controle da temperatura e da umidade do local que custa aproximadamente R\$ 700,00. Por outro lado, o microcontrolador utilizado para desempenhar o papel de CLP no ambiente inteligente custa cerca de R\$ 140,00, ou seja, uma economia de 80%.

Ademais, com uso do *OpenPLC*, a facilidade de programação é mantida, sendo possível a programação dos dispositivos com as linguagens padrões de CLP. Nas figuras a seguir, pode-se observar a programação do microcontrolador responsável pelo controle do ambiente interno da maquete em linguagem *Ladder*:

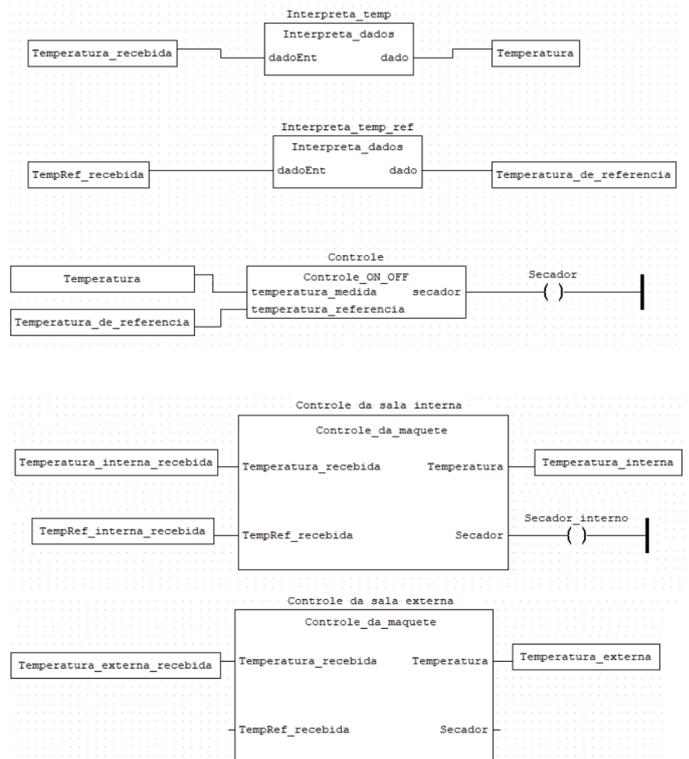


Figura 11. Programação em linguagem Ladder.

Como a linguagem *Ladder* foi desenvolvida com o intuito de possuir familiaridade com comandos elétricos, sendo uma linguagem de programação visual, torna-se mais fácil e acessível a reconfiguração do dispositivo de controle do ambiente.

Por outro lado, no desenvolvimento do ambiente inteligente foi observada uma limitação ao se utilizar o *OpenPLC*. Tal limitação consiste na dificuldade de se implementar controladores mais sofisticados, como um controlador

antecipativo ou preditivo. Sabendo disso, para esse tipo de controle, existem outras alternativas para a automação predial mais eficientes como, por exemplo, a programação do microcontrolador integrado com um sistema operacional, o que permite o desenvolvimento de projetos mais complexos.

## 5. CONCLUSÕES

Por meio do teste desenvolvido para analisar o ambiente inteligente implementado, foi possível se observar que o controle do local foi bem sucedido, de forma que a temperatura média do ambiente correspondesse à temperatura de referência desejada.

Ademais, ao se analisar a inserção do CLP de código aberto no projeto, nota-se que isso permitiu o uso de um dispositivo mais barato e facilmente configurável. Entretanto, durante o desenvolvimento da pesquisa, evidenciou-se uma certa dificuldade em implementar controles mais sofisticados por meio do uso desta ferramenta.

Por outro lado, ao se avaliar a comunicação dos dispositivos com o uso da tecnologia *LoRa*, evidenciou-se uma comunicação bem sucedida que dispensa a necessidade de conexões físicas entre os dispositivos.

Já ao se analisar a conexão com a nuvem e o supervisor criado com o *Node-RED*, foi observada a possibilidade de se melhorar ainda mais o controle do ambiente inteligente com o uso de informações advindas da internet, implementando-se controladores mais precisos e de maior complexidade do que o controlador liga-desliga.

## REFERÊNCIAS

- Alves, T. e Morris, T. (2018). OpenPLC: An IEC 61,131–3 compliant open source industrial controller for cyber security research. *Computers and Security*, 78, 364–379.
- Alves, T.R., Buratto, M., De Souza, F.M., e Rodrigues, T.V. (2014). OpenPLC: An open source alternative to automation. *Proceedings of the 4th IEEE Global Humanitarian Technology Conference, GHTC 2014*, 585–589.
- Carli, R., Cavone, G., Othman, S.B., e Dotoli, M. (2020). IoT based architecture for model predictive control of HVAC systems in smart buildings. *Sensors (Switzerland)*, 20(3), 1–18.
- Che, W.W., Tso, C.Y., Sun, L., Ip, D.Y., Lee, H., Chao, C.Y., e Lau, A.K. (2019). Energy consumption, indoor thermal comfort and air quality in a commercial office with retrofitted heat, ventilation and air conditioning (HVAC) system. *Energy and Buildings*, 201, 202–215.
- Ferrández-Pastor, F.J., Mora, H., Jimeno-Morenilla, A., e Volckaert, B. (2018). Deployment of IoT edge and fog computing technologies to develop smart building services. *Sustainability (Switzerland)*, 10(11), 1–23.
- Ferreira, I.V., Bigheti, J.A., e Godoy, E.P. (2019). Development of a Wireless Gateway for Industrial Internet of Things Applications. *IEEE Latin America Transactions*, 17(10), 1637–1644.
- Hanif, M., Mohammad, N., e Harun, B. (2019). An Effective Combination of Microcontroller and PLC for Home Automation System. *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019*, 2019(Icasert), 19–24.
- Lu, C.Y., Chen, F.H., Hsu, W.C., Yang, Y.Q., e Su, T.J. (2020). Constructing home monitoring system with Node-RED. *Sensors and Materials*, 32(5), 1701–1710.
- Misal, S.R. (2020). MQTT and Bluetooth. (Iccmc), 79–82.
- Moreno, M.V., Úbeda, B., Skarmeta, A.F., e Zamora, M.A. (2014). How can we tackle energy efficiency in iot based smart buildings? *Sensors (Switzerland)*, 14(6), 9582–9614.
- Perković, T., Šolić, P., Zargariasl, H., Čoko, D., e Rodrigues, J.J. (2020). Smart Parking Sensors: State of the Art and Performance Evaluation. *Journal of Cleaner Production*, 262.
- Rinaldi, S., Bellagente, P., Camillo Ciribini, A.L., Chiara Tagliabue, L., Poli, T., Giovanni Mainini, A., Speroni, A., Blanco Cadena, J.D., e Lupica Spagnolo, S. (2020). A cognitive-driven building renovation for improving energy efficiency: The experience of the elisir project. *Electronics (Switzerland)*, 9(4).
- Teizer, J., Neve, H., Li, H., Wandahl, S., König, J., Ochner, B., König, M., e Lerche, J. (2020). Construction resource efficiency improvement by Long Range Wide Area Network tracking and monitoring. *Automation in Construction*, 116(June 2019).
- Vieira, G., Barbosa, J., Leitao, P., e Sakurada, L. (2020). Low-cost industrial controller based on the raspberry pi platform. *Proceedings of the IEEE International Conference on Industrial Technology, 2020-February*, 292–297.
- Zhang, X., Pipattanasomporn, M., Chen, T., e Rahman, S. (2020). An IoT-Based Thermal Model Learning Framework for Smart Buildings. *IEEE Internet of Things Journal*, 7(1), 518–527.