Desenvolvimento de um *Gateway OPC* para aplicações de Automação no Sistemas Elétrico de Potência

Gabriel Teixeira Zanchin [*] **Samuel Lessinger** [**]
Rodrigo Marques de Figueiredo [***] **Lúcio Renê Prade** [****]
Alzenira da Rosa Abaide [†]

[*] *Faculdade de Engenharia Elétrica, Universidade do Vale do Rio dos Sinos, RS, (e-mail: gabrielzanchin@hotmail.com)*
[**] *Faculdade de Engenharia Elétrica, Universidade do Vale do Rio dos Sinos, RS, (e-mail: samulessinger@unisinos.br)*
[***] *Faculdade de Engenharia de Controle & Automação, Universidade do Vale do Rio dos Sinos, RS (e-mail: marquesf@unisinos.br)*
[****] *Faculdade de Engenharia Elétrica, Universidade do Vale do Rio dos Sinos, RS, (e-mail: luciorp@unisinos.br)*
[†] *CEESP, Universidade Federal de Santa Maria, RS, (e-mail: alzenira@ufsm.br)*

Abstract: The advancement of communication network technologies in recent years has enabled the emergence of a large number of protocols and equipment that require the use of intermediary hardware to integrate the nodes of a plant or automation system. The need to adapt networks to the modern scenario, perform the interoperability of these networks, and make them communicate in a transparent way is a great challenge. With the establishment of the Industrial Internet and Internet of Things, it is possible to implement solutions to solve this problem. In this context, this work was to develop an *OPC Gateway* for automation, capable of reading and writing values from a plant or automation systems and making them available for user access through client software. The hardware operation will be described based on the message exchange tests carried out at the end of the research, demonstrating its efficiency and usefulness.

Resumo: O avanço das tecnologias de redes de comunicação nos últimos anos possibilitou o surgimento de um grande número de protocolos e equipamentos que demandam o uso de hardwares intermediários para integração dos nós de uma planta ou sistema de automação. A necessidade de adequar redes ao cenário moderno, realizar a interoperabilidade dessas redes, e faze-las comunicarem-se de maneira transparente é um grande desafio. Com o estabelecimento da Internet Industrial e Internet das Coisas é possível implementar soluções que se proponham a solucionar esse problema. Nesse contexto o objetivo desse trabalho foi desenvolver um *Gateway OPC* para automação, capaz de realizar a leitura e escrita de valores de uma planta ou sistemas de automação e disponibilizá-los para acesso de usuários através de um software cliente. O funcionamento do hardware será descrito com base nos testes de troca de mensagens realizados ao término da pesquisa, demonstrando sua eficiência e utilidade.

Keywords: Automation; *Gateway*; Networks; Protocols; Electric Power System
Palavras-chaves: Automação; *Gateway*; Redes; Protocolos; Sistema Elétrico de Potência

1. INTRODUÇÃO

A comunicação entre elementos nas redes de automação é um tópico da Engenharia que vem se desenvolvendo nos últimos anos, apresentando constantemente inovações e aplicações. As diferentes necessidades decorrentes de cada contexto demandaram que as redes de comunicação desenvolvessem uma grande variedade de tecnologias e padrões, de maneira a atender setores distintos da sociedade, como nas telecomunicações, em redes industriais e na automação dos sistemas elétricos de potência. (Group, 2020), (AUTOMATION, 2020).

Dentro do contexto de automação dos sistemas elétricos de potência e automação na indústria, o grande número de equipamentos e dispositivos, de diferentes fabricantes e operando sob diferentes protocolos, demandam a utilização de dispositivos intermediários que assegurem a comunicação correta e segura entre os diversos pontos de um sistema de automação, sendo estes dispositivos conhecidos como *Gateways* (Keller, 2016).

Esse trabalho aborda o desenvolvimento de um *Gateway* para estabelecer a comunicação no cenário de automação, tanto industrial como em redes de distribuição.

O objetivo geral é o desenvolvimento de um hardware que funcione como *Gateway OPC UA (Open Platform Communications Unified Architecture*, ou Plataforma Aberta de Comunicação com Arquitetura Unificada) para gerenciar o acesso a um sistema de automação, de modo que o recurso possa ser utilizado por um número maior de usuários e não fique limitado ao uso através do software proprietário do fabricante.

Inicialmente, o problema fora delimitado no acesso via computador pessoal de valores registrados em controladores *OPC* e dispositivos conectados fisicamente ao *Gateway*. No decorrer do trabalho são também apresentados métodos para escrita de parâmetros utilizando a arquitetura proposta, tornando o escopo de utilização mais abrangente.

Após essa introdução, a sessão de contextualização indicará trabalhos com aspectos próximos aos desenvolvidos e conteúdos teóricos necessários ao entendimento. Após, são apresentados os tópicos de metodologia e implementação, resultados e conclusões.

2. CONTEXTUALIZAÇÃO

Um protocolo de comunicação é um conjunto de regras estabelecido para que possa existir a troca de informações entre dispositivos. Esse conjunto de regras pode apresentar variações de um protocolo para outro, no entanto em geral estabelece três conceitos fundamentais, que são sintaxe, semântica e *timing*. A sintaxe diz respeito ao formato dos dados que serão trocados (quantidade de bits), e que costuma ser exemplificada através de um *dataframe*. A semântica trata do significado dos bits, realizando uma interpretação de cada trecho – leitura, escrita, etc. Por fim, *timing* se refere a quando os bits devem ser enviados ou recebidos, e também com que velocidade se dará a conexão (Forouzan, 2010).

Os protocolos de comunicação podem ser classificados como abertos ou fechados. São considerados protocolos abertos aqueles que podem ser utilizados por qualquer usuário ou fabricante de *Hardware/Software* na implementação de seu produto. Esses protocolos são largamente divulgados em meios públicos e possuem suas normas técnicas acessíveis. São considerados protocolos fechados aqueles que não são disponibilizados ao usuário geral e nem a outros fabricantes de hardware que não sejam o proprietário (Forouzan, 2010).

O modelo *OSI (Open Systems Interconnection)* é um modelo de arquitetura de sistemas de comunicação criado com o objetivo de facilitar as regras de padronização para criação de redes industriais, uma vez que a vasta quantidade de equipamentos existentes no mercado dificultava a comunicação entre dispositivos de fabricantes diferentes em uma mesma rede. Esse modelo é organizado em uma divisão de sete camadas distintas porém relacionadas entre si, conforme exibido na Figura 1 (Forouzan, 2010).

O protocolo *OPC*, (*OLE for Process Control*) foi criado em 1996 por um grupo de companhias com o objetivo de resolver os problemas relacionados a intercomunicação de dispositivos instalados em níveis diferentes de uma rede de automação.

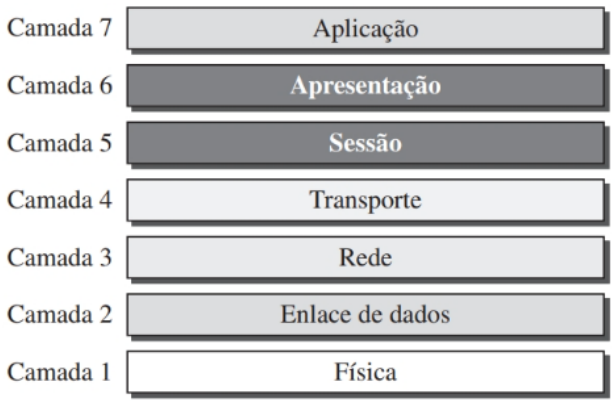

Figura 1. Modelo *OSI*. Adaptado de (Forouzan, 2010).

À época, e de acordo com os requisitos dentro do contexto de aplicações industriais, foram desenvolvidas três principais especificações para o protocolo: Acesso de Dados *(Data Access, ou "DA")*, Alarme & Eventos ("A&E") e Acesso a Histórico de Dados *(Historical Data Access, ou "HDA")* (Mahnke et al., 2009).

Com a popularização do *OPC DA* clássico e do surgimento de novas demandas, o protocolo sofreu alterações para tornar-se um padrão na indústria, removendo a necessidade de utilizar tecnologias proprietárias e transformando-se em uma arquitetura unificada com a descrição *UA (Unified Architecture)*.

Atualmente, o *OPC UA (Open Platform Communications Unified Architecture)* é um protocolo de comunicação no qual diversos tipos de dispositivos e sistemas podem ser configurados para que se comuniquem, realizando troca de mensagens no formato cliente/servidor.

A arquitetura *OPC UA* define a existência de máquinas Servidores e Clientes trabalhando em parceria, com cada sistema podendo receber diversos Servidores e Clientes que interagem concomitantemente. O Servidor *OPC UA* modela o *endpoint* das interações entre cliente e servidor (FOUNDATION, 2021).

Na arquitetura, os Objetos reais *(Real Objects)* representam objetos físicos (por exemplo, um dispositivo de campo) ou de software que podem ser consultados pela aplicação do Servidor. Esses objetos são organizados dentro de uma estrutura conhecida como *AddressSpace*, que por sua vez é modelada como um conjunto de Nós *(Nodes)*; cada nó representa um objeto real, que pode ser acessado pelos Clientes (FOUNDATION, 2021).

A função principal do *AddressSpace* é a de padronizar a forma com que servidores representam os Objetos para acesso dos clientes. A Figura 2 ilustra o modelo de Objeto utilizado pelo protocolo, que é descrito na forma de Variáveis e Métodos. As Variáveis representam valores armazenados atribuídos ao Objeto, que podem ser utilizadas para leitura e escrita, enquanto os Métodos são funções que podem retornar ao cliente um resultado. Esse modelo permite a referência entre Objetos, o que faz com que o Nós do *AddressSpace* possam ser organizados em hierarquias (FOUNDATION, 2021).

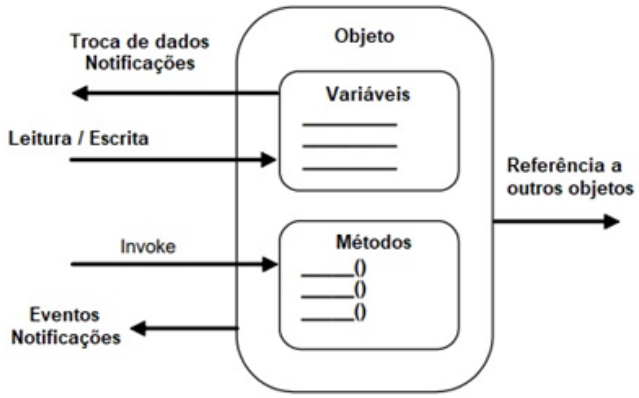

Figura 2. Modelo de Objeto *OPC UA*. Adaptado de (FOUNDATION, 2021).

O conjunto de Nós são organizados no *AddressSpace* conforme o esquema da Figura 3, através de Atributos e Referências. Os Atributos fornecem um nome, uma descrição e o tipo de dado que o Nó carrega, que são fornecidos no momento em que o Nó é instanciado no *AddressSpace* (FOUNDATION, 2021).

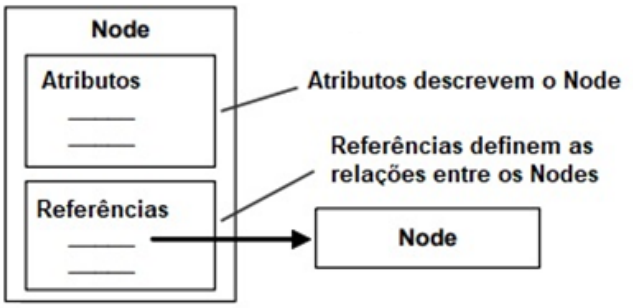

Figura 3. Modelo de Nó *OPC UA*. Adaptado de (FOUNDATION, 2021).

As Referências são definidas pelo Servidor, e cada uma é instanciada através de um *ReferenceType Node*. O Nó que contém a Referência é conhecido como *SourceNode*, enquanto o Nó que é referenciado é conhecido como *TargetNode* – que pode estar localizado tanto no mesmo *AddressSpace* quanto no de outro servidor *OPC UA* (FOUNDATION, 2021).

As principais vantagens na utilização de uma interface *OPC* são: autonomia dos fabricantes na manufatura de equipamentos e desenvolvimento de softwares, facilidade para configuração do tipo de dados trocados, escalabilidade, orientação a objetos e permissão para acesso simultâneo as informações guardadas pelo Servidor *OPC* (AUTOMATION, 2020).

O *Gateway* é um dispositivo responsável por realizar a conexão entre duas ou mais redes de comunicação e que atua para que não ocorram problemas de comunicação ocasionados por arquiteturas, padrões ou protocolos de comunicação diferentes. Ele é o agente que garante a interoperabilidade dos elementos de rede conectados entre si (Group, 2020).

No caso de um *Gateway OPC UA*, sua função é a de coletar dados de algum dispositivo de controle compatível com o protocolo, e instanciá-los no *AddressSpace* do Servidor, permitindo que outros Clientes realizem a leitura e escrita dos valores de variáveis através dos serviços *OPC UA*.

Conforme Ausberger (2019), o *Gateway* precisa ter implementado em sua aplicação funções que realizem tarefas de comunicação (tais como conexão, reconexão, tratamento de erros, etc.), bem como procura de estruturas de dados e verificação da integridade dessas estruturas.

A operação do *Gateway* ocorre de maneira que, iniciada a conexão com a estrutura de dados que se tem interesse, a aplicação passa a atualizar com o dispositivo alvo de forma sincronizada. Ao receber a requisição de um Cliente, o *Gateway* registra a requisição e a coloca em uma pilha de sincronizações, que resolvida irá retornar ao Cliente seu pedido (Ausberger, 2019).

O espectro de funções que um *Gateway* pode suportar dentro da automação industrial vem aumentando, junto do estabelecimento de novas tecnologias voltadas para Internet Industrial *(IIoT)*. O conceito moderno compreende dispositivos que realizam interface de instrumentos que se encontram na base da pirâmide de automação com o armazenamento dos dados em nuvem. Esses produtos são capazes de ir além de sua função de traduzir protocolos e oferecem aplicações completas, tais como análise em tempo real de dados unificados em um servidor rodando em nuvem (Group, 2020), (Keller, 2016).

Atualmente, a evolução da tecnologia em redes Industriais e *Wireless* aponta a tendência de se trabalhar com sistemas que utilizam da tecnologia sem fio na comunicação entre dispositivos eletrônicos inteligentes. As vantagens para isso passam desde a uma redução nos custos de cabeamento, maior confiabilidade na troca de mensagens, até a robustez necessária para que opere em conjunto com outros dispositivos *IoT*, com aplicação em diversos novos setores, como os sistemas de distribuição elétrica inteligentes.

No uso aplicado em *IoT*, os *Smart Gateways* podem agir para suprir o problema que surge ao se trabalhar com tantos dispositivos heterogêneos (muitos de baixo custo), que trocam dados em vários formatos. Um dos objetivos em *IoT* é criar casas inteligentes para que coletivamente formem cidades inteligentes; isso somente será possível se todos os nós dentro da rede conseguirem decodificar as mensagens trocadas de maneira correta (Mastilak et al., 2018).

O funcionamento de um hardware desse tipo é proposto por Mastilak et al. (2018), um *Smart Gateway* voltado para comunicação de dispositivos de baixo custo. Uma aplicação web fornece interface para comunicação com dispositivos externos, informações de instrumentos de campo armazenados em um banco de dados. O formato de arquivo escolhido para empacotamento pelos autores foi o *JSON*, mas este conceito pode ser sofrer alterações em outros ambientes.

A aplicação que controla os dispositivos de campo é composta por seis módulos: encriptação, autenticação, retransmissão (que faz o tratamento de eventuais pacotes perdidos), controlador de interface de rede, planejamento e coleta de dados/formatação. O módulo de autenticação verifica se os dispositivos disponíveis na conexão estão

aptos a criarem sessão com o *Smart Gateway*; em caso afirmativo, este dá o sinal e permite que o equipamento acesse com uma senha. Uma etapa adicional de troca *hash* é efetuada (Mastilak et al., 2018).

Outro trabalho que apresenta proposta de um *Gateway* para uso com aplicação no cenário de *IoT* é feito por Keller (2016). O trabalho aborda o desenvolvimento de um *middleware* para integração de redes de automação distribuídas que operam sob o protocolo Modbus *RTU*, utilizando o protocolo de *IoT MQTT (Message Queing Telemetry Transport)*. Esse estudo levou em consideração o sistema de geração de energia solar fotovoltaica, geograficamente espalhado numa região, para centralizar o monitoramento numa única planta de controle virtual (Keller, 2016).

Atualmente, as redes elétricas inteligentes *Smart Grids* em colaboração com a geração de energia elétrica distribuída, consistem em um tópico de ponta nas discussões acerca do planejamento de um ambiente mais sustentável. As vantagens para o uso vão desde a economia em transmissão e perdas reduzidas a diversidade na matriz energética da planta. Automação de subestações remotas também consiste em outro tópico de grande interesse (Keller, 2016).

3. METODOLOGIA & IMPLEMENTAÇÃO

A *Raspberry Pi*, exibida na Figura 4, consiste em um computador compacto, com tamanho próximo de um cartão de crédito, desenvolvido pelo britânico Eben Upton, inicialmente com o objetivo de ser utilizado no aprendizado de crianças dentro do ambiente de programação e robótica. Os primeiros protótipos da *Raspberry* se tratavam de placas com capacidade limitada, de modo que Upton e demais colegas fundaram a *Raspberry Pi Foundation*, e em meados de 2012 lançaram o primeiro modelo da placa com processamento semelhante aos conhecidos hoje (Upton, 2017).

Figura 4. Raspberry Pi 3. Adaptado de (Upton, 2017).

O modelo escolhido para ser utilizado é a *Raspberry Pi 3* Modelo B, que roda diversas distribuições Linux e compatível com Windows 10, além de outros sistemas operacionais desenvolvidos especialmente para plataformas de *IoT*.

O processador embutido na placa é um Broadcom BCM2837 arquitetura 64 bits, com memória RAM de 1 GB. Ela conta com portas *USB* que permitem a conexão com dispositivos genéricos, saída *HDMI* para monitores, cartão de memória *SD* e alimentação via cabo micro *USB*. Conta com conector de entradas e saídas de 40 pinos, podendo ser expandida com o uso de diversos módulos extras, rádio *Wireless*, *Wi-Fi* e *Bluetooth*, tornando uma alternativa conveniente em aplicações sem fio (Upton, 2017).

O *Gateway* permitirá o acesso aos dados de simulação de uma planta através dos computadores que estiverem conectados na rede local.

Para a fase inicial de projeto, foram definidos alguns *Hardwares* e ferramentas de *Software* que são utilizados na solução do problema. A Figura 5 apresenta uma ilustração da rede que se busca implementar, definindo a *Raspberry Pi 3* como plataforma utilizada com vista nas facilidades de implementação expostas anteriormente.

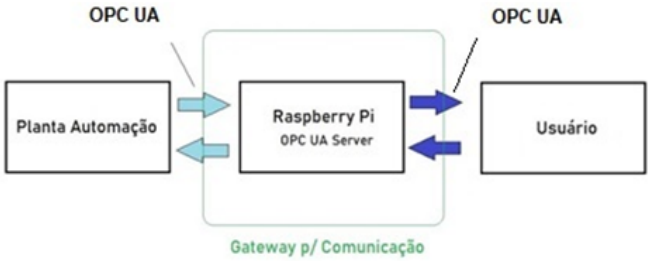

Figura 5. Diagrama de blocos.

Um Servidor *OPC UA* é composto por três elementos principais: Objetos, Aplicação Servidor e o Espaço de Endereçamento. Os objetos dizem respeito a objetos físicos que podem ser consultados pela aplicação Servidor; exemplos de objetos são os próprios dispositivos terminais de redes *fieldbus*, como um *timer*. A aplicação é o software que roda a ferramenta Servidor *OPC*, uma interface para atuar em conjunto das saídas *OPC*. O espaço de endereçamento guarda as informações de quais pontos da rede são acessíveis pelo Servidor e quais referências uns possuem com os outros (FOUNDATION, 2021).

O escopo deste trabalho está limitado no desenvolvimento da aplicação Servidor. Isso se dará através da implementação de um código *Python* que irá ser executado na *Raspberry Pi* 3. Para simular o Servidor original e o Cliente *OPC UA* serão utilizados softwares disponibilizados gratuitamente.

O *FreeOPCUA* consiste em um projeto de código aberto que foi desenvolvido para prover aos usuários de *Python* uma série de recursos na implementação de servidores e clientes *OPC UA*. A instalação da biblioteca na *Raspberry* é feita com o pacote instalador *pip*, pelo Terminal do Linux com o comando *"sudo pip 3 install freeopcua"*. Por dependência de outras bibliotecas, é necessário fazer antes a instalação do pacote *cryptography*. O algoritmo que é executado na *Raspberry Pi* 3 foi escrito em *Python* e é exibido na Figura 6.

O acesso aos dados pelo usuário é realizado através do software *UaExpert*. Ele é um programa gratuito disponível para *Windows* e *Linux* e que foi projetado para ser um cliente teste em redes *OPC UA*. Possui recursos de acesso a dados em tempo real ou armazenados em histórico, alarmes e condições, bem como funções de invocar métodos.

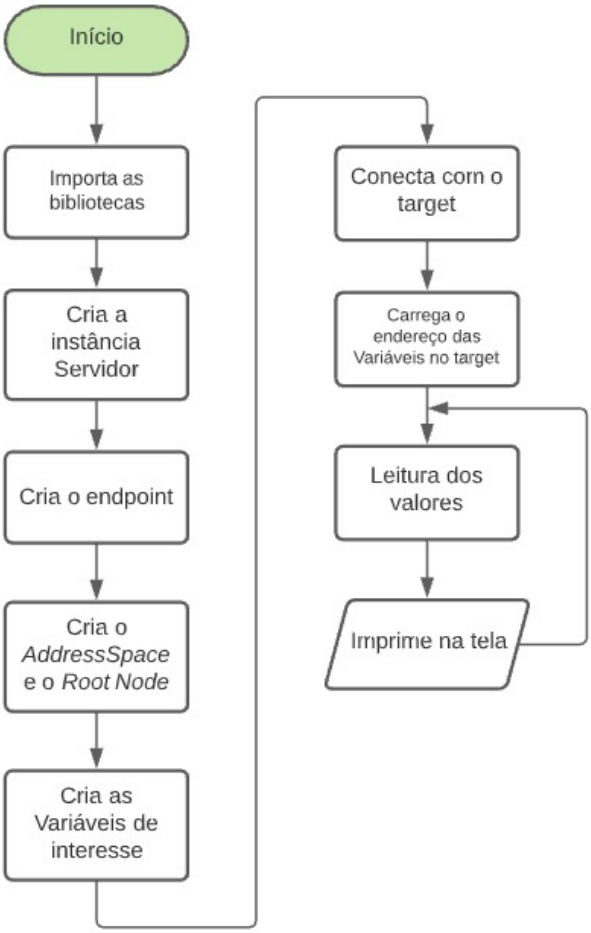

Figura 6. Fluxograma do Servidor.

O *Prosys OPC UA Simulation Server* é um servidor gratuito, *Standalone* e multiplataforma, desenvolvido pela finlandesa Prosys, que permite a simulação de estruturas de dados geradas por dispositivos que possuem *OPC UA* integrados, como o *CLP* da planta que se deseja ter acesso. O software permite a configuração de diversos ambientes de simulação, e oferece uma série de sinais predefinidos que podem ser vinculados a objetos e variáveis dentro do contexto *OPC* (Prosys, 2021).

4. RESULTADOS

O algoritmo do programa escrito para realizar a comunicação com o *Target OPC UA* inicia com a importação das bibliotecas *Server* e *Client OPC UA*, e realizando a criação da instância servidor na *Raspberry Pi* 3, disponibilizando um *endpoint* para clientes se conectarem. É feita também a importação da biblioteca *RPi.GPIO*, responsável por habilitar os pinos de entrada e saída do módulo *Raspberry Pi* 3.

Para leitura e armazenamento dos dados dentro do espaço de endereçamento do *Gateway* foi criado um Objeto de nome "Parâmetros", que receberá as variáveis "Contador", *"Random"*, "Senoide" e *"Setpoint"*.

Com o aplicativo *Prosys OPC UA Server*, apresentado na Figura 7, sendo executado na rede, o *Gateway* realizou uma série de leituras dos sinais e valores gerados pelo simulador. Para validação inicial, foi instanciado no *AdressSpace* do *Gateway* um Nó que recebe e armazena os valores de um contador, um gerador de números aleatórios, uma senoide e um valor de referência (*setpoint*).

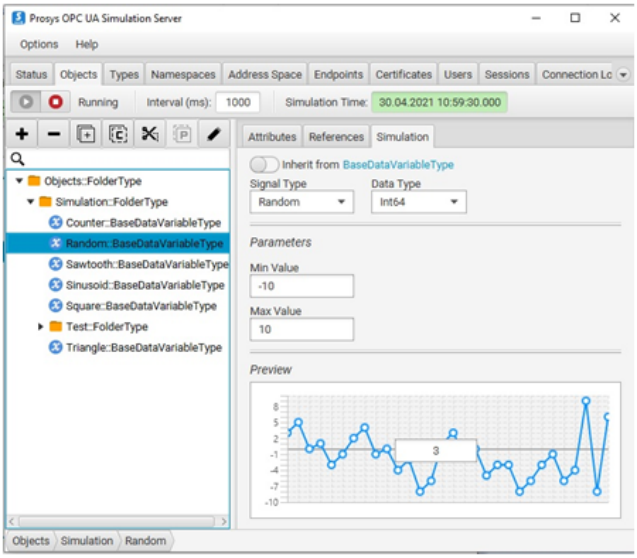

Figura 7. Tela do *Prosys OPC UA Server*, simulando uma planta.

Na Figura 8 apresenta os valores lidos pelo *Gateway* e impressos na tela do Terminal, indicando a capacidade de reconhecer as mensagens do simulador *OPC UA*.

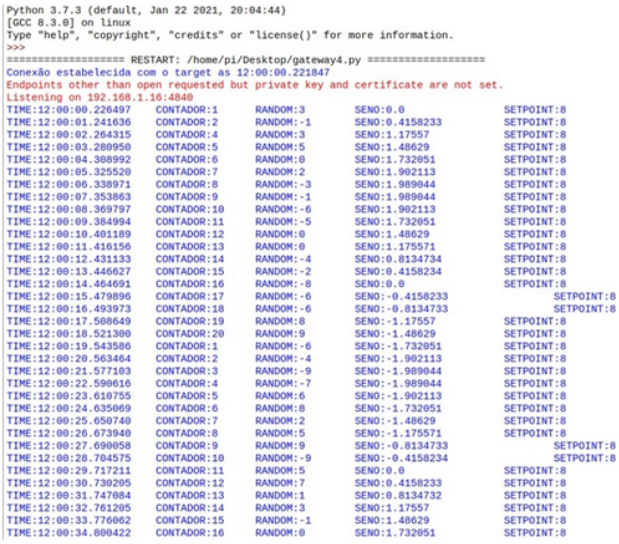

Figura 8. Saída de terminal da *Raspberry Pi 3*, exibindo leituras.

Para o teste de escrita dos clientes no *Gateway*, foi criado no servidor uma variável de controle chamada *"Set Point"*. Essa variável foi utilizada para fazer o acionamento de um *LED* com as *GPIOS* da *Raspberry Pi* 3.

Esse teste foi uma maneira de demonstrar visualmente o procedimento de escrita no Espaço de Endereçamento do *Gateway*, além de também demonstrar o uso de recursos de I/Os oferecidos pela plataforma. Dentro do código da aplicação *Gateway*, foi criada uma condição para que o *LED* acendesse quando o usuário escrevesse o valor 10

na variável de *SetPoint*. O resultado é apresentado na Figura 9. A *Raspberry Pi* 3 é conectada à rede pelo módulo *Wi-Fi* integrado no dispositivo.

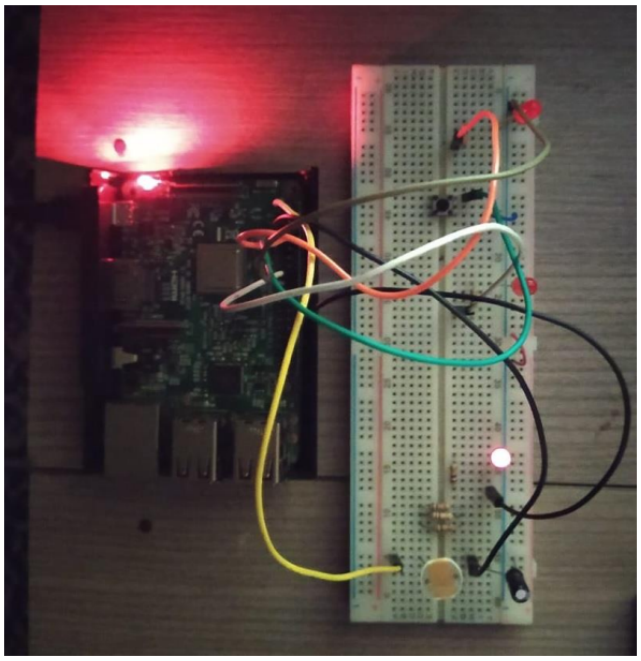

Figura 9. Montagem para confirmar funcionamento.

Para alterar o valor de *SetPoint*, o usuário utiliza o *UaExpert*. Na mesma janela de visualização, é dado um duplo clique em cima do valor que se deseja escrever. O valor é atualizado no *Gateway* ao pressionar Enter no cliente. A Figura 10 registra a troca de valor armazenado sendo feita pelo cliente, fazendo a troca do valor e comandando o acionamento do *LED* anteriormente apresentado.

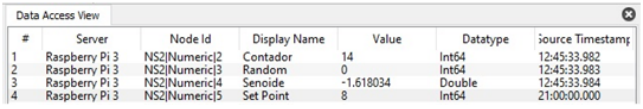

Figura 10. Escrita de valores pelo simulador.

Os resultados obtidos no ambiente de simulação, com softwares que emulam um controlador *OPC UA* transmitindo seus dados numa rede local possibilitaram constatar a viabilidade do projeto do *Gateway* para ser utilizado em aplicações reais. Adicionalmente, os dispositivos conectados as *GPIOs* do *Gateway* exemplificaram funcionalidades que podem ser utilizadas para expandir o leque de aplicações.

5. CONCLUSÃO

O trabalho apresentado tratou do desenvolvimento de um *Gateway* para acesso de dados no âmbito de automação industrial, mais especificamente em plantas que possuam recursos *OPC UA*. O problema foi delimitado na solução primeira de se obter dados de leitura de simuladores e componentes conectados fisicamente ao *Gateway*, emulando dispositivos de campo.

Com o avanço das tecnologias de redes *Fieldbus* cada vez mais voltadas para aplicação em conjunto da Internet 4.0 e de sistemas distribuídos. Isso faz com que a aplicação do *Hardware* abordado nesse trabalho tenha possibilidades de escalar para uso em outros ambientes, inclusive com outros protocolos de comunicação utilizados na indústria e que não foram mencionados.

Neste trabalho, o uso de ferramentas *Freeware* e *Open Source* foi explorado, tornando a metodologia de prototipação e desenvolvimento de grande implementação prática, pois facilita a execução futura do método em centros de operação e planejamento das companhias do setor elétrico.

Futuramente, deseja-se realizar melhorias para o uso do *Gateway* conforme as sugestões: possibilitar a troca de dados via protocolo *DNP3 (Distributed Network Protocol 3)* proporcionando integração com equipamentos presentes em subestações; implementação no código de ferramentas que disponibilizem os dados de leitura via Web, eliminando a restrição da rede local; desenvolver uma aplicação que registre o histórico de dados lidos pelo *Gateway* para acesso posterior.

REFERÊNCIAS

Ausberger, T; Stetina, M. (2019). General methodology for building opc ua gateways. In *16th IFAC Conference on Programmable Devices and Embedded Systems PDES 2019*. IFAC 2019. URL `https://www.sciencedirect.com/science/article/pii/S2405896319326291`.

AUTOMATION, U. (2020). Motivation for the opc ua. URL `http://documentation.unifiedautomation.com/uasdkdotnet2.0.0L2OpcUaMotivation.html`. Acesso em: 15/12/2020.

Forouzan, B. (2010). *Comunicação de Dados e Redes de Computadores*. AMGH.

FOUNDATION, O. (2021). Opc ua specification. URL `https://opcfoundation.org/developer-tools/specifications-unified-architecture`. Acesso em: 10/11/2020.

Group, A.A. (2020). What are industrial network gateways? URL `https://www.arcweb.com/blog/what-industrial-network-gateways`. Acesso em: 26/12/2021.

Keller, A. (2016). *Internet das coisas aplicada à indústria: dispositivo para interoperabilidade de redes industriais*. Master's thesis, Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo. Programa de Pós-Graduação em Engenharia Elétrica - Mestrado.

Mahnke, W., Leitner, S., and Damm, M. (2009). *OPC Linfei Architecture*. Springer.

Mastilak, L., Galinski, M., Kotuliak, I., and Ries, M. (2018). Improved smart gateway in iot. In *16th International Conference on Emerging eLearning Technologies and Applications*. ICETA 2018. URL `https://www.researchgate.net/publication/329645004_Improved_Smart_Gateway_in_IoT`.

Prosys (2021). Opc ua simulation server. URL `https://www.prosysopc.com/products/opc-ua-simulation-server/`. Acesso em: 16/04/2021.

Upton, E.; Halfacree, G. (2017). *Raspberry Pi: Guia do Usuário*. Alta Books.