

Controlador Lógico Programável como um Serviço: Uma Proposta para a Indústria 4.0

Michel M. Fernandes*. Jeferson A. Bigheti**.
Ricardo P. Pontarolli*. Eduardo P. Godoy*

*Universidade Estadual Paulista (Unesp), Sorocaba – SP, Brasil
(michel_m_fernandes@hotmail.com, pasquati@ifsp.edu.br, eduardo.godoy@unesp.br).

**SENAI, Lençóis Paulista - SP, Brasil
(jefersonbigheti@terra.com.br)

Abstract: The industry 4.0 (I4.0) emerges as the evolution of production systems, with the objective of making feasible and standardizing access from several equipment through management and decision-making systems, leading to the development of flexible and adaptable technologies to industry changes. A tendency to promote this interoperability has been the adoption of Service Oriented Architectures (SOA) and Cloud Computing in I4.0. In this context, this paper proposes a new approach for the use of Programmable Logic Controllers (PLC) as a service focusing on control and automation applications in I4.0. This paper presents the development of PLC and Visualization (Dashboard) microservices, which can be executed on different platforms such as computers, embedded systems and in the cloud. A control experiment of an industrial plant using these microservices is discussed and proves the feasibility of the proposal. The methods and technologies used in the development of this paper are presented, as well as the advantages of using this microservice approach for the PLC.

Resumo: A Indústria 4.0 (I4.0) surge como a evolução dos sistemas produtivos, com o objetivo de viabilizar e padronizar o acesso às informações dos diversos equipamentos pelos sistemas de gerenciamento e de tomada de decisões, conduzindo ao desenvolvimento de tecnologias flexíveis e interoperáveis. Uma tendência para promover essa interoperabilidade têm sido a adoção de arquiteturas orientadas a serviços (SOA) e da Computação em Nuvem na I4.0. Diante desse contexto, este trabalho propõe uma nova abordagem para o uso de Controladores Lógico Programáveis (CLP) como um serviço com foco em aplicações de automação e controle na I4.0. Este trabalho apresenta o desenvolvimento dos microserviços CLP e Visualização (Dashboard), os quais podem ser executados em diferentes plataformas como computadores, sistemas embarcados e em nuvem. Um experimento de controle de uma planta industrial usando os microserviços é discutida e comprova a aplicabilidade da proposta. Os métodos e tecnologias utilizados no desenvolvimento deste trabalho são apresentados, bem como demonstradas as vantagens dessa abordagem de uso do CLP.

Keywords: Industry 4.0, Molecular Framework; Service Oriented Architecture (SOA), Cloud Automation

Palavras-chaves: Indústria 4.0, Framework Molecular; Arquitetura Orientada a Serviço; Automação em Nuvem

1. INTRODUÇÃO

A Indústria 4.0 (I4.0) é um conceito que representa uma evolução dos sistemas de produção atuais a partir da convergência entre a automação, tecnologias operacionais (TO) e Tecnologia da Informação (TI). A maioria das tecnologias necessárias para a implementação da I4.0 já existe, o desafio é promover a integração entre essas tecnologias, visando obter uma nova realidade produtiva, onde tudo estará conectado para que sejam tomadas as melhores decisões de produção, custo e segurança, sob demanda e em tempo real (Bigheti *et al.*, 2018).

O desenvolvimento de sistemas industriais por meio da Internet das Coisas Industrial (IIoT - *Industrial Internet of*

Things) e I4.0 enfrenta os desafios de promover a integração entre tecnologias, equipamentos e sistemas de automação alocados em diferentes níveis hierárquicos de sistemas industriais, proporcionando a descentralização, modularidade e independência dos sistemas. Nesse sentido, um paradigma recente tem sido a automação colaborativa através do compartilhamento de serviços, obtendo uma arquitetura flexível, escalável e integrada à rede (Delsing, 2017). A Arquitetura Orientada a Serviços (SOA - *Service Oriented Architecture*) permite o fornecimento de serviços em nuvem alocados em diferentes dispositivos, gateways ou sistemas, facilitando e padronizando as interações entre eles, podendo assim ser considerada um caminho promissor para os desafios da implantação da I4.0 (Bigheti, Fernandes, Godoy, 2019).

A SOA é uma arquitetura de sistemas de TI na qual se busca fragmentar as aplicações convencionais em “Serviços” (Xiao, Wijegunaratne, Qiang, 2017). Na prática, trata-se de uma arquitetura modular, na qual os elementos devem ser compostos de acordo com as necessidades do negócio. Neste tipo de sistema, cada serviço deve ser passível de descoberta e independente quanto à plataforma e à linguagem (Theorin, Hagsund, Johnsson, 2014), com o objetivo de obter baixo acoplamento de um serviço em relação ao outro, escalabilidade e a capacidade de manter uma arquitetura distribuída. Dentro deste contexto surge a ideia de micros serviços, ou seja, serviços com poucas responsabilidades, pequenos e autônomos que podem trabalhar de forma independente ou em conjunto com outros serviços.

Nos últimos anos, diversos trabalhos desenvolveram arquiteturas SOA e avaliaram sua utilização em aplicações industriais no contexto de IIoT e I4.0. Na SOA, cada elemento dos diferentes níveis hierárquicos da arquitetura industrial da ISA-95 não mais se comunica somente com as camadas adjacentes e é somente um fornecedor de dados para a camada superior. Na Fig. 1 é possível notar que há uma migração da arquitetura legada hierárquica para uma arquitetura SOA em nuvem, onde todos os recursos, equipamentos e sistemas são disponibilizados numa mesma infraestrutura em nuvem de forma padronizada e interoperável.

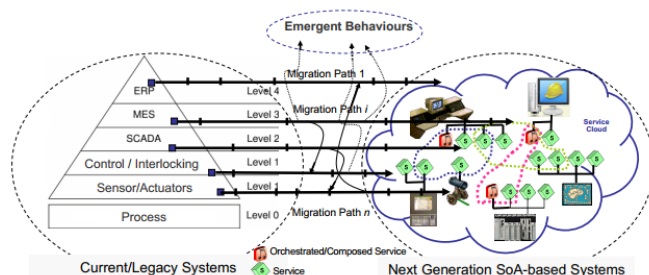


Fig. 1 Arquitetura SOA Aplicada na Automação de um Processo (Fonte: Colombo *et al.*, 2017)

Diversos trabalhos discutem o uso de SOA para aplicações industriais (Jammes *et al.*, 2014), (Blomstedt *et al.*, 2014), (Leitão, Colombo and Karnouskos, 2016), (Espí-Beltrán, Gilart-Iglesias and Ruiz-Fernández, 2017). Espí-Beltrán, Gilart-Iglesias and Ruiz-Fernández (2017) utilizam o padrão REST para desenvolvimento de uma arquitetura SOA industrial. Bigheti *et al.* (2018) apresentam uma Arquitetura Orientada a Microserviços (MOA) em nuvem para aplicações de IIoT e I4.0, discutindo as características, detalhes operacionais, vantagens e potências de aplicação. Bigheti, Fernandes and Godoy (2019) utilizam micros serviços para o controle de processos, discutindo parâmetros de latência e jitter da comunicação.

Ainda que uma grande quantidade de trabalhos relacionados ao desenvolvimento e implementação de SOA para aplicações industriais possa ser encontrado na literatura, o mesmo não acontece quando se foca em Arquiteturas Orientadas a Microserviços (MOA). Di Francesco, Lago and Malavolta (2018) apresentam uma revisão sobre a adoção industrial de MOA e concluem afirmando que o uso de MOA ainda é

bastante incipiente. A maioria dos trabalhos existentes focam em proposições de arquiteturas, discussão de vantagens e desvantagens de uso e análise dos desafios para migração e implantação industrial de MOA (Di Francesco, Lago and Malavolta, 2018), (Sarkar, Vashi and Abdulla, 2018).

A ideia de virtualizar equipamentos de automação, como um Controlador Lógico Programável (CLP) foi inicialmente apresentada em Givehchi *et al.* (2017). Baseando-se nas possibilidades que as arquiteturas baseadas em serviços oferecem, este trabalho apresenta o desenvolvimento de Controladores Lógico Programáveis (CLP) como um micros serviço com foco em aplicações de automação e controle na I4.0. Uma explicação sobre as tecnologias e infraestrutura de software e hardware utilizados, com foco no framework Molecular de micros serviços é descrita. Uma aplicação de controle de uma planta industrial virtual é apresentada utilizando os micros serviços desenvolvidos: Controlador Lógico Programável (CLP) e Visualização (Dashboard).

2. MICROSERVIÇOS E FRAMEWORK MOLECULAR

2.1 Microserviços

Na SOA, os serviços são módulos de negócio que possuem interfaces de conexão que são invocadas via mensagens (Dustdar, Papazoglou, 2008). Essas interfaces disponibilizam recursos sem que a implementação dos serviços seja conhecida, sendo esses serviços processos independentes que são descritos, disponibilizados, localizados e invocados por meio de redes de comunicação. Os serviços podem interagir através de simples respostas a requisições de processos até a execução de ações e tarefas complexas que exigem uma comunicação entre provedores e consumidores de serviços.

O conceito de serviços permite que elementos que compõem uma aplicação possam ser clientes ou servidores, a depender da necessidade do processo, e que haja interação, baseada em troca de mensagens, entre quaisquer elementos dos níveis hierárquicos de um processo industrial (Moraes, 2017). A arquitetura SOA também permite que qualquer aplicação industrial, no contexto de IIoT e I4.0, possa ser rapidamente composta pela seleção e combinação de novos serviços e funcionalidades disponibilizadas como um serviço em nuvem (Bigheti, Fernandes, Godoy, 2019).

Evolução da SOA, os micros serviços compõem arquiteturas de software em que as aplicações são decompostas em serviços de baixo acoplamento, oferecendo modularidade. Os micros serviços são um padrão arquitetural que objetiva criar sistemas escaláveis e com a capacidade de permitir rápidas mudanças dos modelos de negócio na nuvem (Butzin, Golatowski, Timmermann, 2016). Quando se discutem arquiteturas baseadas em serviços, um ponto que deve ser levado em conta é o tipo de organização que se deve adotar para que todo o conjunto funcione de forma satisfatória. Richards (2015) apresenta dois conceitos de composição de serviços: a orquestração e a coreografia. A Fig. 2 apresenta um esquema comparativo entre eles.

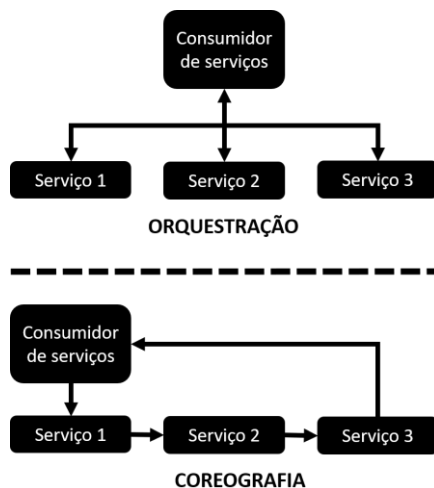


Fig. 2 Comparativo entre orquestração e coreografia de serviços. (Fonte: Adaptado de Richards, 2015)

Na orquestração, existe sempre um processo central (mestre), sendo um serviço ou uma atividade de negócio que coordena a chamada de outros serviços para compor uma função de maior relevância. Para este tipo de composição, cada serviço participante não tem conhecimento de que faz parte de uma composição de serviço de maior hierarquia, com exceção do processo mestre. Somente o processo mestre controla a aplicação e tem a responsabilidade de coordenar a execução dos serviços, de forma que seja obedecida uma ordem predefinida para a execução.

Na coreografia, os serviços são pré-determinados antes da sua execução. Quando um serviço é acionado e envia uma mensagem, outros serviços podem estar programados antecipadamente para receber ou não essa mensagem e dispararem outras ações. Na composição de serviços por coreografia, não existe um processo mestre que controla e coordena os demais processos, sendo a execução, portanto, descentralizada. Nessa configuração, cada processo envolvido tem o conhecimento de que faz parte de uma composição de serviços e que precisa interagir com outros serviços/aplicações de maneira ordenada para que a composição resultante tenha sucesso.

2.3 Molecular

O *Molecular* é um framework de código aberto utilizado na criação de microserviços em linguagem *JavaScript* para a plataforma *Node.js*. Ele ajuda a construir serviços eficientes, confiáveis e escaláveis, e fornece muitos recursos para construir e gerenciar microserviços (Molecular, 2018). Um framework representa uma estrutura base ou uma plataforma de desenvolvimento, que contém ferramentas, bibliotecas, sistemas e componentes que simplificam o processo de desenvolvimento de uma solução específica.

A Fig. 3 apresenta a estrutura do framework *Molecular*, na qual os serviços são executados em “nós” individuais que se comunicam via serviço de comunicação (Transporter), ou seja, um intermediário de envio de mensagens.

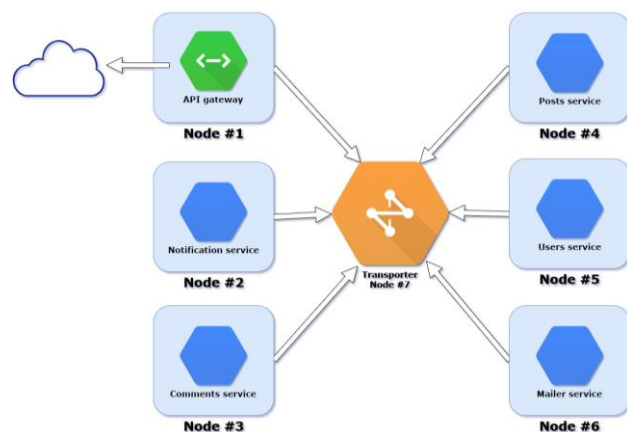


Fig. 3 Arquitetura Orientada a Microserviços do Molecular (Fonte: Molecular, 2018)

Na arquitetura do *Molecular* não existe qualquer tipo de hierarquia ou prioridade entre os microserviços que a compõem. Uma grande vantagem desse framework é que todos os microserviços desenvolvidos possuem disponível um recurso automático de registro e descoberta, sendo assim, todos os serviços existentes são informados após a criação de um novo serviço ou após a disponibilização de uma nova funcionalidade em um serviço. O framework *Molecular* proporciona o desenvolvimento de um estilo arquitetônico em que as aplicações são decompostas em serviços de baixo acoplamento, oferecendo modularidade e tornando mais simples a sua implementação.

3. TECNOLOGIAS APLICADAS

O trabalho desenvolvido faz uso de ferramentas tecnológicas que são interconectadas para que seja possível aplicar o conceito do uso de serviços e da arquitetura orientada a serviços. Nesta seção serão apresentadas essas ferramentas bem como uma breve introdução do funcionamento de cada uma, para melhor compreensão do estudo em questão.

3.1 Factory I/O

A virtualização de processos industriais permite reproduzir ou simular o funcionamento de uma planta industrial em um ambiente digital. Para testes e validação do presente trabalho foi necessário implementar essa tecnologia através do software *Factory I/O*. O *Factory I/O* é um simulador de fábrica 3D utilizado para aplicar tecnologias de automação. É um software que permite construir rapidamente uma fábrica virtual usando uma seleção de peças industriais comuns. Geralmente o *Factory I/O* é utilizado como uma plataforma de treinamento de Controlador Lógico Programável (CLP), uma vez que os CLPs são os controladores mais comuns encontrados em aplicações industriais. No entanto, também pode ser usado com microcontroladores, SoftPLC, Modbus, entre muitas outras tecnologias.

3.2 OpenPLC

O OpenPLC é um controlador lógico programável de código aberto, tanto em software quanto em hardware, padronizado de acordo com a norma IEC 61131-3, que define a arquitetura básica de software e as linguagens de programação para CLPs. O OpenPLC têm sido usado para implementação de funcionalidades de um Controlador Lógico Programável (CLP) em vários tipos de aplicação (Open PLC, 2019). O servidor do OpenPLC pode rodar localmente com o hardware que faz a aquisição das entradas e saídas, ou pode usar um hardware remoto para aquisição de dados em rede.

A lógica de operação da planta industrial foi desenvolvida no OpenPLC Editor, um software que permite escrever programas de CLP que são executados no OpenPLC. Os programas podem ser desenvolvidos no padrão: Lógica Ladder (LD - *Ladder Logic*), Diagrama de Blocos Funcionais (FBD - *Function Block Diagram*), Lista de Instruções (IL - *Instruction List*), Texto Estruturado (ST - *Structured Text*) e Gráfico de Funções Sequenciais (SFC - *Sequential Function Chart*).

3.3 Node-RED

Parte do sistema desenvolvido nesse estudo consiste na implementação de um sistema para monitorar as variáveis de entradas e saídas do CLP (OpenPLC) desenvolvido com um serviço. Para isto foi escolhido o Node-RED. O Node-RED é uma ferramenta de programação para conectar dispositivos de hardware, Interface de Programação de Aplicativos (API - *Application Programming Interface*) e serviços online. Ele fornece um editor baseado em um navegador web que utiliza uma abordagem de programação gráfica. O Node-RED possui blocos de funções do framework *Molecular*, que é utilizado para a implementação do microserviço utilizado neste trabalho. Além disso, o Node-RED disponibiliza blocos específicos para a construção de dashboard, o que facilita o desenvolvimento de um sistema de supervisão de estado de variáveis de um dispositivo.

4. MATERIAIS E MÉTODOS

4.1 Microserviço CLP

O microserviço CLP é responsável por integrar as funcionalidades de um CLP na arquitetura do *Molecular*. A proposta é adicionar o OpenPLC como serviço na estrutura do *Molecular*. O OpenPLC une as funcionalidades de um CLP com os benefícios de uma arquitetura aberta de hardware e software. O objetivo desse serviço é executar lógicas e sequenciamento de tarefas que são executadas em CLPs convencionais, bem como a aquisição e o monitoramento de dados de entradas e saídas do controlador. No entanto, o diferencial dessa proposta é que ao ser disponibilizado como um serviço, o microserviço CLP pode ser implantado em múltiplas plataformas como computador, sistema embarcado ou nuvem, conforme mostrado na Fig. 4.

Na estrutura do microserviço CLP implementado com o framework *Molecular*, são disponibilizadas ações relacionadas às funcionalidades principais do serviço: programação do controlador, monitoramento de variável de entrada e

monitoramento de variável de saída. As ações de monitoramento de I/O do microserviço podem ser atreladas a diferentes tipos de I/O, como entradas e saídas analógicas e digitais, sendo que a única diferença é o tamanho do dado (em bits) da variável de interesse, sendo 1 bit para variáveis digitais e 16 bits para variáveis analógicas. A interface entre o código do microserviço no *Molecular* e as variáveis de entradas e saídas do OpenPLC é feita usando comunicando Modbus TCP.

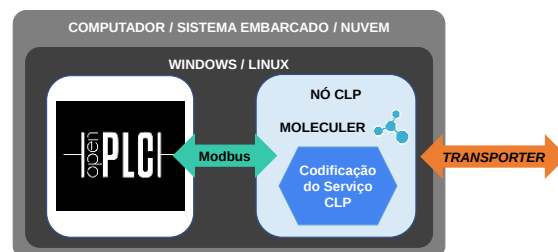


Fig. 4 Estrutura do Microserviço CLP

4.2 Arquitetura Proposta

A aplicação apresentada neste trabalho consiste de uma arquitetura de microserviços distribuída do *Molecular*, como é apresentado na Fig. 5. Trata-se de uma composição de serviços por orquestração, pois os microserviços trabalham com o mecanismo de pergunta/resposta, ou seja, o serviço Dashboard envia uma requisição ao serviço CLP, que por sua vez retorna a resposta ao serviço Dashboard.

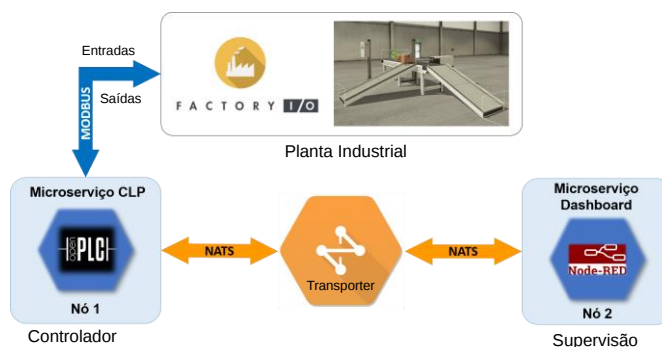


Fig. 5 MOA da aplicação do microserviço CLP

Na aplicação da Fig. 5, o *Factory I/O*, software que simula a planta industrial, se comunica via protocolo Modbus TCP diretamente com o OpenPLC, que é responsável por controlar o funcionamento da planta industrial. Toda a lógica de controle do CLP foi desenvolvida em Ladder por meio do *OpenPLC Editor*.

O microserviço CLP (Fig. 3) foi implantado num sistema embarcado (Nó 1). Dessa forma, é possível comunicar seus dados de interesse através do *Transporter* NATS para o microserviço Dashboard (Nó 2), implantado num computador usando o Node-RED. No caso em questão, os dados de interesse do microserviço CLP são os estados das variáveis de entrada e saída do OpenPLC. Isso possibilita monitorar o comportamento do CLP em tempo real, a partir de qualquer dispositivo que contenha acesso a um navegador web e à

infraestrutura rede em que os dispositivos de interesse estejam conectados.

5. RESULTADOS

A concepção desse trabalho foi possível através da integração das tecnologias descritas. A Fig. 6 demonstra os dispositivos e software utilizados, bem como a estrutura à qual foram aplicados. Dois dispositivos são utilizados, sendo um computador (PC) e um sistema embarcado Raspberry Pi.

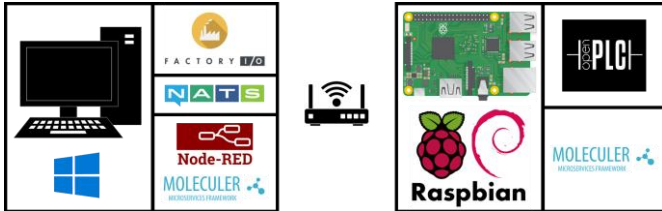


Fig. 6 Estrutura de hardware e software para testes do CLP com um Serviço

No PC está instalado o sistema operacional Windows. Nesse sistema operacional estão instalados o software Factory I/O, o transporter NATS e o Node-RED. No Node-RED foi desenvolvido o microserviço Dashboard, responsável por requisitar e disponibilizar os dados do microserviço CLP através de uma interface de visualização criada.

No sistema embarcado está instalado o sistema operacional Raspbian (Linux), que contém o software OpenPLC e o microserviço CLP. O OpenPLC foi instalado em uma placa Raspberry Pi, que é um mini computador que possui pinos de entradas e saídas que podem ser conectados à dispositivos externos como por exemplo, sensores, reles, conversores, entre outros, possibilitando desenvolver aplicações voltadas à área de automação.

O objetivo deste experimento era demonstrar a viabilidade de se aplicar o CLP como um serviço, portanto a planta industrial utilizada é virtual, como mostra a Fig. 7.

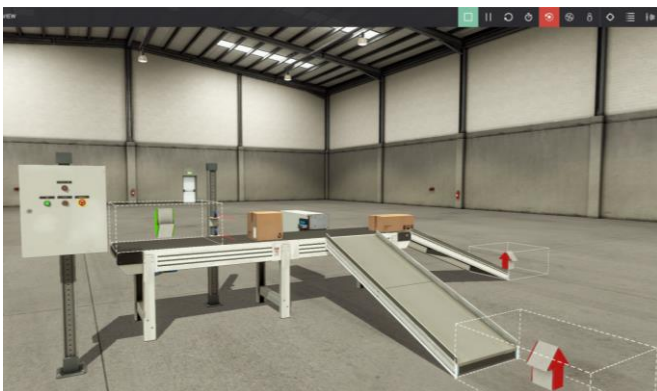


Fig. 7 Planta Industrial em Simulação no Software Factory I/O

Esse cenário de simulação de planta industrial consiste em selecionar caixas de dois tamanhos diferentes, e um atuador desvia a caixa mais alta do curso da esteira para a rampa lateral. A planta virtual é composta de um painel elétrico com botoeiras de comandos de Liga, Desliga e Emergência. Um

sinaleiro, também inserido no painel, indica quando o botão de Emergência está acionado. A esteira transportadora de caixas é acionada através das botoeiras contidas no painel elétrico. Dois modelos de caixas distintos são inseridos aleatoriamente, onde a seta verde apontada para baixo está inserida. As caixas são selecionadas pelos sensores instalados ao lado da esteira. Quando os dois sensores são acionados ao mesmo tempo, a lógica construída no OpenPLC habilita o atuador após um determinado tempo, para que a caixa alta seja desviada do seu curso pela rampa lateral. Caso somente o sensor inferior seja acionado, o atuador não altera seu estado, permitindo que as caixas baixas prossigam por todo o curso da esteira, deslizando pela rampa localizada na extremidade da esteira. As setas vermelhas apontadas para cima removem da cena as caixas que deslizam pelas rampas.

O uso do Node-RED facilitou criar o microserviço Dashboard, pois ele contém blocos do Moleculer e blocos para a elaboração de dashboard, o que simplifica a elaboração tanto do microserviço quanto da interface gráfica. O microserviço Dashboard foi implementado de forma a funcionar como uma aplicação interna do Moleculer, ou seja, que se comunica com os demais microserviços através do Transporter (NATS). Duas bibliotecas principais foram instaladas no Node-RED para a construção do microserviço Dashboard. A primeira contém ferramentas específicas para a construção de dashboards denominada *node-red-dashboard*. A segunda contém blocos com funções do framework Moleculer denominada *node-red-contrib-moleculer*.

A Fig. 8 ilustra o fluxo construído para requisitar os dados das entradas do OpenPLC e alguns dos blocos do Node-RED utilizados para a programação dos microserviços com o framework Moleculer e do dashboard.

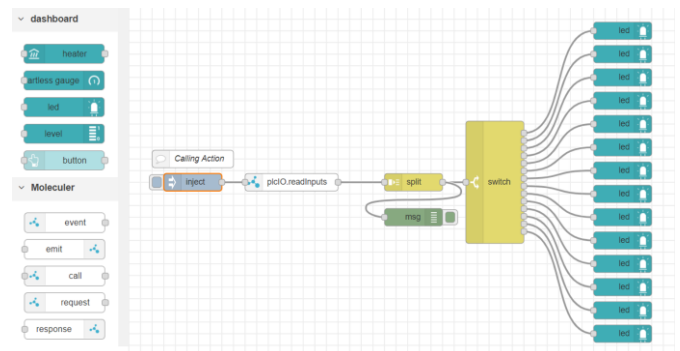


Fig. 8 Blocos de programação do Node-RED e fluxo de requisições de dados das entradas do OpenPLC

A Fig. 9 mostra o dashboard criado para monitorar o estado das entradas e saídas do OpenPLC controlando a planta industrial simulada pelo Factory I/O. Isso torna possível monitorar os sinais de sensores e atuadores da planta industrial até mesmo por meio de dispositivos móveis, como por exemplo, um smartphone ou tablet. O teste operacional realizado pode ser conferido através do link: https://www.youtube.com/watch?v=fYHESMS8GiU&list=UUZ2t_UExwkvAm1jUEgoLc8A.



Fig. 9 Tela de monitoramento do microserviço Dashboard

O experimento realizado com os microserviços, ainda que simplificado, apresentou confiabilidade e desempenho compatíveis para uma aplicação de automação, não apresentando erros operacionais ou falhas de comunicação durante os testes.

A estrutura do microserviço é um paradigma diferente das arquiteturas industriais tradicionais em termos de desenvolvimento, manutenção e implantação. As dificuldades são mais relacionadas à aprendizagem de novos conceitos e ambientes de desenvolvimento do que a complexidade do uso dos microserviços em si. Um outro ponto a ser destacado é a complexidade de gerenciamento em grandes aplicações contendo muitos serviços. O uso de um framework como o Molecular fornece um modelo padrão para o desenvolvimento de serviços, juntamente com o suporte a linguagens de programação diferentes. A manutenção (alterar/atualizar) e a implantação (execução/replicação) podem ser feitas online e remotamente através da infraestrutura de rede.

A facilidade de replicação e reuso dos microserviços atrelada às funcionalidades do framework Molecular de balanceamento das requisições entre múltiplas instâncias de um mesmo serviço na arquitetura, fornecem um potencial interessante para o desenvolvimento de aplicações de automação com redundância, ainda que não tenha sido alvo neste trabalho.

Trabalhos futuros focarão na realização de experimentos com os microserviços desenvolvidos para aplicações de controle de processos em malha fechada.

6. CONCLUSÕES

Este trabalho apresentou uma abordagem de uso de um Controlador Lógico Programável como um serviço disponibilizado em uma arquitetura para aplicações de automação na Indústria 4.0. Esta forma de implementação incorpora uma série de vantagens das arquiteturas orientadas a serviço para aplicações de automação, como a criação de aplicações industriais descentralizadas, com grande modularidade, escalabilidade e independência entre sistemas.

Um experimento de automação de uma planta industrial foi utilizado para demonstrar a aplicabilidade dos microserviços CLP e Dashboard criados. Uma vantagem interessante verificada é que toda a comunicação em rede entre os serviços é abstraída de desenvolvimento, pois acontece de forma automática com a utilização do framework de microserviços. A possibilidade de se ter o controlador como um serviço em nuvem, como apresentado nesse trabalho, foca em aplicações em que não se exige determinismo. Além disso, seu grande benefício se encontra na não necessidade de ter controladores físicos na planta, sendo somente necessária a utilização de dispositivos remotos de aquisição de dados em conjunto com o controlador virtualizado para compor a aplicação.

AGRADECIMENTOS

Os autores agradecem ao apoio: 2018/19984-4, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

REFERÊNCIAS

- Bigheti, J. A. *et al.* (2018) 'Proposta De Arquitetura Orientada a Microserviços Para Aplicações De Internet Das Coisas Industrial', *Proceedings XXII Congresso Brasileiro de Automática*. doi: 10.20906/cps/cba2018-0971.
- Bigheti, J. A., Fernandes, M. M. and Godoy, E. D. P. (2019) 'Control as a Service: A Microservice Approach to Industry 4.0', *2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings*. IEEE, pp. 438–443. doi: 10.1109/METROI4.2019.8792918.
- Blomstedt, F. *et al.* (2014) 'The arrowhead approach for SOA application development and documentation', *IECON Proceedings (Industrial Electronics Conference)*. IEEE, pp. 2631–2637. doi: 10.1109/IECON.2014.7048877.
- Butzin, B., Golatowski, F. and Timmermann, D. (2016) 'Microservices approach for the internet of things', *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2016-Novem. doi: 10.1109/ETFA.2016.7733707.
- Colombo, A. W. *et al.* (2017) 'Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution', *IEEE Industrial Electronics Magazine*. IEEE, 11(1), pp. 6–16. doi: 10.1109/MIE.2017.2648857.
- Dustdar, S. and Papazoglou, M. P. (2008) 'Services and Service Composition – An Introduction (Services und Service Komposition – Eine Einführung)', *it - Information Technology*, 50(2), pp. 86–92. doi: 10.1524/itit.2008.0468.
- Espí-Beltrán, J. V., Gilart-Iglesias, V. and Ruiz-Fernández, D. (2017) 'Enabling distributed manufacturing resources through SOA: The REST approach', *Robotics and Computer-Integrated Manufacturing*. Elsevier Ltd, 46(November 2016), pp. 156–165. doi: 10.1016/j.rcim.2016.09.007.
- Di Francesco, P., Lago, P. and Malavolta, I. (2018) 'Migrating Towards Microservice Architectures: An Industrial

Survey', *Proceedings - 2018 IEEE 15th International Conference on Software Architecture, ICSA 2018*. IEEE, pp. 29–38. doi: 10.1109/ICSA.2018.00012.

Jammes, F. *et al.* (2014) 'Promising technologies for SOA-based Industrial Automation Systems', *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach*, 9783319056, pp. 89–109. doi: 10.1007/978-3-319-05624-1_4.

Leitão, P., Colombo, A. W. and Karnouskos, S. (2016) 'Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges', *Computers in Industry*, 81, pp. 11–25. doi: 10.1016/j.compind.2015.08.004.

Moleculer (2018) *What is Moleculer?* Available at: <https://moleculer.services/docs/0.13/> (Accessed: 10 October 2018).

Moraes, E. C. (2017) 'Desenvolvimento de Interfaces Baseadas em Serviço para Integração de Sistemas Heterogêneos na Manufatura', p. 241.

OpenPLC (2019). *The Open PLC project*. Available at: <https://www.openplcproject.com/> (Accessed: 5 May 2019).

Richards, M. (2015) *Microservices vs Service Oriented Architecture*, *O'Reilly Media*. Available at: <http://ir.obihiro.ac.jp/dspace/handle/10322/3933>.

Sarkar, S., Vashi, G. and Abdulla, P. P. (2018) 'Towards Transforming an Industrial Automation System from Monolithic to Microservices', *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. IEEE, 2018-Septe, pp. 1256–1259. doi: 10.1109/ETFA.2018.8502567.

Theorin, A., Hagsund, J. and Johnsson, C. (2014) 'Service orchestration with OPC UA in a graphical control language', *19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2014*. IEEE, pp. 1–6. doi: 10.1109/ETFA.2014.7005351.

Xiao, Z., Wijegunaratne, I. and Qiang, X. (2017) 'Reflections on SOA and Microservices', *Proceedings - 4th International Conference on Enterprise Systems: Advances in Enterprise Systems, ES 2016*. IEEE, pp. 60–67. doi: 10.1109/ES.2016.14.