



Universidade Federal da Bahia
Instituto de Matemática / Escola Politécnica

Programa de Pós-Graduação em Mecatrônica

**UMA ARQUITETURA ORIENTADA A
SERVIÇO VISANDO O SUPORTE À
AUTOMAÇÃO INDUSTRIAL INTEGRADA E
DISTRIBUÍDA**

Tiago de Oliveira Silva

DISSERTAÇÃO DE MESTRADO

Salvador
26 de Março de 2018

TIAGO DE OLIVEIRA SILVA

**UMA ARQUITETURA ORIENTADA A SERVIÇO VISANDO O
SUPORTE À AUTOMAÇÃO INDUSTRIAL INTEGRADA E
DISTRIBUÍDA**

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Mecatrônica.

Orientador: Herman Augusto Lepikson, Dr. Eng

Salvador
26 de Março de 2018

Ficha catalográfica elaborada pelo Sistema Universitário de Bibliotecas (SIBI/UFBA),
com os dados fornecidos pelo(a) autor(a).

Silva, Tiago de Oliveira

Uma arquitetura orientada a serviço visando o
suporte à automação industrial integrada e distribuída /
Tiago de Oliveira Silva. -- Salvador, 2018.
68 f. : il

Orientador: Herman Augusto Lepikson.

Dissertação (Mestrado - Pós-graduação em Mecatrônica) -
- Universidade Federal da Bahia, Escola Politécnica,
2018.

1. Automação industrial. 2. Controle Preditivo. 3.
Arquitetura orientada a serviço. 4. Serviços web . 5.
OPC UA. I. Lepikson, Herman Augusto. II. Título.

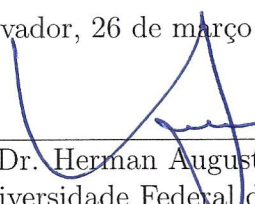
TERMO DE APROVAÇÃO

TIAGO DE OLIVEIRA SILVA

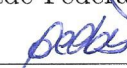
UMA ARQUITETURA ORIENTADA A SERVIÇO VISANDO O SUPORTE À AUTOMAÇÃO INDUSTRIAL INTEGRADA E DISTRIBUÍDA

Esta Dissertação de Mestrado foi julgada adequada à obtenção do título de Mestre em Mecatrônica e aprovada em sua forma final pelo Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia.

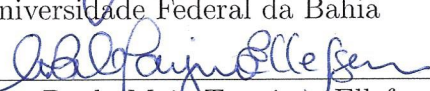
Salvador, 26 de março de 2018



Prof. Dr. Herman Augusto Lepikson
Universidade Federal da Bahia



Prof. Dr. Antônio Cezar de Castro Lima
Universidade Federal da Bahia



Dr^a Ana Paula Maia Tanajura Ellefsen

AGRADECIMENTOS

A **Deus** por me permitir concluir este trabalho mesmo com as lutas e dificuldades encontradas.

Aos meus pais, **Maria Imperatriz e Valdomiro Silva**, pelo carinho e incentivo. Itens que foram indispensáveis para a minha vida pessoal e profissional.

A minha namorada, **Alana Pereira** pela paciência, dedicação e apoio.

A minha irmã, **Amanda Catariny** pela amizade e carinho.

Ao meu orientador **Prof. Herman**, pelas muitas reuniões e sugestões neste trabalho. Ao **Prof. Leizer** pelos conselhos e orientações.

A minha amiga, **Mariane** pela companhia durante a pesquisa e participação nos momentos difíceis do desenvolvimento deste sistema.

Aos amigos do **CTAI** que apoiaram diretamente ou indiretamente esta pesquisa. Em especial **Lúcia, Marcos, Márcio Fontana, Emílio, Filipe, Márcio Martins, Yuri e Flávio**.

Aos amigos **Orlando, Odilon e Ricardo**, obrigado pelas excelentes dicas e auxílios que foram fundamentais para a conclusão e realização deste trabalho.

A toda equipe do LEA que de maneira indireta ajudou nos momentos de diversão e brincadeiras.

A todos que direta ou indiretamente me apoiaram para a realização desta pesquisa.

"Mesmo quando eu andar por um vale de trevas e morte, não temerei perigo algum, pois tu estás comigo; a tua vara e o teu cajado me protegem."

—SALMOS (23:4)

RESUMO

Maior poder de processamento, maior capacidade de comunicação, maior interoperabilidade e maior integração são as características principais que norteiam as tecnologias atuais. A disseminação desses conceitos em ambientes industriais tem ocorrido sob uma dinâmica inovadora, em que o aumento na competitividade e a disputa por novos mercados determinam uma mudança de comportamento em escala mundial. Este comportamento leva à percepção de que o mundo está vivendo um intenso momento de transição, o qual, nas últimas décadas, caracterizou-se como uma quarta revolução industrial. Desta forma, é necessário criar condições para que a transição inevitável dos modelos industriais clássicos para um modelo que se projeta em um futuro próximo ocorra, de forma a permitir que dispositivos legados e cada vez mais distribuídos coexistam com novas tecnologias e princípios. Um dos problemas mais relevantes a ser enfrentado é o da interoperabilidade entre sistemas heterogêneos em chão de fábrica, principalmente quando se tratam de sistemas distribuídos. O presente trabalho tem por objetivo apresentar uma arquitetura que permita a interoperabilidade entre dispositivos legados com os novos conceitos inseridos pela nova Revolução Industrial. Para tanto, foi desenvolvido um sistema capaz de permitir a operação de dispositivos de campo por aplicações clientes remotamente, permitindo interconectividade, interoperabilidade e transparência. A proposta deste trabalho é utilizar uma arquitetura orientada a serviços para integrar as aplicações do chão de fábrica, de forma modular e escalável.

Palavras-chave: SOA, OPC UA, Controle preditivo, *Web Services*

ABSTRACT

The advances in processing power, communication, interoperability and integration capability have characterized the current technologies. Spreading those concepts into industrial environments has materialized a new dynamic, where the growth of the competitiveness for new and larger markets determine new practices in a global level. Those practices demonstrate an important transition moment experienced by the world during the last decades, also called the Fourth Industrial Revolution. Hence, creating conditions capable of making the transition from classical industrial models to a new model is an inevitable need. This new model, which must be a reality in a near future, is characterized by the coexistence of legacy devices and increasingly distributed new technologies. One of the most relevant issues being faced is the interoperability between heterogeneous systems and factory floor, mainly when distributed systems are considered. The current work aims to demonstrate a middleware architecture that allows the interoperability between legacy devices and the concepts existing in the new industrial revolution. Along these lines, a system to permit operation of field devices by client applications was developed, through which legacy devices might be handled with interoperability, interconnectivity and transparency features. Wherefore, this work proposes a servicesoriented architecture in order to integrate the factory floor applications with modularity and scalability.

Keywords: SOA, OPC UA, Predictive control, Web Services

SUMÁRIO

Lista de Figuras	xv
Capítulo 1—Introdução	1
1.1 Objetivo Geral	5
1.2 Objetivos específicos	6
1.3 Estrutura da Dissertação	6
Capítulo 2—Referencial Teórico	7
2.1 Indústria e Revoluções Industriais	8
2.2 Contextualização da indústria 4.0	12
2.3 Tecnologias habilitadoras	14
2.3.1 Internet das Coisas	14
2.3.2 Internet dos Serviços	15
2.3.3 <i>Big data</i>	16
2.3.4 Sistemas Ciberfísicos	17
2.3.5 Fábrica inteligente	17
2.4 Padrões para a integração e comunicação	17
2.4.1 <i>Web Services</i>	21
2.4.2 OPC	25
2.5 Trabalhos relacionados e contribuições	28
Capítulo 3—Plataforma Aberta para integração via web service	35
3.1 Servidor OPC UA	41
3.2 Aplicação Cliente	42
3.3 Síntese	44

Capítulo 4—Validação Experimental	45
4.1 Descrição da planta para validação experimental	45
4.2 Identificação do Modelo Matemático da Planta	47
4.3 Desenvolvimento da aplicação cliente MPC utilizando SOA	48
4.4 Discussão dos resultados obtidos	51
4.5 Síntese	52
Capítulo 5—Conclusões	53
5.1 Produção Científica Relacionada	54
5.2 Recomendações e Trabalhos futuros	54
Referências Bibliográficas	55
Apêndice A—Apêndice Códigos Sistemas web service OPC UA client	63
Apêndice B—Apêndice Códigos matlab	67

LISTA DE FIGURAS

2.1	Histórico da revolução industrial.	9
2.2	Pirâmide Clássica da Automação.	10
2.3	Reconstrução da pirâmide da automação.	12
2.4	Comunicação na arquitetura orientada a serviços.	19
2.5	Mapa conceitual.	23
2.6	Arquitetura SOAP	24
2.7	Especificações do OPC UA.	27
3.1	Arquitetura IWSOP.	36
3.2	Funcionalidades da plataforma	37
3.3	Mediador <i>Web</i> cliente	37
3.4	Diagrama UML do aplicativo	40
3.5	Servidor OPC UA.	41
3.6	Estrutura do <i>Software TOPServer</i>	42
3.7	Construção do cabeçalho	43
4.1	Planta piloto	46
4.2	Sinal construído para a identificação.	47
4.3	Aplicação cliente MPC.	48
4.4	Trajetória do MPC para controle regulatório	49
4.5	Trajetória do MPC para controle servo	50
4.6	Planta piloto	50

LISTA DE ABREVIATURAS

CCE - *Component Core Editor* (Núcleo Editor de componente)

CLP - Controlador Lógico Programável

CNC - Comando Numérico Computadorizado

CPS - *cyber physical systems* (Sistema Ciberfísicos)

CTAI - Centro de Capacitação Tecnológica em Automação Industrial

DCOM - *Distributed Common Object Model* (Modelo de Objeto Comum Distribuído)

ERP - *Enterprise Resource Planning* (Planejamento de Recursos Empresariais)

FMC - Célula de Fabricação Flexível

HTTP - Protocolo de Transferência de Hipertexto

HMI ou *IHM* - Interface Homem Máquina

I4 - Indústria 4.0

IoS - *Internet of Service* (Internet dos serviço)

IoT - *Internet of Things* (Internet das coisas)

IP - *Internet protocol* (Protocolo de internet)

IWSOP - Plataforma aberta para integração via *web service*

JSON - *Java Script Object Notation* (Notação de Objeto Javascript)

MES - Sistemas de Execução da Manufatura

MPC - Controlador Preditivo: *Model Predictive Control*

MV - Variável manipulada

MVC - Modelo, visualização e controle

OPC - *Object Linking and Embedding (OLE) for Process Control* (Vinculação e incorporação de objetos (OLE) para controle de processo)

OPC – UA - *Open Platform Communications Unified Automation* (Arquitetura Unificada OPC)

PI - Controlador do tipo Proporcional e Integral

PID - Controlador do tipo Proporcional, Integral e Derivativo

PSO - *Particle Swarm Optimization* (Otimização de enxame de partículas)

PV - Variável de processo

RFID - Identificação por rádio frequência

REST - *Representational State Transfer* (Transferência de Estado Representacional)

SCADA - Sistema de supervisão e aquisição de dados

SDCD - Sistema Digital de Controle Distribuído

SIGE ou *SIG* - Sistema Integrado de Gestão Empresarial

SOA - *Service Oriented Architecture* (Arquitetura Orientada a Serviços)

SOAP - *Simple Object Access Protocol* (Protocolo de acesso a objetos simples)

SP - Valor de referência ou *Setpoint*

TI - Tecnologia de Informação

W3C - *World Wide Web Consortium* (Consórcio gestor da Internet)

WSDL - Linguagem de descrição de serviços *web*

XML - *Extensible Markup Language* (Linguagem de Marcação Extensível)

Capítulo

1

INTRODUÇÃO

O aumento das formas de consumo e a diversificação dos produtos e mercados vêm impactando a diversificação nos meios de produção ao longo dos anos. Desta forma, o setor industrial vem se tornando um ambiente complexo, no qual a produção de bens tem se alterado devido às novas necessidades impostas por um mercado cada vez mais globalizado e competitivo. Dentre as principais modificações na forma de produção, observa-se a necessidade crescente de desenvolver em larga escala, produtos com um ciclo de vida mais curto, com elevados padrões de qualidade, que visam atender aos mais diversos mercados, com uma redução no custo final e que possibilita manter a competitividade em um sistema de mercado, no qual coexistem pequenas empresas e grandes corporações, e como resultado final, na busca constante da redução do custo de produção (SCHWAB; MIRANDA, 2016).

Dentre o conjunto de ferramentas utilizadas para gerir essa complexidade, sistemas de automação industrial têm representado uma alternativa capaz de agregar flexibilidade na produção, por meio da integração de diferentes setores da indústria. A necessidade de integração pode ser descrita, dentre outros aspectos, como o modo significativo que a troca de informações entre os diversos setores e partes interessadas nos processos de produção industrial pode variar. Esta variação ocorre na forma de comunicação, no fluxo e volume de dados, no custo de processamento e em aspectos de segurança da informação (DUNCAN, 2004).

Um sistema de automação industrial é aquele que utiliza sistemas mecânicos e computacionais, com o objetivo de aperfeiçoar processos produtivos. O seu surgimento está ligado ao surgimento da eletrônica e da própria informática (GROOVER, 2011). Por este

motivo, as sucessivas evoluções ocorridas no campo da informática têm impacto, ainda que posteriormente, no aperfeiçoamento dos sistemas de automação industrial.

Com o melhor poder de processamento, é possível utilizar equipamentos que possuem a capacidade de executar lógicas mais complexas de forma rápida e eficaz. À medida que novas tecnologias emergem, o conhecimento se acumula e novos desafios podem ser alcançados. Com relação aos sistemas de automação industrial, se observa que a evolução tecnológica permite a criação de novos métodos e processos de produção, melhora a capacidade de inovação em recursos humanos, aumenta a produtividade, permite a tomada de decisões em tempo real, ou seja, um trabalho preditivo preventivo que possibilita aproveitar as oportunidades e superar os desafios. Soma-se a esse cenário o respeito a conceitos de segurança e diminuição da interferência humana na execução de parte dos processos.

A fabricante de computadores industriais Siemens, por exemplo, comercializava em 2010, o IPC427C, uma linha de computadores industriais de alto desempenho que utilizava um processador Intel de 2 núcleos, já defasado há 5 gerações de processadores, quando comparado aos computadores pessoais comercializados no mesmo ano (GLOBAL, 2018). Para facilitar o entendimento, em 2010, se comercializavam computadores pessoais com processadores de 6 núcleos e com um desempenho de aproximadamente 4 vezes maior que o utilizado naquele computador industrial.

De forma semelhante, em janeiro de 2018, a Siemens comercializava IPC427E, que utilizava o processador lançado 3 anos antes e que possuía 4 núcleos, enquanto os processadores da época já contavam com 18 núcleos (INTEL, 2017). Essa comparação serve para demonstrar que, embora os computadores industriais de alto desempenho não necessitem dos processadores de última geração, a evolução desses processadores passa, ainda que um pouco mais lentamente, a ser incorporada pelos sistemas industriais e torna o primeiro, uma ferramenta importante na evolução do segundo. Todos os impactos gerados pelas modificações que o setor industrial vem sofrendo para se manter competitivo e inovador, criam um ambiente propício para a atualização dos sistemas e padrões utilizados para a troca das informações entre os diferentes ambientes.

Outra importante consequência que a evolução dos sistemas computacionais causa nos sistemas de automação industrial é o aumento do fluxo de dados circulante na rede industrial. Como os computadores industriais estão mais velozes e de utilização mais ampla, a informação passa a ser processada localmente e transferida diretamente para as partes interessadas. Este novo comportamento na integração industrial diminuiu a importância de um servidor central, permitiu um maior compartilhamento de recursos, permitiu que a informação fosse gerada e processada de forma local e, conseqüentemente, permitiu a

modularização dos setores industriais, de acordo com o tipo de recurso presente. Por outro lado, as velocidades de comunicação das redes industriais tradicionais passaram a ser um limitador na transferência de informações. Dessa forma, as redes de computadores não industriais e mais velozes passaram a ser utilizadas nos setores ligados à produção industrial.

A necessidade de compartilhamento de recursos gerou a criação dos Sistemas distribuídos, localizados em computadores interligados em rede. Os diversos sistemas computacionais se comunicam e coordenam suas ações apenas enviando mensagens entre si. O uso de sistemas de forma distribuída se justifica pela necessidade de se obter recursos ou serviços de forma compartilhada em uma mesma arquitetura, seja esta automotiva ou empresarial. A utilização de sistemas distribuídos no ambiente industrial tem como propósito inserir novas soluções no ambiente de produção como, por exemplo, transmitir as informações geradas no chão de fábrica aos diversos ambientes da organização, de forma segura, rápida e uniforme (COULOURIS et al., 2013). No entanto, transmitir informações rapidamente entre os diferentes ambientes é complexo, pois as arquiteturas existentes possuem diferentes padrões de comunicação e possuem restrições de integração nas suas redes, devido a uma heterogeneidade de *softwares* e *hardwares*, (COLOMBO; BANGEMANN; KARNOUSKOS, 2013).

Ao se observar a evolução dos sistemas computacionais na indústria nos últimos 10 anos, percebe-se que houve um expressivo aumento no poder de processamento, uma maior distribuição de sistemas em várias unidades de processamento, uma utilização de algoritmos complexos aptos a melhorar significativamente a automatização de processos e a diminuição da intervenção humana em ambientes que exigem repetitividade, controle de atuação e segurança. Tais aspectos, juntamente com a recente dinâmica da citada acerca da complexidade dos sistemas de produção industrial atual, têm sido constantemente definidos como a quarta revolução industrial (SCHWAB; MIRANDA, 2016; RESEARCH, 2013).

A quarta revolução industrial é caracterizada pelo uso de novas tecnologias conectadas que colaboram para criar um ambiente de produção industrial integrado tanto à cadeia produtiva quanto ao mercado consumidor. Para alcançar seus objetivos, ela se utiliza de recursos e serviços de forma distribuída, inserido-os em uma arquitetura de automação industrial, na cadeia de fornecedores de matéria-prima que sejam capazes de se modificar, preferencialmente, de forma automatizada, para se adaptar a alterações impostas por oscilações nas demandas previstas, como, por exemplo, alterações climáticas, governamentais, logística etc. Neste cenário, as novas tecnologias em ascensão são a robótica,

a inteligência artificial, a manufatura aditiva, a análise de volumes massivos de dados (*big data*) e toda gama de tecnologia recente que vem sendo utilizada para permitir a existência deste novo modelo industrial. Porém, a quarta revolução industrial não se resume ao uso dessas tecnologias de forma isolada, mas pela capacidade de convergência que pode ocorrer entre elas.

A sua base vem sendo construída por meio de planos estratégicos elaborados pelos governos de vários países, em conjunto com especialistas da área. Um dos seus objetivos é melhorar a competitividade da sua indústria, tanto no seu mercado interno como em um sistema globalizado. Independente do movimento e plano estratégico adotados pela quarta revolução industrial, observa-se que uma característica comum a tais processos é a necessidade de tornar a estrutura tradicional da fábrica mais cooperativa e independente, de forma a se adaptar autonomicamente às constantes modificações do ambiente complexo no qual está inserida.

Essa revolução tem como fundamentos a digitalização cada vez mais ubíqua, associada à comunicação em rede e a *hardwares* cada vez mais baratos. As principais tecnologias habilitadoras são a internet das coisas (IoT – *Internet of Things*), a internet dos serviços (IoS – *Internet of Services*) e os sistemas ciberfísicos. A junção dessas tecnologias habilita um ambiente propício para evoluir nos conceitos de fábrica inteligente, em que as máquinas trocam informações entre si e acessam os sistemas com o objetivo de uma produção automatizada. Segundo Hermann, Pentek e Otto (2016), existe a necessidade de se desenvolver uma arquitetura adequada para a conectividade entre os diferentes clientes e prestadores de serviços na indústria.

A capacidade dos dispositivos de trocar informação entre si, para que seja possível a execução de um ou mais serviços sob a forma de cooperação automatizada, é definida como internet das coisas. Cada dispositivo presente neste contexto possui uma interface de comunicação própria, normalmente de baixo custo, baixo consumo de energia e exigência de baixo custo computacional. Estes dispositivos trabalham com protocolos específicos e implementam características de segurança de acesso próprio. No tocante à cooperação de suas ações, se estabelece de uma forma conceitualmente simplificada, que essa troca de informações referentes aos diversos serviços realizados entre os diversos conjuntos de dispositivos de internet das coisas é caracterizada como internet dos serviços.

A quarta revolução industrial tem como tendência o uso de sistemas cada vez mais inteligentes, com um alto grau de automatização e com mínimo de dependência de ações humanas. Segundo o estudo elaborado por Reinhard, Jesper e Stefan (2016), este modelo de indústria tem uma previsão de investimento de U\$900 bilhões até o ano de 2020.

As características atreladas a este novo modelo são: a flexibilidade, a descentralização da tomada de decisão, sistemas interoperáveis e uso de aplicações orientadas a serviço. Com estas características é possível estabelecer bases para uma integração horizontal e vertical, reconstruindo o modelo de gestão e operação dos processos industriais. Estando inseridos em um novo e mais flexível ambiente de produção, no qual, as quantidades variam de acordo com a demanda, a tomada de decisão é descentralizada, os processos industriais precisam de autogerenciamento e de acesso aos diferentes sistemas que regem o funcionamento de uma empresa. Por autogerenciamento entende-se que a fábrica terá a capacidade de se reorganizar para gerir o seu funcionamento, seus estoques, demandas e atuar para corrigir possíveis eventualidades (RÜSSMAN et al., 2015; SHROUF; ORDIERES; MIRAGLIOTTA, 2014).

De acordo com Rüssman et al. (2015), para que os recursos cooperem entre si existe uma crescente necessidade de comunicação entre os elementos atuantes no processo de produção desta nova revolução. São destacados como elementos atuantes os componentes para sensoriamento, máquinas, produtos e pessoas. A IoT e IoS são tecnologias que inserem os diferentes elementos e sistemas neste movimento, no entanto, é necessário utilizar uma estrutura que permita que os diferentes ambientes de uma indústria troquem ou acessem informações, utilizando uma arquitetura padrão.

Na indústria atual, quem ainda não se adaptou ao modelo da quarta revolução, provavelmente utiliza uma estrutura de automação hierarquizada (GROOVER, 2011). Isto implica que a criação de aplicações e o método como as informações são transmitidas entre os diferentes níveis da automação utilizem padrões de desenvolvimento que não permitem a modularização de sistemas. Tal impedimento, por promover um alto acoplamento entre os setores, tende a dificultar o acontecimento de mudanças pontuais, coibindo a flexibilização dos processos.

Por este motivo, Moraes (2017) afirma que é necessário quebrar a tradicional estrutura hierárquica da automação atual e prover novas arquiteturas que possam incluir as novas tecnologias de comunicação do setor industrial. A arquitetura orientada a serviços (SOA), como descrito por Delsing et al. (2012), se mostra como uma abordagem plausível para a criação de ambientes de reestruturação, por ser capaz de utilizar tecnologias que interliguem os diferentes sistemas industriais.

1.1 OBJETIVO GERAL

A partir das considerações levantadas acerca da necessidade de utilizar arquiteturas que possibilitem a integração entre os diferentes níveis da automação, surge a seguinte questão:

é possível que uma arquitetura orientada a serviços, utilizando as características da quarta revolução industrial, integre os diferentes ambientes e setores industriais da planta industrial, desde o chão de fábrica? A partir desta hipótese, é possível estabelecer uma investigação dos conceitos de SOA como suporte à integração da quarta revolução industrial. Como objetivo principal desta investigação, se propõe a criação de uma arquitetura SOA capaz de conectar as informações do chão de fábrica até o nível de controle da planta. Para alcançar este objetivo geral, serão estabelecidos os seguintes objetivos específicos:

1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver um sistema de comunicação para aquisição de dados, disponibilizando-os de forma padronizada para uso em uma arquitetura distribuída e orientada a serviço;
- Estruturar a arquitetura de automação, utilizando os conceitos da quarta revolução para orientá-la a serviço;
- Apresentar uma arquitetura de comunicação para a aplicação de técnicas de controle avançado;
- Validar a arquitetura proposta em uma planta industrial por simulação.

1.3 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está organizado em quatro capítulos e as considerações finais. O presente capítulo apresenta algumas considerações sobre a área de investigação e aborda os objetivos deste estudo. O segundo apresenta um referencial teórico e uma revisão de trabalhos relacionados aos conceitos da quarta revolução industrial e sobre as tecnologias utilizadas neste trabalho como SOA, OPC UA (*Open Platform Communications Unified Automation*) e *Web Services*, descrevendo o estado da arte. O terceiro capítulo apresenta o desenvolvimento da plataforma de estudos e a arquitetura de automação utilizada para os testes. O quarto capítulo é responsável pela validação da arquitetura orientada a serviço, por meio de uma plataforma aplicada ao controle. O quinto capítulo apresenta as conclusões com os resultados alcançados, as contribuições da dissertação e sugestões para trabalhos futuros.

REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico sobre a evolução dos diversos tópicos tratados nesta dissertação e tem por objetivo contextualizar os novos conceitos utilizados para o desenvolvimento deste trabalho.

A seção 2.1 apresenta um preâmbulo da criação da Indústria 4.0 (I4), demonstrando como as diversas modificações tecnológicas caracterizaram as revoluções industriais ocorridas ao longo da história. Discorre também sobre como o avanço tecnológico permitiu uma alteração na pirâmide de automação, na qual a introdução do conceito de modularização permitiu modificações no gerenciamento e na manipulação dos processos industriais. Com base nessas informações, a seção 2.2 contextualiza a quarta revolução industrial e estabelece o termo I4 para descrever o modelo seguido pelas fábricas que participarem dessa revolução, além de estabelecer os princípios base para a sua implantação.

A seção 2.3 demonstra as novas tecnologias empregadas nesse contexto e como essas interagem para permitir a transição do modelo tradicional de uma fábrica para o modelo da I4, cunhando, desta forma, o conceito de fábricas inteligentes. Com bases nessas novas tecnologias, a seção 2.4 discorre sobre os padrões de comunicação utilizados para prover a integração dos dispositivos e dos sistemas inseridos nessa revolução, inclusive descrevendo os padrões utilizados no presente trabalho. Por fim, é feito um levantamento de diversos trabalhos relacionados à pesquisa, com o intuito de enquadrar o presente documento no cenário do atual estado da arte.

2.1 INDÚSTRIA E REVOLUÇÕES INDUSTRIAIS

De acordo com Schwab e Miranda (2016), o período atual da humanidade se caracteriza pelo surgimento e aperfeiçoamento de tecnologias que trazem mudanças significativas nos atuais processos de produção industrial. As mudanças e suas tecnologias são significativas por causarem, sob certos aspectos, uma fusão dos ambientes físicos, digitais e biológicos presentes nas formas modernas de se prover produtos e serviços. Alguns autores, Behrendt et al. (2017), MDIC and MCTIC (2016), Rüssman et al. (2015), também apontam que essas mudanças apresentam um caráter disruptivo e irreversível, quando comparadas com os paradigmas que caracterizavam as formas de produção industrial e promoção de serviços até o ano de 2010. De posse dos argumentos expostos e baseado nos parágrafos seguintes, é possível justificar que o período atual da sociedade se caracteriza por estar inserida em uma revolução industrial em curso.

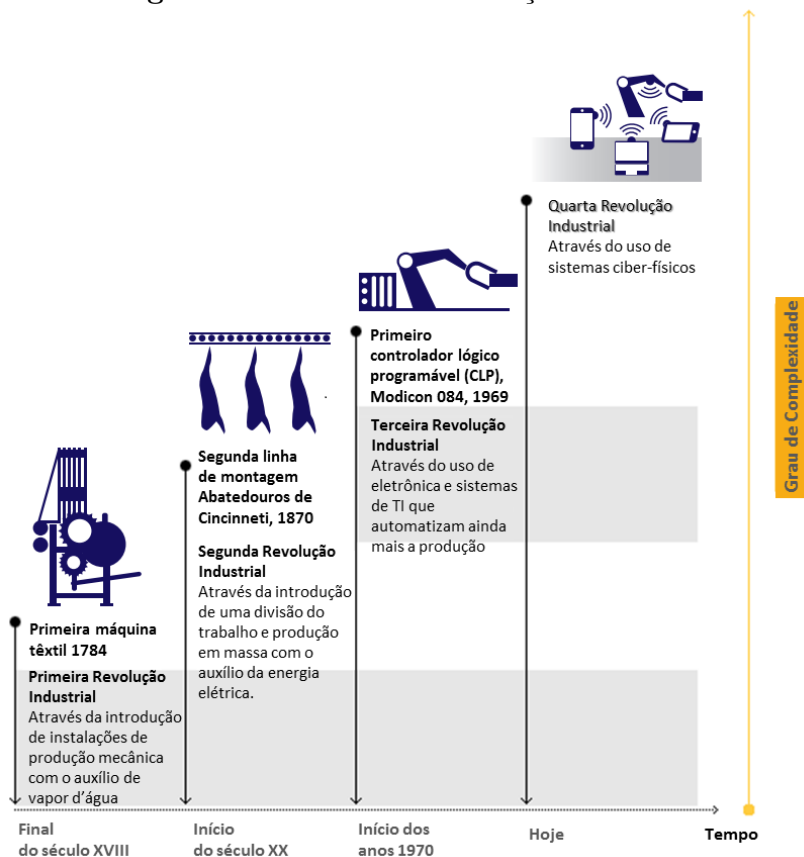
Baseado no contexto histórico das modificações industriais, como demonstrado na Figura 2.1, compreende-se hoje que os novos paradigmas representam a 4^a revolução industrial. O que caracteriza cada uma dessas ditas revoluções são os importantes ganhos de produtividade observados pela introdução de novas tecnologias. Na primeira revolução ocorreu o surgimento das primeiras máquinas movidas a vapor e, com elas, energia multiplicada para operar sistemas cada vez mais mecanizados, substituindo a força humana ou animal. Na segunda revolução, com o uso da energia elétrica e do petróleo como combustível, mas, principalmente, com o advento das linhas de produção e com a divisão do trabalho em funções elementares, deu-se início à era da produção em massa.

Com o surgimento dos primeiros controladores e robôs industriais, que rapidamente evoluíram para os conhecidos CLPs (Controlador Lógico Programável), CNCs (Comando Numérico Computadorizado) e robôs manipuladores, junto com a disseminação das técnicas japonesas de gestão da produção, as quais são atualmente conhecidas como manufatura enxuta, ocorreu uma nova e grande mudança no setor industrial, com novos e importantes ganhos de produtividade, justificando o que passou a ser chamado de terceira revolução industrial. Atualmente, com a digitalização cada vez mais ubíqua e com o barateamento dos meios de comunicação e do *hardware*, viabilizou-se os sistemas ciberfísicos, que são dispositivos mecatrônicos¹ com habilidades de comunicação e conectividade. Surgem daí os dispositivos móveis, sistemas microprocessados e aplicativos para os mais variados usos. As inúmeras e poderosas possibilidades advindas dessa profusão de tecnologias estão a

¹Dispositivos que abrigam, de forma integrada, acionamentos e encapsulamento mecânicos com a necessária eletrônica e sistema de controle com *software* embarcado para cumprir funções antes discretizadas.

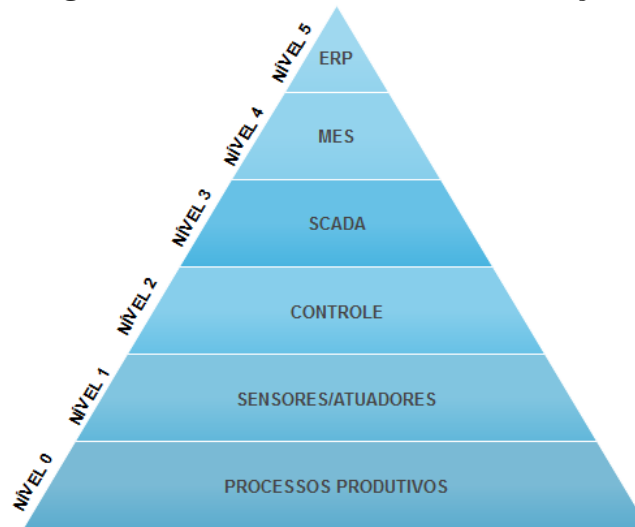
ensejar o início da quarta revolução industrial. A Figura 2.1 ilustra essas revoluções.

Figura 2.1 Histórico da revolução industrial.



Fonte: Research (2013).

A Pirâmide de automação, Figura 2.2, visa organizar de forma hierárquica o funcionamento do modelo clássico de indústria. Esta organização demonstra que os diferentes níveis possuem diferenças em relação ao tipo e ao volume de informação. No modelo clássico de infraestrutura industrial, existe uma separação nas redes de comunicação entre os níveis e setores de um ambiente industrial. Para que os sistemas de gestão acessem os dados da produção são necessárias manobras e adaptações, isto ocorre em virtude da falta de parametrização entre os sistemas de aquisição de dados e os diversos padrões de rede. Com a nova revolução, este modelo antigo começa a se modificar, pois uma das propostas é o uso de padrões de comunicação abertos e interoperáveis, de forma que a informação seja melhor acessada por outros setores de um ambiente industrial ou empresarial (MORAES; CASTRUCCI, 2007; CUCINOTTA et al., 2009).

Figura 2.2 Pirâmide Clássica da Automação.

Fonte: Elaborado pelo autor baseado em Colombo, Bangemann e Karnouskos (2013).

Este novo período é impulsionado pela integração do mundo digital e suas novas tecnologias nas empresas. Essa integração tem forçado os modelos clássicos, quando seus diferentes processos eram ligados por uma interdependência de execução, a se adaptarem e se tornarem modulares. Neste conceito de módulos, os processos passam a adquirir uma independência. Isso significa que as ações passam a ser executadas com um baixo acoplamento e uma alta coesão, o que transforma o modo como as informações são trocadas nas estruturas hierárquicas atuais.

O uso de sistemas modulares permite que o cotidiano em uma indústria se torne mais integrado e habilite uma estrutura na qual é possível uma reconfiguração dos processos mais rapidamente. Como descrito por Colombo, Bangemann e Karnouskos (2013), sistemas em que os processos são modularizados, ou seja, mais coesos e menos acoplados, são mais facilmente modificados e adaptados quando existe uma perturbação. Isso implica que a transformação da arquitetura clássica de automação para os novos modelos de arquitetura presentes na I4 permite que a informação, antes delimitada em níveis hierárquicos rígidos e interdependentes, passe a ser disponibilizada de forma descentralizada, acessível em vários setores em uma indústria, permitindo uma readaptação capaz de manter a execução do processo. Isso refletirá em um sistema cuja estrutura de comunicação tende a ser universal, independente de fabricantes, com acesso a informações distribuídas na vertical e horizontal (RESEARCH, 2013).

Nos sistemas clássicos atuais, os setores podem ser fisicamente isolados e o seu acesso

ocorre apenas localmente e por um membro autorizado pela sua cadeia hierárquica (SILVA; FILHO; MIYAGI, 2015). Porém os sistemas modulares também são responsáveis pela modificação estrutural de alguns aspectos dentro da indústria. Como os diversos elementos (*hardwares/softwares*) passam a trocar informações com outros elementos e em diferentes setores, se faz necessário uma capacidade de comunicação em rede. Cada elemento passa a compartilhar a sua funcionalidade com outros presentes na rede (MARZULLO, 2009). Toda essa troca ocorre sob aspectos específicos de controle e segurança, normalmente na forma de requisição e resposta. Nesta nova arquitetura, o compartilhamento de recursos, ações ou funcionalidades, sob um gerenciamento seguro e controlado é definido como serviço.

As implicações criadas por essa abertura representam uma mudança sistemática nas características de segurança da rede industrial. Criptografia de dados, implementação de políticas de segurança e acesso, utilização de protocolos de comunicação mais complexos e robustos, controles de segurança geral que sejam capazes de impedir invasão e também de garantir que setores que possuem acessos limitados não executem ações indesejadas por outros setores, são alguns dos diversos exemplos de como a implementação de um sistema modularizado implica a observação especial e dinâmica dos aspectos de controle e segurança, largamente utilizados em sistemas de computadores em rede, porém que não representam os mesmos riscos aos ambientes industriais não modularizados (WAIDNER; KASPER, 2016).

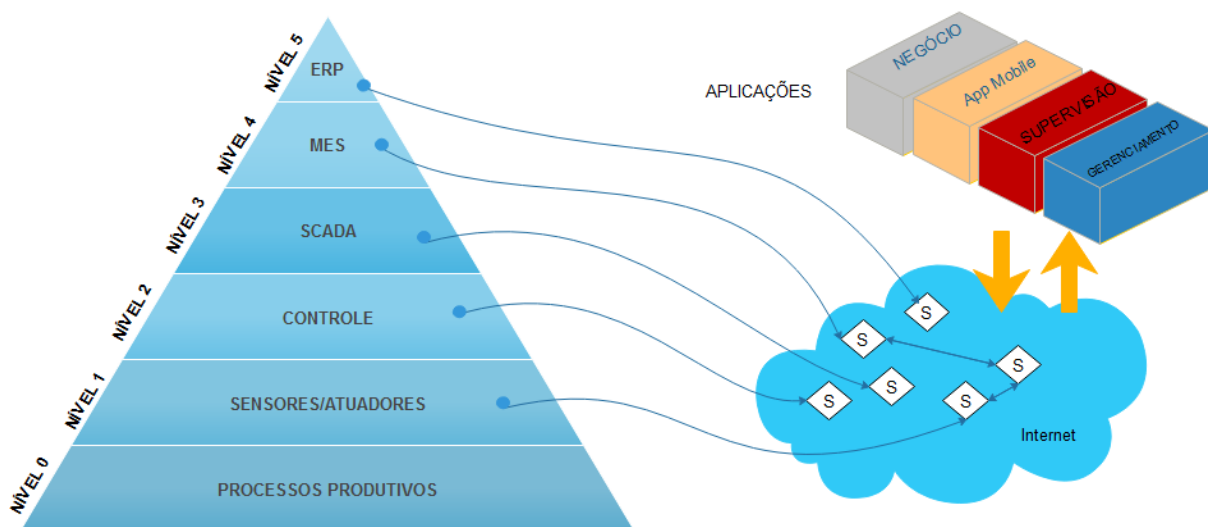
No entanto, a utilização de sistemas modulares simplifica a necessidade de treinamento de pessoas, quando comparados com os modelos industriais atuais. A popularização e o dinamismo de tecnologias atuais trazem soluções de uso intuitivo, de maneira que, em certos casos, dispensam-se manuais.

Mesmo com as modificações inseridas pelo novo conceito de modularização industrial, o conceito e a necessidade de controle continuam representando, ainda que de forma generalizada, os mesmos aspectos existentes na indústria clássica. A camada de controle presente na pirâmide de automação continua sendo a entidade que atua no processo, com o intuito de preservar as restrições e manter a otimização da operação. Entretanto se observa que características implementadas pela modularização, mais especificamente na ampliação da comunicação entre os diversos setores e os diversos *hardwares* e *softwares*, criam a necessidade de inserir atributos que podem influenciar as restrições que garantem tal otimização e, conseqüentemente, as aplicações de controle.

Com o intuito de demonstrar a definição do conceito modularização, a Figura 2.3 apresenta uma estrutura de automação sendo reconstruída, quando diferentes informações

presentes nos diversos níveis da hierarquia clássica passam a fazer parte de um único arcabouço conceitual, capaz de ser acessado de forma independente por diversos setores ou sistemas modularizados, respeitando as métricas de segurança e controle que têm como função garantir a execução adequada dos processos (COLOMBO; BANGEMANN; KARNOUSKOS, 2013).

Figura 2.3 Reconstrução da pirâmide da automação.



Fonte: Adaptado de Colombo, Bangemann e Karnouskos (2013).

2.2 CONTEXTUALIZAÇÃO DA INDÚSTRIA 4.0

Neste cenário de mudanças, é necessário que pequenas e grandes corporações atualizem os seus sistemas e aproximem seus colaboradores para os novos conceitos. Com estas necessidades, iniciativas de programas estruturantes para enfrentar os desafios da 4ª revolução industrial vêm sendo tomadas pelos governos de vários países, a exemplo dos EUA (SCIENCE; COUNCIL, 2012), Alemanha (RESEARCH, 2013), China (Aileen Jin, 2015) e Brasil (MDIC and MCTIC, 2016). Tais propostas têm em comum o uso de novas tecnologias com o emprego da internet para intercâmbio de informações, planejando permitir a comunicação entre um número ilimitado de pessoas e dispositivos.

Em janeiro de 2016, o tema Quarta Revolução Industrial foi a principal questão do Fórum Econômico Mundial na cidade de Davos (Suíça) e diversos países participantes, principalmente os do continente Europeu, discutiram sobre o avanço e os efeitos que

a nova revolução provocará no mundo, além de definir como nomenclatura para essa revolução, o termo Indústria 4.0.

O termo “Indústria 4.0”, “smart factory”, “intelligent factory”, “factory of the future” são termos que descrevem uma visão do que será uma fábrica no futuro (MACKENZIE, 2016). Nesta dissertação, adotamos o acrônimo I4 para designar o conceito. Nesta perspectiva, a fábrica terá um maior poder computacional, os sistemas ciberfísicos processarão algoritmos inteligentes para tornar os sistemas flexíveis e terão a capacidade de tomar ações para atender a diferentes clientes. Para que tais ações possam ser executadas, é necessário que a indústria cumpra uma série de princípios básicos. Hermann, Pentek e Otto (2016) trazem seis princípios base para a criação de um ambiente da I4 e, adicionalmente, esses visam suportar a sua criação de aplicações as quais podem ser inseridas nesse novo contexto.

- Capacidade em Tempo Real - os sistemas devem ser capazes de prover mecanismos que permitam coletar e analisar os dados no momento em que estes são gerados, da mesma forma que deve ser possível, baseado em regras pré-estabelecidas, tomar decisões com reflexo imediato. Além da aplicação de dados em tempo real para tomada de decisão, utilizam-se essas informações para entender o comportamento do processo e seu desempenho. Como exemplo de aplicação deste princípio, Bauters et al. (2018) utiliza câmeras conectadas via IoT e interligadas a um sistema de análises, para medir o desempenho do trabalhador e melhorar os ciclos de trabalho realizados em um sistema flexível de montagem manual.
- Orientação a Serviço - utilização de uma arquitetura de *software* que permita construir aplicações que sejam disponibilizadas e utilizadas como serviços, aliadas aos conceitos de IoS. Em resumo, este princípio tem por objetivo criar uma arquitetura que permita que outros equipamentos/sistemas da rede acessem e utilizem as funções disponíveis. Com isto, ocorre a integração entre as diferentes camadas de *softwares* de um mesmo ambiente.
- Modularidade - os sistemas que são modulares têm a capacidade de serem adaptáveis a mudanças de requisitos, substituindo ou criando módulos individuais.
- Descentralização - tem por objetivo fazer com as decisões possam ser tomadas de forma distribuída por equipamentos locais ou remotos, de acordo com as necessidades da produção em tempo real, sem a intervenção humana.

- Interoperabilidade - é a capacidade de um determinado sistema se comunicar com diversos outros, utilizando a internet das coisas e dos serviços.
- Virtualização - é a criação de uma cópia virtual da fábrica, a fim de permitir a simulação e a avaliação dos processos de todos os setores sem impactar produção.

2.3 TECNOLOGIAS HABILITADORAS

A quarta revolução ou I4.0 está concentrada na melhoria dos processos e ambientes industriais. Este novo período é impulsionado por uma série de novas tecnologias que permitem mudanças significativas na estrutura tradicional de uma fábrica. Tais tecnologias, quando implantadas, alteram o funcionamento dos processos, permitindo que um sistema fabril tenha independência de execução. Outra característica é a sua capacidade de criar uma conectividade entre os equipamentos, setores administrativos e sistemas de produção. A união eficaz destas tecnologias permite que seja possível criar uma gestão integrada das diferentes operações presentes em uma fábrica (SILVA; FILHO; MIYAGI, 2015).

2.3.1 Internet das Coisas

No que tange a internet das coisas, pode-se afirmar que um dos objetivos é conectar objetos físicos, ambientes e equipamentos à rede mundial de computadores. O termo IoT foi criado por Kevin Ashton, em 1999, e a sua primeira aplicação foi na cadeia de suprimentos para a identificação por rádio frequência (RFID) (COELHO, 2016). Tendo seu uso alavancado pelo surgimento do protocolo de internet versão 6 (IPV6), que ampliou consideravelmente, a quantidade possível de pontos de conexão IP (internet *protocol*), possibilitando a inserção de mais equipamentos na rede (FEREIRA et al., 2017).

De acordo com Research (2013), a integração entre internet das coisas e dos serviços nas indústrias iniciou a quarta revolução industrial. Essa integração permitiu que sistemas e dispositivos interagissem em rede, de modo que as informações alcançassem qualquer ambiente da indústria ou do mundo. Algumas vantagens dessa integração são: melhoria dos resultados da indústria com redução de custos operacionais, aumento da produtividade, abertura de mercados e melhoria dos produtos. Apesar das vantagens dessa integração, há alguns desafios a serem vencidos, para que seu uso seja pleno no ambiente industrial. Um deles é, segundo o MDIC and MCTIC (2016), o uso de padrões abertos na internet das coisas. Isso é necessário para que sistemas ciberfísicos e dispositivos se comuniquem entre si de forma interoperável.

A grande quantidade de dados gerados e manipulados (*Big Data*) representa outro

grande desafio, pois exige técnicas específicas que garantam a segurança, a velocidade de tratamento de dados e seu armazenamento de forma estruturada (ZHOU; YAO; ZHANG, 2017). Desenvolvimento de *middleware*, ou seja, os sistemas que promovem a interconexão de diferentes serviços, interfaceando protocolos e tecnologias, promovendo a autenticação e a identificação das diversas entidades presentes na rede, o que também representa um desafio para o uso de IoT. Os *middlewares* devem executar seu papel com um consumo controlado de recursos computacionais. Como citado por Gilchrist (2016), os *middlewares* fornecem serviços adicionais para diversos níveis de *software* e *hardware*, correlacionam dados para processamento e armazenamento e possuem papel indispensável no provisionamento e na atualização de tarefas de forma remota. Por exemplo, em apenas 30 minutos de funcionamento dos motores de um voo da Boeing, 10 *terabytes* de dados são produzidos.

2.3.2 Internet dos Serviços

Atualmente, os serviços evoluem cada vez mais e não se limitam pela simples tarefa de substituir pessoas, que executam trabalhos repetitivos, por sistemas e máquinas ((CORDEIRO; RANCHORDAS; SHISHKOV, 2011), pg.3). A quarta revolução trouxe a necessidade de que modelos, arquiteturas e tecnologias orientados a serviços, que antes eram amplamente utilizados por profissionais de TI (Tecnologia da Informação), se adequassem ao ambiente industrial e ao da automação.

Para Hermann, Pentek e Otto (2016), a IoS tem como objetivo disponibilizar as ações da cadeia de fornecedores, bem como as diversas formas de comunicação e interação entre os diversos usuários e consumidores, através de canais específicos na internet. Este novo modelo de negócio permite uma variação dinâmica da distribuição das atividades da cadeia de valor individual: aplicações de *software*, plataformas de programação, visualização de dados e soluções para infraestrutura (servidores, armazenamento e dados na nuvem) são exemplos das aplicações comuns de IoS já disseminados pela I4. A possibilidade de disponibilizar serviços na internet tem como benefício direto a redução de tempo para troca de informações e consequente tomada de decisões sobre os processos.

Estas modificações, aliadas com a adoção de computação ubíqua e inteligência artificial, permitem a criação de serviços inteligentes nas empresas, com impacto direto nos alicerces que apoiam a sua competitividade. Entretanto a adoção de IoS na estrutura de produção implica mudanças nas infraestruturas de segurança da informação e de armazenamento dos dados para todos os envolvidos no processo (cadeia de fornecedores, usuários, clientes), isso pode representar grandes investimentos em treinamento, reestruturação de

hardware/software e até no comportamento dos envolvidos e no posicionamento no mercado.

2.3.3 Big data

O conceito de *Big Data* se refere a grandes quantidades de dados que são armazenados e processados (COELHO, 2016). As tecnologias IoT e IoS inseriram em uma arquitetura comum diferentes equipamentos, sistemas e pessoas. Isto permitiu que informações adquiridas em tempo real controlassem o processo, com o intuito de otimizar sua execução através da tomada de decisão em tempo real, porém a quantidade de dados gerados cresce exponencialmente. Estima-se que, atualmente, existem 6 bilhões de celulares em utilização, e esta é apenas uma das diversas fontes de geração de dados no mundo. É muito comum que companhias de tecnologia armazenem petabytes de informações, e como os dados são gerados por humanos e máquinas concorrentemente, estas companhias sempre necessitam aumentar seus *datacenters* e construir arquiteturas que sejam capazes tanto de armazenar essas informações, quanto de avaliar todo esse volume de forma rápida, eficaz e sem que o investimento em infraestrutura tenha que crescer na mesma velocidade que o acúmulo de dados (LUO, 2017). Porém existe uma grande dificuldade de equalizar estes fatores, uma vez que cada pessoa é um gerador de dados ambulante e existe uma grande dificuldade de coletar toda essa gama de dados e extrair uma grande quantidade de informações úteis (BRYNJOLFSSON; MCAFEE, 2012).

Embora tenha o grande volume de dados manipulados como sua principal característica, *Big Data* também está associado à capacidade de extrair informações de um volume de dados inicialmente desorganizados e o valor não está bem definido. Volume, Velocidade, Variedade, Veracidade e Valor, também conhecido como os 4V's, são as 5 maiores características dos processos de *Big Data*. Enquanto os 3 primeiros têm relação direta com a aquisição de dados, a veracidade representa o grau de confiabilidade que se pode atribuir ao dado coletado e valor ao resultado que sua manipulação pode trazer para o processo (LUO, 2017).

Para Carvalho e Bernardino (2017), as empresas atuais têm um grande desafio que é como utilizar esta grande quantidade de dados gerada para se tornarem mais competitivas. Em resumo, os setores industriais precisarão de profissionais capacitados para lidar com os problemas que grandes volumes de dados irão gerar e com isto, utilizar os resultados para um enriquecimento do processo.

2.3.4 Sistemas Ciberfísicos

No que tange o sistema Ciberfísico (CPS), pode-se afirmar que este é um sistema multidimensional e complexo que combina computação, rede e ambiente físico (KHAITAN; MCCALLEY, 2015). Estes sistemas possuem a maior parte dos seus elementos constituídos por *hardware* embarcado, que são utilizados para o monitoramento dos processos físicos (GARAY, 2012).

Tais sistemas são construídos em três bases, computação, comunicação e controle. Conforme apresentado por Liu et al. (2017), estas bases são integradas, de forma que haja uma colaboração profunda entre os elementos participantes, a fim de construir sistemas confiáveis, eficientes e inteligentes.

O desenvolvimento de um sistema Ciberfísico está baseado na integração de sensores com comunicação em rede. A partir destes elementos, é possível criar sistemas computacionais colaborativos, com um alto grau de automação, com capacidade cibernética em todos os componentes físicos e obter reorganização/reconfiguração dinâmica e comunicação em múltipla escala (KHAITAN; MCCALLEY, 2015).

Para Liu et al. (2017), as características que um CPS deve possuir são: confiabilidade, concorrência, execução em tempo real, capacidade de processamento de dados, distribuição, autonomia, segurança, reorganização dinâmica e heterogeneidade. Com estas características, é possível alcançar vantagens na produção de bens e serviços na indústria.

2.3.5 Fábrica inteligente

De acordo com Research (2013), o IoT e IoS tornam possível a criação de redes que incorporam o processo de fabricação para transformar os ambientes fabris tradicionais em ambientes que disponibilizam a informação e distribuem os serviços de forma controlada. A fabricação inteligente, no contexto da I4, pode ser definida como uma fábrica na qual CPS se comunica através da IoT e usa o IoS para auxiliar pessoas e máquinas com a execução de tarefas (HERMANN; PENTTEK; OTTO, 2016). Dessa forma, a produção poderá ser automatizada e autônoma, com a mínima intervenção humana e com suas informações sendo transmitidas em tempo real.

2.4 PADRÕES PARA A INTEGRAÇÃO E COMUNICAÇÃO

Os sistemas industriais possuem uma heterogeneidade de tecnologias para a comunicação dos diferentes setores, equipamentos e processos. O principal motivo desta heteroge-

neidade é a tentativa dos fornecedores em impor padrões proprietários para gerar dependência e, desta forma, garantir um fluxo constante de vendas em um determinado segmento de mercado. Outros geradores de heterogeneidades de tecnologias de comunicação na indústria são os problemas de compatibilidade entre os sistemas com equipamentos legados presentes no mesmo ambiente que sistemas modernos. Esses motivos criaram novos requisitos para o desenvolvimento de tecnologias, de forma a solucionar problemas que ocorrem na integração dos sistemas, como utilização de tecnologias elaboradas sob o modelo de desenvolvimento *Open Source*.

Para criar um cenário compatível com a I4, é necessário que os diferentes sistemas e equipamentos se comuniquem de forma padronizada e interoperável. Chen e Doumeingts (2008) definem que integração é o processo de garantir o relacionamento entre entidades necessárias para alcançar um ou mais objetivos, podendo ser dividida em três níveis:

- Integração de negócios: coordenação de funções para gerenciar, controlar e monitorar os processos de negócios.
- Integração física: interconexão entre dispositivos através de redes;
- Integração de aplicativos: ocorre entre aplicativos, sistemas e banco de dados;

Um princípio de projeto fundamental para integração de sistemas é garantir que um determinado tipo serviço seja fornecido por apenas um provedor de serviço. Este princípio auxilia a modularização e diminui o acoplamento dos diferentes setores. Porém individualizar completamente todos os serviços executados dentro de uma companhia representa um desafio associado aos problemas típicos da heterogeneidade citada anteriormente (SADTLER et al., 2006).

A modularização é um princípio importante na I4. Por este motivo, o SOA vem sendo utilizado para resolver problemas de integração, que ocorrem pela heterogeneidade entre sistemas legados que precisam se comunicar, por processos que deverão ser flexibilizados para a correta implantação de IoT e por modelar os processos que serão integrados (MARKS; BELL, 2006). A implementação do SOA surge com o objetivo de encontrar um modelo de arquitetura ampla, que permita a integração entre os módulos de negócios, dispositivos e aplicativos.

O modelo SOA é um arquétipo para criar serviços em um sistema (ERL, 2008). De acordo com Eliasson et al. (2013), o SOA é uma abordagem utilizada para desenvolver e gerenciar sistemas de maneira distribuída, cujas funções são separadas em blocos autônomos definidos como serviços. Neste contexto, serviço é uma unidade de *software*

embarcado que tem uma lógica de controle ou de funções que respondem a solicitações específicas (ver configuração apresentada na Figura 2.4).

O SOA funciona baseado na oferta, busca e uso de serviços na rede. Segundo Mendes et al. (2008), esta arquitetura fornece uma plataforma de comunicação que permite a compatibilidade em sistemas heterogêneos, utilizando o princípio do baixo acoplamento com a interoperabilidade dos serviços.

Figura 2.4 Comunicação na arquitetura orientada a serviços.



Fonte: Baseada em Mendes et al. (2008).

O SOA não é uma arquitetura concreta, mas um conceito de arquitetura corporativa que se utiliza de um conjunto de serviços embarcados em interfaces tecnológicas específicas. Essas interfaces e serviços servem para promover a integração entre as diferentes estratégias dos setores de uma empresa e as tecnologias responsáveis por garantir a execução dessas estratégias em um determinado domínio. Isto significa que a implementação de SOA em um ambiente de produção tem como objetivo principal promover a criação de aplicações flexíveis e modulares para troca unificada de informações e, conseqüentemente, garantir que os setores responsáveis pela gestão estratégica do negócio possam executar suas ações, dispondo dos recursos tecnológicos necessários e presentes na empresa.

Além dessa integração entre a gestão estratégica e a estrutura tecnológica, Erl (2008) explica que o SOA também padroniza a comunicação (contrato de serviço), diminui a dependência e promove a interoperabilidade entre os serviços (baixo acoplamento), possibilita a modularização de serviços, facilitando, dessa forma, a reutilização, abstração,

descoberta e composição. Por fim, o excesso de informações pode prejudicar a disponibilidade de um serviço, portanto esses devem ser projetados para que permaneçam ativados apenas quando necessários (*Stateless*). Esses princípios definem o funcionamento de SOA e demonstram que esse é um paradigma capaz de construir sistemas mais duráveis (COLOMBO; BANGEMANN; KARNOUSKOS, 2013), (KOMODA, 2006).

Apesar das inúmeras vantagens, a adoção de uma arquitetura SOA pode ser visualizada como um grande desafio. De modo geral, sua implementação é mais evolucionária do que revolucionária, e por esse motivo, o processo tem início, mas não tem um fim bem estabelecido. Os desafios da implementação do SOA podem ser divididos em três grandes pilares: pessoas, processos e tecnologias. Todos têm como dilema a correta escolha de quando e por onde iniciar a sua implementação, pois os setores terão seus processos modificados, os agentes deverão modificar seus comportamentos e novas tecnologias devem ser adquiridas. Normalmente, qualquer modificação envolve custos de treinamento e deve ser adaptada aos processos existentes, de forma a impactar minimamente o seu funcionamento.

A adoção de SOA causa impacto nas pessoas e tecnologias envolvidas nos processos, pois inicialmente exige uma conscientização da necessidade de adoção de novos papéis, modificação de processos existentes, adoção de novos padrões de comunicação e segurança e implantação de novos procedimentos com novos ambientes. Por vezes, a falta de um entendimento mais profundo dos artefatos tecnológicos a serem adotados, bem como um conhecimento prévio dos sistemas legados presentes na empresa, pode criar uma resistência à mudança capaz de comprometer a implantação de toda a arquitetura (KNUTSSON; GLENNOW, 2015).

Diante deste contexto, é necessário construir um ambiente industrial integrado e compatível com a nova revolução industrial (ZARTE et al., 2016). Como mecanismos para criação de ambientes que utilizem os princípios de SOA, são selecionados por esta dissertação as tecnologias *web service* e OPC UA. O uso dessas tecnologias se justifica por serem formas de implementação do SOA, capazes de integrar dispositivos de automação legados a novos sistemas, e também de superar a heterogeneidade de *hardwares* e *softwares* que coexistem no ambiente industrial.

As tecnologias *Web* deixam de ser apenas um compartilhador de informações e passam a cooperar com o SOA, de forma a permitir o compartilhamento e a operação remota dos serviços (MARKS; BELL, 2006). É comum perceber a importância das tecnologias capazes de distribuir os diversos serviços em diferentes ambientes para diferentes usuários que, por sua vez, podem utilizar diferentes dispositivos de conexão e diferentes padrões

de comunicação, desde que todos estejam, sob algum aspecto, compatíveis com o modelo de comunicação. Por esse motivo, foi desenvolvido um padrão de comunicação e conectividade capaz de interligar dispositivos/*hardwares* com os *softwares* do chão de fábrica, denominado OPC UA. O uso desse padrão, aliado à tecnologia *web service*, permite que informações estejam disponíveis em diversos ambientes de uma corporação.

2.4.1 Web Services

De forma geral, uma orientação a serviços é um paradigma de projetos que tem como objetivo a criação de unidades lógicas de solução compostas individualmente, porém que podem ser utilizadas coletivamente como apoio à realização de objetivos estratégicos específicos. Como descrito anteriormente, o serviço é um conjunto de competências que possui um contexto funcional próprio definido por regras específicas, expressas sob a forma de um contrato de serviço (ERL, 2008), (KONSTANTINOV, 2014). Neste contexto, um contrato de serviço compreende as definições (normalmente, publicadas na forma de um documento escrito) que descrevem o serviço, sua funcionalidade e características técnicas de interface e utilização. Quando um serviço é descrito com o intuito de funcionar através de uma rede de computadores baseada na *web* (internet ou intranet) cria-se um *web service*.

Os *web services* são sistemas utilizados para a integração e comunicação de diferentes aplicações, seu surgimento foi devido à necessidade de uma comunicação interoperável entre sistemas modulares e distribuídos.

O *World Wide Web Consortium* (W3C) é uma comunidade internacional responsável por organizar e desenvolver os padrões *Web*, e tem como objetivo desenvolver protocolos que garantam a popularização e o crescimento desta tecnologia. No tocante a *web service*, a W3C descreve e desenvolve as características que tais sistemas necessitam para a manter a sua funcionalidade e comportamento, frente ao ambiente heterogêneo em que estes estão inseridos. A W3C também garante a neutralidade de fornecedor, ou seja, diferentes fornecedores podem desenvolver e garantir o funcionamento de seus próprios sistemas *web*, desde que respeitem os elementos que descrevem fundamentais da arquitetura: Conceito, relacionamento e modelos (BOOTH et al., 2004).

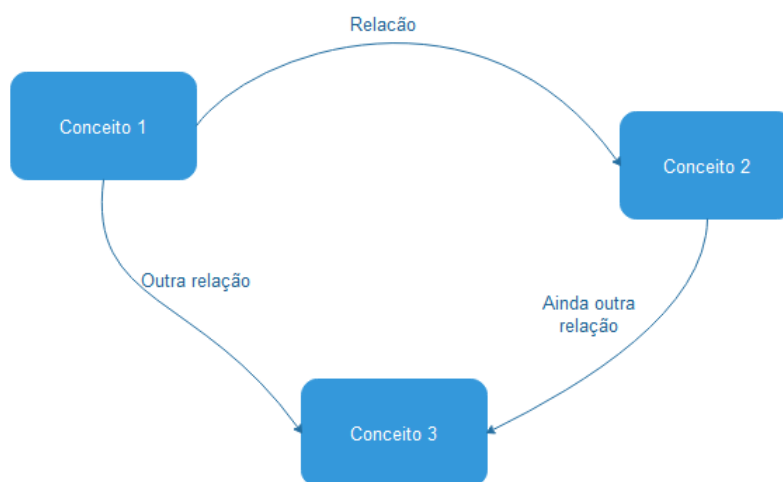
- Conceito - qualquer entidade que esteja envolvida na execução de um serviço, dentro do contexto de *web service*, possui um conceito que a classifica e a diferencia dos outros envolvidos. Conceito é uma característica abstrata associada aos objetos envolvidos na realização de um serviço, serve de suporte às ações que esses objetos

executam para alcançar seus objetivos e como eles devem se relacionar com os outros objetos presentes no processo. Por este motivo, o conceito de um objeto em um determinado *web service*, pode se diferenciar do conceito de um mesmo objeto, ou tipo de objeto, presente em outro *web service*. Por exemplo, o objeto mensagem tem um conceito claro: uma unidade básica de informação enviada de um agente para outro, ainda que seu conteúdo só passe a ter significado completo após a análise de várias outras mensagens (situação comum em que uma mensagem faz parte de uma sequência de mensagem) e que suas necessidades de relacionamento e validação sejam diferentes em diferentes *web services*. No tocante ao conceito do objeto mensagem, fica claro o seu relacionamento entre os objetos agentes (agente que envia e agente que recebe a mensagem). Em algumas situações, os objetos envolvidos em um *web service* não possuem seu conceito tão claro, por exemplo, as pessoas e as organizações, e desta forma, a descrição de seus conceitos e seus relacionamentos também pode não ter uma descrição clara, sendo descrita a partir de suas diversas atribuições, de seus relacionamentos e modelos.

- Relacionamentos – expressa a forma com que os conceitos se relacionam, ou seja, a troca de dados entre os objetos envolvidos no *web service*. Relacionamento tem uma ligação direta com os conceitos dos objetos, pois, como descrito anteriormente, dois objetos, ainda que semelhantes conceitualmente, podem ter diferenças que serão expressas em seus relacionamentos com os outros objetos. Como suporte a esse conceito, normalmente, é criado um mapa conceitual, que é a disposição dos conceitos e relacionamentos de forma gráfica, a fim de demonstrar de forma clara não apenas os objetos e suas realizações, mas também a particularidades de seus relacionamentos. A Figura 2.5 demonstra a relação entre os diferentes conceitos de um *web service*.
- Modelos – as descrições de conceito e de relacionamento mencionadas anteriormente necessitam ser organizadas, de forma a separar os aspectos particulares dos objetos presentes na arquitetura. Esta reunião de características deve ser capaz de permitir que todos os interessados nos conceitos e nos relacionamentos possam interagir entre esses, de forma organizada e segura. Embora os modelos possam compartilhar conceitos, a sua função principal é encapsular e expor aspectos significativos dentro da arquitetura *web service*. Modelo orientado a mensagens, por exemplo, tem como papel descrever o *web service* a partir da perspectiva da troca de mensagens, enquanto o modelo orientado a serviço foca nos conceitos fundamentais que garantem

a execução dos serviços. De uma forma geral, o W3C estabelece um meta-modelo da arquitetura *web service* da seguinte forma: Modelo Orientado a políticas, Modelo Orientado a Recursos, Modelo Orientado a mensagens e Modelo Orientado a serviços.

Figura 2.5 Mapa conceitual.



Fonte: Baseada em Booth et al. (2004).

Para o desenvolvimento dos conceitos relacionados aos *web service*, existem diferentes padrões. Lecheta (2015) lista os padrões mais conhecidos e utilizados, que são os desenvolvidos em SOAP ou REST.

A) SOAP

SOAP (*Simple Object Access Protocol*) é um protocolo para troca de dados e informações estruturadas em sistemas descentralizados e distribuídos. Este padrão de comunicação é especificado e mantido pela W3C (BOOTH et al., 2004). O SOAP utiliza a linguagem de marcação extensível (XML - *Extensible Markup Language*) como formato da troca de mensagem de comunicação e foi projetado para chamar aplicações remotas através de RPC ou troca de mensagens, através de uma abordagem que seja independente da plataforma de comunicação e linguagem de programação (LECHETA, 2015).

A arquitetura de comunicação baseada em requisições e respostas SOAP segue uma estrutura que tem a interação entre três partes: o servidor, o cliente e o registro. O registro define uma WSDL que é uma linguagem de descrição de serviços para serviços

web e publica no seu registro ou disponibiliza para os consumidores. O cliente busca no registro a descrição do serviço para se conectar no servidor e através disto, pode interagir com o serviço. A Figura 2.6 ilustra a interação entre estas diferentes entidades.

Figura 2.6 Arquitetura SOAP



Fonte: Baseada em Booth et al. (2004).

Paik et al. (2017) define que uma mensagem SOAP é um documento XML comum, contendo as seguintes partes:

- Elemento *Envelope* - determina o início/fim da mensagem e como o xml será transformado em uma mensagem SOAP e como será traduzida pelo *web service*.
- Elemento *Header* - este é opcional e contém todos os atributos da mensagem, normalmente, nele vão informações relativas ao *web service* que irá receber a mensagem.
- Elemento *Body* - este contém as informações que serão enviadas e a definição das falhas que podem ocorrer na transmissão.
- Elemento *Fault* - este elemento é opcional e nele pode conter códigos e mensagens erro que podem ocorrer na transmissão e na recepção de documentos.

B) REST

Web services baseados no padrão REST (*Representational State Transfer*), tem sido uma das abordagens mais utilizadas no mercado, pois os seus formatos de mensagens são mais leves e enxutos para implementação (LECHETA, 2015). O REST não restringe o formato da mensagem, apenas o comportamento dos elementos envolvidos. Gr e Pfrommer (2015), explica que o REST tem como uma das suas características a adaptação do formato da mensagem para melhor funcionamento do sistema, sendo os mais usuais o XML e JSON (*Java Script ObjectNotation*).

Paik et al. (2017), afirma que este estilo ainda não possui um padrão oficial, mas existe um conjunto de conceitos-chave para a construção de serviços *web*. O REST surgiu de uma tese de doutorado e vem ganhando espaço no desenvolvimento de sistemas, devido à sua mensagem ser mais leve, gerando um desempenho superior ao SOAP. Sua implementação permite o desenvolvimento e a criação de aplicações em sistemas embarcados, *web services* locais, aplicações na nuvem etc.. A nova revolução industrial trouxe uma série de novas tecnologias para distribuir sistemas de controle e monitoramento, devido a isto, utilizar de padrões que permitam a troca rápida de mensagens tem sido o objetivo de muitas pesquisas.

A comunicação via REST utiliza mensagens via HTTP (Protocolo de Transferência de Hipertexto) com os métodos *GET*, *POST*, *PUT* e *DELETE* para se comunicar e não necessita criar novos protocolos e códigos. Lecheta (2015) define estes métodos.

- GET - Obter algum recurso ou alguns recursos;
- POST - É utilizado para enviar dados ao *web service*, com objetivo de inserir informações;
- PUT - Atualiza registros ou informações;
- DELETE - É utilizado para excluir algum registro.

O protocolo REST não possui estado entre a troca de dados, ou seja, cada comunicação realizada é independente e padronizada. Isto faz com que o sistema implementado neste padrão seja escalável, pois não necessita de armazenar os dados em cache (PAIK et al., 2017).

2.4.2 OPC

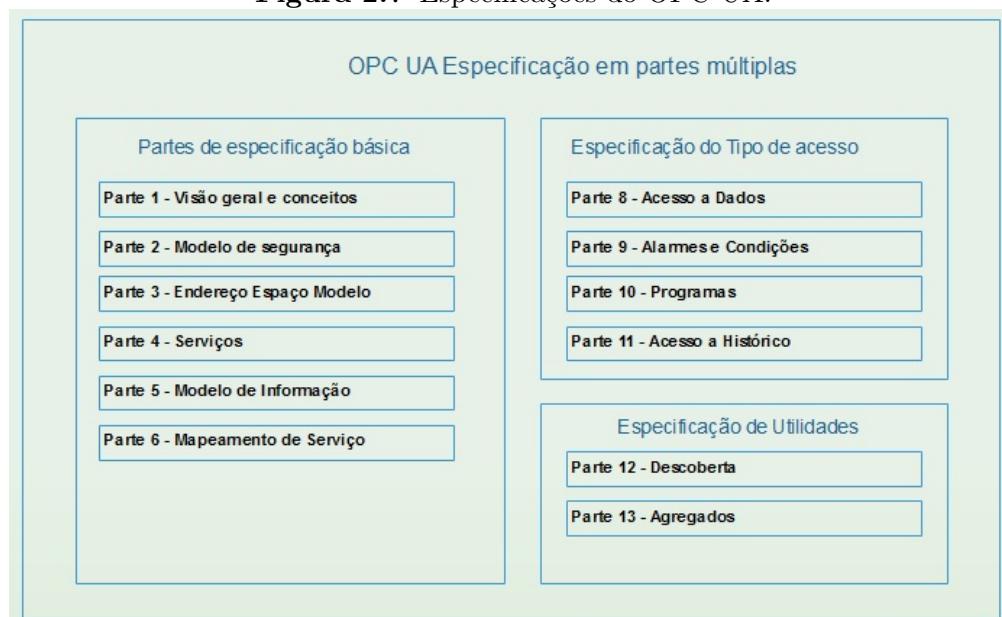
O aumento da demanda por comunicação, monitoramento e controle dos diversos equipamentos em um ou mais ambientes industriais fez surgir diferentes *drivers* de dispositivos. O excesso de *drivers* de dispositivos gerou a necessidade de interoperabilidade e padronização que facilitassem a integração entre equipamentos e sistemas. Para amenizar esta situação, diferentes fabricantes de tecnologia e usuários se uniram para criar um padrão de comunicação interoperável que permitisse a troca segura e confiável dos dados. Nesta união, é criada uma fundação denominada *OPC Foundation*, que regulamenta e define um padrão de comunicação industrial. Baseado nos padrões já existentes *ObjectLinking*

o *Embedding (OLE)* e *Distributed Common Object Model (DCOM)*, ambos de propriedade da Microsoft, é criado o *OLE for Process Control (OPC)* (FOUNDATION, 2016a), (SCHWARZ; BORCSOK, 2013).

O OPC se tornou um grande sucesso ao entrar no mercado em 1996. Sua utilização supriu as principais necessidades relacionadas a um padrão de comunicação que integrasse dispositivos e sistemas necessários na indústria. Contudo, os *softwares* e *hardwares* vêm sendo atualizados para sistemas mais complexos, com novos requisitos como: o uso de arquiteturas unificadas, modelagem da informação e independência de sistema operacional. Devido a essas novas características e aos novos desafios que surgiram com a introdução do SOA nos sistemas de fabricação, a *OPC Foundation*, juntamente com a Comissão Internacional de Eletrotécnica (IEC – *International Electrotechnical Commission*), teve a necessidade de desenvolver um novo padrão de comunicação, o qual fosse de código aberto (*Open Source*), com arquitetura unificada e orientada a serviço denominado *Open Platform Communication Unified Architecture (OPC UA)*. Desde o início de seu desenvolvimento, o novo padrão OPC UA foi desenvolvido com o intuito de se tornar uma norma, posteriormente denominada IEC62541. Recentemente, o antigo padrão OPC foi rebatizado de OPC Clássico (FOUNDATION, 2016a; MAHNKE; LEITNER; DAMM, 2009; SCHWARZ; BORCSOK, 2013). O padrão OPC UA diferencia-se do OPC Clássico, principalmente pela presença dos seguintes requisitos (MAHNKE; LEITNER; DAMM, 2009):

- Acesso aos dados unificado: a unificação do OPC clássico (DA, HDA e AE) com OPC UA permite que um equipamento compatível com a especificação do OPC UA possa se comunicar com outro equipamento que tenha especificações do OPC clássico. Ademais, em uma única requisição é possível ter todas as informações, como: alarmes, dados e eventos.
- Mecanismos de segurança: melhores recursos de segurança, como autenticação, criptografia e auditoria. Esses recursos permitem que a segurança tenha certa flexibilidade e podem ser utilizados em qualquer nível da pirâmide.
- Múltiplas plataformas: o OPC UA pode ser desenvolvido para qualquer sistema operacional e em sistemas embarcados.
- Suporte a estruturas de dados complexos: modelagem avançada das informações e inserção de dados mais complexos.

O padrão OPC UA, como descrito na norma IEC 62541, pode ser detalhado em um conjunto de 13 especificações organizadas em 3 grupos, como o ilustrado na Figura 2.7.

Figura 2.7 Especificações do OPC UA.

Fonte: Baseada em (MAHNKE; LEITNER; DAMM, 2009).

Segundo Mahnke, Leitner e Damm (2009), a especificação parte 1 define uma visão geral sobre o padrão, enquanto a parte 2 descreve os requisitos/modelos de segurança para sua implementação, ambas não são normativas. Para o desenvolvimento de uma aplicação OPC UA é muito importante entender o funcionamento das partes 3, 4, 5 e 8:

- Parte 3 (Endereço Espaço Modelo) - esta parte descreve os blocos de construção e itens para a configuração do espaço de endereçamento do servidor OPC UA (UNIFIED et al., 2015).
- Parte 4 (Serviços) - esta especificação representa as possíveis interações entre o servidor e o cliente. Esta parte também apresenta o comportamento cliente-servidor e os dados comuns que são usados como parâmetros nos serviços (UNIFIED; SPECIFICATION; RELEASE, 2015b).
- Parte 5 (Modelo de Informação) - Esta parte descreve como o espaço de endereços OPC UA, nós e referências são usados (ARCHITECTURE; PART; RELEASE, 2015).
- Parte 6 (Mapeamento de Serviço) - esta especificação define o mapeamento dos serviços para mensagens, mecanismos de segurança aplicados às mensagens e como

os dados são transferidos entre clientes e servidores (MAHNKE; LEITNER; DAMM, 2009).

- Parte 7 (Perfis) - esta parte especifica perfis que estão disponíveis para clientes-servidores. Esses perfis fornecem serviços ou funcionalidades que são usados para certificação e testes de funcionalidade dos produtos OPC UA (FOUNDATION, 2016b).
- Parte 8 (Acesso a dados) - esta parte define os tipos de variáveis, características, como usar os dados de automação e características específicas como unidades de engenharia (UNIFIED; SPECIFICATION; RELEASE, 2015c).
- Parte 9 (Alarmes e Condições) - esta parte especifica o uso do OPC UA para que possa ter acesso a alarmes e condições. Além disso, fornece a representação da informação de alarmes e condições no espaço de endereço do padrão (UNIFIED; SPECIFICATION; RELEASE, 2015a).
- Parte 10 (Programas) - esta parte especifica um modelo de informações para definir uma base de máquinas de estados para a execução, manipulação e monitoramento de programas (UNIFIED; SPECIFICATION; RELEASE, 2015a).
- Parte 11 (Acesso a Histórico) - esta parte especifica o uso dos serviços de acesso ao histórico e como demonstrar informações sobre configuração de dados e histórico de eventos com o uso OPC UA (UNIFIED; SPECIFICATION; RELEASE, 2012).
- Parte 12 (Descoberta) - esta parte da especificação descreve como os servidores podem ser achados na rede e como um cliente pode estabelecer conexão (UNIFIED; SPECIFICATION; RELEASE, 2015b).
- Parte 13 (Agregados) - esta parte define o uso de funções agregadas para aplicações em OPC UA. Os agregados são utilizados para o acesso histórico e para o monitoramento dos valores em tempo real (MAHNKE; LEITNER; DAMM, 2009).

2.5 TRABALHOS RELACIONADOS E CONTRIBUIÇÕES

A compreensão dos conceitos mencionados até este ponto é de fundamental importância para a aplicação da proposta levantada nesta investigação. No entanto, percebe-se que uma ampla pesquisa acerca dos fundamentos do arquétipo da arquitetura orientada a serviço, juntamente com a definição dos alicerces da indústria 4.0; são tão importantes

quanto uma análise profunda sobre os trabalhos desenvolvidos, com o intuito de abordar a mesma temática e elucidar a mesma hipótese, bem como tangenciar os objetivos (geral e específicos) mencionados no presente documento. Para tanto, foi feita uma revisão das diferentes tecnologias utilizadas na indústria, bem como uma revisão da literatura em diferentes trabalhos acadêmicos, com o objetivo de aproximar o processo utilizado na nova revolução. Estes trabalhos ajudaram a fundamentar os conceitos desenvolvidos na dissertação e a melhor orientar as contribuições trazidas pela arquitetura aqui proposta.

Delsing et al. (2011) discute diferentes pontos de vista sobre a migração parcial de um sistema legado de monitoramento e controle de processos para as novas tecnologias de SOA. As suas considerações são baseadas nas alterações da pirâmide de automação considerações de *hardware/software* necessárias para garantir a migração. Com uma abordagem sobre o acesso à orquestração de serviços, à composição e ao agrupamento de serviços complexos e dispositivos heterogêneos, à criação de sistemas colaborativos baseados em uma extensa rede baseada em nuvem e à manutenção das ferramentas e tecnologias existentes, o autor descreveu um ponto de vista do topo para a base da pirâmide (*Top - Down*).

O autor cita que adoção de SOA pode causar uma sobrecarga na comunicação dos sistemas devido à quantidade de mensagens que os sensores trocam para atualizar os controladores. Ele descreveu a necessidade de distinguir os dados trocados a partir de requisições, ou sob demanda, e a transferência de dados baseada em eventos, o que foi descrito como visão da base para o topo da pirâmide (*botton - up*). Esta separação serviu para prevenir uma sobrecarga da rede devido ao uso de forma generalizada do SOA. Porém o autor também sugere uma melhor investigação na troca de mensagens na rede de sensores.

Posteriormente, Delsing et al. (2012) descreve a migração das funcionalidades de uma aplicação de um ambiente Sistema de supervisão e aquisição de dados (SCADA), integrado a um Sistema controle distribuído (DCS), para uma arquitetura baseada em serviço. O autor afirma que a utilização do padrão SOA aplicado ao controle e à automação permite que tecnologias, ferramentas e métodos possam ter sua interoperabilidade melhorada devido ao desacoplamento entre a funcionalidade e implementação. Como consequência direta, percebe-se um aumento na transparência de toda a infraestrutura, incluindo as ferramentas de desenvolvimento de sistemas e serviços. A maior contribuição da migração de um sistema de controle de processos legado para um sistema SOA é a manutenção das funcionalidades, enquanto permite uma atualização gradual, altamente integrada de uma forma estruturada, respeitando os sistemas proprietários e as estruturas abertas.

Karnouskos et al. (2012) define quais serviços básicos uma abordagem de SOA precisa para apoiar a comunicação e transformar uma fábrica em um ambiente de interação e colaboração. O autor cita a necessidade de detalhar e priorizar serviços essenciais que irão compor os sistemas futuros, dentre estes, são citados como necessidades básicas o monitoramento, o gerenciamento, o gerenciamento de dados e integração etc. Algumas considerações deste trabalho são a modularização dos diferentes processos, *softwares* e como eles interagem entre si e com os diferentes atores dentro de uma infraestrutura de automação e de gerenciamento colaborativo de camadas cruzadas, a partir de tecnologias-chaves e disruptivas.

Também é citado que uma fábrica pode ser transformada em uma “nuvem de serviços”, a partir da virtualização, ou seja, é criado um ambiente colaborativo, no qual existem uma interação e uma alocação dinâmica de recursos através de componentes-chaves de *hardware* e *software*. Isto torna o gerenciamento de uma rede de dispositivos e sistemas heterogêneos parte fundamental para a criação de uma arquitetura de larga escala, baseada em rede. Abordagem de desenvolvimento de *software* para sistemas industriais simples será modificada, a fim de contemplar o enfoque SOA, em que funcionalidades comuns podem ser agrupadas em uma plataforma de funcionalidades e tal coleção fará parte de uma arquitetura baseada em serviços.

Com a difusão da arquitetura orientada a serviços do setor industrial, alguns padrões de integração tiveram de ser modificados para se adaptarem a este novo ambiente. Para esclarecer sobre as diferenças e as evoluções do padrão OPC Clássico para OPC UA, Konstantinov (2014) apresenta as arquiteturas, descreve o funcionamento a partir de uma investigação sobre a conectividade para uma simulação em tempo de execução em sistemas de automação. Seu trabalho teve como objetivo investigar a conectividade através do padrão OPC UA e seu estudo de caso foi integração entre o modelo virtual e real de uma planta de manufatura da Festo em ambiente laboratorial. Como contribuição para o atual estado da arte, o autor descreveu uma metodologia para monitorar, gerir e simular o seu processo em execução. Desta forma, integrou os diferentes *softwares* (proprietários e livres) com diversos ambientes de produção, virtualização, denominado CCE (*Component Core Editor*) e gestão da planta. O autor concluiu que o OPC UA é uma tecnologia promissora para a integração de dispositivos e sistemas para gestão dos processos e citou que ainda existem recursos que podem ser investigados na integração entre o OPC UA e o ambiente de virtualização CCE.

Também foi feito um levantamento sobre os trabalhos que abordaram o uso de modelos para a geração de equipamentos de forma reutilizável e padronizada através de SOA

e OPC UA (HENSEL et al., 2016) e (REBOREDO; KEINERT, 2013). Estes artigos citam o controle de processo de aplicações em larga escala, a necessidade de unificação e representação de dados e topologia em SOA, ao mesmo tempo, em que embasam o uso do espaço de endereçamento do OPC UA para criação de modelos genéricos de equipamentos, com o objetivo de diminuir o tempo gasto para a criação de informações em bases de dados. Reboredo e Keinert (2013) discorre sobre a dificuldade e a importância de se obter informações dos dispositivos de campo para o nível de gerenciamento, e sugere que o paradigma SOA promete ser capaz de garantir a demanda de competitividade e de sistemas de produção flexível na manufatura discreta. Porém, salienta que SOA não pode ser aplicável em todo o dispositivo de campo e, por este motivo, apresenta um estudo sobre a integração de dispositivos incompatíveis com o padrão SOA, existentes em uma rede de dados e serviços *fieldbus*, com uma infraestrutura SOA, com a utilização de uma arquitetura do tipo OPC UA.

Embora o autor não demonstre uma implementação de execução de comandos via arquitetura OPC UA e a coloque como sugestão de trabalhos futuros, ele descreve que a plataforma de integração semântica de dispositivos de campo permite a utilização de um controlador integrado ou um dispositivo de campo integrado com o servidor OPC UA. Hensel et al. (2016) descreve o conceito de acoplamento de modelos de simulação já existentes, a fim de permitir a simulação de modelos individuais, criando um ambiente de cosimulação. Desenvolve uma análise sobre a utilização do OPC UA como uma tecnologia genérica do tipo *middleware* nos cenários descritos pela indústria 4.0. O autor demonstra a vantagem, o alto grau de flexibilização e descrição semântica em ambientes de cosimulação com a integração de OPC UA, embora com a desvantagem do baixo desempenho devido à agregação de servidor. O autor não demonstra medições referentes à sobrecarga de comunicação de rede típica da aplicação OPC UA e SOA em sistemas legados, nem analisa a integração com sistemas de automação reais.

De acordo com Girbea et al. (2014), a evolução do mundo atual faz com que a inovação remodele toda a estratégia de fabricação de uma empresa, com o objetivo de maximizar o lucro. O seu trabalho traz a proposta de uma arquitetura industrial que tem por objetivo melhorar o desempenho das fábricas. Nessa arquitetura, são utilizadas as tecnologias OPC UA e *web services*, para estruturar o sistema de ordens de serviços. O seu principal objetivo é fazer com que um pedido gerado pelo setor empresarial maximize a criação de ações automatizadas que não dependam da ação humana, criando um sistema de produção CPS. Através desta abordagem, são demonstradas as vantagens na utilização de uma arquitetura que permita a integração entre os diferentes ambientes de produção,

além de permitir o uso de uma estrutura modularizada, com regras definidas que servem para o processo não entrar em falha.

Dada a escassez de soluções livres que utilizem o padrão OPC UA no matlab, os pesquisadores Elfaham et al. (2016) desenvolveram bibliotecas em c++ com o código *open62541* para a comunicação com dispositivos de controle. Este desenvolvimento permitiu integrar o ambiente matlab com o CLP de uma célula de combustível para implementar o controle regulatório do tipo PID (Proporcional, Integral e Derivativo). Após o desenvolvimento do sistema, foi feito um estudo de caso dividido em três fases:

- A primeira corresponde aos testes, com o objetivo de identificar o modelo da planta. Nesta fase, são feitas coletas dos dados de entrada e saída, até encontrar o melhor modelo que descreva o sistema.
- A segunda fase corresponde à fase de prototipagem. Através desta etapa, os testes de operação são iniciados e o objetivo é encontrar a melhor sintonia dos ganhos do controle PID.
- A última fase corresponde à operação da planta que se subdivide em execução do controle e análise/armazenamento dos dados. Esta etapa monitora o processo, gerando informações para serem utilizadas na manutenção e rastreamento de falhas.

Como conclusão, os autores demonstraram que o padrão OPC UA pode ser utilizado para controle em malha fechada e ratificaram a viabilidade na aplicação deste padrão para a integração desta tecnologia com controle de processos.

Moraes (2017) demonstra as vantagens e os benefícios da aplicação de SOA e apresenta alguns estudos de caso com diferentes tecnologias para uma produção baseada em serviços. As tecnologias empregadas neste estudo possibilitaram a integração do chão de fábrica com os elementos da produção e não abrange outros níveis da pirâmide da automação como cadeia produtiva, logística e sistemas corporativos.

Foram apresentadas as tecnologias OPC UA, DPWS, gSoap, as quais permitiram receber/acessar informações via *web services*. O autor cita ainda que através das soluções baseadas em serviços é possível reduzir a dificuldade na integração de sistemas automatizados, gerando ganhos nos ambientes de produção, de forma a torná-los mais flexíveis. Além do benefício citado, o autor relata no seu trabalho limitações e problemas encontrados, como escassez de produtos que utilizem a abordagem de serviços, limitação na comunicação da troca de mensagens via *web services* e dificuldade em escolher a melhor solução aberta para protocolo de comunicação. Como contribuições, este trabalho deixou

considerações e metodologias para serem seguidas, as quais auxiliam a construção e a integração de serviços na presente revolução industrial.

Kim e Sung (2017) desenvolveram um *framework* para a comunicação de aplicações IoT industriais, baseadas em PLC. Com a utilização de um sistema padrão de comunicação de PLC, denominado *PLCOpen*, os autores implementaram as funções de cliente e servidor de uma rede industrial, utilizando um *software* de código aberto, capaz de dar suporte à norma IEC61131-3. Desta forma, foi possível demonstrar a conexão entre um servidor OPC UA externo com uma plataforma de *software* PLC, a qual criava uma lista de atividades recebidas, avaliava as variáveis do PLC e devolvia os resultados para que o servidor OPC UA encaminhasse para os clientes OPC UA. O sistema foi avaliado em um teste de mesa, no qual um robô cartesiano de 3 graus de liberdade (3-DOF) era controlado remotamente por um cliente OPC UA.

Sangeetha et al. (2018) demonstra a substituição de um sistema de controle em cascata *stand-alone* por um sistema de IoT embarcado em um servidor dedicado. O sistema físico apresentado pelos autores consiste em um controle de nível e fluxo, desenvolvido em um kit didático, no qual controles regulatórios PI (Proporcional integral) e PID atuam sobre uma válvula de controle vazão, a fim de manter o controle de nível e fluxo respectivamente. Os autores também desenvolveram um sistema *Web Service* e adicionaram dispositivos de rede Wi-Fi, com o intuito de permitir o controle da planta remotamente, criando, assim, uma aplicação real de IoT.

Para que fosse possível contornar os problemas causados pela latência de rede inerentes à associação do IoT com os controles clássicos, os autores implementaram um controle avançado do tipo MPC (*MPC model predictive control*) associado a uma técnica da inteligência artificial, conhecida como Método de Enxame de Partícula (PSO – *Particle Swarm Optimization*). A aplicação do MPC com o PSO foi capaz de reduzir os atrasos imprevisíveis presentes em transferência de dados via protocolo IP, conhecidos como *Feed Forward* e *Feedback*. O autor valida a aplicação, descrevendo as características do *Web Service* e afirma que foi possível resolver os problemas de segurança e configurações de vários usuários.

Os estudos descritos no presente capítulo sugerem que a integração entre os diversos serviços de automação, sistemas legados, redes de computadores industriais, novos dispositivos de IoT e novos paradigmas de arquitetura e modelos industriais, possuem uma sólida relevância na implementação da quarta revolução industrial. Percebe-se portanto, que o crescente número de publicações relacionadas à integração entre OPC UA, SOA, IoS e IoT no último ano, (CUPEK et al., 2017; FORSSSTROM; JENNEHAG, 2017; IA-

TROU; URBAS, 2017; MORGAN; O'DONNELL, 2017), bem como as necessidades de avanços capazes de consolidar os alicerces relacionados à comunicação, à integração e à operação dos processos industriais nessa nova revolução, demonstra que ainda existe uma lacuna de exploração que precisa ser preenchida. Conseqüentemente, o presente trabalho se enquadra neste contexto, com o intuito de apresentar uma maneira adequada de se obter um controle avançado de processos industriais, a partir da utilização de *web service*, SOA e OPC UA, o que permite a integração via um mediador entre *softwares* e *hardwares*.

PLATAFORMA ABERTA PARA INTEGRAÇÃO VIA WEB SERVICE

No capítulo anterior, foi apresentado o atual cenário de desenvolvimento e contextualizações relacionadas à nova revolução industrial, bem como uma análise sobre os trabalhos acadêmicos recentes, os quais contribuíram como base para a construção do conhecimento utilizado nesta pesquisa.

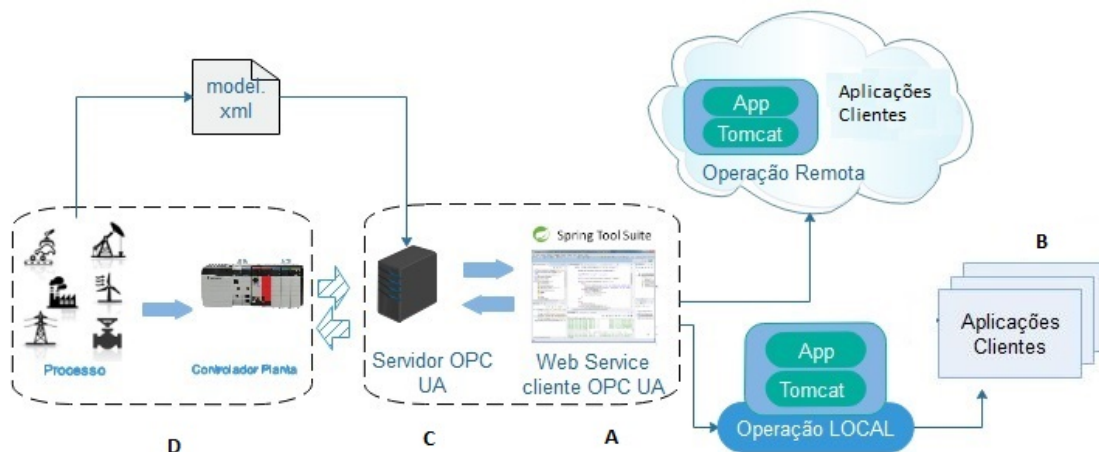
O presente capítulo tem como objetivo apresentar as soluções desenvolvidas para permitir a utilização do SOA como suporte à integração da I4, bem como demonstrar todos os métodos utilizados para alcançar tanto o objetivo principal quanto os objetivos secundários descritos na introdução deste trabalho. Com o intuito de atingir os objetivos citados, foi necessário construir uma plataforma de comunicação capaz de integrar qualquer processo, com qualquer aplicação presente em uma planta industrial, desde que o processo seja mapeável no servidor OPC UA e que a aplicação possua uma conexão do tipo cliente no mesmo servidor. Por razões óbvias de gerenciamento e segurança, também é necessário que o processo e a aplicação possuam uma relação de afinidade.

A Figura 3.1 ilustra a arquitetura da plataforma aberta para integração via *web service* (IWSOP), esta utiliza o novo padrão de comunicação o OPC UA que funciona como servidor/cliente no modo local ou remoto. No que se refere à disponibilização dos dados em rede, utilizou-se o sistema *web service* cliente OPC UA, desenvolvido nesta pesquisa e, através deste, é possível que qualquer cliente habilitado para a leitura em formatos HTTP possa acessar as informações. A comunicação realizada entre dispositivos, controladores e clientes é através da rede de comunicação Ethernet Industrial, utilizando protocolo TCP/IP.

De modo geral, as etapas necessárias para a correta utilização da plataforma estruturada através do SOA são:

- Modelar as informações no espaço de endereçamento do OPC UA. Esta modelagem é a criação de *tag's* com informações dos controladores dentro do servidor. Seguindo as especificações do OPC UA parte 3 e 6.
- Estabelecer a comunicação entre o servidor OPC UA com o *web service* cliente OPC UA. O servidor OPC UA deve ser o mesmo a ser configurado no servidor cliente OPC UA.
- Interagir com o processo mapeado.

Figura 3.1 Arquitetura IWSOP.

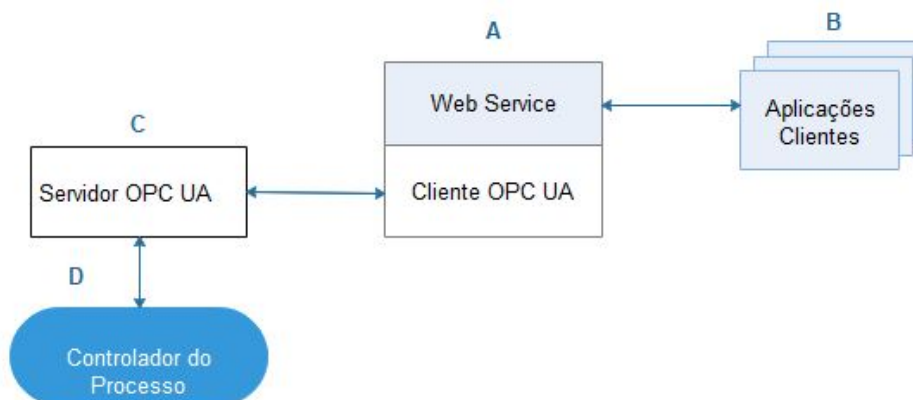


Desta forma, a plataforma desenvolvida possui as seguintes funcionalidades:

- Funcionalidade de interface entre as aplicações clientes e o servidor OPC UA.
- Caracterização da aplicação ser modularizável, de acordo com as definições do I4.
- Criação de um servidor OPC UA capaz de interligar, em sentido amplo, os dispositivos de campo e as aplicações clientes.
- Mapeamento dos diferentes processos e seus respectivos controladores dentro do servidor OPC UA, com o intuito de prover a integração com as aplicações clientes.

A proposta da plataforma é fornecer uma infraestrutura para integração e interoperabilidade que permita distribuir algumas funcionalidades e ações que são utilizadas e realizadas de forma local. Para facilitar a compreensão das etapas de desenvolvimento, a Figura 3.2 ilustra de forma genérica as funcionalidades descritas e as interações entre seus componentes.

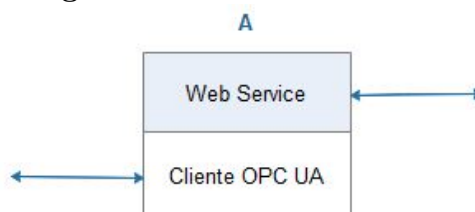
Figura 3.2 Funcionalidades da plataforma



O uso deste sistema de forma distribuída permite que sejam criados módulos com diferentes funcionalidades, como: visualização, controle avançado e gerenciamento remoto. De uma forma simplificada, uma aplicação remota encapsula comandos (escrita e leitura) em uma mensagem que será transferida para um sistema mediador.

Este sistema tem a função de traduzir a informação recebida por uma determinada aplicação cliente para o formato do servidor OPC UA que executará as ações diretamente nos dispositivos de campo. O mediador que está ilustrado na Figura 3.3 é composto por duas partes principais: um *web service* e um cliente OPC UA encapsulado.

Figura 3.3 Mediador *Web* cliente



O sistema mediador é um *web service* que possui em sua implementação um cliente OPC UA capaz de integrar as aplicações clientes com os servidores OPC UA, bem como permitir a criação dos referidos módulos. A construção deste mediador foi baseada, principalmente nas ideias e abordagens dos trabalhos de Girbea et al. (2014), Delsing et al. (2012), Colombo, Bangemann e Karnouskos (2013).

Do trabalho de Girbea et al. (2014), foi extraída a ideia de desenvolver um sistema *web service* para criação de uma plataforma de integração entre os controladores e os diferentes módulos existentes para operação de um processo. A partir das contribuições de Colombo, Bangemann e Karnouskos (2013) e de Delsing et al. (2012), foi possível desenvolver a arquitetura orientada a serviços, para interligar os diferentes níveis da pirâmide da automação, com o objetivo de interagir com aplicações modularizadas.

O mediador de serviços foi projetado no ambiente de desenvolvimento *Spring tools suite*, o qual utiliza a linguagem de programação *java*. A utilização do *Spring* é justificada pela presença de um servidor *TomCat* de forma nativa, ou seja, a sua utilização não está condicionada à necessidade de instalações adicionais. Dentre outras características, o servidor *TomCat* foi escolhido por ser um servidor *Web* que tem a capacidade de funcionar tanto de forma independente, quanto em associação com outros servidores, dos mais diversos modelos e características.

No tocante à sua versatilidade, o servidor *TomCat* é conhecido como um contêiner de *Servlet*, uma classe especial *java* capaz de responder a qualquer tipo de requisição. Como descrito em Pivotal (2017), as principais funcionalidades do *spring* são:

- O *spring MVC* é um *framework* que auxilia no desenvolvimento de aplicações *web* robustas, flexíveis e com uma separação no tratamento da requisição. MVC é um acrônimo para modelo, visualização e controle, que representa um padrão de programação, também conhecido como programação em três camadas, nas quais são definidos os papéis de cada parte da aplicação, de forma a simplificar sua manutenção;
- Suporte para banco de dados – o *spring* possui tecnologias de conexão a bancos de dados de forma nativa, que simplificam o acesso às tecnologias de armazenamento de dados;
- Injeção de dependências - essa é uma forma de manter o baixo acoplamento entre as classes de um mesmo projeto, ou seja, é um tipo de mudança de controle, significando que uma classe não é mais responsável por criar ou buscar os objetos dos quais depende.

REST foi o padrão de comunicação utilizado pelo *framework* na construção do mediador, os critérios para a utilização deste padrão foram: facilidade de codificação, reutilização de código, vasta documentação sobre a utilização da ferramenta e realização de teste de forma simplificada. Além dessas características, foi avaliado que o REST possui o protocolo *Stateless*, um protocolo de comunicação no qual cada conexão com o servidor é considerada uma transação independente, o que diminui o tráfego de rede, uma vez que o servidor não necessita armazenar o estado de cada sessão com cada cliente envolvido.

O *Spring tool* possui ainda uma série de dependências administradas pelo Maven. Esse administrador de dependências é uma ferramenta automatizada que compila os códigos criados e adicionalmente descarrega as bibliotecas e *plugins* de um ou mais repositórios no computador ou na internet. O maven utiliza um arquivo xml que contém uma estrutura com todas as informações do *software*, como ordem de compilação, versão, diretórios necessários e bibliotecas. Este código define a estrutura de compilação e as principais bibliotecas necessárias para criar o mediador. (Apêndice A)

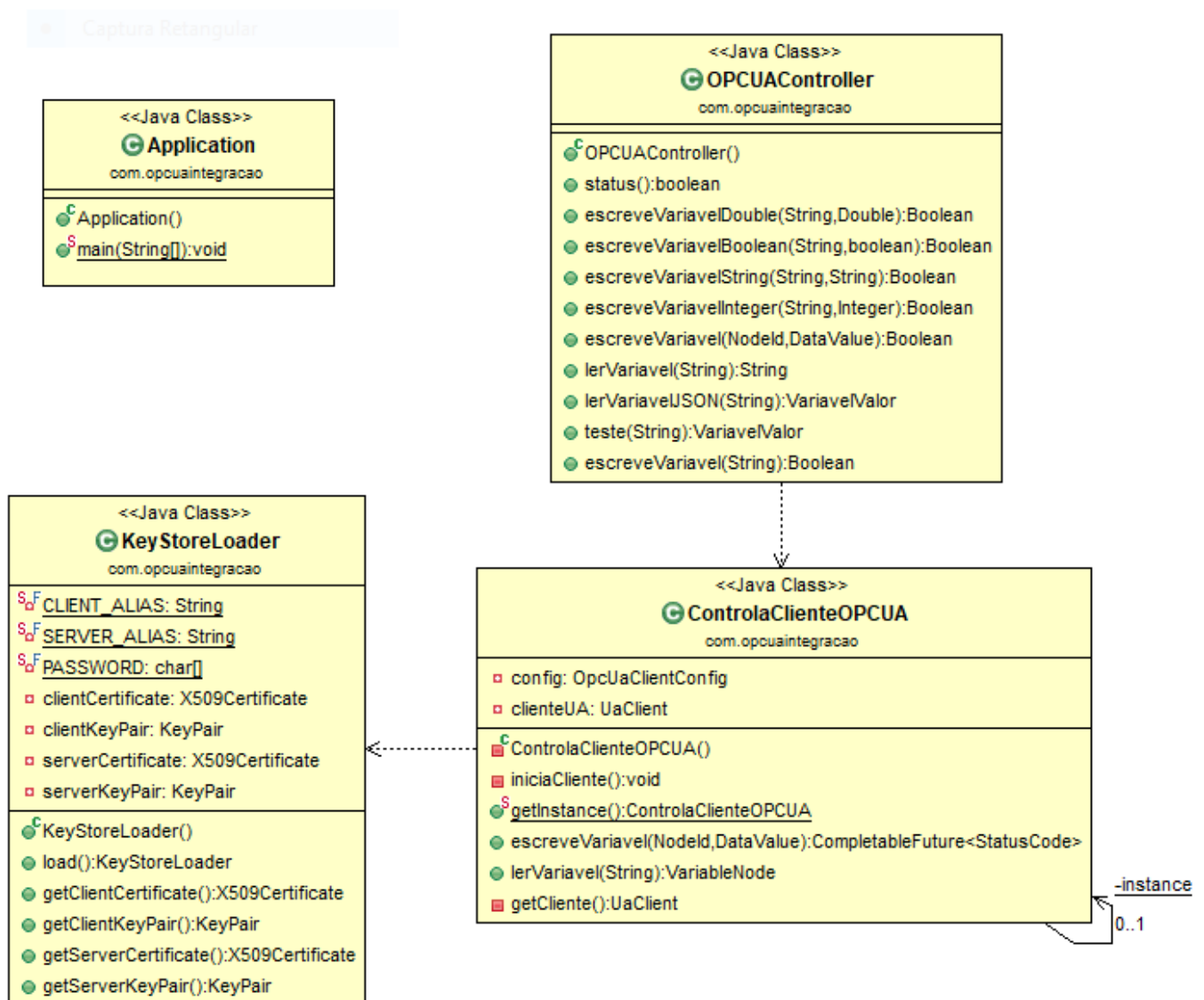
As classes foram desenvolvidas a partir do código aberto do projeto Milo, um grupo de pesquisa que desenvolve classes *java* para servidores OPC UA na plataforma eclipse. A Figura 3.4 ilustra os códigos desenvolvidos e adaptados para o sistema mediador. Este foi dividido em quatro classes:

- *Application*: classe utilizada para criar a aplicação de execução do *web service* na aplicação servidor. Esta classe é utilizada como uma classe de configuração geral.
- *OPCUAController*: classe desenvolvida para prover a comunicação entre o lado *web* com o lado OPC UA. Ela recebe requisições (*request*) em formato *string*, as converte para o formato padrão do servidor OPC UA, e as envia para a classe *ControlaClienteOPCUA*. No caminho inverso, essa classe recebe os dados do formato OPC UA, converte para *string* e envia para as diversas aplicações no formato de resposta (*response*). Utiliza-se do padrão de comunicação REST, no qual os aplicativos clientes se conectam para troca de dados. Esta classe possui dois tipos de métodos, os que permitem a leitura de dados dos dispositivos em campo e os que permitem a execução de ações nesses dispositivos. O *Web service* pode receber e enviar dados das aplicações nos formatos xml e json.
- *ControlaClienteOPCUA*: esta classe é utilizada para a conexão do servidor OPC UA com a classe *OPCUAController*. Para isto, ela inicia e finaliza a conexão do cliente dentro do servidor OPC UA. Com o propósito de prover a segurança do

processo de troca de informações, esta classe necessita de uma autenticação na classe *KeyStoreLoader*.

- *KeyStoreLoader*: esta classe é utilizada para carregar os arquivos de criptografia e dados de acesso de usuário, utilizados nas conexões com o servidor OPC UA.

Figura 3.4 Diagrama UML do aplicativo



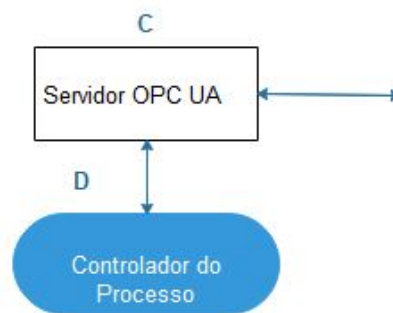
3.1 SERVIDOR OPC UA

O servidor OPC UA foi implementado a partir de dados disponibilizados pela Fundação OPC (FOUNDATION, 2016b) e pelo trabalho de pesquisa de (KONSTANTINOV, 2014). Konstantinov (2014) realizou uma investigação no padrão OPC UA para a conectividade com um sistema de simulação em tempo real.

Como parte de sua pesquisa, o autor fez uma análise de diferentes servidores de comunicação OPC UA para a conexão com CLP's do fabricante *Siemens*. De forma semelhante, o presente trabalho se utilizou de CLP's e inversores da Empresa *Rockwell Automation* para interligar os processos, executados em dispositivos de campo, ao servidor OPC UA. Os dados que devem ser monitorados constantemente são a vazão, o nível dos tanques e a velocidade da bomba.

Para que possam ser visualizados, gerenciados e executados por aplicações, os dispositivos de campo devem estar mapeados no servidor OPC UA. O termo mapear um dispositivo significa registrar o endereço de um dispositivo responsável por um determinado processo, podendo ser um sensor ou um atuador, no servidor. Desta forma, uma aplicação remota, devidamente conectada ao servidor OPC UA, consegue executar comandos específicos nos dispositivos de campo através deste endereço, por exemplo, ações de leitura e escrita. Esta ação está implícita na Figura 3.5.

Figura 3.5 Servidor OPC UA.



O *software TopServer*, desenvolvido pela empresa *Software Toolbox*, foi escolhido por ser um servidor de comunicação com um grau de versatilidade satisfatório, podendo ser personalizável por tipo de indústria (manufatura, óleo & gás e automação residencial). Este servidor de comunicação permite que os dispositivos de campo sejam integrados a diferentes clientes (como *Web services*, MES (Sistemas de Execução da Manufatura), Scada, HMI) através de uma grande lista de *drives* e também de vários componentes de

comunicação (OPC DA, OPC AE, OPC UA, DDE) presentes na sua biblioteca, como demonstrado na Figura 3.6.

Figura 3.6 Estrutura do *Software TOPServer*.

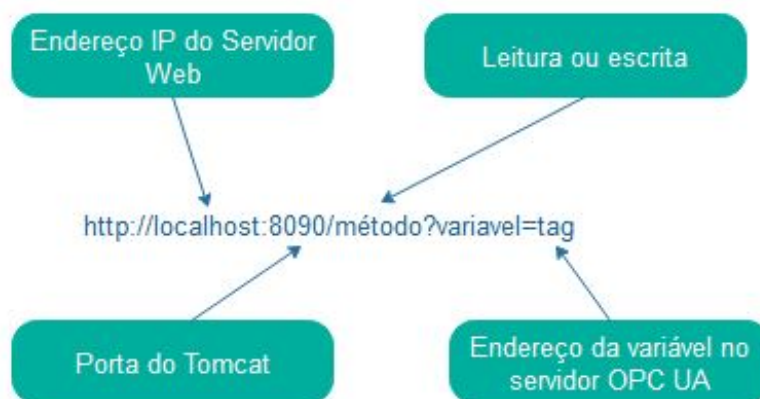


Fonte: Software Toolbox (2016)

3.2 APLICAÇÃO CLIENTE

Uma aplicação cliente é qualquer aplicativo que tem o objetivo de executar ações nos dispositivos de campo. Para isto, ela deve ser capaz de conectar o sistema mediador e enviar comandos para o servidor que gerencia o dispositivo. Desta forma, a aplicação cliente deve conhecer previamente o endereço do dispositivo de interesse, bem como as ações que esses podem executar. Para a arquitetura desenvolvida, a requisição enviada para o servidor OPC UA deve ser encapsulada em uma mensagem no formato XML ou JSON.

Uma parte importante na transferência de comandos pelo mediador é a construção do cabeçalho da mensagem. A mensagem vem com o endereço do servidor *web*, a porta que identifica o serviço, o método que representa a ação a ser executada e o endereço do dispositivo mapeado no servidor. A Figura 3.7 ilustra o cabeçalho utilizado pelos aplicativos para acesso ao dados no servidor OPC UA via o mediador.

Figura 3.7 Construção do cabeçalho

Embora exista uma gama de aplicações que podem ser implementadas como clientes, com as mais diversas funcionalidades e objetivos, no presente trabalho, a aplicação cliente desenvolvida foi um controlador avançado do tipo MPC (*Model Predictive Control*). Como definido por Campos e Teixeira (2006), controle avançado é qualquer estratégia de controle que atue sobre a camada de controle regulatório (básico) da planta. O uso de controle avançado em processos industriais está relacionado com o aumento dos benefícios já alcançados pelo controle regulatório.

Isto significa ganhos como eficiência energética, aumento na segurança e estabilidade do processo, manipulação mais eficaz nas variáveis de processo etc., já passíveis de serem obtidos pelos controles básicos, podem alcançar valores acima da otimização provida por esses controles, a partir da aplicação das estratégias de controle avançado.

As características tecnológicas presentes na I4 permitem que as técnicas de controle avançado possam ser integradas com os conceitos, como: modularização, aquisição de dados em tempo real, interoperabilidade, virtualização e orientação a serviços.

O controle MPC pertence a uma classe de controladores avançados que, a partir de um modelo pré-definido do processo, é capaz de prever o comportamento futuro de um sistema. A predição do comportamento das variáveis controladas ocorre com base em dados adquiridos em tempo real, que são analisados juntamente com um histórico de dados para prever uma sequência de comportamentos futuros até um determinado número de passos, ainda não executados, denominado horizonte de controle.

À medida que o tempo passa, dados previstos no horizonte de predição passam a ser

comparados com os estados atuais das variáveis, e caso sejam identificadas alterações, o comportamento futuro é modificado, ao mesmo tempo em que o horizonte previamente definido tem novos passos adicionados. Desta forma, o controle preditivo trabalha com o conceito de horizonte móvel (CAMACHO; ALBA, 2013; ROSSITER, 2003; MACIEJOWSKI, 2002).

A ideia de utilizar um controle avançado em uma plataforma orientada a serviços é demonstrar a eficácia da aplicação desta técnica em um ambiente modularizado, com acesso via um mediador *web service*. Outra característica que justifica sua utilização é a capacidade de transformar uma planta com tecnologia legada, em um novo ambiente de estudos integrado e distribuído.

3.3 SÍNTESE

Este capítulo apresentou o desenvolvimento da plataforma aberta para integração via *web service* para aquisição e controle dos dados e das informações. A estrutura da arquitetura está dividida em um mediador *web* cliente OPC UA, responsável por intermediar os dados de leitura e escrita que são enviados dos diferentes clientes para o servidor OPC UA, que por sua vez, mapeia os dispositivos de campo para atuar no processo selecionado.

Além disto, através desta arquitetura, os equipamentos legados foram integrados e distribuídos em uma rede de comunicação ethernet. A abordagem proposta é utilizar uma arquitetura e organizá-la para integrar o processo, cliente, indústria, com a operação, diferentes ambientes e equipamentos à luz dos princípios da indústria 4.0.

VALIDAÇÃO EXPERIMENTAL

A validação dos conceitos apresentados no desenvolvimento desta dissertação é realizada através de um estudo de caso, no qual as técnicas de SOA são aplicadas sobre uma Planta para controle e monitoramento de variáveis em diferentes tanques. Desenvolvida no Centro de Capacitação Tecnológica em Automação Industrial da UFBA (CTAI), essa planta representa uma forma didática de execução de um processo industrial de controle das seguintes variáveis: pressão, nível, vazão e temperatura.

No presente trabalho, uma aplicação cliente foi desenvolvida com o intuito de aplicar um controle avançado do tipo MPC sobre um dispositivo de campo, a fim de exercer o controle de nível de forma remota e modularizada. O desenvolvimento desta aplicação cliente teve como objetivo demonstrar, de forma sucinta, a aplicação de técnicas de SOA, bem como a utilização destas técnicas nos moldes da I4. Desta forma, este estudo de caso demonstra que o presente trabalho contempla o quarto objetivo específico definido no primeiro capítulo.

4.1 DESCRIÇÃO DA PLANTA PARA VALIDAÇÃO EXPERIMENTAL

O objetivo principal da planta é demonstrar, didaticamente, a aplicação de diferentes malhas de controle, utilizando equipamentos e ferramentas industriais, além de permitir a criação de novas abordagens nas diferentes áreas de pesquisa, como automação, redes de comunicação, controle e instrumentação.

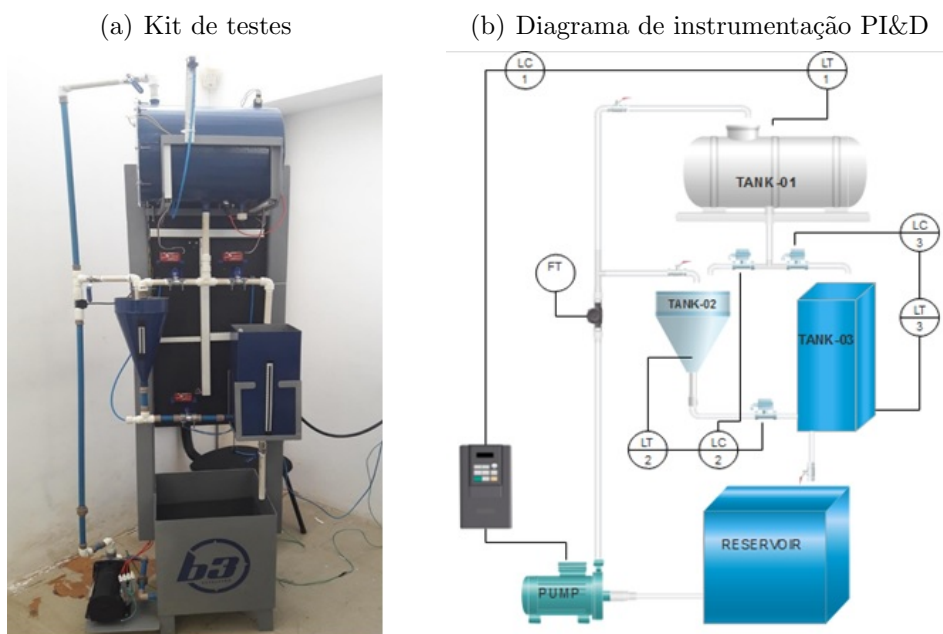
O processo em estudo busca controlar o nível dos diferentes tanques. Assim, utiliza a técnica de controle preditivo para demonstrar aplicabilidade dos conceitos base da quarta

revolução industrial: tempo real para aquisição das informações da planta, modularidade com a criação de aplicativos para monitoramento e controle, interoperabilidade para que qualquer *hardware* ou *software* orientado a serviço interaja com o sistema e descentralização dos níveis da pirâmide da automação.

O estudo de caso é constituído por três tanques: o tanque cilíndrico, retangular e cônico (ver Figura 4.1(a)). O tanque cilíndrico horizontal (tanque 1) localizado no topo da estrutura da planta, com capacidade de 28 litros, o qual foi fabricado com chapa de aço para ter robustez e durabilidade. A estrutura possui dimensões de 15cm de raio, altura de 40cm e espessura de 3mm. O tanque 1 está equipado com medidor de vazão na entrada, sensor de nível (LT-1) e na saída possui duas válvulas de controle (LC-2 e LC-3). O tanque cônico (tanque 2) possui raio de 10cm, altura de 30cm e espessura de 3mm. Este tem a capacidade de 3,1 litros e na sua estrutura tem sensores de medição de nível LT-2.

O funcionamento do tanque é através da abertura da válvula de saída do tanque 1 (LC-2). O tanque retangular (tanque 3), por sua vez, possui lado de 20cm, altura de 40cm e espessura de 3mm. Este dispõe de um volume total de 16 litros e tem seu nível controlado pela válvula de saída do tanque 1 (LC-3). O sistema é alimentado por um reservatório que tem a capacidade acima da soma dos três tanques, e está localizado na base da planta, utiliza uma bomba centrífuga trifásica controlada por inversor de frequência. A Figura 4.1(b) ilustra o esquema descrito.

Figura 4.1 Planta piloto



- Inversor de frequência da *Rockwell automation* modelo *POWERFLEX 40*;
- Controlador Lógico Programável da *Rockwell automation* modelo *Compaclogix L32E*;
- Três válvulas proporcionais tipo servo-motor;
- Três sensores de nível modelo Motorola MPX5010;

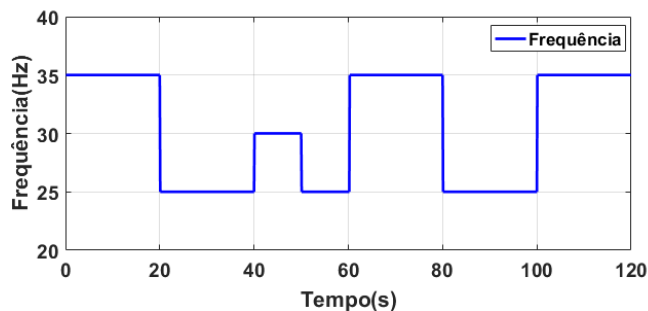
4.2 IDENTIFICAÇÃO DO MODELO MATEMÁTICO DA PLANTA

Segundo Aguirre (2007), para que se possa desenvolver um controlador para um sistema dinâmico, é necessário se obter um modelo matemático que descreve o comportamento desse sistema. Desta forma, a identificação do sistema dinâmico ocorreu a partir da análise computacional dos dados de entrada e saída obtidos de forma experimental. A relação entre esses dados, juntamente com as perturbações ocorridas no processo de coleta de dados, permitiu se obter o modelo matemático que descreve o comportamento da planta desenvolvida.

Para que fosse possível aplicar um controle do tipo MPC sobre a variável de nível, os dados utilizados para a coleta foram os valores do nível, velocidade da bomba e porcentagem de abertura da válvula, correspondendo respectivamente à entrada, à saída e à perturbação da planta de processo. O tempo de amostragem utilizado foi de 0,01s, pois foi o tempo mínimo que a arquitetura do sistema desenvolvida suportava sem ter problemas de perda de dados. Aguirre (2007) define que o tempo de amostragem pode ser entre 5 e 10 vezes inferior à menor constante de tempo do processo.

Com a constante do tempo de amostragem definida, foi elaborado um sinal de excitação para revelar características dinâmicas em torno da região de operação escolhida. O sinal construído é um conjunto de diferentes degraus na frequência de operação da Bomba centrífuga, como ilustrado na Figura 4.2.

Figura 4.2 Sinal construído para a identificação.



Realizado no Tanque 1, o experimento consistiu em aplicar diferentes sinais de degrau, na planta em malha aberta, em torno de um ponto de operação. Para isso, uma rotina foi desenvolvida no *software* Matlab para executar o experimento por um tempo aproximado de 120 segundos, com um total de 10 diferentes coletas. Após a coleta dos dados, os modelos matemáticos foram obtidos através do processo de identificação de sistema para um modelo empírico.

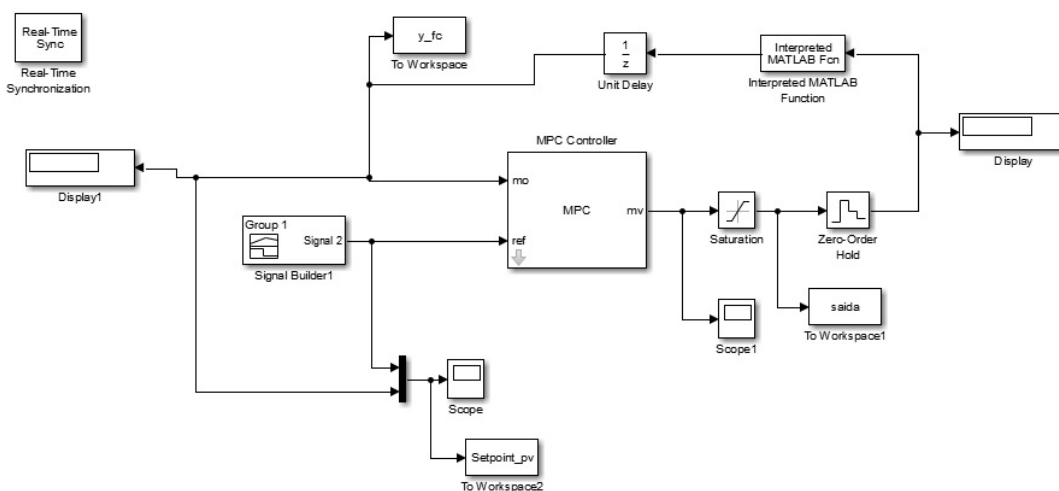
A equação 4.1 refere-se ao modelo do tanque 1 e a sua estimação com a rotina desenvolvida em matlab foi de 88,92%. Este modelo será utilizado para o projeto do controlador preditivo.

$$\begin{aligned} x(k+1) &= 0.9995x(k) + 0.0313\Delta u(k) \\ y(k) &= 0.0173x(k) \end{aligned} \quad (4.1)$$

4.3 DESENVOLVIMENTO DA APLICAÇÃO CLIENTE MPC UTILIZANDO SOA

A partir do modelo de espaço de estados obtidos na seção anterior, é possível projetar o controlador preditivo da planta didática. Com o modelo do controlador preditivo, é possível desenvolver as funções de leitura e escrita do módulo de controle que serão utilizadas no dispositivo de campo. Para isto, foi utilizada a ferramenta de modelagem de sistemas dinâmicos *simulink*. A Figura 4.3 ilustra o diagrama de blocos *simulink* que acessa os dados da arquitetura IWSOP desenvolvida no terceiro capítulo desta dissertação.

Figura 4.3 Aplicação cliente MPC.



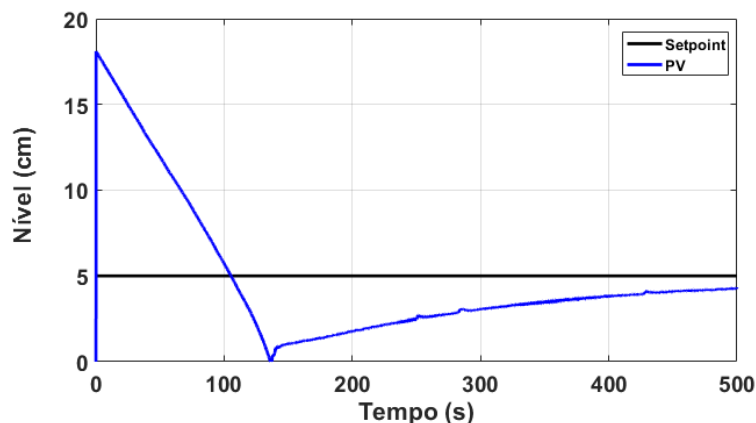
No diagrama de blocos apresentado na Figura 4.3, que utiliza a arquitetura IWSOP, é necessário executar uma função de leitura para acessar as informações via *web*, através dele, as informações que serão controladas e lidas pelo controlador preditivo desenvolvido podem ser configuradas (APÊNDICE B).

Para confirmar a validação do controlador preditivo desenvolvido e avaliar a capacidade deste em seguir as referências dadas, foram feitos testes para o caso regulatório e servo, seguindo as definições de (CAMPOS; TEIXEIRA, 2006):

- No caso regulatório, o *setpoint* é fixo em um valor, a variável de processo tenta se manter o mais próximo possível deste ponto e seu objetivo principal é tentar repelir as perturbações.
- No caso servo, existe uma mudança natural do estado do *setpoint*. Neste caso, o controle deve atuar, mantendo a saída, de acordo com o novo *setpoint*, ou seja, deve seguir uma série de trajetórias com o mínimo de erro.

O gráfico da Figura 4.4 ilustra a rastreabilidade do controlador MPC para o caso do controle regulatório na arquitetura IWSOP. Para o caso regulatório, a ação de controle é iniciada com o tanque em 18cm e o *setpoint* é fixado em 5cm. Assim, percebe-se que a PV (nome dado à variável de processo) segue o valor do *setpoint* e tende a controlar o processo em mais de 500s.

Figura 4.4 Trajetória do MPC para controle regulatório

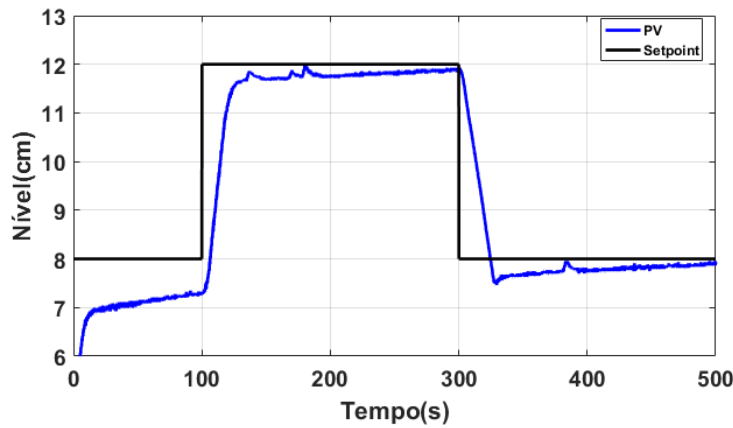


Para o caso servo, o *setpoint* é iniciado em 5cm de altura de nível. Em um determinado momento, o *setpoint* é modificado para 8cm e a contagem de tempo de controle é iniciada entre 0 e 100s. Esta ação serve para medir quando a variável manipulada (MV) inicia o

conjunto de ações para rastrear a alteração do *setpoint*, a bomba é acionada com o valor de vazão máxima do sistema até que a PV possa alcançar o valor do estado estacionário (8cm) e reduzida para manter-se neste estado.

Entre os tempos 100 e 300s, um *setpoint* de 12cm é definido e o controlador deve rastrear-lo novamente. Neste momento, como descrito no gráfico da Figura 4.5, percebe-se um melhor desempenho no controlador MPC. Após 300s, há um *setpoint* de 8cm e uma nova ação do controlador MPC. Desta vez, a rotação da bomba é reduzida e sua atuação demonstra um desempenho semelhante ao ocorrido entre 8 e 12cm. O teste do caso servo é finalizado em 500s de duração.

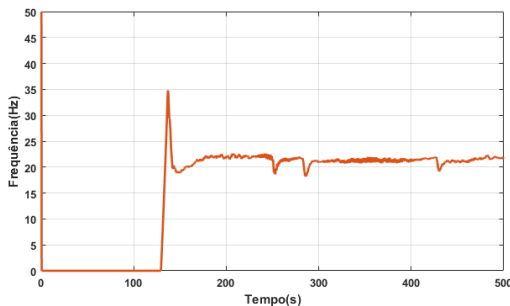
Figura 4.5 Trajetória do MPC para controle servo



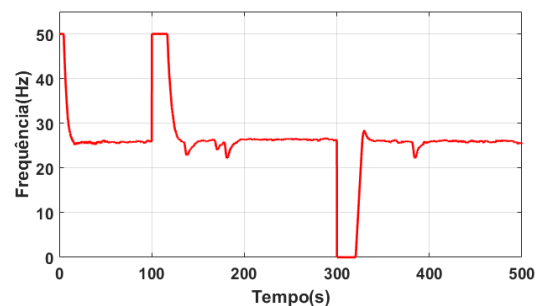
Observa-se, através dos gráficos das Figuras 4.6(a) e 4.6(b), o comportamento da variável manipulada(MV) do controlador tanto para o caso regulatório, quanto para o caso servo, demonstrando que não existem oscilações suficientemente bruscas que, com o passar do tempo, poderão diminuir a vida útil do atuador.

Figura 4.6 Planta piloto

(a) Trajetória da MV para controle regulatório



(b) Trajetória da MV para controle servo



4.4 DISCUSSÃO DOS RESULTADOS OBTIDOS

Com base na plataforma IWSOP descrita no capítulo 3 e validada através do estudo de caso apresentado, foi possível obter vantagens como: interoperabilidade, modularização e informações em tempo real.

Essas vantagens permitem que o usuário tenha acesso ao sistema de forma transparente, sendo necessário apenas conhecer as estruturas de acesso ao mediador. A interoperabilidade permite que os sistemas se comuniquem para que, através da troca de informações, seja possível a cooperação de ações de forma eficiente e segura, diminuindo, assim, as implicações presentes em ambientes com alta heterogeneidade. O SOA padroniza e modulariza as estruturas, permitindo que os usuários tenham a capacidade de consumir as informações e serviços compartilhados.

A plataforma IWSOP admite que qualquer aplicação habilitada possa enviar ou receber informações via o mediador. A modularidade é demonstrada através de um estudo de caso que utiliza uma aplicação de controle MPC. Este conceito é benéfico, pois a estrutura modular suporta que aplicações sejam adicionadas ou retiradas, de forma a não causar grandes impactos e, adicionalmente, favorece a integração dos diferentes níveis da estrutura de uma indústria.

O aplicativo cliente MPC foi desenvolvido para demonstrar que o sistema é eficaz e os dados são acessados em tempo real. As ações de controle do tipo regulatório e servo foram bem-sucedidas e demonstraram que a plataforma é flexível para o uso em diferentes aplicações. Ademais permite a implantação de qualquer técnica de controle que respeite os limites de tempo.

Essa limitação no tempo de execução impacta negativamente no desempenho da aplicação para acesso aos dados abaixo de centessegundos. Isto restringe o uso para situações de controle que necessitem de comunicação abaixo desta base de tempo. Tais limitações são provenientes dos *hardwares* que hospedaram a aplicação e de características relativas ao servidor OPC UA utilizado. Porém isso não impede que a plataforma seja explorada por outras técnicas de controle existentes.

O uso do sistema de forma distribuída e modularizada implica que as informações transmitidas necessitem de uma camada de criptografia dentro das estruturas utilizadas para o compartilhamento da informação. Os servidores OPC UA e *web* permitem adicionar camadas de proteção na sua estrutura, fazendo com que a transmissão e o acesso à informação sejam protegidos. Com o uso desta camada, é possível aumentar a confiabilidade do sistema e, conseqüentemente, minimizar o risco de acessos não autorizados às informações importantes do processo. Apesar de existir estas formas de proteção, este

trabalho não analisou tais características.

4.5 SÍNTESE

Neste capítulo, foram apresentados a criação e o teste de um estudo de caso, no qual uma aplicação cliente foi desenvolvida, com o intuito de executar um controle avançado MPC em um dispositivo de campo, a partir da arquitetura IWSOP.

Para viabilizar a conclusão desse desenvolvimento, uma planta didática foi utilizada. Nesta, um controle de nível foi executado remotamente a partir de um módulo cliente MPC implementado em *simulink*, interligado via um mediador para acessar um servidor OPC UA.

A modelagem matemática da planta foi desenvolvida a partir do modelo de espaço de estados, via identificação de sistemas dinâmicos e utilizado para projetar o controlador MPC.

Os testes foram executados, a fim de avaliar o desempenho da plataforma no contexto de controle servo e regulatório. Para isso, utilizou-se uma variável de entrada, uma variável de saída e uma variável de perturbação.

Esses resultados, portanto, não apenas confirmam a viabilidade do desenvolvimento de aplicações clientes com a utilização de SOA em um sistema industrial, como também colaboram com trabalhos similares na área.

CONCLUSÕES

Através do desenvolvimento deste trabalho, observou-se que os cenários industriais precisam ser aprimorados para que uma empresa continue sendo competitiva. A mudança de paradigma e das tecnologias é um importante ponto de partida para alterar a forma como as informações são disponibilizadas em uma empresa.

Deve-se observar que já está em curso uma revolução industrial capaz de modificar os alicerces básicos da indústria clássica. A quarta revolução industrial tem modificado o comportamento em toda a cadeia produtiva, forçando uma constante reavaliação na pirâmide de produção, inserindo, principalmente, os conceitos de modularização e descentralização da capacidade de tomada de decisão. Influenciado pela implantação de sistemas ciberfísicos, IoT, IoS e as conhecidas tecnologias habilitadoras.

Dessa forma, foi demonstrado que uso de Arquitetura Orientada a Serviço via *web services* e OPC UA são ferramentas que possibilitam uma reestruturação, de forma padronizada, da pirâmide da automação. A capacidade de intercomunicação provida por *web services*, juntamente com o padrão de comunicação unificado do OPC UA, permitem que vários dispositivos se comuniquem, formando uma rede heterogênea modularizada, intercambiável e que seja capaz de interligar, com segurança, o ambiente de produção e as tecnologias típicas do ambiente empresarial.

A partir dos conceitos investigados, foi possível estruturar uma arquitetura de automação, que tenha a capacidade de implementar os conceitos da arquitetura orientada a serviço, desenvolvendo um sistema mediador, capaz de interligar um *web service* com um servidor OPC UA e permitir a ação de diferentes aplicações clientes em dispositivos de campo.

Dessa forma, a arquitetura IWSOP foi validada partir da aplicação de um controle avançado em um dispositivo de campo, permitindo que um sistema legado pudesse ser inserido no conceito de orientação a serviço.

5.1 PRODUÇÃO CIENTÍFICA RELACIONADA

Durante o desenvolvimento desta pesquisa alguns artigos (aceitos, publicados e apresentados) foram desenvolvidos:

1. Correia, M. D., Silva, T. O., Lepikson, H. A., 2016. Plataforma aberta para a comunicação entre dispositivos de campo e clientes de diferentes níveis em sistemas de manufatura dinâmicos e integrados pela automação. Congresso Nacional de Engenharia Mecânica - CONEM, Fortaleza-CE.

2. Silva, T. O., Correia, M. D., Reis, R. A., Lepikson, H. A., Schnitman, L., 2017. Desenvolvimento de uma interface *web* baseada em uma arquitetura orientada a serviços visando o gerenciamento de uma planta de petróleo. 9^o Congresso Brasileiro de P/&D em Petróleo e Gás- Pdpetro, Maceió-AL.

5.2 RECOMENDAÇÕES E TRABALHOS FUTUROS

Algumas sugestões de trabalhos futuros a partir das informações apresentadas nesta pesquisa são:

- Embarcar os códigos desenvolvidos em uma *raspberry* para testes com plataformas portáteis;
- Investigar a camada de criptografia e segurança de acesso às informações;
- Realizar a orquestração de serviços de um processo;
- Criar um sistema de banco de dados na nuvem para análise de grandes dados;
- Fazer um estudo comparativo com outras técnicas de controle;
- Testar a plataforma desenvolvida em outros processos.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aileen Jin. *Overview*. 2015. Access date: 3 out. 2017. Disponível em: <<http://www.controleng.com/single-article/made-in-china-2025-chinese-government-aims-at-industry-40-implementation/fa41045cb073e0f10734e858f67b15c6.html>>.
- Software Toolbox. *Overview*. 2016. Access date: 1 Ago. 2016. Disponível em: <<http://www.softwaretoolbox.com/topserver/index.html>>.
- AGUIRRE, L. *Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, 2007. ISBN 9788570415844. Disponível em: <<https://books.google.com.br/books?id=f9IwE7Ph0fYC>>.
- ARCHITECTURE, O. P. C. U.; PART, S.; RELEASE, I. M. *OPC Unified Architecture Specification - Part 5: Information Model*. 2015.
- BAUTERS, K. et al. Automated work cycle classification and performance measurement for manual work stations. *Robotics and Computer-Integrated Manufacturing*, Elsevier Ltd, v. 51, n. October 2017, p. 139–157, 2018. ISSN 07365845. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0736584516303787>>.
- BEHRENDT, A. et al. Digital Manufacturing Capturing sustainable impact at scale. n. June, p. 1–24, 2017.
- BOOTH, D. et al. Web Services Architecture. *W3C Note*, v. 22, n. February, p. 1—98, 2004. ISSN 13583948. Disponível em: <<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/{\%}5Cnhttps://www.w3.org/TR/ws-arch/{\#}dis>>.
- BRYNJOLFSSON, E.; MCAFEE, A. Big Data : The Management Revolution. n. October, 2012.
- CAMACHO, E. F.; ALBA, C. B. *Model Predictive Control*. Springer London, 2013. (Advanced Textbooks in Control and Signal Processing). ISBN 9780857293985. Disponível em: <<https://books.google.com.br/books?id=tXZDAAAQBAJ>>.
- CAMPOS, M.; TEIXEIRA, H. *Controles típicos de equipamentos e processos industriais*. Edgard Blücher, 2006. ISBN 9788521203988. Disponível em: <https://books.google.com.br/books?id=d30_MgAACAAJ>.
- CARVALHO, G.; BERNARDINO, J. The internet of things and big data: Future trends. In: *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.: s.n.], 2017. p. 1–4.

- CHEN, D.; DOUMEINGTS, G. Architectures for enterprise integration and interoperability : Past , present and future. v. 59, p. 647–659, 2008.
- COELHO, P. M. N. *Rumo à Indústria 4.0*. 65 p. Tese (Doutorado) — Universidade de Coimbra, 2016. Disponível em: <https://estudogeral.sib.uc.pt/jspui/bitstream/10316/36992/1/TesePedroCoelhoRumo{\{a\}Industria4}>.
- COLOMBO, A. W.; BANGEMANN, T.; KARNOUSKOS, S. A system of systems view on collaborative industrial automation. *Proceedings of the IEEE International Conference on Industrial Technology*, p. 1968–1975, 2013.
- CORDEIRO, J.; RANCHORDAS, A.; SHISHKOV, B. *Software and Data Technologies: 4th International Conference, ICSOFT 2009, Sofia, Bulgaria, July 26-29, 2009. Revised Selected Papers*. Springer Berlin Heidelberg, 2011. (Communications in Computer and Information Science). ISBN 9783642201158. Disponível em: <https://books.google.com.br/books?id=-un6LaftKnwC>.
- COULOURIS, G. et al. *Sistemas Distribuidos - 5ed: Conceitos e Projeto*. Bookman Editora, 2013. ISBN 9788582600542. Disponível em: <https://books.google.com.br/books?id=6WU3AgAAQBAJ>.
- CUCINOTTA, T. et al. A Real-Time Service-Oriented Architecture for Industrial Automation. v. 5, n. 3, p. 267–277, 2009.
- CUPEK, R. et al. Performance evaluation of redundant OPC UA architecture for process control. *Transactions of the Institute of Measurement and Control*, v. 39, n. 3, p. 334–343, 2017. ISSN 01423312.
- DELSING, J. et al. A migration approach towards a SOA-based next generation process control and monitoring. *IECON Proceedings (Industrial Electronics Conference)*, p. 4472–4477, 2011. ISSN 1553-572X.
- DELSING, J. et al. Migration of industrial process control systems into service oriented architecture. *38th Annual Conference of the IEEE Industrial Electronics Society (IECON2012), 25-28 October 2012, Montreal, Canada*, p. 5790–5796, 2012. ISSN 1553-572X.
- DUNCAN, W. R. *A Guide to the Project Management Body of Knowledge 3.0*. [s.n.], 2004. 388 p. ISBN 1-930699-45X. Disponível em: www.pmi.org.
- ELFAHAM, H. et al. Full Integration of MATLAB / Simulink with Control Application Development using OPC Unified Architecture. p. 371–376, 2016.
- ELIASSON, J. et al. A SOA-based framework for integration of intelligent rock bolts with internet of things. *Proceedings of the IEEE International Conference on Industrial Technology*, p. 1962–1967, 2013.
- ERL, T. pdf. *Service Oriented Architecture: Principles of Service Design (The Prentice Hall Service-Oriented Computing Series) by Thomas Erl*. Prentice Hall, 2008. ISBN 9780132344821. Disponível em: [55000/4638a31e209e582b69dbdb95e1cbbfde](https://doi.org/10.1002/9780132344821).

- FEREIRA, I. V. et al. *PROPOSTA DE UM MODELO PARA A APLICAÇÃO DA INTERNET DAS COISAS INDUSTRIAL*. Porto Alegre-RS: [s.n.], 2017.
- FORSSTROM, S.; JENNEHAG, U. A performance and cost evaluation of combining OPC-UA and Microsoft Azure IoT Hub into an industrial Internet-of-Things system. *GIoTS 2017 - Global Internet of Things Summit, Proceedings*, 2017.
- FOUNDATION, O. *OPC Overview Definitions and Interfaces*. 2016. Disponível em: <https://opcfoundation.org/developer-tools/specifications-classic/data-access>.
- FOUNDATION, O. *What is OPC?* 2016. Disponível em: <https://opcfoundation.org/about/what-is-opc/>.
- GARAY, J. R. B. *CyberSens : Uma Plataforma Para Redes de Sensores em Sistemas Ciber-Físicos*. Tese (Doutorado) — Universidade de São Paulo, 2012.
- GILCHRIST, A. *Industry 4.0: The Industrial Internet of Things*. [S.l.: s.n.], 2016. 1–243 p. ISSN 19891490. ISBN 978-1-4842-2047-4.
- GIRBEA, A. et al. Design and implementation of a service-oriented architecture for the optimization of industrial applications. *IEEE Transactions on Industrial Informatics*, v. 10, n. 1, p. 185–196, Feb 2014. ISSN 1551-3203.
- GLOBAL, S. *Industrial Automation - Products & Services - Siemens Global Website*. 2018. Disponível em: <https://www.siemens.com/global/en/home/products/automation.html>.
- GR, S.; PFROMMER, J. A RESTful Extension of OPC UA. *IEEE*, n. 01, 2015.
- GROOVER, M. P. *Automação industrial e sistemas de manufatura*. Pearson Education do Brasil, 2011. ISBN 9788576058717. Disponível em: <https://books.google.com.br/books?id=GBrluAAACAAJ>.
- HENSEL, S. et al. Co-simulation with OPC UA. *IEEE International Conference on Industrial Informatics (INDIN)*, p. 20–25, 2016. ISSN 19354576.
- HERMANN, M.; PENTEK, T.; OTTO, B. Design principles for industrie 4.0 scenarios: A literature review. In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. [S.l.: s.n.], 2016. p. 3928–3937. ISSN 1530-1605.
- IATROU, C. P.; URBAS, L. Efficient OPC UA binary encoding considerations for embedded devices. *IEEE International Conference on Industrial Informatics (INDIN)*, p. 1148–1153, 2017. ISSN 19354576.
- INTEL. *Comparação de especificações de produtos Intel®*. 2017. Disponível em: <https://ark.intel.com/PT-BR/compare/90618,36695>.

KARNOUSKOS, S. et al. A SOA-based architecture for empowering future collaborative cloud-based industrial automation. *IECON Proceedings (Industrial Electronics Conference)*, p. 5766–5772, 2012. ISSN 1553-572X.

KHAITAN, S. K.; MCCALLEY, J. D. Design techniques and applications of cyberphysical systems: A survey. *IEEE Systems Journal*, v. 9, n. 2, p. 350–365, June 2015. ISSN 1932-8184.

KIM, W.; SUNG, M. OPC-UA Communication Framework for PLC-based Industrial IoT Applications. *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation - IoTDI '17*, p. 327–328, 2017.

KNUTSSON, M.; GLENNOW, T. *Challenges of Service-Oriented Architecture (SOA) - From the public sector perspective*. Tese (Doutorado) — Lund University, 2015.

KOMODA, N. Service oriented architecture (SOA) in industrial systems. *Industrial Informatics, 2006 IEEE International Conference on*, p. 1–5, 2006. Disponível em: <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=4053>.

KONSTANTINOV, S. *Investigating OPC UA Connectivity for Runtime Simulation of Automation Systems*. Tese (Doutorado) — University of Applied Sciences Emden/Leer, 2014.

LECHETA, R. *Web services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google*. NOVATEC, 2015. ISBN 9788575224540. Disponível em: <https://books.google.com.br/books?id=AyyLCgAAQBAJ>.

LIU, C. et al. Methodology of Cyber-Physical Systems. p. 230–246, 2017.

LUO, C. Survey of Parallel Processing on Big Data. 2017.

MACIEJOWSKI, J. M. *Predictive Control: With Constraints*. Prentice Hall, 2002. (Pearson Education). ISBN 9780201398236. Disponível em: <https://books.google.com.br/books?id=HVY58c7K>.

MACKENZIE, H. *The Smart Factory of the Future – Part 1*. 2016. Disponível em: <http://www.belden.com/blog/industrialethernet/The-Smart-Factory-of-theFuture-Part-1.cfm>.

MAHNKE, W.; LEITNER, S.-H.; DAMM, M. *OPC Unified Architecture*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. 1–30 p. ISSN 1098-6596. ISBN 978-3-540-68898-3. Disponível em: <http://ebooks.cambridge.org/ref/id/CBO9781107415324A009http://link.springer.com/10.1007/978-3-540-68899-0>.

MARKS, E. a.; BELL, M. *Executive 's Guide to Service-Oriented Architecture*. [S.l.: s.n.], 2006. ISBN 9780471768944.

MARZULLO, F. *SOA na Prática*. NOVATEC, 2009. ISBN 9788575222010. Disponível em: <https://books.google.com.br/books?id=h5WjAwAAQBAJ>.

MDIC and MCTIC. *Perspectivas de especialistas brasileiros sobre oportunidades e desafios para a manufatura avançada no Brasil*. Brasília, 2016. 67 p. Disponível em: <http://www.abinee.org.br/informac/arquivos/maavmdic.pdf>.

MENDES, J. M. et al. Service-oriented control architecture for reconfigurable production systems. In: *2008 6th IEEE International Conference on Industrial Informatics*. [S.l.: s.n.], 2008. p. 744–749. ISSN 1935-4576.

MORAES, C. C. de; CASTRUCCI, L. P. *Engenharia de automação industrial*. LTC, 2007. ISBN 9788521615323. Disponível em: <https://books.google.com.br/books?id=irN1PgAACAAJ>.

MORAES, E. C. *Desenvolvimento de interfaces baseadas em serviço para integração de sistemas heterogêneos na manufatura*. Tese (Doutorado) — Universidade federal da bahia, 2017.

MORGAN, J.; O'DONNELL, G. E. Enabling a ubiquitous and cloud manufacturing foundation with field-level service-oriented architecture. *International Journal of Computer Integrated Manufacturing*, v. 30, n. 4-5, p. 442–458, 2017. ISSN 13623052.

PAIK, H.-y. et al. *Web Service Implementation and Composition Techniques*. [s.n.], 2017. ISBN 978-3-319-55540-9. Disponível em: <http://link.springer.com/10.1007/978-3-319-55542-3>.

Pivotal. *Spring tools suite*. 2017. Access date: 20 agosto. 2017. Disponível em: <https://spring.io/>.

REBOREDO, P.; KEINERT, M. Integration of discrete manufacturing field devices data and services based on OPC UA. In: IEEE. *Industrial EleCtronics Society, IECON 2013-39th Annual Conference of the IEEE*. [S.l.], 2013. p. 4476–4481.

REINHARD, G.; JESPER, V.; STEFAN, S. Industry 4.0: Building the digital enterprise. *2016 Global Industry 4.0 Survey*, p. 1–39, 2016. ISSN 0196-9722. Disponível em: www.pwc.com/industry40.

RESEARCH, F. M. o. E. A. 001.Recommendations for implementing the strategic. n. April, 2013.

ROSSITER, J. A. *Model-Based Predictive Control: A Practical Approach*. CRC Press, 2003. (Control Series). ISBN 9780203503966. Disponível em: <https://books.google.com.br/books?id=owznQTI-NqUC>.

RÜSSMAN, M. et al. Industry 4.0. *The Boston Consulting Group*, p. 20, 2015.

SADTLER, C. et al. *Patterns : SOA Foundation Service Connectivity*. 1st. ed. Poughkeepsie NY: IBM, 2006. ISBN 9788184049848. Disponível em: <https://pdfs.semanticscholar.org/3f33/00c5f3ce2bfb3dd056723414ffebb5bab150.pdf>.

SANGEETHA, A. L. et al. Particle swarm optimization tuned cascade control system in an Internet of Things (IoT) environment. *Measurement*, Elsevier, v. 117, n. December 2017, p. 80–89, 2018. ISSN 02632241. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0263224117307856>.

SCHWAB, K.; MIRANDA, D. *A Quarta Revolução Industrial*. EDIPRO, 2016. ISBN 9788572839785. Disponível em: <https://books.google.com.br/books?id=0wgcvgAACAAJ>.

SCHWARZ, M. H.; BORCSOK, J. A survey on OPC and OPC-UA: About the standard, developments and investigations. *2013 24th International Conference on Information, Communication and Automation Technologies, ICAT 2013*, 2013.

SCIENCE, N.; COUNCIL, T. A national strategic plan for advanced manufacturing. 2012. Disponível em: <http://www.whitehouse.gov/ostp/nstc>.

SHROUF, F.; ORDIERES, J.; MIRAGLIOTTA, G. Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In: *2014 IEEE International Conference on Industrial Engineering and Engineering Management*. [S.l.: s.n.], 2014. p. 697–701. ISSN 2157-3611.

SILVA, R. M. da; FILHO, D. J. S.; MIYAGI, P. E. MODELAGEM DE SISTEMA DE CONTROLE DA INDUSTRIA 4.0 BASEADA EM HOLON, AGENTE, REDE DE PETRI E ARQUITETURA ORIENTADA A SERVICOS. *XII SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE*, v. 2015, 2015.

UNIFIED, O. P. C.; SPECIFICATION, A.; RELEASE, C. *OPC Unified Architecture Specification - Part 9: Alarms & Conditions*. 2015.

UNIFIED, O. P. C.; SPECIFICATION, A.; RELEASE, D. *OPC Unified Architecture Specification - Part 12: Discovery*. 2015.

UNIFIED, O. P. C.; SPECIFICATION, A.; RELEASE, D. A. *OPC Unified Architecture Specification - Part 8: Data Access*. 2015.

UNIFIED, O. P. C.; SPECIFICATION, A.; RELEASE, H. A. *OPC Unified Architecture Specification - Part 11: Historical Access*. 2012.

UNIFIED, O. P. C.; SPECIFICATION, A.; RELEASE, P. *OPC Unified Architecture Specification - Part 10: Programs*. 2015.

UNIFIED, O. P. C.; SPECIFICATION, A.; RELEASE, S. *OPC Unified Architecture Specification - Part 4: Services*. 2015.

UNIFIED, O. P. C. et al. *OPC Unified Architecture Specification - Part 3: Address Space Model*. 2015.

W Aidner, M.; Kasper, M. Security in Industrie 4.0: Challenges and Solutions for the Fourth Industrial Revolution. In: *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. San Jose, CA, USA: EDA Consortium, 2016. (DATE '16), p. 1303–1308. ISBN 978-3-9815370-6-2. Disponível em: <http://dl.acm.org/citation.cfm?id=2971808.2972112>).

Zarte, M. et al. Building an Industry 4.0-compliant lab environment to demonstrate connectivity between shop floor and IT levels of an enterprise. In: IEEE. *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. [S.l.], 2016. p. 6590–6595.

Zhou, J.; Yao, X.; Zhang, J. Big Data in Wisdom Manufacturing for Industry 4.0. In: *2017 5th International Conference on Enterprise Systems (ES)*. [S.l.: s.n.], 2017. p. 107–112.

APÊNDICE CÓDIGOS SISTEMAS WEB SERVICE OPC UA CLIENT

Arquivo de configuração do xml que carrega as principais dependências do *web service*.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi=
  "http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd" >
4 <modelVersion>4.0.0</modelVersion>
5
6 <groupId>com.opcuaintegracao</groupId>
7 <artifactId>OPCUAIntegracao</artifactId>
8 <version>0.0.1-SNAPSHOT</version>
9 <packaging>jar</packaging>
10
11 <name>OPCUAIntegracao</name>
12 <description>OPCUA Integracao</description>
```

```
13
14 <parent>
15 <groupId>org.springframework.boot</groupId>
16 <artifactId>spring-boot-starter-parent</artifactId>
17 <version>1.4.3.RELEASE</version>
18 </parent>
19
20 <dependencies>
21 <dependency>
22 <groupId>org.springframework.boot</groupId>
23 <artifactId>spring-boot-starter-web</artifactId>
24 </dependency>
25 <dependency>
26 <groupId>org.springframework.boot</groupId>
27 <artifactId>spring-boot-starter-test</artifactId>
28 <scope>test</scope>
29 </dependency>
30 <dependency>
31 <groupId>com.jayway.jsonpath</groupId>
32 <artifactId>json-path</artifactId>
33 <scope>test</scope>
34 </dependency>
35
36 <!-- https://mvnrepository.com/artifact/org.eclipse.milo/sdk-client -->
37 <dependency>
38 <groupId>org.eclipse.milo</groupId>
39 <artifactId>sdk-client</artifactId>
```

```
40 <version>0.1.3</version>
41 </dependency>
42
43 <dependency>
44 <groupId>org.eclipse.milo</groupId>
45 <artifactId>stack-core</artifactId>
46 <version>0.1.3</version>
47 </dependency>
48
49 </dependencies>
50
51 <properties>
52 <java.version>1.8</java.version>
53 </properties>
54
55
56 <build>
57 <plugins>
58 <plugin>
59 <groupId>org.springframework.boot</groupId>
60 <artifactId>spring-boot-maven-plugin</artifactId>
61 </plugin>
62 </plugins>
63 </build>
64
65 <repositories>
66 <repository>
```

```
67 <id>spring-releases</id>
68 <url>https://repo.spring.io/libs-release</url>
69 </repository>
70 </repositories>
71 <pluginRepositories>
72 <pluginRepository>
73 <id>spring-releases</id>
74 <url>https://repo.spring.io/libs-release</url>
75 </pluginRepository>
76 </pluginRepositories>
77 </project>
```

APÊNDICE CÓDIGOS MATLAB

Código da interpreted function(leitura e escrita via xml)

```
1 function [Valout] = Funtion_opcua_2(input2)
2
3 variavel='Channel2.Device2.NIVEL_TANQUE_CILINDRICO_CENTIMETROS';
4
5 mv = input2;
6 mvnovo = num2str(mv);
7
8 urlread(strcat('http://localhost:8090/escreveVariavelDouble?variavel=
Channel2.Device2.mv&valor=',mvnovo));
9 urlread(strcat('http://localhost:8090/escreveVariavelDouble?variavel=
Channel2.Device1.mv2&valor=',mvnovo));
10
11 %%%%%%%%% Read webserver
12 [nome, valor] = lerVariavel(variavel);
```



```
13 pv = str2double(valor);
14 pvnovo = urlread('http://localhost:8090/lerVariavel?variavel=Channel2.Device2.NIVEL_-
TANQUE_CILINDRICO_CENTIMETROS');
15 pv = str2double(pvnovo);
16
17 pvnovo = urlread('http://localhost:8090/lerVariavel?variavel=Channel2.Device1.Nivel');
18 Valout = pv;
19 end
20
```

Código de funcionamento via json

```
1 function [nome, valor] = lerVariavel(variavel)
2 json = urlread(strcat('http://localhost:8090/lerVariavelJSON?variavel=', variavel))
3 jsonSplit = strsplit(json, '');
4 nome = jsonSplit(4);
5 valor = jsonSplit(8);
6 end
```