

UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

**Classificação de *Single Board Computer* para Embarque
de Sistemas Ciberfísicos Baseado na Arquitetura 5C**

Braian Konzgen Maciel

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Eder Mateus Nunes Gonçalves

Rio Grande, 2019

Ficha catalográfica

M152c Maciel, Braian Konzgen.

Classificação de *Single Board Computer* para embarque de sistemas ciberfísicos baseado na arquitetura 5C / Braian Konzgen Maciel. – 2019.

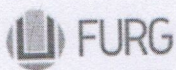
70 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2019.

Orientador: Dr. Eder Mateus Nunes Gonçalves.

1. Indústria 4.0 2. Sistemas Ciberfísicos 3. *Single Board Computer* 4. Comunicação 5. OPC UA I. Gonçalves, Eder Mateus Nunes II. Título.

CDU 004.2



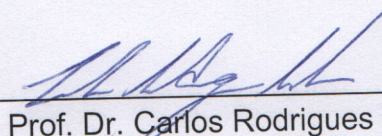
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

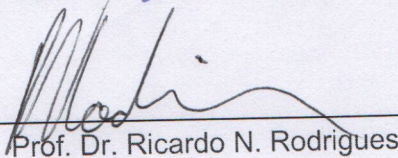
CLASSIFICAÇÃO DE *SINGLE BOARD COMPUTER* PARA EMBARQUE DE
SISTEMAS CIBERFÍSICOS BASEADOS NA ARQUITETURA 5C

Braian Konzgen Maciel

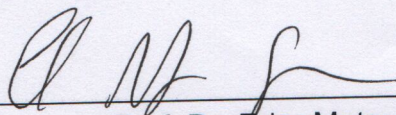
Banca examinadora:



Prof. Dr. Carlos Rodrigues Rocha



Prof. Dr. Ricardo N. Rodrigues



Prof. Dr. Eder Mateus Gonçalves
Orientador(a)

RESUMO

MACIEL, Braian Konzgen. **Classificação de *Single Board Computer* para Embarque de Sistemas Ciberfísicos Baseado na Arquitetura 5C**. 2019. 70 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

A Indústria 4.0, ou manufatura avançada, está promovendo uma nova revolução industrial através da integração de tecnologias computacionais e de comunicação. O desenvolvimento dos Sistemas Ciberfísicos e o emprego da Internet das Coisas e Serviços podem ser considerados como peças chave para o desenvolvimento dessa revolução. Quanto aos Sistemas Ciberfísicos (CPS), a arquitetura 5C apresenta-se como uma das mais relevantes na literatura, mas apenas enumera as funções necessárias a constituição de um CPS, sem dar maiores detalhes de desenvolvimento e implementação. Por outro lado, a utilização do Computador de Placa Única (*Single Board Computer - SBC*) já encontra aplicações em diversas áreas como controle industrial de eletrônicos de consumo, instrumentos médicos, comunicação em rede, automóveis e outros dispositivos portáteis. O presente trabalho tem por objetivo principal, classificar dispositivos SBC que ofereçam suporte a implementações que atendam os requisitos da Indústria 4.0, mais especificamente como plataforma de hardware para a arquitetura 5C. Essa classificação é feita a partir da definição de um conjunto de funções mínimo que estes dispositivos devem oferecer para que possam ser integrados a uma rede de dispositivos CPS. Um estudo de caso demonstra quais são essas funcionalidades mínimas e o uso de alguns SBCs dentro destas condições.

Palavras-chave: Indústria 4.0, Sistemas Ciberfísicos, Single Board Computer, Comunicação, OPC UA.

ABSTRACT

MACIEL, Braian Konzgen. **Single Board Computer Classification for Boarding of Cyber Physical Systems Based on Architecture 5C**. 2019. 70 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Industry 4.0, or advanced manufacturing, is promoting a new industrial revolution through the integration of computing and communication technologies. The development of Cyber Physical Systems (CPS) and the use of the Internet of Things and Services can be seen as key to the development of this new revolution. As for Cyber Physical Systems, the 5C architecture is one of the most relevant in the literature, but only lists the functions necessary to build CPS, without giving further development and implementation details. On the other hand, the use of Single Board Computer (SBC), already finds applications in industrial control of consumer electronics, medical instruments, network communication, automobiles and other portable devices. The main objective of this work is to classify SBC devices that support implementations that meet the requirements of Industry 4.0, more specifically as hardware platform for the 5C architecture. This classification is made from the definition of a minimum set of functions that these devices must offer so that they can be integrated into a network of CPS devices. A case study demonstrates what these minimum functionalities are and the use of some SBCs within these conditions.

Keywords: Industry 4.0, Cyber Physical Systems, Single Board Computer, Communication, OPC UA.

LISTA DE FIGURAS

Figura 1	Evolução da indústria. Fonte: Adaptado de Wahlster (2013)	11
Figura 2	Capacidade de conexão da Internet das coisas. Fonte: Razzaque et al. (2016)	16
Figura 3	Sistema ciberfísico representado através da combinação de tecnologias emergentes. Fonte: Adaptado de Oztemel & Gursev (2018)	18
Figura 4	Arquitetura 5C para CPS com funções e atributos de cada camada. Fonte: Adaptado de (Lee et al. 2015)	18
Figura 5	A composição conceitual das Fábricas Inteligentes. Fonte: Adaptado de Wahlster (2013)	20
Figura 6	Componentes Fundamentais de um sistema com padrão MTConnect. Fonte: Adaptado de MTConnect (2008)	22
Figura 7	Possíveis implementações utilizando o padrão MTConnect. Fonte: Adaptado de Oliveira (2017))	24
Figura 8	Arquitetura OPC Clássico e suas Especificações. Fonte: Adaptada de Venturelli (2018)	25
Figura 9	Possíveis implementações da arquitetura Cliente-Servidor. Fonte: Venturelli (2018)	26
Figura 10	Diferentes componentes dos clientes e servidores OPC UA. Fonte: Adaptado de Leitner & Mahnke (2006))	27
Figura 11	Arquitetura CPS baseada na combinação de SoA e Computação em nuvem. Fonte: Colombo et al. (2015)	31
Figura 12	Arquitetura para CPS baseado em sistemas multiagentes. Fonte: Leitão et al. (2015).	32
Figura 13	CPS baseado em SMA para integração de qualidade e controle de processos aplicados em uma indústria de máquinas de lavar de proposto em Rodrigues et al. (2012).	33
Figura 14	Arquitetura geral do CPPS. Fonte: (Garcia, Irisarri, Pérez, Estévez & Marcos 2016)	33
Figura 15	Estudo de caso IEC 61499 e arquivo de configuração XML do OPC-UA. Fonte: García, Irisarri, Pérez, Estévez, Orive & Marcos (2016)	34
Figura 16	Arquitetura de Arduino Yun com controle de temperatura e aplicação OPC UA. Fonte: Müller et al. (2017)	35
Figura 17	Procedimento para solicitação de dados do cliente OPC UA e resposta do servidor OPC UA. Fonte: Müller et al. (2017)	35
Figura 18	Arquitetura proposta para <i>plug&produce</i> hierárquico. Fonte: Profanter et al. (2017)	36

Figura 19	Processo de descoberta com vários níveis de hierarquia. Profanter et al. (2017)	37
Figura 20	Arquitetura de Hardware proposta em Vimos et al. (2016).	38
Figura 21	Conversão do formato de dados feito pelo OPC Gateway para transmissão de dados. Fonte: Adaptado de Vimos et al. (2016)	38
Figura 22	Associação dos módulos com a arquitetura CPS 5C.	43
Figura 23	Arquitetura de software genérica disposta em módulos com a inserção de funcionalidades OPC UA.	44
Figura 24	Exemplo de topologia genérica do sistema.	45
Figura 25	Resultado em segundos do teste <i>Benchamrks C-Ray</i>	49
Figura 26	Resultado em segundos do teste <i>Benchamrks Encode-Flac</i>	50
Figura 27	Resultado em segundos do teste <i>Benchamrks MAFFT</i>	51
Figura 28	Resultado em segundos do teste <i>Benchamrks Smallpt</i>	52
Figura 29	Resultado em C/s do teste <i>Benchamrks BlowFish</i>	53
Figura 30	Resultado em MFlops do teste <i>Benchamrks Himeno</i>	54
Figura 31	Resultado em S/s do teste <i>Benchamrks OpenSSL</i>	55
Figura 32	Resultado em segundos dos <i>Benchamrks</i> aplicados nos dispositivos SBC's.	56
Figura 33	Grupos dispositivos SBCs.	57
Figura 34	Demanda computacional exigida.	57
Figura 35	Planta didática <i>Smar PD3</i>	59
Figura 36	Diagrama de representação da aplicação do estudo de caso	59
Figura 37	Arquitetura de hardware.	60
Figura 38	Servidor contendo informações do transmissor de nível.	60
Figura 39	Servidor contendo informações do posicionador de válvula.	61
Figura 40	Topologia utilizada para aplicação.	62
Figura 41	Resposta do sistema.	63

LISTA DE ABREVIATURAS E SIGLAS

ACL	<i>Linguagem de Comunicação de Agentes</i>
CLP	<i>Controlador Lógico Programável</i>
CNC	<i>Computer Numeric Control</i>
CPPS	<i>Cyber-Physical Production System</i>
CPS	<i>Cyber-Physical System</i>
ERP	<i>Enterprise Resource Planning</i>
ESP	<i>Enterprise Service Bus</i>
FIPA	<i>Foundation for Physical Agents</i>
HTTP	<i>HyperText Transfer Protocol</i>
IESP	<i>Enterprise Service Bus Inteligente</i>
IMA	<i>Independent Meta Agents</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
MAS	<i>Multi Agent Systems</i>
MES	<i>Sistema de Execução de Manufatura</i>
M2M	<i>Machine-to-Machine</i>
NIST	<i>Instituto Nacional de Padrões e Tecnologia</i>
PA	<i>Product Agents</i>
PID	<i>Controlador Proporcional, Integral, Derivativo</i>
PTA	<i>Product Type Agents</i>
SMA	<i>Sistema Multiagente</i>
SBC	<i>Single Board Computers</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SOA	<i>Service Oriented Architecture</i>
TCP	<i>Transfer Control Protocol</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
1.2	Organização do Trabalho	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Indústria 4.0	14
2.1.1	Princípios da Indústria 4.0	14
2.1.2	Componentes da Indústria 4.0	15
2.2	<i>Single Board Computer</i> - SBC	20
2.3	Padrões para comunicação e aquisição de dados no contexto da Indústria 4.0	21
2.3.1	MTConnect	21
2.3.2	Protocolo OPC	24
2.3.3	OPC UA x MTConnect	28
2.3.4	Bibliotecas de código aberto OPC UA	28
3	TRABALHOS RELACIONADOS	30
3.1	Aplicações CPS industriais, arquiteturas, plataformas de hardware e comunicação	30
3.2	Análise dos trabalhos relacionados	39
4	ARQUITETURA PROPOSTA E DEFINIÇÃO DE HARDWARE	42
4.1	Arquitetura de Software	42
4.2	Topologia geral de hardware	45
4.3	Classes ou grupos de hardware para o CPS	45
4.3.1	Benchmarks	46
5	ESTUDO DE CASO	58
5.1	Estrutura de hardware do estudo de caso	58
5.2	Implementação Módulo de comunicação	59
5.2.1	Implementação dos Servidores OPC UA	60
5.3	Implementação do Módulo de Configuração	61
5.4	Validação do Estudo de Caso	61
6	CONSIDERAÇÕES FINAIS	64
	REFERÊNCIAS	66

1 INTRODUÇÃO

A evolução do ambiente industrial é fundamental para o suprimento das demandas humanas. No momento em que surgem novas necessidades e desafios, inovações tecnológicas precisam emergir para suprir tais exigências.

Ao longo da história o cenário industrial passou por três grandes transformações: a 1ª, 2ª e 3ª Revolução Industrial. Atualmente alguns movimentos como a Indústria 4.0 ou Manufatura Avançada sinalizam uma nova revolução, dados os diversos avanços das técnicas de manufatura, computação, robótica e automação.

A Indústria 4.0 ou Manufatura Avançada vem sendo amplamente divulgada e explorado por diversos pesquisadores nos últimos anos. Weyer et al. (2015), Browne et al. (1995), Tien (2012), apresentando conceitos, princípios e valores para promover uma transição entre o modelo atual e viabilizar a próxima revolução industrial.

A 1ª Revolução Industrial, ocorrida em meados do século XVIII ocorreu através da mecanização da produção, com a invenção da máquina a vapor. Já a 2ª Revolução Industrial, no século XX, teve como característica a introdução da eletricidade, viabilizando a produção em massa na linha de montagem idealizada por Henry Ford.

A 3ª Revolução Industrial ocorreu após a Segunda Guerra Mundial, e introduziu elementos de eletrônica, tecnologia da informação e automação nesse cenário. Elementos como o Controlador Lógico Programáveis (CLP) e da Tecnologia da Informação (TI) aplicadas no chão de fábrica (Rodrigues et al. 2016).

De acordo com Kagermann et al. (2013), a introdução do programa alemão denominado *Industrie 4.0* deu início ao desafio que se apresenta como a de 4ª Revolução Industrial. Assim máquinas e componentes inteligentes podem comunicar-se sem intervenção humana. Nesse cenário, decisões no chão de fábrica já são tomadas pelas próprias máquinas, a partir de informações fornecidas em tempo real. Outra característica importante, que pode ser destacada da Indústria 4.0, é a integração de diversas tecnologias relacionadas ao sistema, com foco na sua representação cibernética. A Figura 1 demonstra essa evolução do sistema industrial com suas principais características.

Além da Alemanha, outros países como Áustria, China e Estados Unidos vem incentivando o desenvolvimento da Indústria 4.0 através de grandes projetos com participação

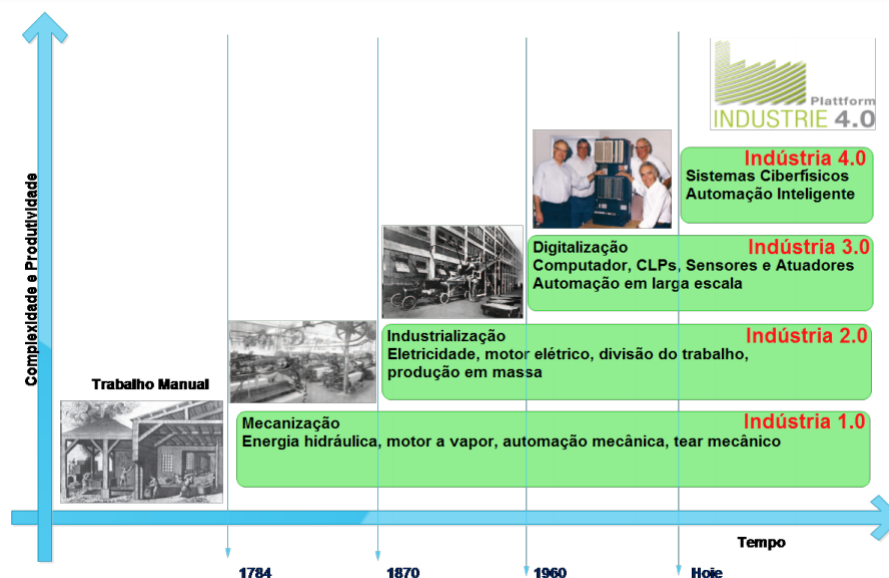


Figura 1: Evolução da indústria. Fonte: Adaptado de Wahlster (2013)

do governo e iniciativas privada, que tem como princípio o desenvolvimento da próxima geração de sistemas industriais (Karnouskos et al. 2016). Dentro deste novo mundo que esta sendo idealizado, alguns conceitos ganham um grande destaque e que influenciam diretamente para o desenvolvimento da 4ª Revolução Industrial, como Sistemas Ciberfísicos, do inglês, *Cyber Physical Systems* (CPS) (Jazdi 2014), Internet das Coisas ou do inglês, *Internet of Things* (IoT) (Shrouf et al. 2014), e com isso contribuem para que tecnologias, já conhecidas sejam aplicadas a manufatura industrial.

Para Barbosa et al. (2016), o CPS e o IoT surgem como pontos fundamentais para a realização da 4ª Revolução Industrial. O CPS conecta o mundo virtual com a realidade física, que integra capacidades de computação, comunicação e armazenamento, podendo operar em tempo real de forma confiável, segura, estável e eficiente.

A IoT é uma infraestrutura de rede global, dinâmica e auto-gerenciada, onde coisas físicas e virtuais possuem identidades, atributos físicos e personalidades virtuais. Essa infraestrutura é capaz de integrar vários dispositivos equipados com sensoriamento, identificação, processamento e comunicação em rede (Da Xu et al. 2014). Em outras palavras, IoT é um conceito tecnológico que representa a conexão dos diversos objetos do cotidiano com a rede mundial de computadores.

Nesse mesmo sentido, Industrial Internet of Things (IIoT) ou Internet Industrial das Coisas surge como a aplicação da IoT no Ambiente industrial. Tendo como características aquisição e acessibilidade de enormes quantidades de dados, em grande velocidade e muito mais eficiente que antes (Jeschke et al. 2017).

A constituição do CPS, por sua vez, pode envolver o uso de várias tecnologias emergentes, como, Arquiteturas Orientadas a Serviços (*Service-Oriented Architecture - SoA*), computação em nuvem (*Cloud Computing*), *Big Data*, *Machine-to-Machine* (M2M) e a

Computação Visual (CV). Essa integração entre tecnologias emergentes visa contribuir para o CPS enfrentar desafios identificados nos princípios da Indústria 4.0, como flexibilidade, robustez, adaptação, configuração e controle distribuído para estes sistemas.

Dentro deste contexto, surgem alguns pré-requisitos que devem ser resolvidos mais próximo ao chão de fábrica, como por exemplo: a capacidade de comunicação e transferência de dados entre diferentes dispositivos (sensores e atuadores) dentro de um ambiente industrial que surge como um dos problemas a serem resolvidos para a nova infraestrutura de indústria. Diversos padrões tem sido desenvolvidos para garantir que equipamentos de diferentes fontes realizem a coleta e transmissão de dados de forma eficiente e segura. Outro ponto que tem um grande destaque neste cenário é a descentralização do controle e incremento de complexidade para a realização de todas as operações.

A literatura apresenta diversas arquiteturas para desenvolvimento de CPS, como arquitetura CPS 5C apresentada por Lee et al. (2015), uma das primeiras arquiteturas propostas para sistemas Indústria 4.0. Embora seja uma proposta com características interessantes, por ser uma forma genérica para execução de CPS complica de ser usada como base para desenvolvimentos reais, uma vez que carece de maiores detalhes para sua concepção, como por exemplo tecnologias, comunicação, segurança, formato de dados, etc.). A proposta apresentada no presente trabalho visa detalhar a implementação para uma aplicação específica.

Além disso, a utilização de *Single Board Computers* em aplicações industriais já é uma realidade. No entanto, estes dispositivos ainda demandam uma melhor análise quanto a requisitos específicos da indústria 4.0, especialmente quando orientados a CPS's. Eles devem apresentar garantias mínimas de executabilidade sob diferentes condições de complexidade computacional, aquisição e armazenamento de dados

1.1 Objetivos

1.1.1 Objetivo Geral

Classificar dispositivos SBC que ofereçam suporte a implementações que apresentem características da indústria 4.0, mais especificamente, a uma implementação da arquitetura 5C (Lee et al. 2015). Esta arquitetura é composta por cinco camadas: configuração, cognição, ciberfísico, conversão de dados para informação e conexão inteligente. Esta classificação visa definir os requisitos mínimos para um SBC executar uma implementação da 5C.

1.1.2 Objetivos Específicos

- Propor uma arquitetura modular baseada na 5C;
- Identificar padrões existentes para comunicação entre dispositivos;

- Escolher um padrão para comunicação e aquisição de dados que seja compatível com a proposta da 4ª Revolução Industrial;
- Implementar o módulo de comunicação, que compreende a aquisição de dados de acordo com o padrão escolhido;
- Propor uma descrição de componentes conceituais e tecnológicos da arquitetura 5C, e suas relações que serão inseridos em cada classe.

1.2 Organização do Trabalho

A organização do trabalho esta descrita da seguinte forma:

- Capítulo 2: compreende uma contextualização profunda de técnicas e tecnologias que servirão de base para o desenvolvimento do trabalho.
- Capítulo 3: apresenta uma revisão bibliográfica de trabalhos relacionados abordando a partir da concepção de Sistemas Ciberfísicos com uma complexidade moderada, passando por CPS de baixo custo até diretamente a implementações voltadas para a aquisição de dados no contexto da 4ª Revolução Industrial.
- Capítulo 4: este capítulo é composto pela proposta, incluindo as arquiteturas software e classificação de dispositivos de hardware .
- Capítulo 5: neste capítulo é apresentado o estudo de caso em que foi aplicado a proposta do trabalho.
- Capítulo 6: Considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta alguns conceitos de técnicas, bem como tecnologias que estão ganhando um grande destaque com o desenvolvimento da próxima Revolução industrial. Estes conceitos servirão como base para o entendimento do trabalho ou como fundamentação para desenvolvimento do mesmo.

2.1 Indústria 4.0

A Indústria 4.0 é vista como um sinônimo de transformação das fábricas da atualidade em fábricas inteligentes com o desafio de flexibilizar a produção, gerando produtos altamente customizados para competir com o mercado global. Os elementos chave do conceito são as máquinas inteligentes, a análise avançada de dados e as transformações do ambiente de trabalho, com a adoção de suporte mais inteligente para as operações de manutenção, promovendo qualidade e segurança (Jazdi 2014). Ou seja, produtos, componentes e máquinas de produção, coletam e compartilham dados em tempo real, o que leva a mudar o sistema de controle centralizado de fábrica para inteligência descentralizada (Shrouf et al. 2014).

2.1.1 Princípios da Indústria 4.0

Para garantir que um sistema se encaixe no movimento da indústria 4.0 alguns autores afirmam que, estes sistemas devem estar de acordo com seis princípios básicos ou pilares desta nova revolução, que são eles (Hermann et al. 2016):

- Capacidade de operação em tempo real: consiste na aquisição e tratamento de dados de forma instantânea, permitindo a tomada de decisões em tempo real.
- Virtualização: propõe a existência de uma cópia virtual das fábricas inteligentes. Permitindo a rastreabilidade e monitoramento remoto de todos os processos por meio dos inúmeros sensores espalhados ao longo da planta.
- Descentralização: a tomada de decisões poderá ser feita pelo sistema ciberfísico de acordo com as necessidades da produção em tempo real. Além disso, as máquinas

não apenas receberão comandos, mas poderão fornecer informações sobre seu ciclo de trabalho.

- Orientação a serviços: utilização de arquiteturas de *software* orientadas a serviços aliado ao conceito de *Internet of Services*.
- Modularidade: produção de acordo com a demanda, acoplamento e desacoplamento de módulos na produção. O que oferece flexibilidade para alterar as tarefas das máquinas facilmente.
- Interoperabilidade: capacidade dos sistemas ciberfísicos, humanos e fábricas inteligentes comunicar-se uns com os outros geralmente através da Internet das Coisas.

A combinação entre alguns componentes da Indústria 4.0 podem garantir o ingresso de sistemas na 4ª revolução industrial, ou seja, garantem a realização dos seis princípios básicos.

2.1.2 Componentes da Indústria 4.0

Com base na revisão da literatura foram identificados quatro componentes chave da Indústria 4.0: Sistemas Ciberfísicos, Internet das Coisas, Internet dos Serviços e Fábricas Inteligentes (Hermann et al. 2016). A seguir, esses quatro componentes serão descritos explicando a relação de cada componente com a Indústria 4.0.

2.1.2.1 Internet das Coisas

O termo *Internet of Things* (IoT) ou internet das coisas tem sido vastamente utilizado ultimamente, mas foi citado pela primeira vez em meados dos anos 1993. Segundo Ashton (2011) o conceito de IoT é habilitar objetos do mundo real como fala, visão, audição, toque, entre outros, para o mundo virtual. Fazendo com que coisas que são inanimadas, com a inserção destas inteligências, possam desempenhar determinadas tarefas de uma forma mais precisa, com maior responsabilidade, colaboração e com aprendizagem.

O IoT pode ser considerado como uma mudança de paradigma para as tecnologias advindas da internet. Trappey et al. (2016) diz que a quantidade de dispositivos tecnológicos habilitados para IoT, em estimativas feitas para o ano de 2014, seria maior que a população mundial. Desta forma uma grande quantidade de novos consumidores, bem como fabricantes estão usufruindo desta nova metodologia principalmente os que trabalham diretamente com o ciber (software, sistemas de dados) e conexões físicas (dispositivos, máquinas, equipamentos) para funcionar efetivamente (Santos et al. 2016).

No contexto da indústria 4.0 a IoT, junto com CPS, surgem como um dos facilitadores chave para essa transformação da indústria atual. O IoT no nível fundamental é um meio de conectar objetos físicos à internet como uma rede que permite aos objetos coletar e trocar informações.

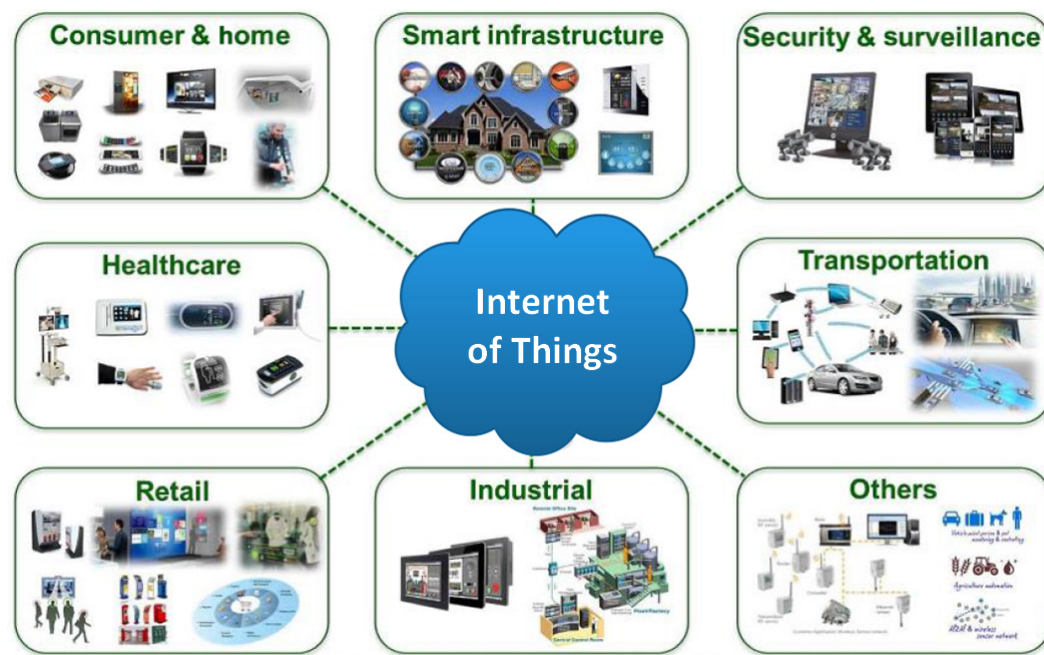


Figura 2: Capacidade de conexão da Internet das coisas. Fonte: Razzaque et al. (2016)

2.1.2.2 *Internet Industrial das Coisas*

Internet Industrial das Coisas (IIoT) refere-se, em particular, aos ambientes industriais, ou seja, se refere principalmente a uma estrutura industrial na qual um grande número de dispositivos ou máquinas são conectados e sincronizados. Fazendo o uso de ferramentas de software e tecnologias de ponta em um contexto M2M e IoT para a Indústria 4.0 *Internet Industrial das Coisas (IIoT)* (2017).

Nesse sentido, pode ser considerado um movimento em direção a máquinas inteligentes, por meio do qual os níveis de precisão das operações envolvidas nos respectivos sistemas são elevados a um nível que não pode ser alcançado por meio de intervenções humanas.

Como benefícios deste novo conceito espera-se que os resultados da aplicação da IIoT contribuam, na conectividade aprimorada e inteligente entre dispositivos ou máquinas, na redução de erros humanos e trabalho manual Jeschke et al. (2017). Outros benefícios são esperados ainda na segurança industrial, no aumento da eficiência geral e na redução de custos, de tempo como de dinheiro, assim como, no controle de qualidade e manutenção aprimorada.

2.1.2.3 *Internet dos Serviços*

A Internet dos Serviços ou *Internet of Services* (IoS) habilita a geração de serviços através da internet, por meio de um modelo de negócio com uma infraestrutura para serviços. Onde serviços são oferecidos e combinados para gerar valor em uma rede com vários fornecedores. O crescimento dos serviços pela internet cria novas variações da

distribuição de valor para os participantes da cadeia em suas atividades individuais Kagermann et al. (2013).

2.1.2.4 *Sistemas Ciberfísicos - CPS (Cyber Physical Systems)*

A composição de Sistemas Ciberfísicos (CPS) compreendem sistemas dotados de uma alta complexidade e multidisciplinares, que tem como característica básica a integração física e virtual de componentes onde agrega-se elementos de computação, para compor a parte cibernética, fazendo relação com fenômenos físicos. Isso inclui principalmente a observação ao contexto, a comunicação e aspectos dos sistemas físicos a partir de uma perspectiva multidisciplinar de controle. Eles permitem ainda o controle instantâneo e a verificação dos *feedbacks* do processo com o objetivo de gerar os resultados esperados (Möller 2016).

Segundo MacDougall (2014), CPSs criam as bases para a criação de uma Internet das Coisas, que combinada com a Internet dos Serviços forma o que é conhecido como Indústria 4.0. A combinação de diversas tecnologias emergentes como mencionado anteriormente são responsáveis por dar suporte aos seis pilares da Indústria 4.0. Com o intuito de fornecer inteligência e adaptação sobre sistemas modulares descentralizados, fornecer interoperabilidade em sistemas distribuídos e heterogêneos (no caso de SoA) Marks & Bell (2008), responsável por permitir o armazenamento massivo de dados e análise de dados de alto desempenho (Sousa et al. 2009). E ainda permitir a implementação de técnicas para entender e extrair conhecimento do grande volume e variedade de dados coletados Yin & Kaynak (2015), *Machine-to-Machine* (M2M) é responsável por possibilitar a interconexão entre dispositivos e realidade aumentada surge como interface para apoiar a integração do humano no circuito. Todas estas características contribuem para que CPS e IoT constituem uma rede inteligente que conecta todas as coisas a Internet, a exemplo de sensores e atuadores, com a finalidade de permitir a comunicação entre esses dispositivos através de protocolos conhecidos, permitindo a identificação, o rastreamento, a localização, o monitoramento e o gerenciamento inteligente dessas entidades (Gubbi et al. 2013).

A pesquisa relacionada a CPS e Internet das Coisas, como essa rede inteligente, esta cada vez mais difundida na atualidade, e com isso, diversas arquiteturas estão em desenvolvimento. Na literatura, uma das arquiteturas mais conhecidas e citadas, denominada 5C, foi proposta por Lee et al. (2015) e serve como base para o desenvolvimento deste trabalho. Arquitetura 5C, baseia-se em modelos de automação de processos já consagrados e será melhor detalhada a seguir.

2.1.2.5 *Arquitetura 5C*

A arquitetura 5C está estruturada em 5 níveis ou camadas e, de certa forma, serve como uma guia para a elaboração de um CPS, onde delimita uma forma de desenvolver

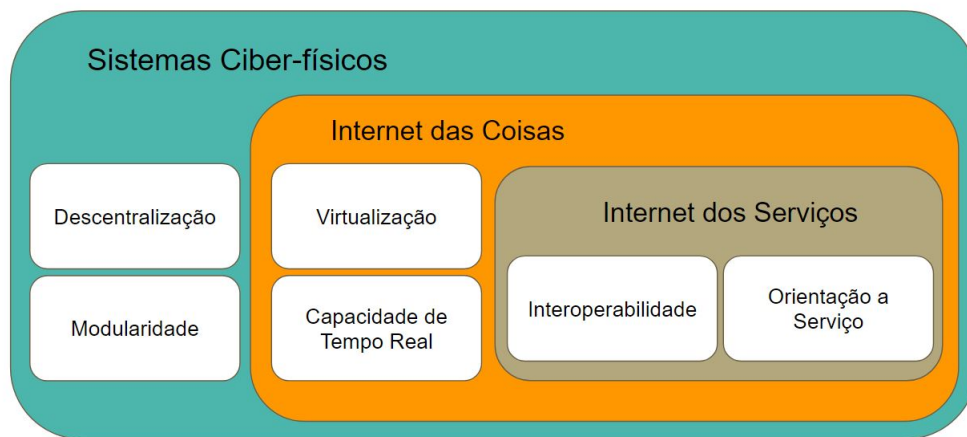


Figura 3: Sistema ciberfísico representado através da combinação de tecnologias emergentes. Fonte: Adaptado de Oztemel & Gursev (2018)

o sistema e aponta questões relevantes para sua implementação. Em geral, um Sistema Ciberfísico consiste em dois componentes funcionais principais (Lee et al. 2015):

1. A conectividade avançada que garante a aquisição de dados em tempo real a partir do mundo físico e o feedback da informação do espaço cibernético;
2. Gerenciamento inteligente de dados, análise e capacidade computacional que constrói o espaço cibernético.

A Figura 4 apresenta a arquitetura 5C para Sistemas Ciberfísico com funções e atributos de acordo com cada camada ou nível. A arquitetura é composta por cinco níveis sendo eles:



Figura 4: Arquitetura 5C para CPS com funções e atributos de cada camada. Fonte: Adaptado de (Lee et al. 2015)

Conexão inteligente: O nível conexão inteligente é o responsável por garantir a conexão entre os dispositivos e é o primeiro ponto que deve ser analisado quando se pensa em aplicações CPS, e é responsável pela aquisição de dados. No entanto esse processo deve garantir que os dados provenientes das máquinas e seus componentes sejam recebidos de forma precisa e confiável. A aquisição de dados pode ser oriunda de diversas fontes, sendo diretamente medida pelos sensores ou obtidos de sistemas de controle ou de fabricação, como ERP (*Enterprise Resource Planning*), MES (*Sistema de Execução de Manufatura*), SCADA (*Supervisory Control and Data Acquisition*) (Lee et al. 2015).

Conversão de dados para informações: O nível de conversão é responsável por receber os dados brutos e transformar em informações úteis, os dados são processados e convertidos em informações significativas. O objetivo desse nível é permitir a autoconsciência para o nível de componentes e máquinas. O processamento de sinal, a extração de recursos e os algoritmos de Gerenciamento de Saúde e Prognóstico (PHM), e análises preditivas são integrados neste nível Lee et al. (2015).

Cibernético: Nível Cibernético é onde toda informação é processada. A comparação entre pares, o compartilhamento de informações, a modelagem colaborativa, os registros de máquina, do tempo da utilização da máquina e do histórico de condições de saúde são analisados. Essas análises alimentam máquinas com capacidade de auto comparação, onde o desempenho de uma única máquina pode ser comparado e avaliado. Com isso, o desempenho da máquina e as informações históricas podem ser medidas para prever o comportamento futuro da máquina. Neste nível, a abordagem do CPS é normalmente usada para avaliar a saúde da máquina em diferentes ciclos ou regimes Lee et al. (2015).

Cognição: O nível de cognição é responsável por gerar um conhecimento profundo do sistema monitorado e posteriormente fornecer informações de raciocínio para correlacionar o efeito de diferentes componentes dentro do sistema. A adequada organização e apresentação dos conhecimentos adquiridos para usuários experientes apoiará a tomada de decisões apropriadas. Para este nível, são necessários gráficos de informação adequados para transferir completamente os conhecimentos adquiridos para os usuários Lee et al. (2015).

Configuração: O nível de configuração é o *feedback* do espaço cibernético ao espaço físico e atua como controle de supervisão para tornar as máquinas auto-configuradas e auto-adaptadas, e onde atua a auto-manutenção. Esta etapa atua como sistema de controle de resiliência para aplicar as decisões corretivas e preventivas, que foi feita em nível de cognição, ao sistema monitorado Lee et al. (2015).

Apesar de a arquitetura 5C ser uma forma de fluxo de trabalho sequencial, de como construir um CPS começando desde a aquisição inicial de dados, a análise, até tomada de decisão, existem ajustes e definições que devem ser feitos na Arquitetura 5C para aproximar em uma aplicação final, o que justifica este trabalho. E assim facilitar a aplicação do CPS de acordo com o que se propõe a indústria 4.0.

2.1.2.6 Fábricas Inteligentes

As tecnologias envolvidas na Fábrica Inteligente constituem-se em características-chaves da Indústria 4.0 (Kagermann et al. 2013). De acordo com Wahlster (2013), a combinação do CPS, IoT e IoS fundamenta a base do desenvolvimento do conceito de fábricas inteligentes. Essas fábricas utilizam massivamente os recursos tecnológicos para promover a eficiência do processo produtivo, agregando valor e reduzindo custos.

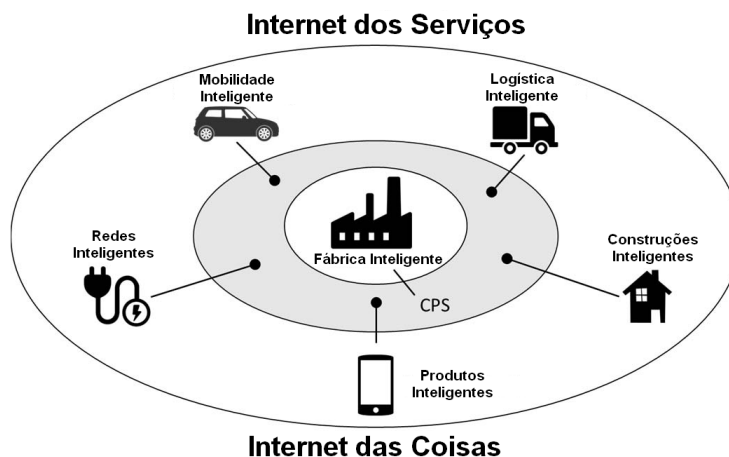


Figura 5: A composição conceitual das Fábricas Inteligentes. Fonte: Adaptado de Wahlster (2013)

A fábrica inteligente é definida como uma fábrica onde pessoas e máquinas interagem e contribuem na execução de suas tarefas. Como visualizado na Figura 5, isso só é possível através de tecnologias que ficam trabalhando no plano de fundo, ou seja, sistemas inteligentes que consideram o contexto, como a posição e o status de um objeto. Esses sistemas são capazes de auxiliar a realização de tarefas baseadas na informação proveniente do mundo físico e virtual.

2.2 Single Board Computer - SBC

Um *Single Board Computer* ou Computador de Placa Única é um computador embarcado de forma reduzida e pronto para uso. Pode ser considerado como uma solução genérica em *hardware* e *software* que pode ser utilizada no desenvolvimento de sistemas embarcados. Muitas vezes, o desenvolvimento de um protótipo é considerado difícil e caro devido aos custos de projeto de *hardware*, *design* e desenvolvimento de *software*, bem como fabricação e construção de *hardware*. O crescente barateamento e miniaturização do hardware na computação levaram a aplicação do Computador de Placa Única no desenvolvimento desses projetos e consequentemente os custos podem ser facilmente reduzidos (Matthews et al. 2016).

Soluções de *hardware* prontas para uso já existem, como o *Raspberry Pi*, que tem

software pronto para uso com Linux embarcado, oferecem uma plataforma completa para o desenvolvimento de produtos finais para diversas aplicações, como médicas, automação industrial, aeroespacial e robótica (Rahman et al. 2009).

Segundo Newark (2014), hoje os SBCs se dividem basicamente em duas categorias: proprietário e código aberto. SBCs Proprietários são aqueles que geralmente são projetados para uso em aplicação final ou como referência de avaliação. São projetos industriais que passam pelos mesmos testes que um produto final requer. Os SBC de código aberto oferecem aos usuários o acesso ao design e layout de hardware e ao código fonte usado na placa. Isso é ideal para todos os usuários, pois eles podem facilmente entender como o software e o hardware operam e adotar o design para atender aos requisitos de projeto.

Considerando os SBC de código aberto mais populares, foi criada uma tabela, analisando características como: capacidade de processamento, memória, armazenamento, dispositivos de E/S, conectividade. Com isso foram definidas classes, mas apenas as informações de capacidade de processamento e memória disponível e serão utilizadas e serão apresentadas na Seção 4.3.

2.3 Padrões para comunicação e aquisição de dados no contexto da Indústria 4.0

Os sistemas para a Indústria 4.0 estão fundamentados na integração de dados e serviços virtualmente localizados que estão vinculados aos recursos físicos da manufatura, através de uma rede inteligente DE OLIVEIRA (2017). Essa integração é garantida através do uso de padrões de referência Kagermann et al. (2013) destacam que a adoção desses padrões apresentam-se como um requisito para a adequada implementação das tecnologias que compõem a quarta revolução industrial. Ao mesmo tempo, a capacidade de conexão entre diferentes máquinas e dispositivos surge também como uma das principais necessidades da Indústria 4.0.

Conseguir interconectar estes dispositivos em uma planta de manufatura facilitará a introdução nesse novo contexto de indústria. Alguns padrões e protocolos foram desenvolvidos nos últimos anos, e com o avanço da tecnologia estão diretamente associados ao novo contexto de indústria Garcia, Irisarri, Pérez, Estévez & Marcos (2016) e com isso são passíveis de aplicação para desenvolvimento de um CPS. Nessas circunstâncias, ganham destaque os protocolos desenvolvidos pelo Instituto MTConnect MTConnect (2008) e OPC Foundation OPC Foundation (2016) que serão apresentados a seguir.

2.3.1 MTConnect

O MTConnect foi criado pelo Instituto MTConnect, com o objetivo inicial de tornar-se o padrão entre os fabricantes de máquinas CNC (*Computer Numeric Control*), assim como o Bluetooth ou o USB, tornaram-se em suas respectivas áreas (Warndorf et al. 2007).

O MTConnect surgiu como um padrão de troca de dados que permite que diferentes entidades e dispositivos em um sistema de manufatura possam compartilhar dados de forma transparente e em um formato comum. Trata-se de um padrão de comunicação aberto para interconexão, capaz de permitir que diferentes dispositivos e equipamentos forneçam seus dados em um formato compreensível, que pode ser lido por qualquer outro dispositivo que utilize este mesmo formato (Vijayaraghavan et al. 2008).

O padrão MTConnect baseia-se em tecnologias padrão da Internet como HTTP (*Hyper Text Transfer Protocol*) e XML (*Extensible Markup Language*), extensibilidade é uma característica chave do padrão MTConnect onde oferece uma representação flexível e amplamente reconhecida e aceita pela troca de dados semiestruturados na máquina. Este padrão é capaz de interconectar a parte mais baixa da cadeia de produção até a parte mais alta do planejamento do processo o que torna adaptável a qualquer tipo de indústria. O MTConnect também é significativo porque permite as principais características que são críticas para sistemas compatíveis com o IoT Industrial. Esses sistemas devem incorporar arquitetura de *software* aberta, protocolos abertos e modelos de dados abertos. Caso contrário, elementos proprietários criarão barreiras para usuários e desenvolvedores de aplicativos que operam no IoT industrial.

Um sistema que implementa o protocolo MTConnect tem cinco componentes fundamentais como visto na Figura 6: Dispositivo, Adaptador, Agente, Rede (network) e Aplicação/Cliente.

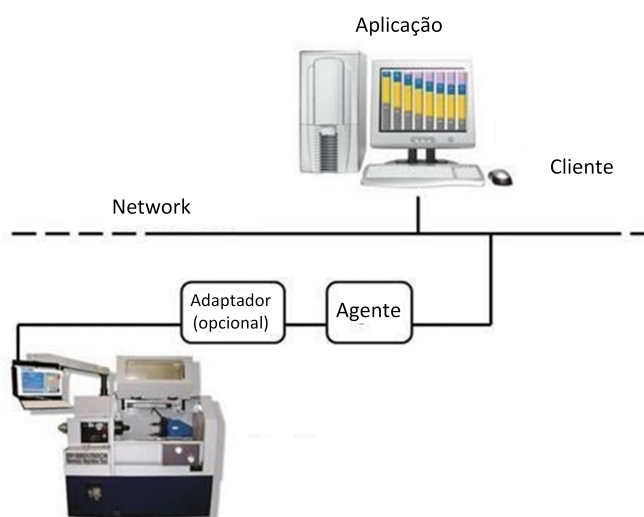


Figura 6: Componentes Fundamentais de um sistema com padrão MTConnect. Fonte: Adaptado de MTConnect (2008)

- **Dispositivo:** pode ser qualquer tipo de ferramenta, processo, planta, máquina, controlador ou equipamento;

- **Adaptador:** é responsável por realizar a “tradução” entre o formato proprietário do equipamento ou máquina para o padrão MTConnect. Pode ser um *software* ou *hardware* que fornece um link de definição de dados dedicados do dispositivo para a definição de dados no padrão do MTConnect. É importante destacar que o adaptador é opcional, pois quando os equipamentos de fabricação já possuem o MTConnect, não se faz necessário que se utilize o adaptador do mesmo;
- **Agente:** o agente pode ser visto como a ponte entre o adaptador (caso exista) e a aplicação. Um dos principais trabalhos do agente é formatar dados MTConnect em XML. Este é o formato que permite que os dados sejam interpretados por quase todos os aplicativos de software projetados para aceitar dados da Internet. O agente também deve fornecer armazenamento intermediário adequado para armazenar dados até que os pedidos o solicitem. O tamanho do *buffer* pode variar de acordo com as necessidades da implementação.

Outro trabalho importante para o agente é fornecer uma interface para a rede para que os dados possam ser acessados por aplicações externas. O agente faz isso usando o HTTP para receber pedidos de dados do cliente, processar esses pedidos e transmitir os dados solicitados;

- **Rede(network):** Network, é responsável por garantir a conectividade entre as fontes de dados (dispositivos) e consumidores de dados (aplicações/clientes), geralmente é uma rede Ethernet. A comunicação na rede usa métodos padrão de comunicação de rede como o protocolo HTTP;
- **Aplicação/Cliente:** Os clientes são *softwares* que se conectam ao agente a fim de requisitar informações com algum propósito. É o componente que consome os dados MTConnect as funções típicas da aplicação são para solicitar, armazenar, manipular e exibir dados usando várias técnicas para produzir informações sobre o dispositivo.

O MTConnect é um padrão desenvolvido com base em uma arquitetura modular, e permite o uso de diferentes ferramentas e métodos para conectar os dispositivos. Segundo Oliveira (2017), existem quatro possíveis arquiteturas para implementações com o padrão MTConnect, com a localização de cada componente no sistema, bem como a forma de comunicação entre eles, e são apresentados na Figura 7.

A primeira opção corresponde ao sistema ideal de aplicação do padrão MTConnect onde o agente e o Adaptador são elementos nativos de fábrica do dispositivo com isso tornando completamente compatível com o padrão. A arquitetura B o dispositivo vem com um Adaptador incorporado, porém o Agente é instalado separado em um HW, funcionando como um servidor (Oliveira 2017). Os dispositivos completamente incompatíveis com o padrão MTConnect estão representados nas arquiteturas C e D e nesses casos é

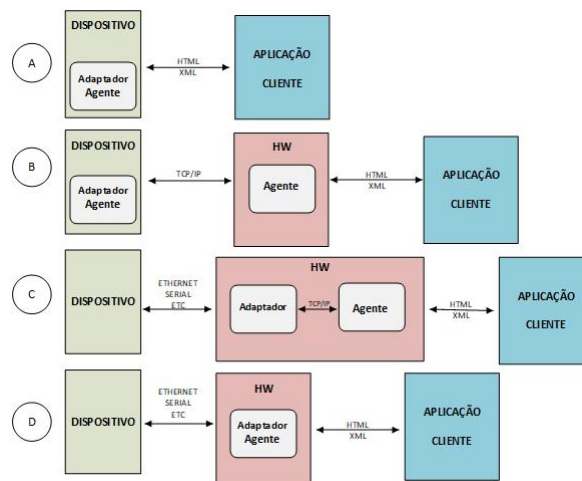


Figura 7: Possíveis implementações utilizando o padrão MTConnect. Fonte: Adaptado de Oliveira (2017))

necessário programar o Agente e o Adaptador. O diferencial da utilização do padrão MTConnect consiste dele suportar diferentes formatos de dados coletados de diferentes máquinas.

A Aplicação inclui uma função chamada Cliente que inicia todos os pedidos de dados MTConnect. O Cliente é uma função de *software* na Aplicação que realmente solicita dados do Agente e traduz esses dados no formato necessário para o aplicação Part (n.d.).

O MTConnect baseia-se nos padrões mais relevantes para a indústria de fabricação e software, o que maximiza a número de ferramentas disponíveis para sua implementação e fornece um alto nível de interoperabilidade com outros padrões e ferramentas nessas indústrias. No contexto da indústria 4.0 o MTConnect surge como solucionador para conexão entre dispositivos físicos de um CPS, atuando no mais baixo nível consegue garantir a aquisição de dados independente do formato ou protocolo de comunicação.

2.3.2 Protocolo OPC

Antes de descrever a Arquitetura Unificada do OPC, é preciso citar o OPC Clássico um dos principais produtos da OPC Foundation. Fundação criada em 1996, como resultado de uma reunião entre algumas empresas foi que contava com membros da Microsoft para dar suporte técnico, para desenvolver este Protocolo novo padrão.

2.3.2.1 OPC Clássico

Antigamente, os diversos dispositivos espalhados no chão de fábrica eram produzidos por inúmeros fornecedores os quais possuíam seus próprios drives e protocolos de comunicação particulares. Logo, surgiu a necessidade de criar um protocolo padrão para garantir a integração e conectividade entre esses dispositivos.

Foi então que surgiu o protocolo OPC (*OLE for Process Control*) Clássico. O OPC tinha como proposta, solucionar problemas de interoperabilidade em sistemas de

automação industrial, tratando a integração dos dados entre os diversos níveis de suas redes.

Na Figura 8 pode-se visualizar a arquitetura do OPC clássico, onde essa primeira versão era exclusiva a tecnologias proprietárias: utilizando OLE (*Object Linking and Embedding*), tecnologia da Microsoft, que permite a conexão entre objetos e dados, através da interface COM e DCOM (*Distributed Component Object Model*) também da Microsoft, assim permitindo a troca de dados entre aplicativos e dispositivos (Venturelli 2018).

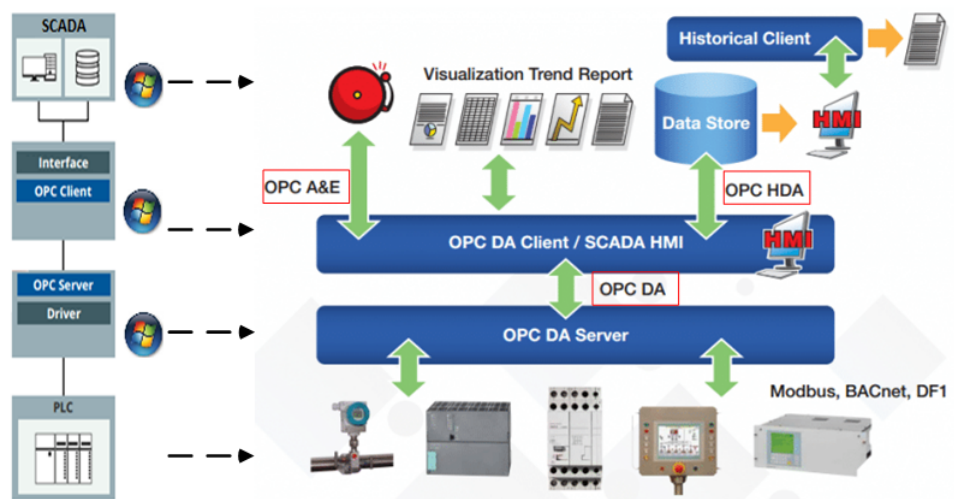


Figura 8: Arquitetura OPC Clássico e suas Especificações. Fonte: Adaptada de Venturelli (2018)

É possível também identificar, na Figura 8, as especificações fornecidas pelo OPC Clássico como:

1. OPC DA (Data Access): Define a troca de dados, incluindo valores, tempo e informações de qualidade.
2. OPC AE (Alarm & Event): Define a troca de informações de mensagem de tipo de alarme e evento, bem como estados variáveis e gerenciamento de estado.
3. OPC HDA (Historical Data Access): Define métodos de consulta e análises que podem ser aplicadas em dados históricos.

Elas resultam na interoperabilidade entre todos os componentes de automação, como, por exemplo, controladores, HMI. Em outras palavras, o OPC serve de interface para a troca de dados entre sistemas e foi especialmente desenvolvido para interfacear com máquinas, equipamentos e dispositivos, diretamente na camada de automação do chão de fábrica (OPC Foundation 2016).

2.3.2.2 OPC Unified Architecture - OPC UA

Algumas insatisfações, como por exemplo limitação do Sistema Operacional, no OPC Clássico motivaram o a criação de um protocolo para suprir esta demanda (Melo

2016).Então, em torno de 2008 surgiu o *OPC Unified Architecture* ou a Arquitetura Unificada do OPC UA.

O OPC UA contém todas as funcionalidades do OPC Clássico como a capacidade de os Servidores proverem acesso a dados atuais e históricos, bem como alarmes e eventos para notificar Clientes de mudanças importantes. Com o avanço da tecnologia, a possibilidade de ser extensível para qualquer sistema operacional foi alcançada. Ou seja, o OPC UA é um padrão independente de plataforma, pelo qual vários tipos de sistemas e dispositivos podem se comunicar pela troca de mensagens entre Clientes e Servidores sobre inúmeros tipos de redes (Venturelli 2018).

O OPC UA define conjuntos de serviços que Servidores podem fornecer, e os Servidores individuais especificam o conjunto de serviços que seus Clientes suportam. A informação é transmitida utilizando as definições de tipos de dados do OPC UA e de sistemas proprietários e os Servidores definem modelos de objetos que os Clientes podem descobrir dinamicamente (OPC Foundation 2016). Em seu núcleo, a OPC UA define:

- Um protocolo assíncrono (baseado em TCP, HTTP ou SOAP) que define a troca de mensagens por meio de sessões, (no topo de) canais de comunicação seguros, (em cima de) conexões brutas;
- Um sistema de tipo para mensagens de protocolo com um esquema de codificação binário e baseado em XML;
- Um meta-modelo para modelagem de informações, que combina orientação a objetos com relações tríplicas semânticas;
- Um conjunto de 37 serviços padrão para interagir com modelos de informações do lado do servidor. A assinatura de cada serviço é definida como uma mensagem de solicitação e resposta no sistema de tipos de protocolo.

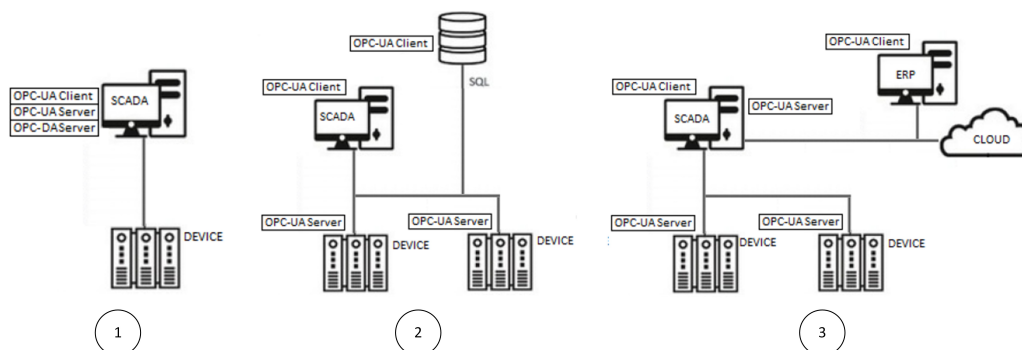


Figura 9: Possíveis implementações da arquitetura Cliente-Servidor. Fonte: Venturelli (2018)

Como mencionado em Venturelli (2018), a arquitetura do sistema OPC UA modela Clientes e Servidores interagindo como parceiros. A Figura 9 apresenta as possíveis

implementações para aplicação do padrão OPC UA no sistema industrial atual. Cada sistema pode conter múltiplos Servidores e Clientes. Cada Cliente pode interagir concorrentemente com um ou mais Servidores e cada Servidor pode interagir concorrentemente com um ou mais Clientes. Uma aplicação pode combinar componentes de Servidor e Cliente para permitir a interação com outros Servidores e Clientes.

Como visto em OPC Foundation (2016) OPC UA especifica um conjunto abstrato de serviços e o mapeamento para uma tecnologia concreta mas em contrapartida não especifica uma API, apenas os formatos de mensagem para dados trocados na conexão. Uma biblioteca de comunicação é usada no lado do cliente e do servidor para codificar e decodificar solicitações e respostas de mensagens. Diferentes bibliotecas de comunicação podem funcionar juntas, desde que usem o mesmo mapeamento de tecnologia. A Figura 10 apresenta uma visão geral dos diferentes componentes dos clientes e servidores OPC UA bem como explica as possíveis interações de servidores OPC UA.

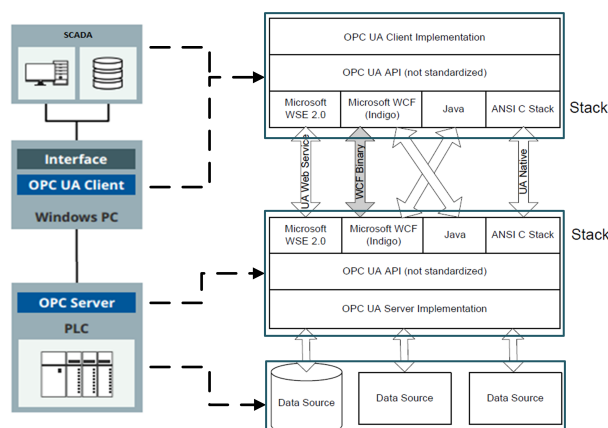


Figura 10: Diferentes componentes dos clientes e servidores OPC UA. Fonte: Adaptado de Leitner & Mahnke (2006)

Cliente OPC UA: A arquitetura do Cliente OPC UA modela o ponto final da interação entre Cliente/Servidor no lado do Cliente. O API do Cliente é o código que implementa a função de Cliente. Ele utiliza a API do Cliente OPC UA para enviar e receber requisições e respostas para o Servidor OPC UA. A API do Cliente OPC UA é uma interface interna que isola o código, da aplicação Cliente, da biblioteca de comunicação (Leitner & Mahnke 2006).

Servidor OPC UA: A arquitetura do Servidor OPC UA modela o ponto final da interação entre Cliente/Servidor no lado do Servidor. A biblioteca de comunicação do lado do servidor entrega as mensagens de solicitação para a implementação do servidor por meio da API do OPC UA. Como a API do OPC UA realiza as especificações de serviço abstratas, ela pode ser a mesma do lado do cliente. A implementação do Servidor aplica a lógica necessária para retornar a mensagem de resposta apropriada e obtém seus dados de algum sistema subjacente. Por exemplo, de um banco de dados, um conjunto de dispositivos ou algum servidor OPC (Leitner & Mahnke 2006).

OPC UA está seguindo exatamente no caminho da quarta revolução industrial, ou Indústria 4.0. Conceitos como comunicação independente de plataforma e fornecedor, segurança de dados, padronização e inteligência descentralizada são comuns entre os dois (Imtiaz & Jasperneite 2013).

Portanto, tecnologias para dispositivos, IoT (Internet of Things) e IIoT (Industrial Internet of Things) já estão completamente disponíveis na Arquitetura Unificada do OPC.

2.3.3 OPC UA x MTConnect

Embora sejam padrões semelhantes em alguns aspectos, eles se diferenciam principalmente em estrutura e propósito, onde cada padrão tem suas vantagens e desvantagens. Da mesma forma, os cenários em que eles devem ser usados não são os mesmos.

O MTConnect é descrito como um conjunto de padrões abertos, sem *royalties*, destinados a promover maior interoperabilidade entre máquinas, dispositivos, sistemas de controle e aplicativos de *software*. Ele traduz dados de ferramentas, de máquinas em uma linguagem comum baseada na Internet, que pode ser interpretada por um aplicativo de *software*. O OPC UA é uma arquitetura unificada baseada no protocolo OPC Clássico para comunicação industrial. É um conjunto aberto e não proprietário de especificações que garante que os sistemas de automação sejam compatíveis entre si (Jasperneite et al. 2015).

Em outras palavras, o OPC UA é um padrão que lida com a disponibilização de dados, enquanto o MTConnect é um padrão que lida com tornar os dados acionáveis. Uma grande diferença entre os protocolos, é que o OPC UA pode ser um protocolo de leitura/escrita, enquanto o MTConnect é somente leitura, e conseqüentemente nada pode ser escrito de volta para a máquina. Com isso a escolha do OPC UA se torna mais viável para aplicação desenvolvida neste trabalho.

2.3.4 Bibliotecas de código aberto OPC UA

Nesta seção, são apresentadas algumas bibliotecas OPC UA de código aberto onde é realizada com o intuito de escolha para a aplicação do estudo de caso. Todas as seguintes bibliotecas OPC UA podem ser compiladas no Linux e no Windows.

1. *FreeOpcUa* (LGPL): é uma biblioteca de servidores e clientes com suporte para a maioria dos conjuntos de serviços básicos do OPC UA, exceto o conjunto de serviços de descoberta. Desde 2015, as contribuições para este projeto diminuíram rapidamente e apenas alguns recursos foram adicionados.
2. *OpenOpcUA*: esta biblioteca fornece uma API (*Application Programming Interface*) para desenvolvimento de servidor e cliente e permite carregar dinamicamente modelos de informações de OPC UA a partir de arquivos XML. Também é testado

com os CTT. Sua principal desvantagem é que uma taxa única é solicitada para acessar a base de código.

3. *ASNeG OpcUaStack*: fornece uma API para servidores e clientes com apenas funcionalidades básicas, ou seja, leitura, gravação e monitoramento de variáveis do OPC UA.
4. *open62541*: Esta biblioteca fornece uma API para servidores e clientes que suportam quase todos os recursos dos diferentes conjuntos de detecção, exceto (inicialmente) o Conjunto de Serviços de Descoberta, incorporado nas versões posteriores. A interface é compatível com a Ferramenta de Teste de Conformidade (CTT) do OPC Foundation. O modelo de informações pode ser gerado automaticamente a partir de arquivos XML. Este projeto é desenvolvido ativamente e novos recursos são constantemente adicionados. Além do Linux e do Windows, muitas plataformas diferentes são suportadas, por exemplo, OS X, QNX, Android e sistemas embarcados.

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos". Existem bibliotecas comerciais como a **AnsiC** da OPC Foundation fornece uma implementação de referência oficial sob uma licença dupla, proprietária dos membros da Fundação OPC e da GPL para todos os demais. Esta pilha é a mais completa. Em março de 2017, a implementação do servidor LDS-ME foi lançada como Beta apenas para Windows.

Ele tem uma licença aberta e muitos dos recursos necessários já estão implementados. Um dos recursos mais importantes, o servidor LDS-ME, ainda não estava incluído na pilha, como é o caso de todas as outras pilhas de código aberto. Além disso, assinatura e criptografia ainda não são suportados pelo *open62541*, mas estão atualmente em desenvolvimento e serão suportados em breve.

3 TRABALHOS RELACIONADOS

Neste capítulo é realizada uma análise de trabalhos que abordam o desenvolvimento de sistemas ciberfísicos, explorando diferentes arquiteturas e características como padrões e tecnologias emergentes para concepção desses sistemas bem como plataformas de *hardware* para aplicação de um CPS.

3.1 Aplicações CPS industriais, arquiteturas, plataformas de hardware e comunicação

O trabalho apresentado em Colombo et al. (2015) é denominado de Projeto SOCRADES e baseia-se no paradigma de “automação colaborativa”, onde cria um ecossistema orientado a serviços. Um sistema em rede é composto por dispositivos inteligentes embarcados que interagem com o ambiente físico Colombo et al. (2015). O comportamento inteligente do sistema de automação é composto por configurações de dispositivos como *Web service* que introduzem partes incrementais da inteligência requisitada. O objetivo do projeto, está em administrar o número de dispositivos inteligentes e dominar a complexidade associada dos objetos físicos com sua representação cibernética.

A arquitetura proposta é apresentada na Figura 11. Os componentes CPS tem suas funcionalidades embarcadas em sistemas como *Web service*, e por meio de um barramento de serviço, que por sua vez, formam uma rede de componentes ou objetos inteligentes, tais funcionalidades são transmitidas para os demais componentes.

Os componentes físicos de *hardware* e *software* tem sua representação cibernética representada na nuvem. Desta forma, os dispositivos inteligentes de automação compreendem o controle, inteligência e lógica para a parte mecatrônica e todas essas funcionalidades podem ser acessadas a partir de qualquer nó da rede e qualquer outro membro/componente da nuvem pode se comunicar e trocar dados e informações diretamente para sistemas de nível superior como SCADA, MES e ERP.

Os autores ressaltam que o maior impacto dos resultados do projeto SOCRADES foram as muitas lições aprendidas sobre como especificar, desenvolver, comissionar e operar uma arquitetura de automação colaborativa totalmente distribuída e orientada a serviços

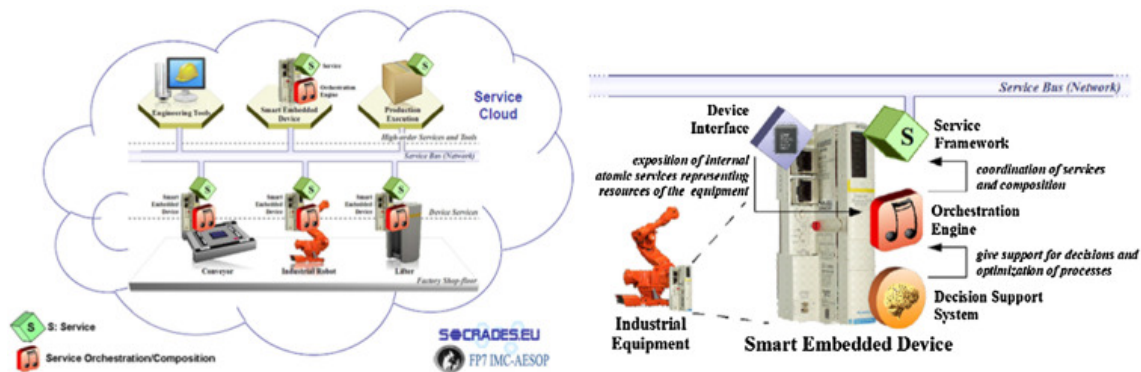


Figura 11: Arquitetura CPS baseada na combinação de SoA e Computação em nuvem. Fonte: Colombo et al. (2015)

inteligentes, que garanta o cumprimento dos requisitos de flexibilidade e reconfigurabilidade rápida, adotados anteriormente (Colombo et al. 2015).

Em Leitão et al. (2015), o trabalho apresentado tem como objetivo desenvolver estratégias de mitigação, com intuito de obter uma resposta mais rápida e adequada, quando deparadas com acontecimentos inesperados na linha de produção. Os autores ainda ressaltam que a aplicação se dá principalmente para produtos complexos e altamente personalizados como aviões e estaleiros. Foi utilizada a combinação de tecnologias como SMA e princípios SoA para desenvolver ferramentas de planejamento, programação e otimização de produção.

Na arquitetura é proposto um barramento de serviço, *Enterprise Service Bus* (ESB), que é usado para suportar a interoperabilidade entre as ferramentas de suporte à decisão SMA. Com a inserção de diferentes módulos como o serviço de Ontologia, Serviço de Transformação de Dados, Gerenciamento de Nó e Gerenciamento do Ciclo de Vida, como resultado tem-se um *Enterprise Service Bus inteligente* (iESB) que por sua vez, garante a interoperabilidade total a toda solução, conforme ilustrado na Figura 12. O ESB também permite a integração de sistemas legados, como ERP, MES ou sistemas SCADA.

A arquitetura implementada foi aplicada em dois casos. Uma fábrica da Airbus, em Hamburgo, para a produção do novo A350 e na fábrica de Iacobucci, em Ferentino, produzindo equipamentos modulares utilizados durante os vôos dos aviões, como máquinas de café, compactadores de lixo, fornos e carrinhos.

No trabalho apresentado em Rodrigues et al. (2012) a arquitetura desenvolvida, visualizada na figura 13, foi aplicada em uma indústria de fabricação de máquinas de lavar. O seu principal objetivo, foi melhorar a produtividade, e a qualidade dos produtos através da integração do controle do processo com a qualidade. Para isso foi implementado um sistema multiagente, no qual foi responsável pelos procedimentos dinâmicos de auto-adaptação, *loops* de controle e mecanismos de auto-otimização.

Foram projetados quatro tipos de Agentes, *Product Type Agents* (PTA), representa os diferentes tipos de produtos que podem ser produzidos na planta; *Product Agents* (PA),

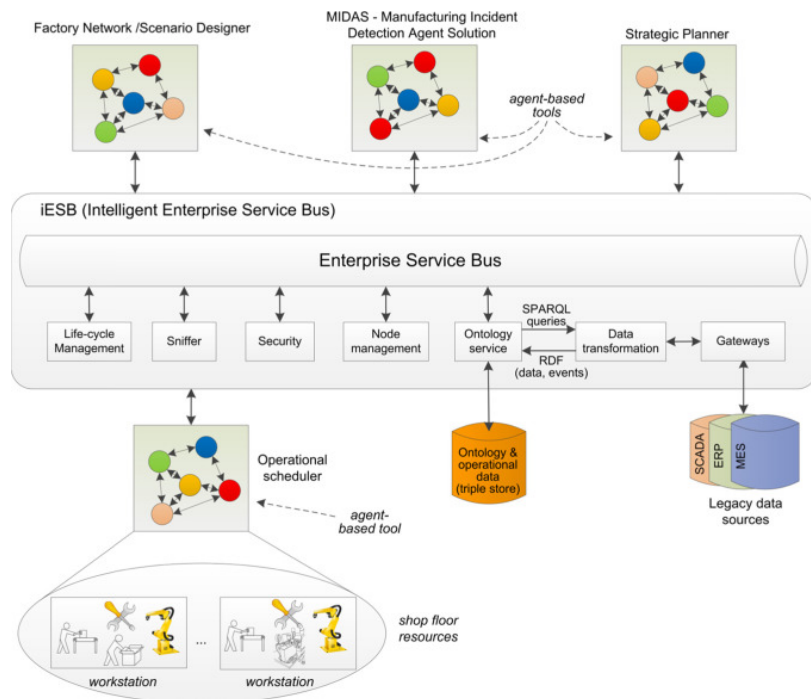


Figura 12: Arquitetura para CPS baseado em sistemas multiagentes. Fonte: Leitão et al. (2015).

representa a produção de instâncias de produto sendo produzidos ao longo da linha de produção; *Resource Agents (RA)*, representa os recursos da linha de produção, estações de processamento, estações de controle de qualidade e operadores; *Independent Meta Agents (IMA)*, IMA: Os agentes coletam os dados do chão de fábrica de forma distribuída e realizam análise de dados em tempo real para ajustar dinamicamente as variáveis de produção, ou seja, os parâmetros de processamento, de operação e de inspeção (Leitão et al. 2016).

Como funcionalidades deste agente, pode-se destacar, a adaptação contínua das estações de processamento e inspeção, a seleção de testes funcionais, a geração de avisos de qualidade e a parametrização do controlador. Toda solução de agente foi projetada utilizando a plataforma JADE, e distribuída em 8 computadores dispostos ao longo do chão de fábrica, sendo conectados por uma rede Ethernet, utilizando o protocolo TCP/IP para comunicação. Para o controle de baixo nível foram utilizados CLP's e os programas executados de acordo com a IEC 61131-3, onde tem como principal função garantir a resposta em tempo real.

Os autores ainda comentam que a integração de tecnologia MAS com dispositivos físicos como CLP, permitem construir componentes ciberfísicos, para integração de processos e controle de qualidade. Os resultados obtidos mostraram um conjunto de benefícios como: aumento de eficiência produtiva, otimização dos parâmetros do processo, bem como melhoria de qualidade do produto (Leitão et al. 2016).

Em García, Irisarri, Pérez, Estévez, Orive & Marcos (2016) e Garcia, Irisarri, Pérez,

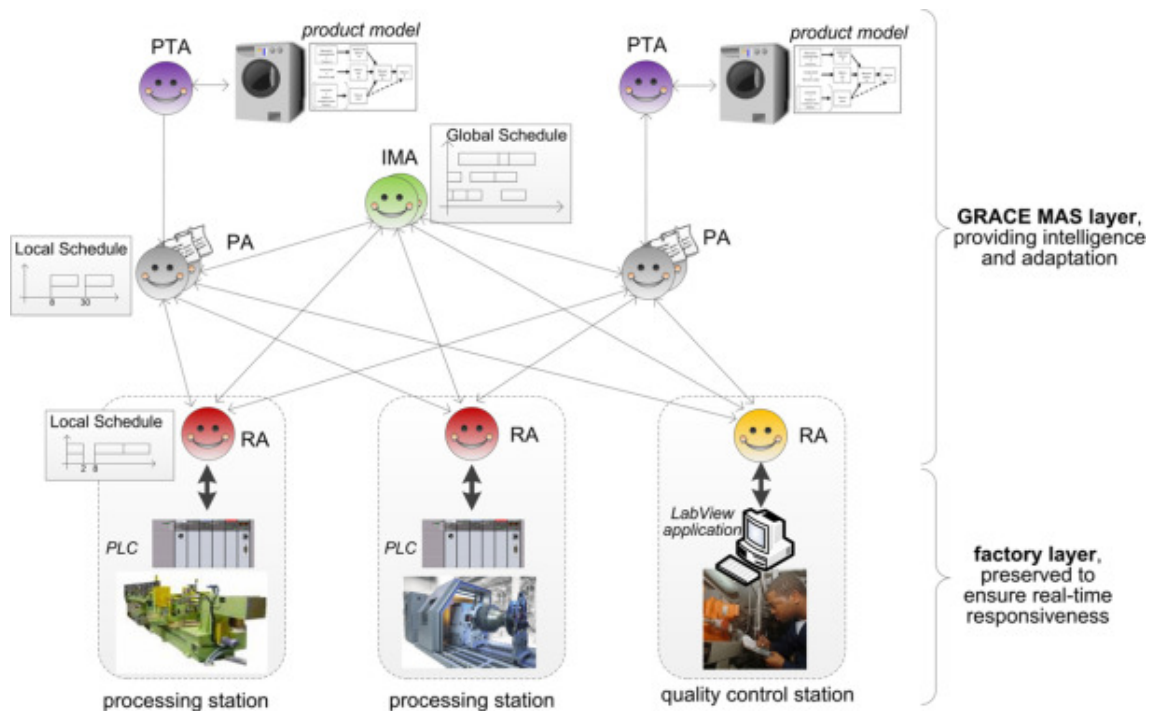


Figura 13: CPS baseado em SMA para integração de qualidade e controle de processos aplicados em uma indústria de máquinas de lavar de proposto em Rodrigues et al. (2012).

Estévez & Marcos (2016) é apresentada uma plataforma de baixo custo para desenvolvimento de um CPPS baseada em um servidor OPC UA e clientes facilmente configuráveis. O principal objetivo deste trabalho é coletar dados de um processo via comunicações Machine-to-Machine (M2M). A abordagem se concentra na integração de redes industriais tradicionais, como, *Modbus/TCP* em um servidor OPC UA sobre o padrão IEC 61499. A arquitetura CPPS base deste trabalho é apresentada na Figura 14, sendo composta por um conjunto de componentes que gerenciam um conjunto de modelos representando o mundo físico, a troca de informações e as informações a serem acessadas.

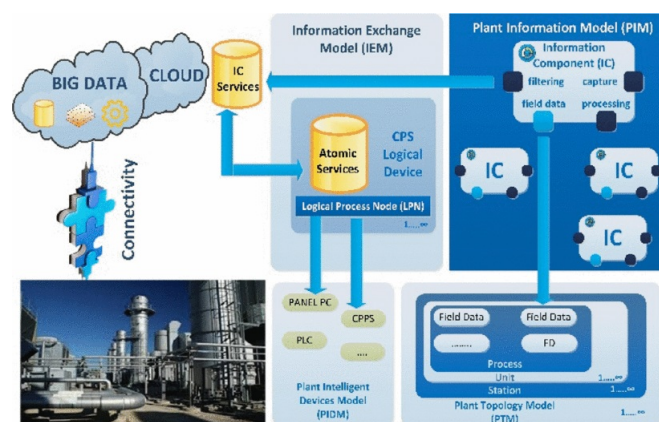


Figura 14: Arquitetura geral do CPPS. Fonte: (Garcia, Irisarri, Pérez, Estévez & Marcos 2016)

O estudo de caso proposto em Garcia, Irisarri, Pérez, Estévez & Marcos (2016) des-

creve um sistema de laboratório que ilustra uma aplicação de automação de uma fábrica em escala. A planta industrial é simulada por uma linha de montagem com três estações da FESTO mostrado na Figura 15.

Como um dos objetivos do trabalho de Garcia, Irisarri, Pérez, Estévez & Marcos (2016) foi proposto uma implementação em plataformas de baixo custo, e para isso as arquiteturas de *hardware* e *software* foram desenvolvidas utilizando SBCs. Um Raspberry Pi 2 modelo B é utilizado como controlador de processo com um servidor OPC UA, um Beagle Bone Black é utilizado como sistema de monitoramento e duas placas Arduino UNO são implementadas como escravos Modbus/TCP para aquisição de dados.

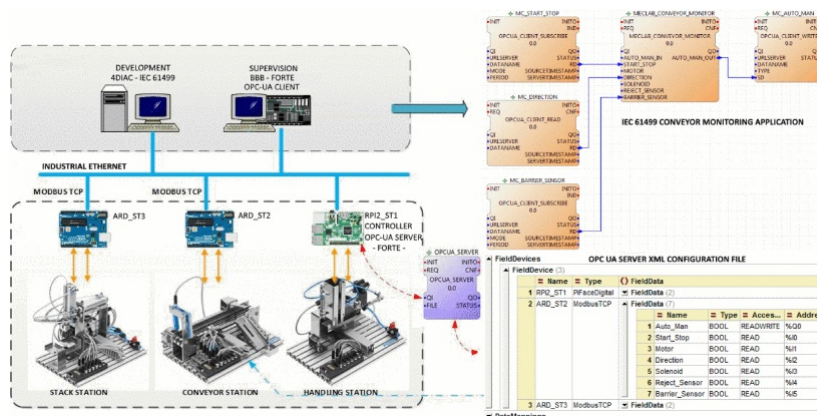


Figura 15: Estudo de caso IEC 61499 e arquivo de configuração XML do OPCUA. Fonte: García, Irisarri, Pérez, Estévez, Orive & Marcos (2016)

O servidor OPC UA e as funcionalidades do cliente foram integrados em uma biblioteca própria implementada usando C++. A biblioteca OPC UA inclui acesso a escravos Modbus/TCP e através dos GPIO do Raspberry Pi 2.

O trabalho apresentado em Müller et al. (2017) segue mesmo princípio onde é apresentada a integração de *software* livre e a aplicação de um servidor OPC UA personalizável em uma placa Arduino Yun usando a biblioteca **Open62541**, que é uma biblioteca aberta e de implementação gratuita do OPC UA. Com o objetivo de proporcionar o desenvolvimento de Sistemas Ciberfísicos de código aberto sem a necessidade de um conhecimento especializado em programação de microcontroladores aproximando aplicativos da Indústria 4.0 para as salas de aula.

A Figura 16 apresenta arquitetura baseada em uma placa de prototipagem Arduino Yun com uma aplicação de controle de temperatura e OPC UA.

O servidor OPC UA é implementado no AR9331 e o programa Arduino (sketch) é executado no ATmega32u4. Com isso toda a parte de comunicação é representada pelo lado Linux, através do AR9331, enquanto o ATmega32u4 representa o lado do Arduino e lida com a interação física com sensores e atuadores. E ainda, a comunicação entre os processadores é via comunicação serial entre si (Müller et al. 2017).

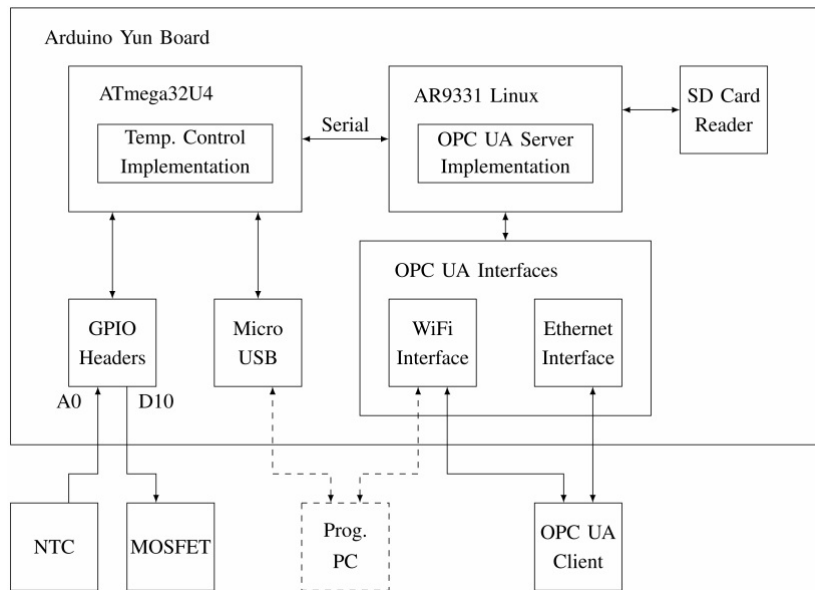


Figura 16: Arquitetura de Arduino Yun com controle de temperatura e aplicação OPC UA. Fonte: Müller et al. (2017)

A implementação do servidor OPC UA foi projetada usando a versão 0.2-rc2 da biblioteca **Open62541**, para a medição de temperatura foi utilizado um termistor NTC EPCOS 100k. A biblioteca PID do Arduino é usada e possibilita o controle em malha fechada. A troca de dados é representada na Figura 17. O procedimento é descrito usando um exemplo: um cliente OPC UA solicita o valor do Parâmetro PID. As etapas de comunicação são representadas por setas numeradas.

1. Um cliente do OPC UA chama o método chamado “*Get PID Parameter P*”.
2. O servidor OPC UA recebe a chamada do método e a manipula. Ele solicita o valor do lado do Arduino via comunicação serial.
3. O lado Arduino recebe o pedido e responde com o valor da variável referenciada.
4. O servidor OPC UA recebe o valor do lado Arduino e o envia como um argumento de saída da chamada de método para o cliente OPC UA.



Figura 17: Procedimento para solicitação de dados do cliente OPC UA e resposta do servidor OPC UA. Fonte: Müller et al. (2017)

Dois clientes OPC UA separados (cliente UaExpert do OPC UA e Android do Prosys OPC UA Client) são usados nesta aplicação, um para definir o valor P do parâmetro PID e o outro para verificar a modificação.

Como resultados desta aplicação os autores salientam que, a implementação aprimora a capacidade do Arduino para comunicação compatível com I4.0. Isso aumenta o potencial do Arduino para aplicações industriais, onde o conceito pode ser usado para desenvolver soluções integradas e personalizadas de acordo com a aplicação Müller et al. (2017).

Em Profanter et al. (2017) foi proposta uma arquitetura hierárquica para um sistema *Plug&Produce* de vários níveis utilizando implementações de código aberto OPC UA para facilitar a integração de novos dispositivos em um sistemas existentes.

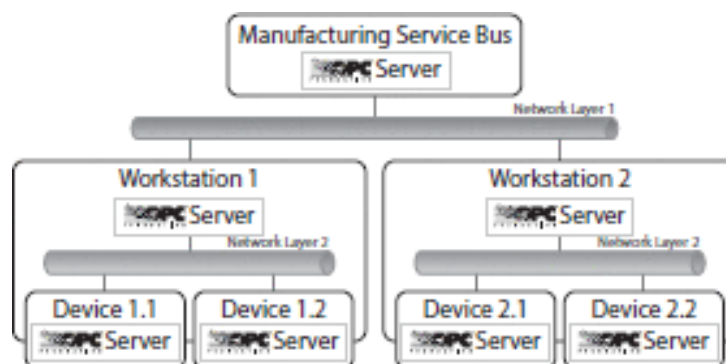


Figura 18: Arquitetura proposta para *plug&produce* hierárquico. Fonte: Profanter et al. (2017)

A arquitetura proposta para a descoberta automática de dispositivos usando os servidores de descoberta local OPC UA com Extensão Multicast (LDS-ME) é visualizada na Figura 18. Esta arquitetura consiste em um "Intelligent Service Bus" (MSB) que é responsável por detectar outros componentes I4.0 na rede e configurar o componente quando ele é conectado.

Os conceitos básicos do OPC UA Discovery Service foram descritos e implementados em diferentes plataformas de código aberto do OPC UA para C/C++ e Java como **open62541** e **Eclipse Milo** por exemplo, por fim uma comparação entre estas plataformas é feita. Para poder criar componentes *Plug&Produce* da Indústria 4.0 foi necessário implementar um conjunto de recursos de acordo com a *OPC UA Specification Part 12* não suportados nas plataformas abertas utilizadas.

A Figura 19 demonstra o funcionamento do processo de descoberta implementado. O barramento de serviço de fabricação controla as estações de trabalho que, por sua vez, controlam os dispositivos.

1. o servidor detecta o servidor LDS-ME através de multicast e registra-se com o lds.
2. o servidor LDS-ME cria um novo cliente para se comunicar com o servidor OPC

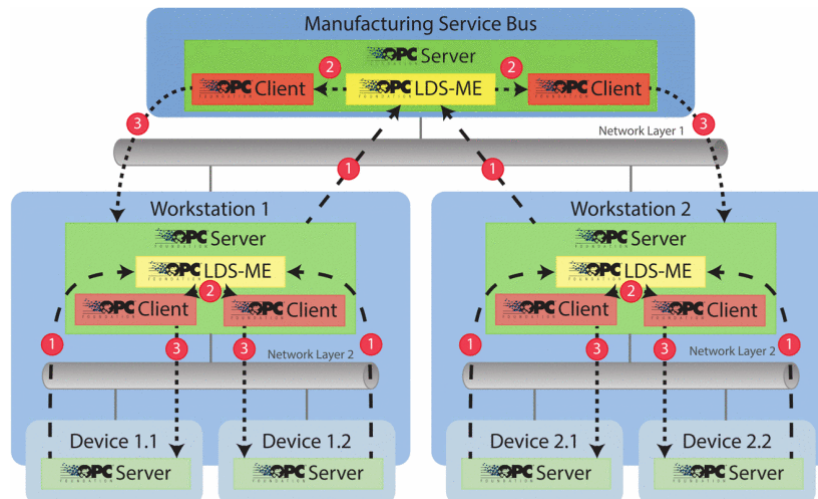


Figura 19: Processo de descoberta com vários níveis de hierarquia. Profanter et al. (2017)

UA.

3. o cliente chama o método de configuração no servidor e controla suas ações.

Para testar as aplicações de acordo com a Figura 19, os autores utilizaram um Desktop-PC com o Linux-Ubuntu, onde o servidor LDS-ME é iniciado diretamente e duas máquinas virtuais são configuradas, para simular as duas estações de trabalho Profanter et al. (2017). Os testes mostraram que ambas as implementações são capazes de se descobrirem mutuamente, mas com uma vantagem significativa para implementação feita com biblioteca **Open62541** que realiza a descoberta dos servidores em menos de 1 segundo.

Em Vimos et al. (2016) foi proposta uma arquitetura baseada no uso do padrão OPC UA e o uso das mensagens MQTT para regular o processo de transmissão de dados entre os sensores e os sistemas de controle de supervisão e aquisição de dados disponíveis na indústria. Os elementos que compõem a arquitetura proposta, apresentados na Figura 20 são:

- Dispositivos ou sensores de *hardware* limitado.
- *Gateway* OPC.
- Servidor OPC.
- Servidor de análise de dados.

Para implementar estes elementos foi utilizada bibliotecas geradas pela própria OPC Foundation, mais especificamente bibliotecas desenvolvidos para JAVA. Para os dispositivos de hardware ou sensores limitados foi utilizado Arduino UNO junto com um módulo de conexão ethernet para a execução de testes de transmissão de dados.

O *Gateway* OPC, apresentado na Figura 21, tem dois papéis basicamente: (1) receber mensagens enviadas por computadores que não podem transmitir dados diretamente

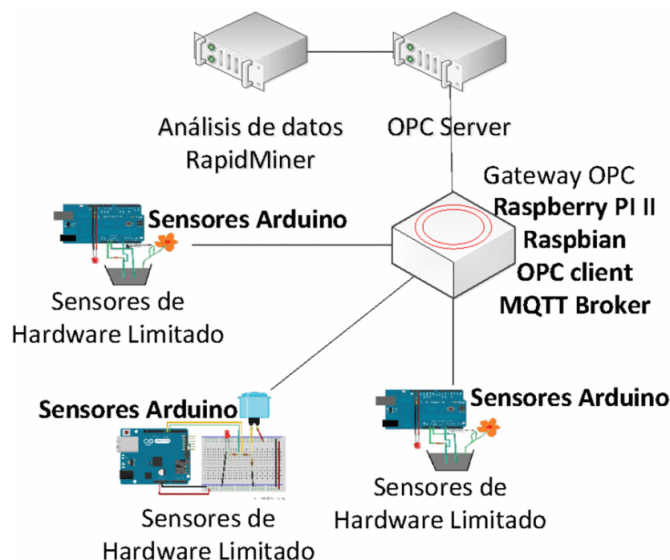


Figura 20: Arquitetura de Hardware proposta em Vimos et al. (2016).

para o servidor OPC usando o padrão, (2) um mediador entre o cliente MQTT e o servidor OPC, pois é responsável por transformar mensagens do formato MQTT à estrutura proposta pelo OPC.

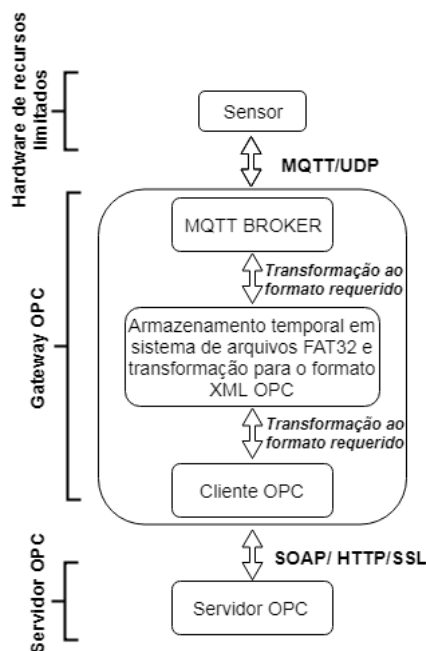


Figura 21: Conversão do formato de dados feito pelo OPC Gateway para transmissão de dados. Fonte: Adaptado de Vimos et al. (2016)

O Servidor OPC UA é responsável por receber e armazenar os dados gerados pelos diferentes sensores. A estrutura dos dados armazenados é baseada em uma modelagem de objetos, que também inclui os métodos necessários para processar eventos, notificações, alarmes ou referências que ajudam a vincular as informações obtidas a cada um dos nós. O Servidor de análise de dados processa e transforma os dados armazenados no Servidor

OPC UA em conhecimento desejado através do uso de algoritmos preditivo ou análise de Inteligência Artificial.

Como resultados desta aplicação os autores salientam o potencial oferecido pelo uso do padrão OPC UA para realizar a transmissão de dados entre os dispositivos desenvolvidos. Onde foi possível verificar que os sensores ou dispositivos desenvolvidos no *hardware* Arduino podem ser integrados em redes mais complexas através do uso de bibliotecas padrão e OPC.

3.2 Análise dos trabalhos relacionados

Nesta seção é realizada uma análise dos trabalhos relacionados apresentados anteriormente. Nos trabalhos propostos por Colombo et al. (2015), Leitão et al. (2015), Rodrigues et al. (2012) são apresentadas arquiteturas para concepção de Sistemas Ciberfísicos que possuem uma alta complexidade, tendo como estudo de caso ambientes industriais reais. Apresentam a utilização de tecnologias que combinam suas características individuais (SoA, *cloud* e MAS) como base para as funções do CPS, assim se justificando com as premissas de concepção dos Sistemas Ciberfísicos. Devido ao fato de ser aplicações mais complexas utilizam como *hardware* componentes mais elaborados como 8 computadores ou até mesmo a nuvem.

Por sua vez, os trabalhos apresentados em García, Irisarri, Pérez, Estévez, Orive & Marcos (2016), Garcia, Irisarri, Pérez, Estévez & Marcos (2016) e Müller et al. (2017), focam na concepção de Sistemas Ciberfísicos de baixo custo baseado no padrão OPC UA. A utilização de tecnologias como M2M e SoA, com a utilização do OPC UA, também pode ser observadas nestas aplicações. Apresentam a implementação em dispositivos SBCs, bem como placas de prototipagem Arduinos para aquisição de dados em seus componentes de *hardware* salientando a compatibilidade desses componentes com a I4.0, porém são aplicados em estudos de caso mais simples se comparado com os anteriores.

Por fim, nos trabalhos apresentados em Profanter et al. (2017) e Vimos et al. (2016) foi utilizado o padrão OPC UA para resolver mais especificamente a descoberta de dispositivos e transmissão de dados entre os componentes do sistema. São utilizados no primeiro caso para implementação um Desktop-PC com Ubuntu-Linux instalado e no segundo caso mais uma vez é utilizado dispositivos SBC combinados com placas de prototipagem Arduino para aquisição dos dados.

Uma relação de cada trabalho com sua aplicação, elencando as tecnologias no âmbito da Indústria 4.0 utilizadas bem como características de sua implementação como Hardware utilizado e demais especificações foram organizadas e apresentadas na Tabela 1 a seguir.

Tabela 1: Análise dos trabalhos relacionados

Trabalhos	Aplicação	Tecnologias I4.0	Implementação(HW e especificações)
Colombo et al. (2015)	Sistemas industriais reais	SoA e Cloud	Dispositivos embarcados inteligentes e paradigmas computacionais como a nuvem
Leitão et al. (2015)	Sistemas industriais reais	SoA e SMA	Não especificado, mas por envolver algoritmos inteligentes e SMA necessitam de um sistema de HW mais elaborado
Rodrigues et al. (2012)	Sistemas industriais reais	SMA	Os SMA foi implementado sob 8 computadores dispostos ao longo do chão de fábrica
García, Irisarri, Pérez, Estévez, Orive & Marcos (2016)	Plantas industriais didáticas	M2M e SoA	Por promover um sistema de baixo custo foram utilizados SBCs, como Raspberry e Beaglebones e paradigmas computacionais como a nuvem.
Garcia, Irisarri, Pérez, Estévez & Marcos (2016)	Plantas industriais didáticas	M2M e SoA	Por promover um sistema de baixo custo foram utilizados SBCs, como Raspberry e Beaglebones e paradigmas computacionais como a nuvem.
Müller et al. (2017)	Aplicação didática	SoA (OPC UA)	Foram utilizados microprocessadores(AR9331) e Placas de prototipagem Arduino.
Profanter et al. (2017)	Descoberta de dispositivos simulação de estações de trabalho	OPC UA	Foram utilizado um Desktop-PC com Linux-Ubuntu, para implementação dos servidores/clientes e duas máquinas virtuais para simulação de estações de trabalho
Vimos et al. (2016)	Comunicação de dispositivos simulação de aplicações industriais	OPC UA	Foram utilizados SBCs, placas de prototipagem Arduino e computadores

Das características apontadas na Tabela 1, vale ressaltar diferentes tipos de *Hardware*, onde sistemas mais complexos, aqueles aplicados em casos industriais, exigem uma plataforma mais elaborada de seus componentes devido ao nível computacional agregado. Porém, a utilização de dispositivos mais simples se torna viável em aplicações de simulação de plantas industriais didáticas mas que se aproximam de exemplos reais, promovendo então a utilização de SBCs de diferentes configurações.

Ao longo da última década, o padrão OPC alcançou ampla adoção na indústria, onde sua versão clássica já esta difundida no modelo de indústria atual resultante da 3ª Revolução Industrial. Com o surgimento de novos conceitos como CPS, IoT, IIoT, a aplicação de padrões de sistemas baseadas em OPC UA vem sendo cada vez mais comum. No contexto de manufatura e automação esses sistemas podem ser referidos a diferentes etapas e processos industriais, por exemplo, de fabricação, design, simulação, controle, entre outros.

Assim o presente trabalho, de certa forma, apodera-se dessas características como tecnologias facilitadoras da Indústria 4.0 fundamentadas na concepção de CPS, bem como a utilização de *Hardware* baseados em computadores de placa única, que estão cada vez mais difundidos na pesquisa para realização desses sistemas. Também acrescenta a essas características uma classificação de dispositivos para que auxilie a escolha de acordo com as especificações de sua aplicação.

4 ARQUITETURA PROPOSTA E DEFINIÇÃO DE HARDWARE

O objetivo desse capítulo é apresentar uma arquitetura para a construção de um CPS utilizando dispositivos embarcados tomando como base os conceitos e premissas da arquitetura CPS 5C e Indústria 4.0. A aplicação de Sistemas Ciberfísicos no contexto de manufatura tem se mostrado muito promissora, embora apresente algumas arquiteturas como a própria 5C e como visto em alguns dos trabalhos relacionados no capítulo 3 carece de arquiteturas, metodologias e estratégias que direcionem a uma implementação mais específica.

A estratégia adotada nessa proposta visa a construção de um ambiente formado por diversos dispositivos embarcados, dotados de um comportamento autônomo. Ela também pode apresentar elementos de controle, através de algoritmos como o PID (*Proporcional Integral Derivativo*) e inteligência artificial, promovendo o controle distribuído e a descentralização.

A partir da definição da Arquitetura de Software e do conhecimento dos benchmarks dos dispositivos SBCs analisados, o objetivo é definir a capacidade mínima de cada dispositivo em rodar os módulos da arquitetura proposta. Nesse sentido, no pior dos cenários, cada SBC deve ser apto a executar pelo menos 1 módulo da arquitetura. Os demais módulos podem ser incluídos de acordo com o benchmark do dispositivo.

4.1 Arquitetura de Software

A arquitetura de software proposta para o desenvolvimento do projeto, está organizada em cinco módulos: configuração, comunicação, inteligência, conversão e cibernético. Visto que a arquitetura foi baseada na arquitetura CPS 5C, apresentada na Seção 2.1.2.5, a Figura 22 relaciona cada módulo com os seus cinco níveis (conexão inteligente, conversão de dados para informação, ciberfísico, cognição e configuração).

O módulo de Comunicação compreende o nível cinco ou conexão inteligente da Arquitetura 5C. É nesse nível que uma massiva aquisição de dados, oriunda de diversos dispositivos, é realizada. Este módulo é responsável pela conexão de diferentes dispositi-

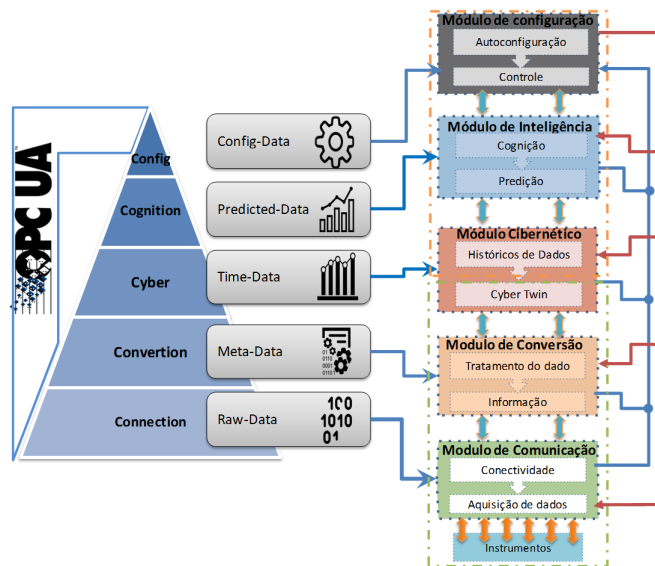


Figura 22: Associação dos módulos com a arquitetura CPS 5C.

vos, protocolos e padrões, garantindo funcionalidades *plug&play*.

O módulo de Conversão é responsável por persistir os dados das variáveis gerenciadas e converter os dados vindos das máquinas ou componentes em informações relevantes.

O módulo Cibernético está relacionado com o nível três da arquitetura 5C. É responsável por gerar e armazenar o modelo topológico bem como garantir uma concentração de dados que servem como base para a criação da representação virtual ou *Twin Model* de componentes e máquinas.

O módulo de inteligência está relacionado com a camada de cognição. Tem como características a utilização de algoritmos inteligentes para previsão de falhas, desempenho e saúde dos componentes e auxiliar a tomada de decisão e o controle distribuído promovido pelo módulo de configuração.

Já o módulo de configuração é responsável pela autoconfiguração de todos os dispositivos bem como a organização de todos os demais módulos da arquitetura. É nele que será implementada toda a interface para interação com os operadores e onde o sistema pode ser ajustado de acordo com a aplicação.

A Figura 23 apresenta a disposição, bem como funcionalidades dos módulos da arquitetura de software. A inserção do padrão OPC UA, serve para garantir que a comunicação entre todos os componentes dispostos no Sistema Ciberfísico seja possível.

- **Módulo de Configuração:** responsável pelas interfaces de configuração de todos os módulos. Nele são definidas interfaces para acessar e definir as propriedades do modelo topológico, propriedades de comunicação, controle, inteligência e armazenamento. Responsável por viabilizar o controle distribuído através de algoritmos de controle como o PID (*Proporcional Integral Derivativo*) ou inteligência artificial (redes neurais). Esse módulo serve para promover o controle distribuído, dotando

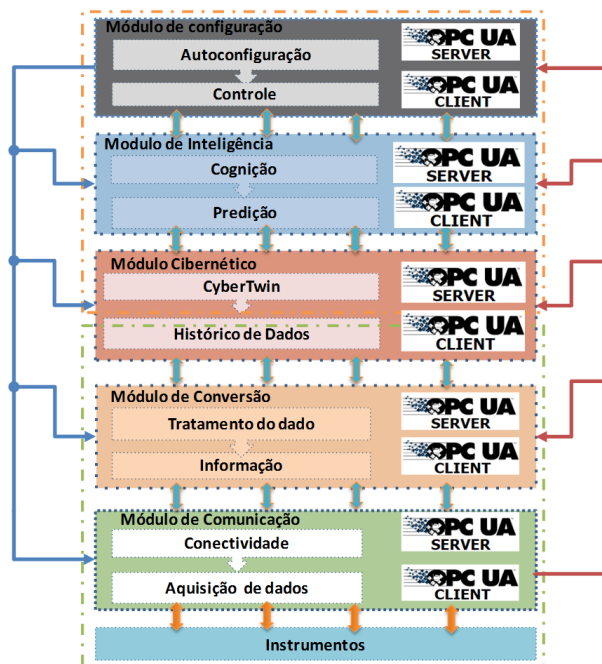


Figura 23: Arquitetura de software genérica disposta em módulos com a inserção de funcionalidades OPC UA.

o dispositivo de capacidade de gerenciar um conjunto de variáveis de forma local e prover as variáveis de controle para outros dispositivos na rede;

- Módulo de Cognição: utiliza algoritmos avançados para apontar falhas e promove o comportamento preditivo e cognitivo do sistema, além de auxiliar a tomada de decisão e controle distribuído;
- Módulo de Cibernético: módulo responsável pelo mapeamento e rastreamento de padrões, comportamentos e o controle de qualidade dos dados. É formado por um *buffer*, que registra a base de dados externa que é habilitada através da aquisição de dados;
- Módulo de Conversão: responsável por persistir os dados das variáveis gerenciadas e por converter os dados vindos das máquinas ou componentes em informações relevantes;
- Módulo de Comunicação: responsável pela aquisição de dados e adaptar os diferentes protocolos de redes industriais para o padrão adotado no sistema, além de viabilizar a interoperabilidade por meio desse padrão. O módulo também deve disponibilizar o modelo com a descrição topológica do equipamento para os demais componentes por meio da rede.

4.2 Topologia geral de hardware

Visto que uma das premissas dos sistemas ciberfísicos é garantir a ampla comunicação entre todos os componentes do sistema, a Figura 24 apresenta de forma genérica a topologia geral para uma implementação da arquitetura de *software* proposta.

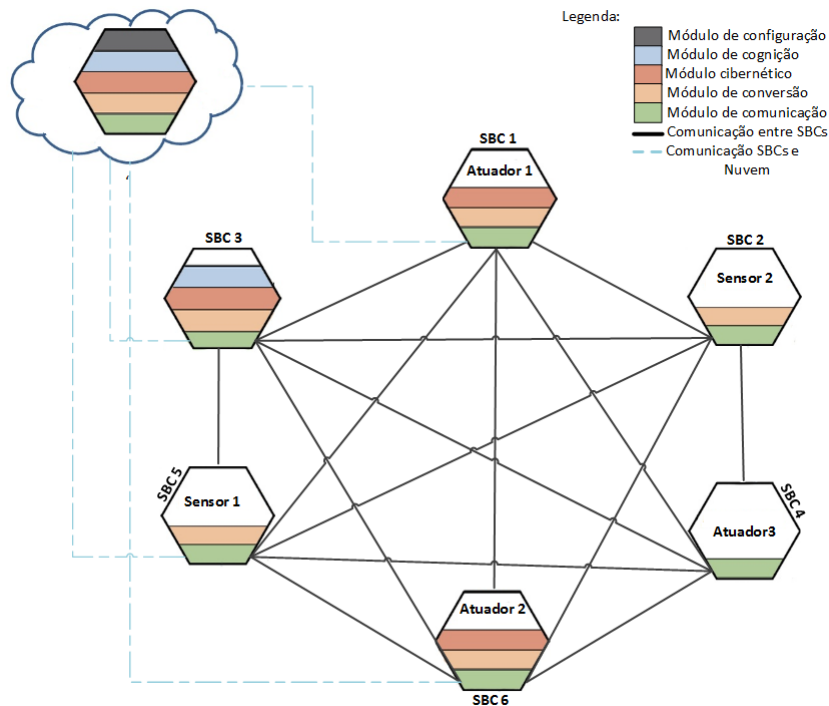


Figura 24: Exemplo de topologia genérica do sistema.

O sistema é composto por diversos dispositivos SBCs dispostos na planta. Cada SBC pode ser associado a um instrumento, sensores, atuadores, entre outros, que serão agregados aos módulos da arquitetura de *software*. De acordo com as necessidades apresentadas pela aplicação, um conjunto SBC/Instrumento pode conter um ou mais módulos da arquitetura.

O único módulo exigido em todos os dispositivos SBCs é o módulo de comunicação, uma vez que garante a conectividade entre todos os demais. A união entre todos os dispositivos físicos, bem como de outros paradigmas computacionais, como a nuvem, realizam a concepção do CPS como um todo.

4.3 Classes ou grupos de hardware para o CPS

Nesta seção será apresentada uma classificação de dispositivos de *hardware* para o desenvolvimento e implementação ou simulação dos módulos da Arquitetura apresentada. Um SBC, como visto anteriormente, é um computador embarcado de forma reduzida e pronto para uso. Pode ser considerado com uma solução genérica em *Hardware* e *Software* que pode ser utilizada no desenvolvimento de sistemas embarcados. Oferecem uma

plataforma completa para o desenvolvimento de produtos finais para diversas aplicações como médicas, automação industrial, aeroespacial e robótica.

Para a classificação, características como Capacidade de processamento, Memória disponível, Dispositivos de E/S, Conectividade e Suporte a *Software* foram analisados. As classes por sua vez foram definidas utilizando apenas as informações de capacidade de processamento e memória disponível de cada *Hardware*, auxiliada por *Benchmarks* que serão apresentados a seguir.

4.3.1 Benchmarks

Existe uma infinidade de usos para dispositivos SBCs hoje em dia. Uma boa maneira de definir qual é o dispositivo mais adequado para sua aplicação, é sujeitá-lo a *benchmark*, como processamento gráfico, consumo de energia, capacidade de processamento na CPU, disponibilidade de memória RAM, entre outros.

A intenção desses testes é dar uma ideia do desempenho relativo entre alguns dispositivos SBCs. Para o auxílio da classificação foi realizada uma série de CPU *Benchmark*. Foram utilizados alguns testes dentro do *Phoronix Test Suite (PTS)* Phoronix (2018), que é uma ferramenta gratuita de *Benchmarking* de código aberto para Linux. Os testes de CPU foram aplicados em SBCs correspondentes as famílias *Odroid*, *Raspberry Pi* e *Banana Pi*. A Tabela 2 apresenta os modelos de SBCs de cada família que foram utilizados para a análise.

Os seguintes testes foram aplicados (Phoronix 2018):

- **John the Ripper Blowfish:** É um teste de criptografia. Seu resultado é a quantidade de criptografias realizadas por segundo. Quanto maior esse valor melhor é o seu desempenho;
- **C-Ray:** É um *raytracer*, ou seja, renderização de imagens tridimensionais projetado para testar o desempenho de ponto flutuante do processador. Seu resultado é em segundos e quanto menor for o tempo de execução melhor;
- **Smallpt:** É um renderizador de iluminação global em C ++ escrito em menos de 100 linhas de código. Seu resultado também é em segundos e quanto menor for o tempo de execução melhor;
- **Himeno:** Este *benchmark* faz medições para Executar grandes *loops* na solução da equação de Poisson usando o método de iteração de Jacobi. O resultado desde teste é dado em FLOPS (operações de ponto flutuante por segundo). Este valor é usado para determinar o desempenho de um computador. Quanto maior a quantidade de FLOPS, melhor o seu desempenho;

Tabela 2: Especificação dos Dispositivos SBC

SBC	Processamento	Entrada/Saída	Memória	Conectividade	Sistema Operacional
Odroid XU4	Samsung Exynos5422 Cortex™-A15 2.0GHz and Cortex-A7 Octa-Core	2x USB 3.0, 1x USB 2.0, 42 Pin : GPIO/IRQ/SPI /ADC,/I2S/I2C	2GB LPDDR3	Porta Ethernet RJ-45 Antena Wi-Fi (Adaptador externo USB)	Ubuntu, Android, ARCHLinux, Debian,
Odroid C2	Amlogic ARM® Cortex® -A53(ARMv8) 1.5Ghz quad core	4X USB 2.0 40 pin GPIOs 7pin I2S/I2C/UART/ADC	2GB LPDDR3	Porta Ethernet RJ-45 Antena Wi-Fi (Adaptador externo USB)	Ubuntu, Android, ARCHLinux, Debian,
Odroid C1+	Amlogic S805 SoC 4 x ARM® Cortex®- A5 1.5GHz	4 x USB 2.0 , 40 pin port(GPIO/UART/ SPI/I2C/ADC) 7pin port (I2S)	1GB DDR3	Gigabit Ethernet USB IEEE 802.11b/g/n 1T1R WLAN com Antena	Ubuntu, Android, Fedora, ARCHLinux, Debian
Raspberry Pi 3	Broadcom BCM2837 4 x ARM Cortex-A53 1.2Ghz	4 x USB 2.0, 40 pin port GPIO / UART / SPI / I2S	1GB LPDDR2	Porta Ethernet RJ-45 Antena Wi-Fi (Adaptador externo USB) Bluetooth 4.1	Raspbian e outros
Raspberry Pi 2	Broadcom BCM2836 4 x ARM® Cortex®-A7 900MHz ARMv7	4 x USB 2.0, 40pin port (GPIO/UART /SPI/I2C/I2S)	1GB LPDDR2	Porta Ethernet RJ-45	Raspbian e outros
Raspberry Pi 1	Broadcom BCM2835 ARM11 700Mhz Single-core	2 x USB 2.0, 40pin GPIO header	512 GB LPDDR2	Porta Ethernet RJ-45	Raspbian e outros
Raspberry Pi 0 W	Broadcom BCM2835 ARM11 de 1GHz Single-core	2x Micro USB 40pin GPIO header	512 GB LPDDR2	Porta Ethernet Wi-Fi: 802.11 b/g/n Bluetooth 4.1	Raspbian e outros
Banana Pi M2U	A31S ARM Cortex A7 quad-core - 1GHz	3x USB 2.0, USB 40 Pins: GPIO, UART, I2C, I2S , SPI , PWN,	1GB DDR3	Wi-Fi 802.11 b/g/n Bluetooth BT4.0	Android,Ubuntu, Debian,Bananian

- **MAFFT**: Este teste realiza um alinhamento de 100 sequências de um determinado tipo de composto químico. Este teste também retorna o resultado em segundos e quanto menor for o tempo de execução melhor;
- **OpenSSL**: O OpenSSL é um kit de ferramentas de código aberto que implementa os protocolos *SSL (Secure Sockets Layer)* e *TLS (Transport Layer Security)*. Este teste mede o desempenho *RSA (Rivest-Shamir-Adleman)* 4096-bit do OpenSSL. O resultado desse teste é dado em sinais por segundo, quanto maior for essa relação melhor o desempenho do computador.
- **Encode FLAC**: Esse teste leva em consideração quanto tempo leva para codificar um arquivo *WAV (WAVE form Audio Format)* de amostra para o formato *FLAC (Free Lossless Audio Codec)* cinco vezes. Este teste também retorna o resultado em segundos e quanto menor for o tempo de execução, melhor.

Estes testes tem como características exigir o máximo de um processador, características essas, que são entendidas como similares e presentes nos sistemas ciberfísicos em geral. Por exemplo, pode-se citar a implementação das camadas da arquitetura 5C onde, à medida que vai se agregando mais inteligência, ou seja, implementando mais camadas da arquitetura, a exigência desses dispositivos também tende a aumentar. Com isso foram obtidos os seguintes resultados:

As Figuras 25, 26, 27 e 28 apresentam os resultados obtidos através dos testes C-Ray, Encode Flac, MAFFT e Smallpt respectivamente, que tem como resultado o tempo de execução dado em segundos. Quanto menor for o tempo de execução melhor o desempenho dos dispositivos.

Na Figura 29 pode-se observar os resultados obtidos através dos testes Blowfish, que é dado em Criptografias por segundo (C/s). Já a Figura 30 apresenta o gráfico de resultados do teste Himeno, que tem como resultado MFLOPs. Por fim, a Figura 31 demonstra o gráfico do teste OpennSSL que resulta em Sinais por segundo(S/s). Nestes três testes o quanto maior for a relação entre C/s, MFLOPs, S/s caracterizam o melhor desempenho dos dispositivos.

Visto que os demais testes aplicados tem como resultado tempo de execução, dado em segundos, uma relação entre os testes Blowfish, Himeno e OpenSSL foi necessária para normalizar as unidades dos resultados facilitando a análise dos dispositivos e consequentemente a definição dos grupos.

Analisando o resultado obtido no teste Blowfish por exemplo, a relação foi feita primeiramente adotando como referência o tempo de execução do teste realizado no dispositivo que apresentou o pior resultado. O SBC Raspberry Pi 1 (C3)¹ (59.4 C/s - 180s)

¹Os SBCs indicados como (C3) são os dispositivos disponíveis no Centro de Ciências Computacionais para realização dos testes.

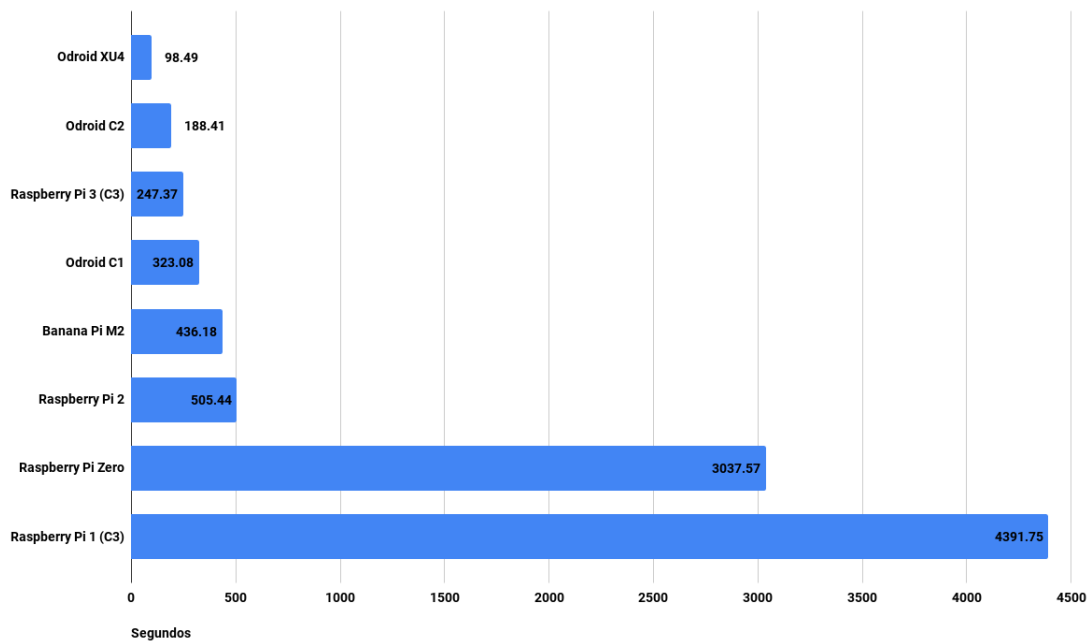


Figura 25: Resultado em segundos do teste *Benchamrks C-Ray*.

com um total de 10692 Criptografias realizadas. Já o Raspberry Pi 3 (C3) realizou cerca 629 C/s - 180s com um total de 113220 criptografias. Sendo assim, é aplicada uma regra de três simples para determinar o tempo de execução do total de criptografias realizadas no Raspberry Pi 1(C3) para o dispositivo SBC Raspberry Pi 3 (C3), onde:

113220 criptografias — 180 segundos

10692 criptografias — X segundos

$$113220X = 180 * 10692 \quad (1)$$

$$X = \frac{1924560}{113220} \quad (2)$$

$$X = 16.998 \quad (3)$$

Assim, é obtido o tempo de aproximadamente 17 segundos para execução de 10692 criptografias no Raspberry pi 3 (C3). Uma regra geral então pode ser descrita para o restante dos dispositivos, da seguinte forma.

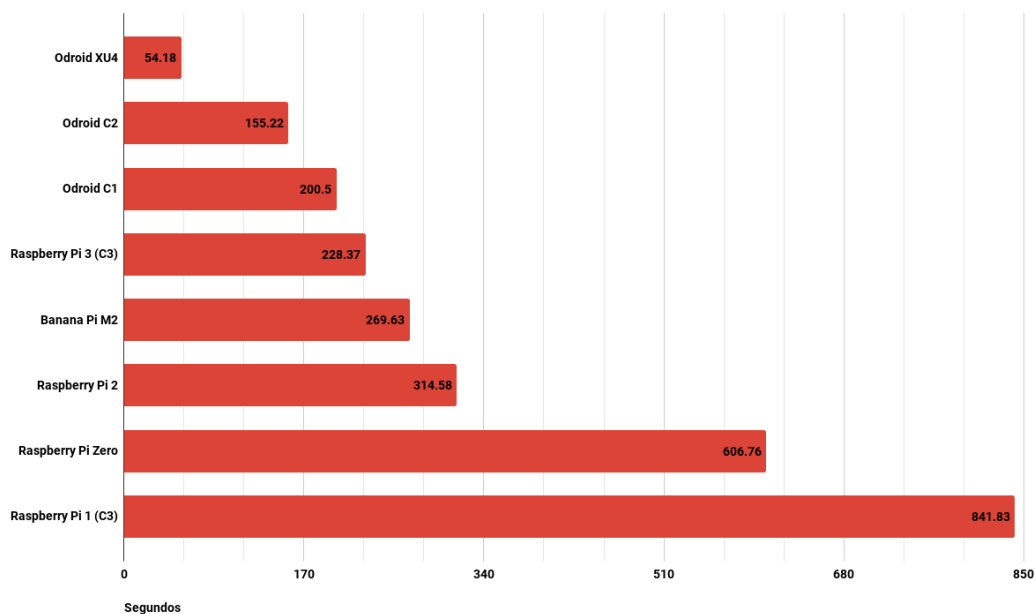


Figura 26: Resultado em segundos do teste *Benchmarks Encode-Flac*.

Total criptografias no SBC — Tempo de execução em segundos

Total criptografias no Rasp 1 — X segundos

O total de criptografias realizadas por dispositivos SBC em 180 segundos está demonstrada na Tabela 3.

Tabela 3: Total de Criptografias por dispositivo SBC

SBC	C/S	Criptografias
Odroid XU4	1043	187740
Odroid C2	866	155880
Raspberry Pi 3 (C3)	629	113220
Odroid C1	494	88920
Banana Pi M2	427	76860
Raspberry Pi 2	387	69660
Raspberry Pi Zero	88.73	15971.4
Raspberry Pi 1 (C3)	59.4	10692

O mesmo procedimento pode ser feito para o teste Himeno e para o teste OpenSSL. No teste Himeno o SBC Raspberry Pi 1 (C3) realizou 8.3 Mflops, em 180s de execução com um total de 1494×10^6 operações realizadas. Já o Raspberry Pi 3 (C3) realizou 72.08 Mflops em 180s totalizou 11257.2×10^6 operações. Sendo assim a mesma regra aplicada no teste anterior pode ser feita, porém analisando agora as operações em ponto flutuante realizadas:

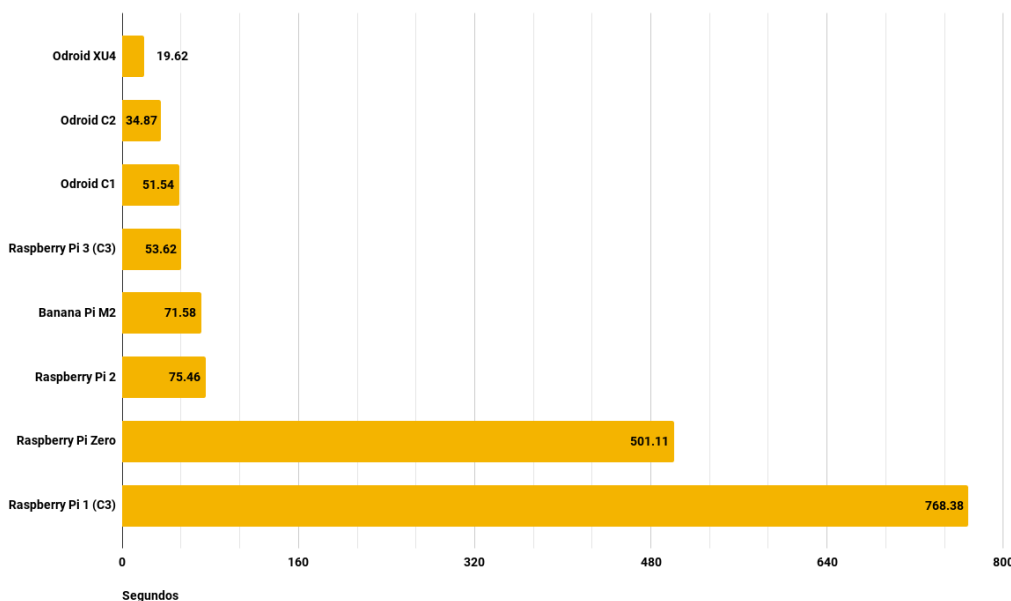


Figura 27: Resultado em segundos do teste *Benchmarks MAFFT*.

Total Operações no SBC — Tempo de execução em segundos

Total Operações no Rasp 1 — X segundos

Substituindo com os valores obtidos temos que:

$$11257.2 \times 10^6 \text{ operações} \text{ — } 180 \text{ segundos}$$

$$1494 \times 10^6 \text{ operações} \text{ — } X \text{ segundos}$$

Assim, é obtido o tempo de aproximadamente 24 s para execução de 1494×10^6 operações de ponto flutuante no Raspberry pi 3 (C3). A Tabela 4 apresenta o total de operações de ponto flutuante obtidos em cada dispositivo SBC.

No teste OpenSSL o SBC Raspberry Pi 1 (C3) processou 1.8 sinais por segundo, o tempo total de execução deste teste foi 66s resultando um total de 118.8 sinais processados. Já o Raspberry Pi 3 (C3) processou 20.55 sinais por segundo tomando os mesmos 66s totalizou 1356.3 sinais processados. Aplicando a mesma regra anterior, tem-se:

Total Sinais no SBC — Tempo de execução em segundos

Total Sinais no Rasp 1 — X segundos

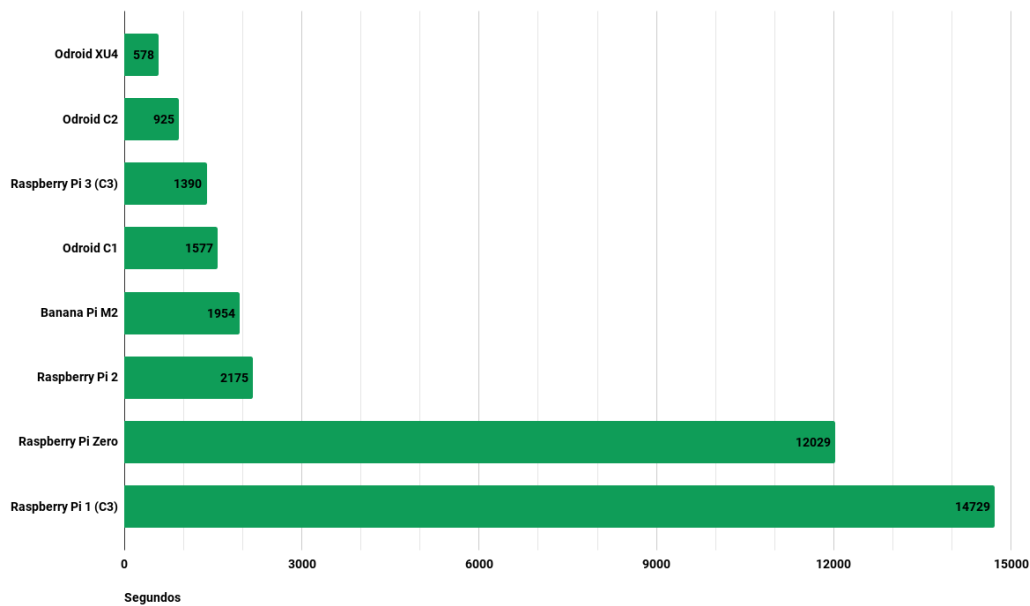


Figura 28: Resultado em segundos do teste *Benchmarks Smallpt*.

Tabela 4: Total de Operações de ponto flutuante por dispositivo SBC

SBC	MFlops	Total de Operações(10^6)
Odroid XU4	164.73	29651.4
Odroid C1	81.97	14754.6
Odroid C2	72.08	12974.4
Raspberry Pi 3	62.54	11257.2
Banana Pi M2	60.08	10814.4
Raspberry Pi 2	33.29	5992.2
Raspberry Pi Zero	10	1800
Raspberry Pi 1	8.3	1494

Substituindo com os valores obtidos temos que:

1356.3 sinais — 66 segundos

118.8 sinais — X segundos

Resultando em um tempo de aproximadamente 5.8 s para o processamento de 118.8 sinais no Raspberry pi 3 (C3). A Tabela 5 apresenta o total de sinais processados por cada dispositivo SBC em 66 segundos.

Uma vez que os dados apresentados nas Tabelas 3,4,5 apresentam o total de criptografias, operações e sinais, para cada dispositivo SBC aplicando-se a regra utilizada para o SBC Raspberry Pi 3 (C3) nos demais SBC's, a Tabela 6 apresenta os resultados convertidos para segundos em todos dispositivos SBCs.

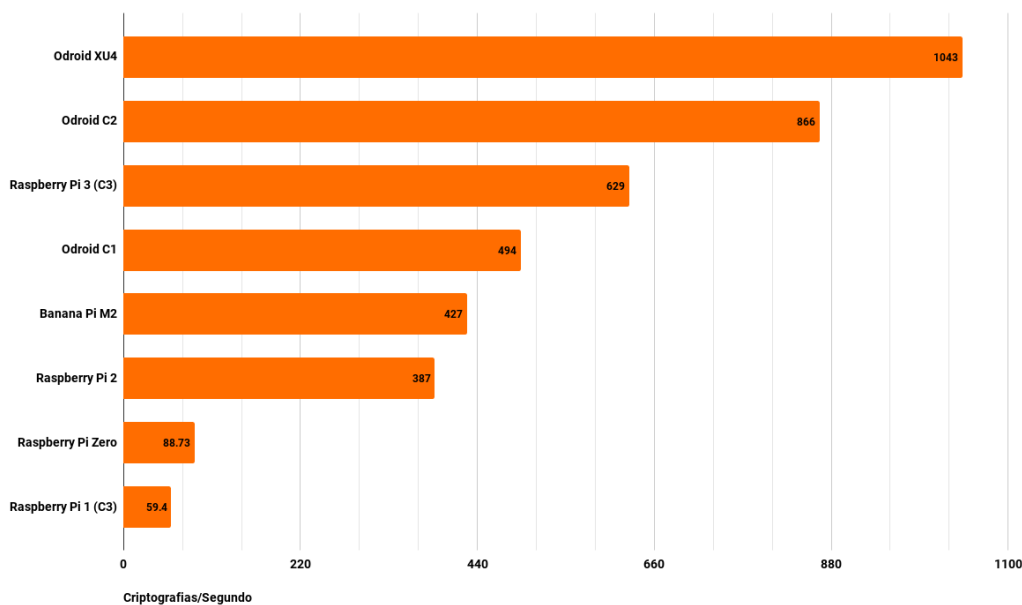


Figura 29: Resultado em C/s do teste Benchmarks BlowFish.

Tabela 5: Total de Sinais processados por dispositivo SBC

SBC	Sinais/segundo	Total de Sinais
Odroid XU4	50.03	3301.98
Odroid C2	21.07	1390.62
Raspberry Pi 3	20.55	1356.3
Odroid C1	18.8	1240.8
Banana Pi M2	13.4	884.4
Raspberry Pi 2	11.93	787.38
Raspberry Pi Zero	2.6	171.6
Raspberry Pi 1	1.8	118.8

Uma vez que todos os testes apresentam a mesma unidade, o gráfico da Figura 32 demonstra os resultados em segundos, dos testes C-Ray, Encode Flac, MAFFT, SmalPT juntamente com os resultados convertidos em segundos dos testes Blowfish, Himeno e OpenSSL, aplicados nos dispositivos SBC. Nesses testes, os melhores dispositivos são aqueles que apresentam um menor tempo de execução. Para auxiliar na definição dos grupos de dispositivos SBC, visto que, os testes agora apresentam a mesma unidade, foi utilizado o somatório entre todos os testes, como pode-se verificar na Figura 33.

Analisando o gráfico da Figura 33 foram definidos quatro grupos distintos (A, B, C e D). Dois grupos são mais notáveis, visto que representam os extremos no gráfico, como melhor e pior desempenhos: o Grupo A e o Grupo D respectivamente. Os grupos B e C, se justificam se analisar os dados de processamento disponíveis na Tabela 2 e com este auxílio, os grupos são definidos a seguir:

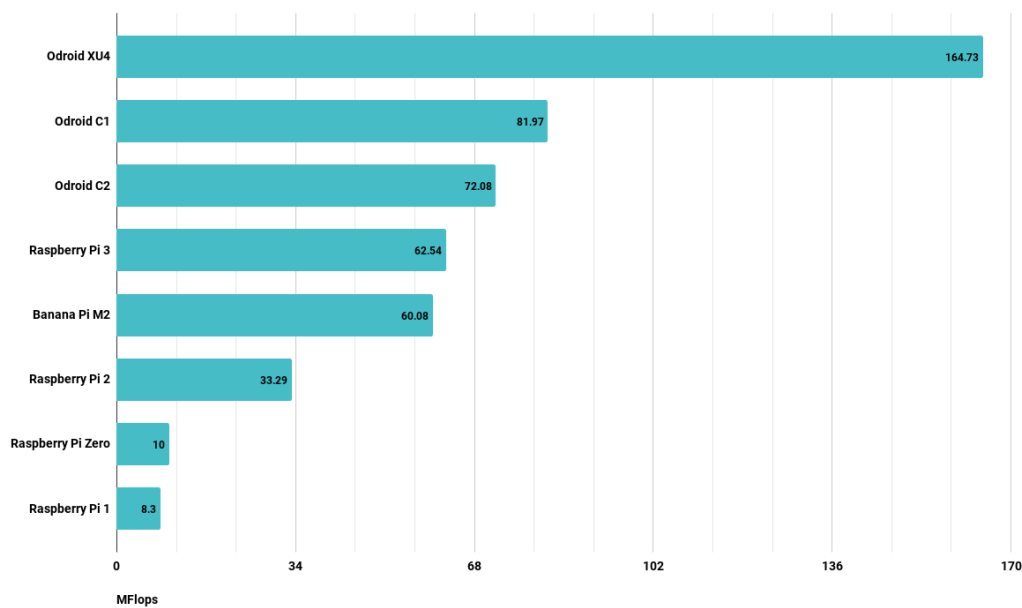


Figura 30: Resultado em MFlops do teste Benchamrks Himeno.

Tabela 6: Relação da transformação dos resultados obtidos nos testes Blowfish, Himeno e OpenSSL para segundos.

SBC	Blowfish		Himeno		OpenSSL	
	<i>C/S</i>	<i>Segundos</i>	<i>MFlops</i>	<i>Segundos</i>	<i>Sinais/segundo</i>	<i>Segundos</i>
Odroid XU4	1043	10.25	164.73	9.07	50.03	2.4
Odroid C2	866	12.35	81.97	18.23	21.07	5.6
Raspberry Pi 3 (C3)	629	17	72.08	20.73	20.55	5.8
Odroid C1	494	21.64	62.54	23.89	18.8	6.3
Banana Pi M2	427	25.04	60.08	24.87	13.4	8.9
Raspberry Pi 2	387	27.63	33.29	44.88	11.93	10.0
Raspberry Pi Zero	88.73	120.5	10	149.4	2.6	45.7
Raspberry Pi 1 (C3)	59.4	180	8.3	180	1.8	66

- Grupo A:** Dentro desta classe encontram-se dispositivos com uma grande capacidade de processamento, com processadores com 8 núcleos podendo chegar a 2 GHz, e com diferentes capacidades de memórias até 2 GB. Através dos gráficos apresentados é possível notar que o modelo SBC Odroid XU4 apresentou os melhores resultados de desempenho;
- Grupo B:** Este grupo tem como característica modelos SBC equipados com processadores quad-core de 1.2 Ghz até 2 GHz e memória até 2 GB. Dos modelos apresentados, Raspberry Pi 3 B, Odroid C1 e Odroid C2 fazem parte desse grupo devido a estabelecerem um intervalo em comum no gráfico;
- Grupo C:** Fazem parte deste grupo dispositivos SBC que dispõem de processadores Quad-Core de 1 GHz e memória de até 1 GB. Os dispositivos testados com estas

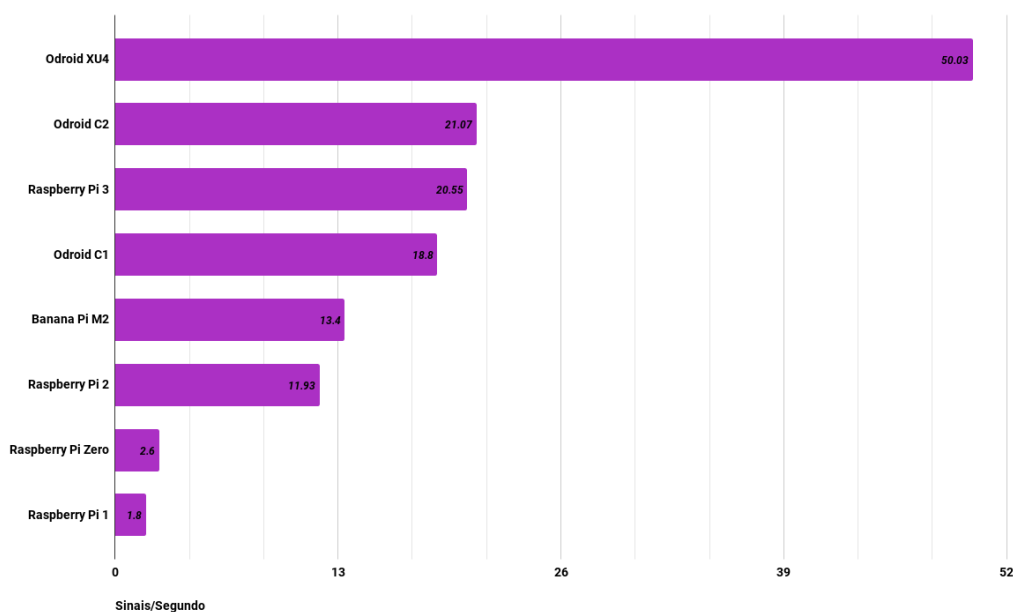


Figura 31: Resultado em S/s do teste Benchmarks OpenSSL.

características e que apresentaram uma semelhança nos resultados na maioria dos testes são os modelos SBC Raspberry Pi 2 e Banana Pi M2;

- **Grupo D:** O Grupo D de SBCs é caracterizada por dispositivos que apresentam uma taxa de processamento com processadores Single-Core de até 1 GHz. Outro ponto característico desta classe é a baixa capacidade de memória podendo chegar até 512 MB. Dos dispositivos testados que representam este grupo estão o Raspberry Pi 0 e o Raspberry Pi 1, que consequentemente apresentaram os piores resultados em todos os testes aplicados;

As características adotadas para definição das classes de HW, capacidade de processamento e memória disponível, servirão como base para implementações de diferentes tipos de aplicações. Como visualizado na Figura 34, admitindo que a demanda de exigência computacional aumenta dadas as atribuições e funcionalidades de cada camada, a escolha certa do dispositivo para um determinada aplicação deve ser realizada.

Em outras palavras, aumentar o nível de computação significa a inserção de tecnologias de software e hardware que auxiliam a realização das funções de cada camada. Uma vez que é compreensível que na camada mais inferior a demanda computacional exigida é menor que as demais camadas.

De acordo com as classes estabelecidas, um conjunto de características como níveis de inteligência, cognição e controle, compatíveis com a estrutura do CPS, serão adicionadas a cada componente da classe, ou até mesmo quantas camadas da arquitetura 5C serão contempladas em cada dispositivo. Pressupõe-se que dispositivos com menor poder computacional, por exemplo o Raspberry Pi 1, seja capaz de executar somente as camadas

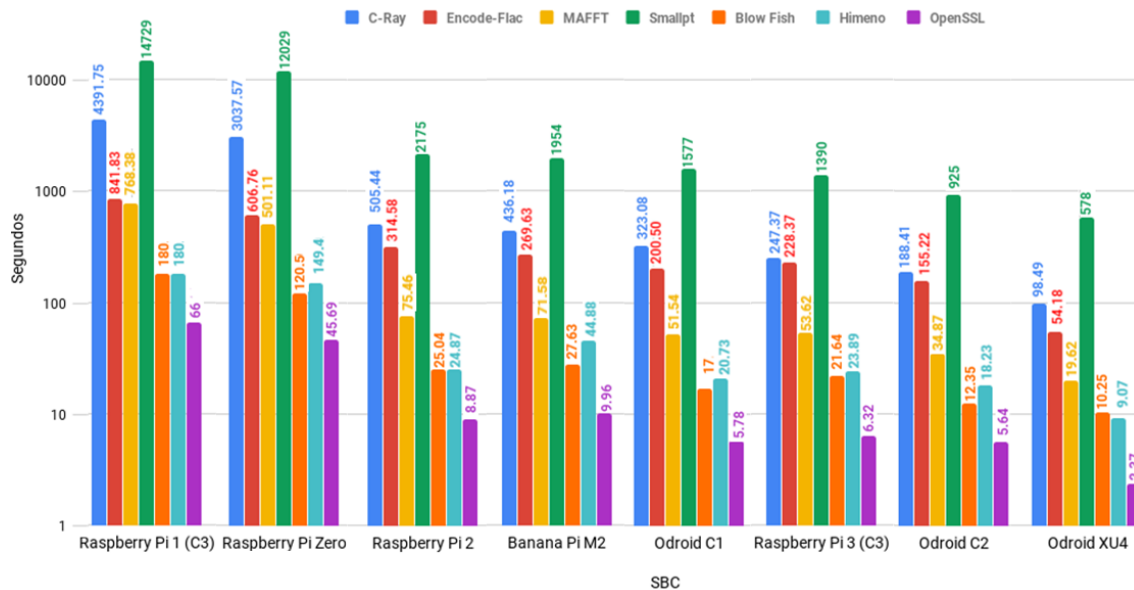


Figura 32: Resultado em segundos dos Benchmarks aplicados nos dispositivos SBC's.

de conexão inteligente e conversão de dados. Para a execução das camadas superiores é recomendado dispositivos com maior poder computacional como por exemplo SBCs dos grupos A e B.

Uma vez classificados os dispositivos SBCs, é possível ter uma noção da capacidade mínima destes dispositivos se comparado aos módulos da arquitetura 5C. Sendo assim, os experimentos da próxima seção demonstram se um conjunto dos dispositivos SBC está apto a rodar pelo menos o módulo de comunicação da arquitetura de software 5C proposta.

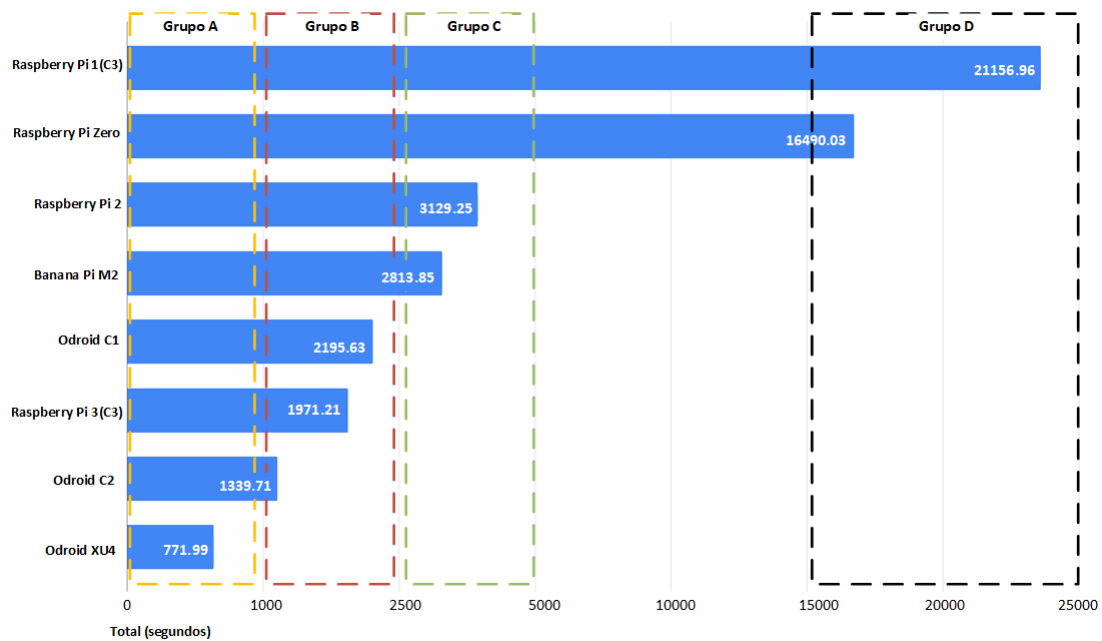


Figura 33: Grupos dispositivos SBCs.

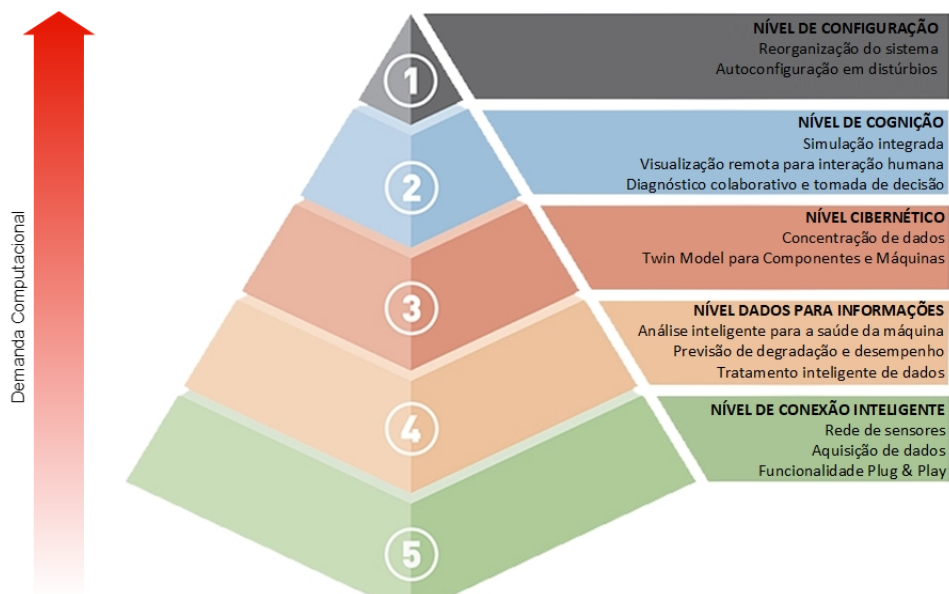


Figura 34: Demanda computacional exigida.

5 ESTUDO DE CASO

O objetivo desse capítulo é descrever um estudo de caso onde a proposta do capítulo anterior será implementada em uma aplicação real no projeto 3DCS (Tecnologias Avançadas de Automação e controle para supervisão de Sub-sistemas Marítimos). O estudo foi desenvolvido no laboratório de pesquisas do NAUTEC (Grupo de Robótica e Automação Inteligente) no C3 (Centro de Ciências Computacionais) da FURG (Universidade Federal do Rio Grande).

Mais especificamente, o objetivo é demonstrar que os dispositivos analisados na seção anterior estão aptos a rodar pelo menos o módulo de comunicação da arquitetura 5C. Isto por que, com este módulo, é possível inserir o dispositivo na rede CPS e quando necessário outros serviços, demanda-los da camada *fog* ou mesmo da nuvem.

5.1 Estrutura de hardware do estudo de caso

Para implementação dos módulos apresentados na arquitetura de *software*, serão utilizados computadores de placa única, ou *Single Board Computers* (SBC) como *Raspberry Pi*. Dos dispositivos analisados pela classificação de *Hardware* apresentada no capítulo anterior, foram utilizados SBCs da família Raspberry Pi que correspondem aos grupos B e D respectivamente. Esses dispositivos apresentam os requisitos entendidos como necessários para este tipo de aplicação.

O cenário de desenvolvimento é aplicado em uma planta industrial didática, apresentada na Figura 35. Essa planta didática é automatizada e constituída por componentes como tanques, bombas, válvulas pneumáticas, CLPs, sensores e atuadores. Esses componentes formam uma estrutura que viabiliza a simulação de processos industriais comuns a várias fábricas de processos.

A Figura 36 apresenta uma visão geral dos componentes que serão utilizados para o estudo de caso. Os componentes estão identificados de acordo com a estrutura apresentada na topologia de hardware do sistema apresentada na seção anterior.

Alguns dos diversos sensores e atuadores presentes na planta didática foram conectados a diferentes Raspberry Pi. Diante da falta de conexão direta entre os sensores e



Figura 35: Planta didática Smar PD3.

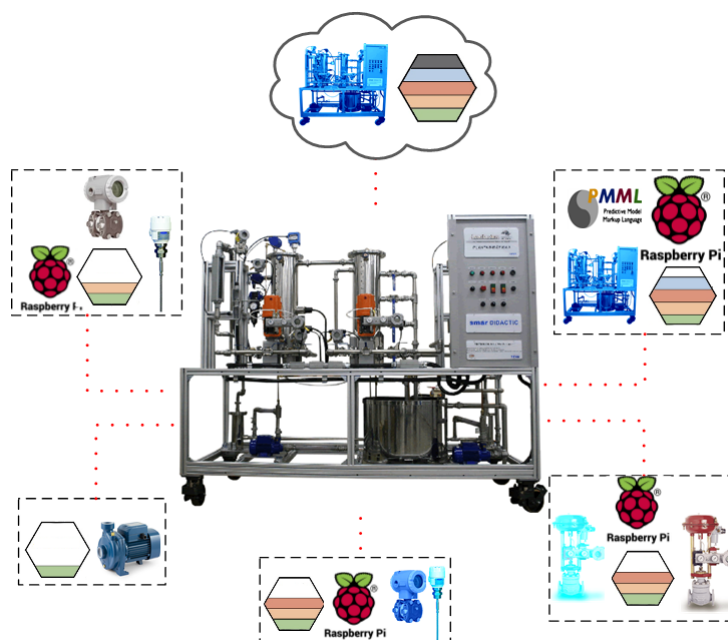


Figura 36: Diagrama de representação da aplicação do estudo de caso

atuadores e o Raspberry Pi foi necessário o desenvolvimento de interfaces que possibilitassem esta comunicação. Com isso, o Raspberry Pi tornou-se capaz de ler e escrever sinais de corrente 4-20 mA e sinais de tensão 0-24 V, ambos comumente encontrados em sistemas de automação. O desenvolvimento dessas interfaces foi realizado em um projeto de graduação de um membro do grupo de pesquisa (Marques Junior 2018). A estrutura de *hardware* é apresentada na Figura 37.

5.2 Implementação Módulo de comunicação

O módulo de comunicação é responsável por garantir a troca segura de dados entre os dispositivos. Para isso, foi escolhido o protocolo OPC UA. O OPC UA é responsável

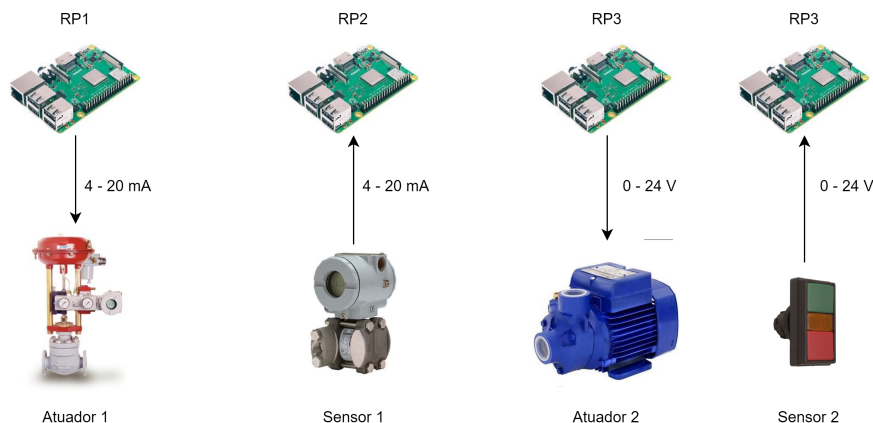


Figura 37: Arquitetura de hardware.

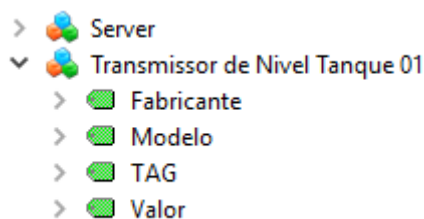


Figura 38: Servidor contendo informações do transmissor de nível.

pela disponibilização dos dados de sensores e pela escrita de um comando de controle por exemplo.

5.2.1 Implementação dos Servidores OPC UA

Devido a existência de diversas bibliotecas OPC UA de código aberto disponíveis, foi realizado um breve levantamento com o intuito de escolha para a aplicação do presente trabalho apresentado na Seção 2.3. Todas as bibliotecas OPC UA analisadas, podem ser compiladas nos sistemas Linux e Windows.

A biblioteca open62541 foi escolhida, por possuir muitos dos recursos já implementados, além de possuir uma comunidade ativa para desenvolvimento de novos recursos e manual com diversos exemplos de implementação. Além disso, assinatura e criptografia ainda não são suportados pelo open62541, mas estão atualmente em desenvolvimento e serão suportados em breve.

A biblioteca fornece modelos de informação de MTConnect e PLCopen, porém para o desenvolvimento desse trabalho decidiu-se pela não utilização de um modelo de informação já definido. Foram criados então dois modelos, um para sensores e outro para atuadores. Os respectivos modelos são exibidos nas Figuras 38 e 39, respectivamente.

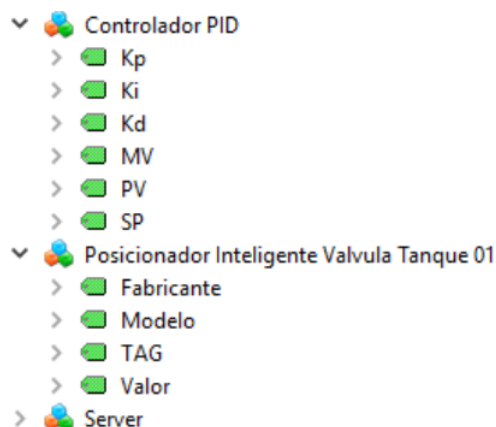


Figura 39: Servidor contendo informações do posicionador de válvula.

5.3 Implementação do Módulo de Configuração

O módulo de configuração usado nesse estudo de caso teve como objetivo controlar o nível de um dos tanques presentes na planta didática. Para isso, foi implementado um Controlador PID utilizando a linguagem de programação C. O módulo de configuração recebe os dados do transmissor de nível, realiza o cálculo do algoritmo PID e escreve o resultado na válvula controladora. A troca dos dados é realizada por meio de clientes e servidores OPC UA.

5.4 Validação do Estudo de Caso

Foram utilizados um Raspberry Pi 1 e um Raspberry Pi 3, respectivamente conectados ao transmissor de nível e ao posicionador de válvula, ambos instalados na planta didática Smar. No Raspberry 1 foi executado o servidor OPC UA, conforme a Figura 38. Ele é responsável por disponibilizar as leituras referentes ao nível do tanque na rede local, assim caracterizando a implementação do módulo de comunicação.

No Raspberry Pi 3, em conjunto ao servidor responsável pelo posicionador de válvula foi adicionado o algoritmo para realizar o controle de nível do tanque. Para realizar a troca de dados entre os servidores é necessário um cliente, que também é executado no Raspberry Pi 3. Os parâmetros do controlador podem ser configurados da maneira desejada, como por exemplo através de um dispositivo móvel, desde que esse esteja conectado à mesma rede. Essa implementação possui as características dos módulos de configuração e comunicação. Na Figura 40 é exibida a disposição e relação dos elementos, bem como o modelo de informação de cada servidor.

Diante disso, foram realizados testes para verificar o funcionamento do sistema proposto. Através do servidor, ilustrado na Figura 39, foi atribuído o valor de 60% para a variável referente ao Set Point. Os dados de SP (*Set Point*), PV (*Process Variable*), MV (*Manipulated Variable*) foram armazenados em uma planilha. Na Figura 41 é apresentada

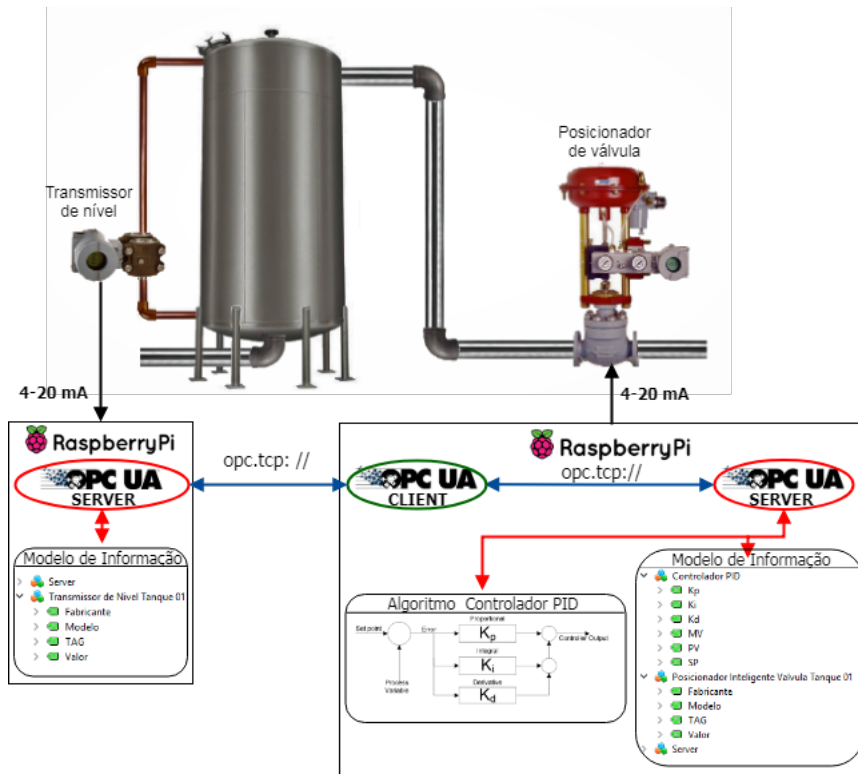


Figura 40: Topologia utilizada para aplicação.

a resposta do sistema.

A partir dos experimentos pode-se concluir que os Raspberry foram capazes de realizar a comunicação através do protocolo OPC UA. Também, verificou-se que o algoritmo de controle apresentou resultados satisfatórios, atingindo o Set Point definido. O módulo de comunicação foi completamente implementado, apresentando as características da camada Smart Connection. O módulo de configuração realizou o controle de nível, assim realizando uma parte dos objetivos da camada de configuração. As demais camadas não foram implementadas.

Vale salientar ainda que o módulo de comunicação implementado tanto no Raspberry Pi 1 como no Raspberry Pi 3, teve seu funcionamento de forma semelhante não importando a classe do dispositivo. Mas uma vez que o nível de computação aumenta a cada camada, dispositivos melhores posicionados na classificação de SBC serão exigidos. Além disso, o uso de OPC UA promove a caracterização de sistemas previstos na Indústria 4.0, tornando o sistema interoperável, modular e descentralizado.

Finalmente, analisando os resultados com os dispositivos utilizados nesse experimento, e os resultados obtidos nos *Benchmarks* apresentados na seção anterior, conclui-se que todos os dispositivos analisados estão aptos a fazer parte de uma rede CPS, desde que alocando a quantidade de módulos da arquitetura de acordo com a capacidade computacional de cada dispositivo. Isto por que, todos os dispositivos utilizados nos experimentos podem rodar tranquilamente a camada de comunicação da arquitetura.

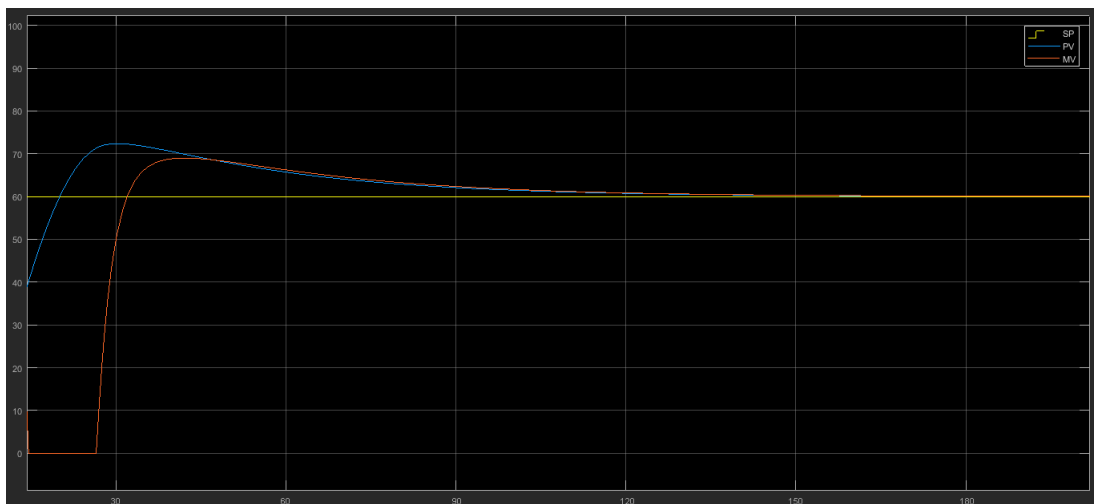


Figura 41: Resposta do sistema.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi realizada uma classificação de dispositivos SBC para aplicação de sistemas ciberfísicos. Estes dispositivos já estão sendo amplamente utilizados em outras áreas, como controle industrial e instrumentos médicos por exemplo. E com avanço da pesquisa estão ganhando espaços em projetos focados na indústria 4.0. Uma breve contextualização em torno da evolução do cenário industrial foi apresentada neste trabalho a fim de demonstrar pontos chave habilitadores da nova revolução industrial.

No capítulo 2, de uma forma mais aprofundada foram apresentados alguns conceitos associados a Indústria 4.0, como CPS, IoT e IoS e tecnologias facilitadoras para realização de fábricas inteligentes. O trabalho também abordou neste capítulo a arquitetura CPS denominada 5C, que apresenta uma forma genérica para execução de CPS em ambientes industriais e que serviu como base para aplicação do estudo, assim como padrões e protocolos de comunicação e aquisição de dados que estão ganhando destaque no contexto da Indústria 4.0. E ainda apresentou definições e aplicações de Computadores de placa Única Ou Single Board Computers .

No capítulo 3 foram apresentados trabalhos relacionados com diferentes abordagens, perspectivas e tecnologias empregada mas com a mesma finalidade de concepção de CPS diretamente, apresentar diferentes aplicações de SBCs no contexto da indústria 4.0. Por fim neste capítulo é realizada uma análise onde alguns desses trabalhos, apresentaram em seu escopo descrições de arquiteturas para sistemas ciberfísicos. Essas arquiteturas foram analisadas, servindo como base teórica para a elaboração da proposta.

No capítulo 4 foi proposta uma arquitetura software modular para o desenvolvimento de um sistema ciberfísico, a arquitetura foi estruturada em cinco módulos: Comunicação, Conversão, Cibernético, Inteligência e configuração. Mas o foco do trabalho foi direcionado para a camada de comunicação onde foi mais detalhada no estudo de caso. Ainda neste capítulo foi apresentada uma classificação de HW onde foram testados alguns SBCs e separados em grupos de acordo com suas características, como processamento e memória, em conjunto com Benchmarks. A intenção desses testes é dar uma ideia do desempenho relativo entre os SBCs. Como resultado, tentou-se ser o mais preciso possível para que possamos comparar o desempenho relativo destes dispositivos. De

acordo com as classes estabelecidas um conjunto de características como níveis de inteligência, cognição e controle, compatíveis com a estrutura do CPS, serão adicionadas a cada componente da classe, ou até mesmo quantas camadas da arquitetura 5C serão contempladas em cada dispositivo.

O capítulo 5 apresentou a aplicação do modelo proposto em um estudo de caso, que foi baseado na implementação do módulo de comunicação em uma planta industrial didática. O módulo foi embarcado em SBCs representantes de dois grupos distintos, o módulo de comunicação foi completamente implementado, apresentando as características da camada Smart Connection. O dispositivo SBC por sua vez mostraram resultados semelhantes para os dois grupos na implementação deste módulo. O módulo de configuração realizou o controle de nível, assim realizando uma parte dos objetivos da camada de configuração. Outro ponto positivo desta implementação é a utilização do OPC UA em todos os módulos tornando o sistema interoperável, modular e descentralizado.

Como trabalhos futuros vale destacar que este trabalho foi direcionado para uma classificação de dispositivos e implementação do módulo de comunicação referente a camada de conexão inteligente da arquitetura 5C, os demais módulos da arquitetura correspondente ao restante das camadas da 5C estão em desenvolvimento em outros trabalhos dentro do Projeto, feito isso, será possível realizar a integração de todos os módulos e consequentemente o CPS como um todo.

REFERÊNCIAS

- A Internet Industrial das Coisas (IIoT)* (2017). accessed:30/06/2018.
URL: <https://www.i-scoop.eu/internet-of-things-guide/industrial-internet-things-iiot-saving-costs-innovation/industrial-internet-things-iiot/>
- Ashton, K. (2011), ‘That ‘internet of things’ thing’, *RFiD Journal* **22**(7).
- Barbosa, J., Leitão, P., Trentesaux, D., Colombo, A. W. & Karnouskos, S. (2016), Cross benefits from cyber-physical systems and intelligent products for future smart industries, *in* ‘Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on’, IEEE, pp. 504–509.
- Browne, J., Sackett, P. & Wortmann, J. C. (1995), ‘Future manufacturing systems—towards the extended enterprise’, *Computers in industry* **25**(3), 235–254.
- Colombo, A. W., Karnouskos, S., Mendes, J. M. & Leitão, P. (2015), ‘Industrial agents in the era of service oriented architectures and cloud based industrial infrastructures’, *Industrial Agents: Emerging Applications of Software Agents in Industry* pp. 67–87.
- Da Xu, L., He, W. & Li, S. (2014), ‘Internet of things in industries: A survey’, *IEEE Transactions on industrial informatics* **10**(4), 2233–2243.
- DE OLIVEIRA, L. E. S. (2017), ‘Concepção de um framework para monitoramento e teleoperação de máquinas-ferramenta cnc via internet aderente à indústria 4.0’.
- García, M. V., Irisarri, E., Pérez, F., Estévez, E. & Marcos, M. (2016), Opc-ua communications integration using a cpps architecture, *in* ‘Ecuador Technical Chapters Meeting (ETCM), IEEE’, IEEE, pp. 1–6.
- García, M. V., Irisarri, E., Pérez, F., Estévez, E., Orive, D. & Marcos, M. (2016), Plant floor communications integration using a low cost cpps architecture, *in* ‘Emerging Technologies and Factory Automation (ETFa), 2016 IEEE 21st International Conference on’, IEEE, pp. 1–4.

- Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. (2013), 'Internet of things (iot): A vision, architectural elements, and future directions', *Future generation computer systems* **29**(7), 1645–1660.
- Hermann, M., Pentek, T. & Otto, B. (2016), Design principles for industrie 4.0 scenarios, *in* 'System Sciences (HICSS), 2016 49th Hawaii International Conference on', IEEE, pp. 3928–3937.
- Intiaz, J. & Jasperneite, J. (2013), Scalability of opc-ua down to the chip level enables "internet of things", *in* 'Industrial Informatics (INDIN), 2013 11th IEEE International Conference on', IEEE, pp. 500–505.
- Jasperneite, J., Neumann, A. & Pethig, F. (2015), 'Opc ua versus mtconnect', pp. 50–55.
- Jazdi, N. (2014), Cyber physical systems in the context of industry 4.0, *in* 'Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on', IEEE, pp. 1–4.
- Jeschke, S., Brecher, C., Meisen, T., Özdemir, D. & Eschert, T. (2017), Industrial internet of things and cyber manufacturing systems, *in* 'Industrial Internet of Things', Springer, pp. 3–19.
- Kagermann, H., Helbig, J., Hellinger, A. & Wahlster, W. (2013), *Recommendations for Implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry; final report of the Industrie 4.0 working group*, Forschungsunion.
- Karnouskos, S., Leitao, P., Ribeiro, L., Lee, J., Strasser, T. & Colombo, A. W. (2016), 'Smart agents in industrial cyber–physical systems', *Proceedings of the IEEE* **104**(5), 1086–1101.
- Lee, J., Bagheri, B. & Kao, H.-A. (2015), 'A cyber-physical systems architecture for industry 4.0-based manufacturing systems', *Manufacturing Letters* **3**, 18–23.
- Leitão, P., Barbosa, J., Papadopoulou, M.-E. C. & Venieris, I. S. (2015), Standardization in cyber-physical systems: The arum case, *in* 'Industrial Technology (ICIT), 2015 IEEE International Conference on', IEEE, pp. 2988–2993.
- Leitão, P., Colombo, A. W. & Karnouskos, S. (2016), 'Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges', *Computers in Industry* **81**, 11–25.
- Leitner, S.-H. & Mahnke, W. (2006), 'Opc ua–service-oriented architecture for industrial applications'.

- MacDougall, W. (2014), *Industrie 4.0: Smart manufacturing for the future*, Germany Trade & Invest.
- Marks, E. A. & Bell, M. (2008), *Service Oriented Architecture (SOA): a planning and implementation guide for business and technology*, John Wiley & Sons.
- Marques Junior, M. R. N. (2018), *Um dispositivo para aquisição de dados no contexto da Indústria 4.0*, Projeto de Graduação em Engenharia de Automação, Universidade Federal do Rio Grande, Rio Grande, Brasil.
- Matthews, S. J., Blaine, R. W. & Brantly, A. F. (2016), Evaluating single board computer clusters for cyber operations, *in* ‘Cyber Conflict (CyCon US), International Conference on’, IEEE, pp. 1–8.
- Melo, P. (2016), ‘Introdução ao opc ua (open platform communications unified architecture)’. accessed:30/06/2018.
URL: <https://www.embarcados.com.br/introducao-ao-opc-ua/>
- Möller, D. P. (2016), Introduction to cyber-physical systems, *in* ‘Guide to Computing Fundamentals in Cyber-Physical Systems’, Springer, pp. 81–139.
- MTConnect (2008), ‘MTConnect Institute’, <http://www.mtconnect.org>. Accessed: 2017-08-13.
- Müller, M., Wings, E. & Bergmann, L. (2017), Developing open source cyber-physical systems for service-oriented architectures using opc ua, *in* ‘Industrial Informatics (INDIN), 2017 IEEE 15th International Conference on’, IEEE, pp. 83–88.
- Newark (2014), ‘A brief history of single board computers’, <http://www.newark.com/wcsstore/ExtendedSitesCatalogAssetStore/cms/asset/pdf/ameri\cas/common/NE14-ElectronicDesignUncovered-Dec14.pdf>. Acessado 18 de Outubro, 2017.
- Oliveira, L. E. S. d. (2017), ‘Concepção de um framework para monitoramento e teleoperação de máquinas-ferramenta cnc via internet aderente à indústria 4.0’.
- OPC Foundation, O. F. (2016), ‘What is opc’. accessed:30/06/2018.
URL: <https://opcfoundation.org/>
- Oztemel, E. & Gursev, S. (2018), ‘Literature review of industry 4.0 and related technologies’, *Journal of Intelligent Manufacturing* pp. 1–56.
- Part, M. (n.d.), ‘1.(2011). the association for manufacturing technology, “getting started with mtconnect: Connectivity guide”’, *White Paper*.

- Phoronix (2018), 'Phoronix test suite', <https://www.phoronix.com/>. Acessado 29 de Maio, 2018.
- Profanter, S., Dorofeev, K., Zoitl, A. & Knoll, A. (2017), Opc ua for plugproduce: Automatic device discovery using lds-me, *in* 'Proceedings of the IEEE International Conference on Emerging Technologies And Factory Automation (ETFA)'
- Rahman, M., Khalib, Z. I. A. & Ahmad, R. (2009), Performance evaluation of pntms: A portable network traffic monitoring system on embedded linux platform, *in* 'Computer Engineering and Technology, 2009. ICCET'09. International Conference on', Vol. 1, IEEE, pp. 108–113.
- Razzaque, M. A., Milojevic-Jevric, M., Palade, A. & Clarke, S. (2016), 'Middleware for internet of things: A survey.', *IEEE Internet of Things Journal* **3**(1), 70–95.
- Rodrigues, L. F., de Jesus, R. A. & Schützer, K. (2016), 'Indústria 4.0: Uma revisão da literatura', *Revista de Ciência & Tecnologia* **19**(38), 33–45.
- Rodrigues, N., Leitão, P., Turrin, C., Pagani, A. & Petrali, P. (2012), Grace ontology integrating process and quality control, *in* 'IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society', IEEE, pp. 4348–4353.
- Santos, B. P., Silva, L., Celes, C., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N. & Loureiro, A. (2016), 'Internet das coisas: da teoria a prática', *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* .
- Shrouf, F., Ordieres, J. & Miragliotta, G. (2014), 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* 'Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on', IEEE, pp. 697–701.
- Sousa, F. R., Moreira, L. O. & Machado, J. C. (2009), 'Computação em nuvem: Conceitos, tecnologias, aplicações e desafios'.
- Tien, J. M. (2012), 'The next industrial revolution: Integrated services and goods', *Journal of Systems Science and Systems Engineering* pp. 1–40.
- Trappey, A. J., Trappey, C. V., Govindarajan, U. H., Chuang, A. C. & Sun, J. J. (2016), 'A review of essential standards and patent landscapes for the internet of things: A key enabler for industry 4.0', *Advanced Engineering Informatics* .
- Venturelli, M. (2018), 'Opc-ua na automação industrial - o padrão de comunicação para interconexão e convergência de dados na indústria'. accessed:30/06/2018.
URL: <https://marcioventurelli.com/2018/06/22/opc-ua-na-automacao-industrial/>

- Vijayaraghavan, A., Sobel, W., Fox, A., Dornfeld, D. & Warndorf, P. (2008), 'Improving machine tool interoperability using standardized interface protocols: Mt connect', *Laboratory for Manufacturing and Sustainability* .
- Vimos, V., Sacoto, E. & Morales, D. (2016), Conceptual architecture definition: Implementation of a network sensor using arduino devices and multiplatform applications through opc ua, *in* 'Automatica (ICA-ACCA), IEEE International Conference on', IEEE, pp. 1–5.
- Wahlster, W. (2013), Industry 4.0: the semantic product memory as a basis for cyber-physical production systems, *in* 'SGAICO Forum: Recent Trends in Artificial Intelligence and Cognitive Science Zürich', Vol. 27.
- Warndorf, P., Fox, A. & Sobel, W. (2007), 'Mtconnect technology white paper. association for manufacturing technology' .
- Weyer, S., Schmitt, M., Ohmer, M. & Gorecky, D. (2015), 'Towards industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems', *IFAC-PapersOnLine* **48**(3), 579–584.
- Yin, S. & Kaynak, O. (2015), 'Big data for modern industry: challenges and trends [point of view]', *Proceedings of the IEEE* **103**(2), 143–146.