

UNIVERSIDADE DE CAXIAS DO SUL

GUILHERME MAFRA SOARES

**DESENVOLVIMENTO DE UM SISTEMA CIBER-FÍSICO COM A PLATAFORMA  
AMAZON WEB SERVICES**

Caxias do Sul  
2019

Guilherme Mafra Soares

**DESENVOLVIMENTO DE UM SISTEMA CIBER-FÍSICO COM A PLATAFORMA  
AMAZON WEB SERVICES**

Trabalho de Conclusão de Curso apresentado  
como parte dos requisitos para a obtenção do  
título de Engenheiro de Controle e Automação  
da Universidade de Caxias do Sul.

Orientador:  
Prof.<sup>a</sup> Dra. Carine Geltrudes Webber

Caxias do Sul  
2019

Guilherme Mafra Soares

**DESENVOLVIMENTO DE UM SISTEMA CIBER-FÍSICO COM A PLATAFORMA  
AMAZON WEB SERVICES**

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para a obtenção do título de Engenheiro de Controle e Automação da Universidade de Caxias do Sul.

Orientador:  
Prof.<sup>a</sup> Dra. Carine Geltrudes Webber

**Aprovado em** \_\_\_\_/\_\_\_\_/\_\_\_\_

**Banca Examinadora**

---

Prof. Dra. Carine Geltrudes Webber(orientadora)  
Universidade de Caxias do Sul - UCS

---

Prof. Me. Patricia Giacomelli  
Universidade de Caxias do Sul - UCS

---

Prof. Dr. Guilherme Holsbach Costa  
Universidade de Caxias do Sul - UCS

Aos meus pais, por tornarem isto possível.

## **AGRADECIMENTOS**

Gostaria de agradecer em especial a minha mãe, que sempre me deu todo o apoio, confiança e suporte, além de ser o meu exemplo para seguir sempre enfrentando todos os obstáculos que possam aparecer em minha vida.

Gostaria de agradecer a Luana Venzon pelo companheirismo e apoio durante todo o período de desenvolvimento deste trabalho.

Aos meus familiares que me deram o todo suporte nos diferentes momentos de minha vida e que foram peças fundamentais durante todo este caminho.

À minha orientadora prof<sup>a</sup>. Dra. Carine Geltrudes Webber, por ter me dado todas as direções, por me incentivar e por sempre demonstrar disposição com o esclarecimento de dúvidas e questionamentos.

A todos que direta ou indiretamente contribuíram para que este trabalho fosse possível. Meu mais sincero obrigado!

*”A competitividade de um país não começa nas indústrias ou nos laboratórios de engenharia.  
Ela começa na sala de aula.”*  
Lee Iacocca

## RESUMO

O objetivo deste trabalho é desenvolver, utilizando uma plataforma de serviços, um sistema ciber-físico. Um sistema ciber-físico consiste em um conjunto de tecnologias que permitem a análise do desempenho de máquinas, assim como a integração de dispositivos físicos com o ambiente virtual, tendo como o foco o aumento de produtividade na indústria. Os conceitos de sistemas ciber-físicos são parte integrante nos conceitos da indústria 4.0, sendo esta a quarta revolução industrial. Os sistemas ciber-físicos permitem a implementação da inteligência artificial, gêmeos digitais e prognóstico da saúde de máquinas. Metodologias vêm sendo estudadas a fim de padronizar a implementação dos conceitos de sistemas ciber-físicos. Neste trabalho, desenvolveu-se um experimento de sistema ciber-físico utilizando a plataforma Amazon Web Service. Seguiu-se a arquitetura 5C, implementando um cenário de CPS. Como resultado, avaliou-se que a plataforma oferece recursos computacionais que favorecem a implementação de um CPS.

**Palavras-chave:** Indústria 4.0. Amazon Web Service. Sistema Ciber-Físico. Arquitetura 5C.

## **ABSTRACT**

The objective of this work is to develop, using a service platform and a cyber-physical system, which consists in a series of technologies that allow the analysis of machine performance, as well as the integration of physical devices with the virtual environment, having as a focus on increasing productivity in the industry. The concepts of cyber-physical systems are a component in the concepts of industry 4.0, this being the fourth industrial revolution. Cyber-physical systems allow the implementation of artificial intelligence, digital twins and prognosis health management for machines. Methodologies have been studied to standardize the implementation of the concepts of cyber-physical systems. In this work, it was developed a experiment for cyber-physical systems using the Amazon Web Service platform. It was based on the 5C architecture, deploying a CPS scene. As result, the platform offer computing resources that favor the deploy of a CPS.

**Keywords:** Industry 4.0. Amazon Web Service. Cyber-Physical System. 5C Architecture.



## LISTA DE FIGURAS

Figura 1 – Linha do tempo da evolução industrial . . . . .	13
Figura 2 – Transição dos conceitos de mecatrônica para CPS. . . . .	18
Figura 3 – Arquitetura básica do CPS. . . . .	19
Figura 4 – Áreas de aplicação de CPS . . . . .	20
Figura 5 – Etapas da arquitetura 5C . . . . .	21
Figura 6 – Técnicas aplicadas nas diferentes camadas da arquitetura 5C. . . . .	22
Figura 7 – Fases de degradação das máquinas . . . . .	23
Figura 8 – Fluxo de funcionamento do PHM . . . . .	23
Figura 9 – Disposição dos sistema no conceito PHM . . . . .	24
Figura 10 – Técnica de três dimensões . . . . .	25
Figura 11 – Tendência evolutiva dos conceitos de DT . . . . .	26
Figura 12 – Digital Twin usando protocolo MQTT. . . . .	28
Figura 13 – Conceito de implementação de DTS. . . . .	29
Figura 14 – Conceito de implementação de DTS. . . . .	31
Figura 15 – Camadas implementadas . . . . .	33
Figura 16 – Robô modelo IRB 1200 . . . . .	34
Figura 17 – Comportamento da produção de leite . . . . .	35
Figura 18 – Rede neural <i>feedforward</i> . . . . .	36
Figura 19 – Rede neural recorrente . . . . .	37
Figura 20 – Modelos de computação em nuvem . . . . .	38
Figura 21 – Serviços da plataforma AWS . . . . .	41
Figura 22 – Arquitetura básica MQTT . . . . .	44
Figura 23 – Conceito de implementação básica do OPC UA. . . . .	45
Figura 24 – Criação do <i>bucket</i> . . . . .	47
Figura 25 – Criação do <i>bucket</i> . . . . .	48
Figura 26 – Ambiente do <i>bucket</i> criado após carregar o <i>dataset</i> . . . . .	49
Figura 27 – Ambiente do <i>SageMaker</i> . . . . .	49
Figura 28 – Ambiente do <i>bucket</i> criado após carregar o <i>dataset</i> . . . . .	50
Figura 29 – Arquitetura final do experimento . . . . .	51
Figura 30 – Ambiente do <i>AWS IoT Core</i> . . . . .	52
Figura 31 – Fim da criação e local dos <i>downloads</i> para o <i>AWS IoT Greengrass Core</i> . . . . .	53
Figura 32 – Configuração inicial <i>AWS Lambda</i> . . . . .	54
Figura 33 – Ambiente de desenvolvimento do <i>AWS Lambda</i> . . . . .	55
Figura 34 – Gráfico de comportamento do <i>dataset</i> de produção de leite . . . . .	56
Figura 35 – Gráfico de comportamento do <i>dataset</i> do robô . . . . .	57
Figura 36 – Imagem adquirida do <i>AWS IoT Core</i> referente ao envio de dados de produção de leite . . . . .	58
Figura 37 – Imagem adquirida do <i>AWS IoT Core</i> referente ao envio de dados do robô . . . . .	58

## **LISTA DE TABELAS**

Tabela 1 – Comparação dos serviços da AWS e do Azure . . . . .	40
--	----

## LISTA DE SIGLAS

AWS	<i>Amazon Web Services</i>
CLP	Controlador lógico programável
CPS	Sistemas Ciber-Físicos
DT	Gêmeo Digital
IA	Inteligência artificial
IaaS	Infraestrutura como um serviço
iCPS	Sistemas Ciber-Físicos inteligente
IoT	<i>Internet of Things</i>
MES	<i>Manufacturing Execution System</i>
ML	Aprendizado de máquina
MQTT	<i>Message Queuing Telemetry Transport</i>
NASA	<i>National Aeronautics and Space Administration</i>
OPC UA	<i>Open Platform Communications Unified Architecture</i>
PaaS	Plataforma como um serviço
PHM	Prognóstico e gestão de saúde
QoS	Qualidade de serviço
RNN	Redes neurais recorrentes
RUL	Tempo de vida restante
SaaS	<i>Software</i> como um serviço

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	HISTÓRIA DA INDÚSTRIA	13
1.2	INDÚSTRIA 4.0	14
1.3	DESAFIOS DA INDÚSTRIA 4.0	14
1.4	OBJETIVOS	15
1.4.1	<b>Objetivo geral</b>	15
1.4.2	<b>Objetivos específicos</b>	15
1.5	JUSTIFICATIVA	15
1.6	ORGANIZAÇÃO DO DOCUMENTO	16
<b>2</b>	<b>BASE TEÓRICA DOS CONCEITOS DE INTELIGÊNCIA ARTIFICIAL NA INDÚSTRIA</b>	<b>17</b>
2.1	SISTEMAS CIBER-FÍSICOS	17
2.2	ARQUITETURA 5C	20
2.3	PROGNÓSTICO E GESTÃO DA SAÚDE DAS MÁQUINAS	22
2.4	GÊMEO DIGITAL	24
2.5	IMPLEMENTAÇÃO DE SISTEMAS GÊMEOS DIGITAIS	26
2.6	TECNOLOGIAS PARA IMPLEMENTAÇÃO DE SISTEMAS CIBER-FÍSICOS E GÊMEOS DIGITAIS	27
2.6.1	<b>Configuração do sistema físico usando gêmeo digital</b>	27
2.6.2	<b>Gêmeo digital no chão de fábrica</b>	28
2.6.2.1	Chão de fábrica físico	29
2.6.2.2	Chão de fábrica virtual	29
2.6.2.3	Sistema de serviço do chão de fábrica	30
2.6.2.4	Dados gêmeos digitais no chão de fábrica	30
2.6.3	<b>Sistema ciber-físico inteligente</b>	30
<b>3</b>	<b>SISTEMAS CIBER-FÍSICOS: CONSTRUÇÃO DE UM EXPERIMENTO</b>	<b>32</b>
3.1	PLANEJAMENTO DA ARQUITETURA DO SISTEMA	32
3.2	SELEÇÃO DE DADOS	34
3.3	CONCEITOS BÁSICOS DAS REDES NEURAIS RECORRENTES	35
3.4	COMPUTAÇÃO EM NUVEM	38
3.5	PLATAFORMA PARA IMPLEMENTAÇÃO DOS CONCEITOS DE CPS	40
3.6	COMUNICAÇÃO NA CAMADA DE CONEXÃO	43
3.6.1	<b>MQTT</b>	43
3.6.2	<b>OPC UA</b>	44
3.7	IMPLEMENTAÇÃO DO EXPERIMENTO UTILIZANDO A AWS	45
3.7.1	<b>Carga de dados no Amazon S3</b>	46
3.7.2	<b>Configuração e treinamento no SageMaker</b>	49
3.7.3	<b>Implementação da camada de conversão</b>	51
3.7.4	<b>Configuração do AWS Lambda e integração com o AWS IoT Greengrass Core</b>	54
3.7.5	<b>Resultados obtidos</b>	55
3.8	CONSIDERAÇÕES FINAIS	59

<b>4 CONCLUSÕES</b> . . . . .	<b>60</b>
4.1 SÍNTESE DO TRABALHO . . . . .	60
4.2 CONTRIBUIÇÕES DO TRABALHO . . . . .	62
4.3 TRABALHOS FUTUROS . . . . .	63
<b>REFERÊNCIAS</b> . . . . .	<b>64</b>

# 1 INTRODUÇÃO

## 1.1 HISTÓRIA DA INDÚSTRIA

O conceito de indústria 4.0, segundo Henning (2013), será o novo desafio dentro das fábricas. A indústria 4.0 é o resultado do avanço tecnológico da computação aplicada aos processos de engenharia e fabricação. A Figura 1 ilustra as evoluções que ocorreram dentro da indústria.

Figura 1 – Linha do tempo da evolução industrial



Fonte: (Ubivis - Internet of Smart Machines, 2019)

A primeira revolução industrial causou grandes impactos ao fim do século 18, apresentando o início da implementação de máquinas dentro da indústria. Durante essa revolução ocorreram mudanças nas metodologias de fabricação, a força humana, até então usada, foi substituída por máquinas que utilizavam força mecânica a vapor (SENTRYO, 2017).

A segunda revolução industrial ocorreu no início do século 20. A tecnologia implementada foi energia elétrica, essa trouxe junto a produção em massa. O início de empresas de manufatura pesada ocorre neste período (SENTRYO, 2017).

A terceira revolução industrial, que iniciou após a metade do século 20, trouxe os primeiros controladores matemáticos para indústria. A digitalização e substituição dos equipamentos puramente elétricos iniciou-se nessa época (SENTRYO, 2017). O primeiro controlador lógico programável (CLP) surgiu nesta transição, conhecido como *Modicon 084*. Atualmente os CLP ainda são amplamente utilizados dentro da indústria, para automação de processos fabris (Library at AutomationDirect, 2019).

Atualmente, a nova revolução industrial, conhecida como indústria 4.0, foi mencionada

pela primeira vez na feira de Hannover em 2011. A quarta revolução industrial traz conectividade e os conceitos de sistemas ciber-físicos (em inglês *Cyber-Physical System*, sigla CPS) para dentro da indústria (SENTRYO, 2017).

## 1.2 INDÚSTRIA 4.0

De forma prática, a indústria 4.0 é vista como uma forma de o próprio ambiente fabril tomar decisões referentes aos métodos a serem executados na produção, tendo em conjunto a integração dos conceitos de Internet das coisas (em inglês *Internet of Things*, sigla IoT) para troca de informações entre máquinas, considerada a quarta revolução industrial (NASCI-MENTO MARQUES JUNIOR et al., 2018).

A indústria 4.0 busca trazer a flexibilidade da produção e aumentar a capacidade de máquinas conforme as mudanças de planejamento. Torna possível a fabricação de pequenos lotes de um produto a custos viáveis, permitindo que produtos sejam desenvolvidos com foco no cliente indiferente do tamanho da demanda. O uso de dados para permitir diferentes análises relacionadas ao equipamento, como por exemplo prognóstico de máquina, é um dos exemplos que podem ser pontuados (Geschäftsstelle Plattform Industrie, 2019).

As integração dos métodos de CPS aos ambientes fabris, gêmeo digital (em inglês *Digital Twin*, sigla DT) e prognóstico e gestão da saúde (em inglês *Prognostics Health Management*, sigla PHM), que foram apresentadas no decorrer deste trabalho, são alguns dos exemplos encontrados para indústria 4.0.

## 1.3 DESAFIOS DA INDÚSTRIA 4.0

A indústria 4.0 traz inúmeras vantagens para os ambientes fabris, porém diferentes desafios foram identificados dos quais soluções ainda estão sendo estudadas no meio acadêmico e por empresas de tecnologia. De acordo com Zhou, Liu e Zhou (2015), desenvolver os conceitos de indústria 4.0 ainda pode levar anos. Os desafios apontados aparecem nas áreas científicas, tecnologia, economia, problemas sociais e política (ZHOU; LIU; ZHOU, 2015).

Desafios no âmbito científico e tecnológico são apontados por Zhou, Liu e Zhou (2015), descritos a seguir:

- Desenvolvimento de dispositivos inteligentes, devido a variedade de segmentos e diferentes necessidades dentro da indústria, assim como os tempos e custos de desenvolvimento para cada setor industrial.
- Cooperação entre diferentes sistemas, dificultando a integração de equipamentos físicos através de protocolos de rede com as demais camadas de infraestrutura, como servidores na nuvem.

- Dificuldade no desenvolvimento de *frameworks* que possam atender a diferentes sistemas físicos, devido a grande variedade de equipamento e máquinas existentes no ambiente fabril.
- Falta de padrões para a validação e teste do CPS implementado, sendo uma etapa importante durante o desenvolvimento de um sistema.
- A análise e processamento de grandes volumes de dados, levando em consideração que máquinas, produtos, produção, equipamentos, entre outros, vão gerar grandes volumes de informação que devem ser processados em tempo real.
- O desenvolvimento de redes de comunicação seguras, devido ao alto número de *hackers*, no qual diferentes meios de invasão são desenvolvidos, sendo que todos os dados que são gerados, podem ser transmitidos através da Internet.

## 1.4 OBJETIVOS

### 1.4.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver um experimento, envolvendo a construção e avaliação de um sistema ciber-físico, utilizando plataformas disponíveis no mercado.

### 1.4.2 Objetivos específicos

Os objetivos específicos deste trabalho são os seguintes:

- Identificar tecnologias computacionais para implementação do conceito de CPS
- Identificar protocolos de comunicação e coleta de dados entre máquinas, sensores e controladores
- Elaborar um cenário de testes, compreendendo a seleção de dados e ferramentas de tratamento apropriadas
- Implementar um CPS simulado, utilizando dados de origem industrial
- Produzir uma análise referente às condições de funcionamento, e resultados obtidos no experimento

## 1.5 JUSTIFICATIVA

Devido ao aumento das necessidades de soluções cada vez mais autônomas, assim como a inevitabilidade da tomada de decisões com faixas de erro cada vez menores dentro de



ambientes fabris, as soluções com aplicação dos conceitos de inteligência artificial se tornam mais atraentes, devido aos ganhos dentro da indústria e as possibilidades de que serão apresentadas ao longo deste trabalho.

Para tal desenvolvimento, o conhecimento em máquinas, equipamentos, protocolos de rede e processos fabris existentes dentro da indústria, podem ser necessários para que aplicações de inteligência artificial no ambiente fabril ocorram de forma mais assertiva. Tais conceitos são estudados em inúmeros momentos dentro dos cursos de diferentes áreas da engenharia, assim como muitos alunos possuem vivências nestas áreas ao decorrer de sua carreira profissional.

## 1.6 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho está organizado em três capítulos. O segundo capítulo apresenta os conceitos de CPS, arquitetura 5C dentro dos conceito de CPS, gêmeo digital (DT) e prognóstico e gestão da saúde (PHM). Desenvolvimentos nas áreas de DT dentro no ambiente fabril e máquinas para indústria, assim como uma solução de CPS descrita como iCPS, são apresentados nesse capítulo.

O terceiro capítulo apresenta a implementação do experimento, com base nos conceitos de CPS, utilizando ferramentas de computação em nuvem. Neste capítulo é apresentada a arquitetura implementada, assim como descrito os limites deste trabalho no desenvolvimento do experimento com base nos conceitos de arquiteturas já existentes. O quarto capítulo apresenta os objetivos atingidos, assim como traz implementações futuras e as contribuições deste trabalho para o desenvolvimento de sistema ciber-físicos.

## **2 BASE TEÓRICA DOS CONCEITOS DE INTELIGÊNCIA ARTIFICIAL NA INDÚSTRIA**

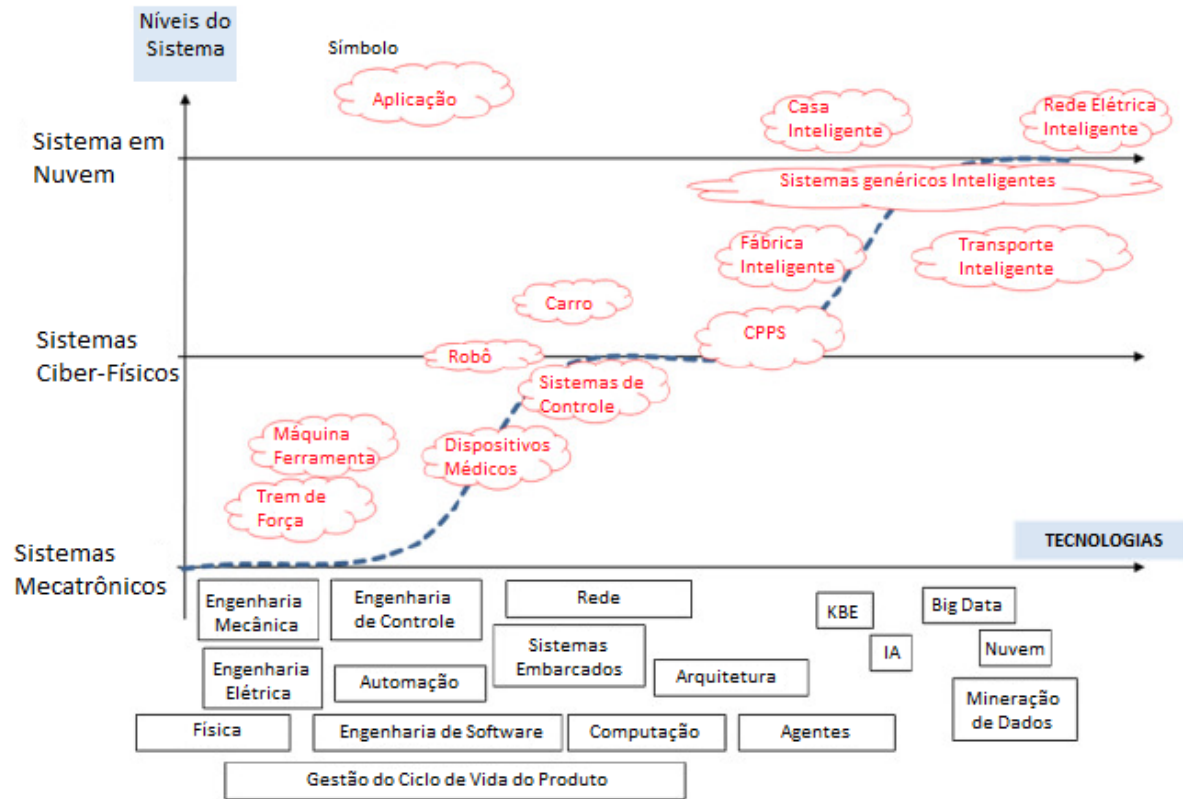
Este capítulo apresenta os conceitos que podem ser utilizados no contexto industrial, para a inserção da Inteligência Artificial no ambiente fabril, em especial tratando-se dos sistemas ciber-físicos (CPS), prognóstico e gestão de saúde (PHM) e gêmeo digital (DT).

### **2.1 SISTEMAS CIBER-FÍSICOS**

Conforme definido por Lee, Bagheri e Kao (2015), CPS é uma combinação de processos físicos e códigos computacionais, possuindo a função de coordenar os sistemas físicos em tempo real.

A quarta revolução industrial integra as redes, trazendo os conceitos de produção inteligente. Essa revolução pode ser tratada como a integração dos conceitos de CPS e IoT (JAMALUDIN; ROHANI, 2018). A indústria 4.0 é a transição dos conceitos de mecatrônica para os conceitos de CPS, ilustrada na Figura 2. Essa transição é necessária, devido aos custos de implementação de sistemas automatizados através dos conceitos de mecatrônica, nos quais profissionais de diferentes disciplinas são necessários, causando atrasos nos desenvolvimentos devido as lacunas existentes entre os conhecimento desses profissionais (JAMALUDIN; ROHANI, 2018).

Figura 2 – Transição dos conceitos de mecatrônica para CPS.

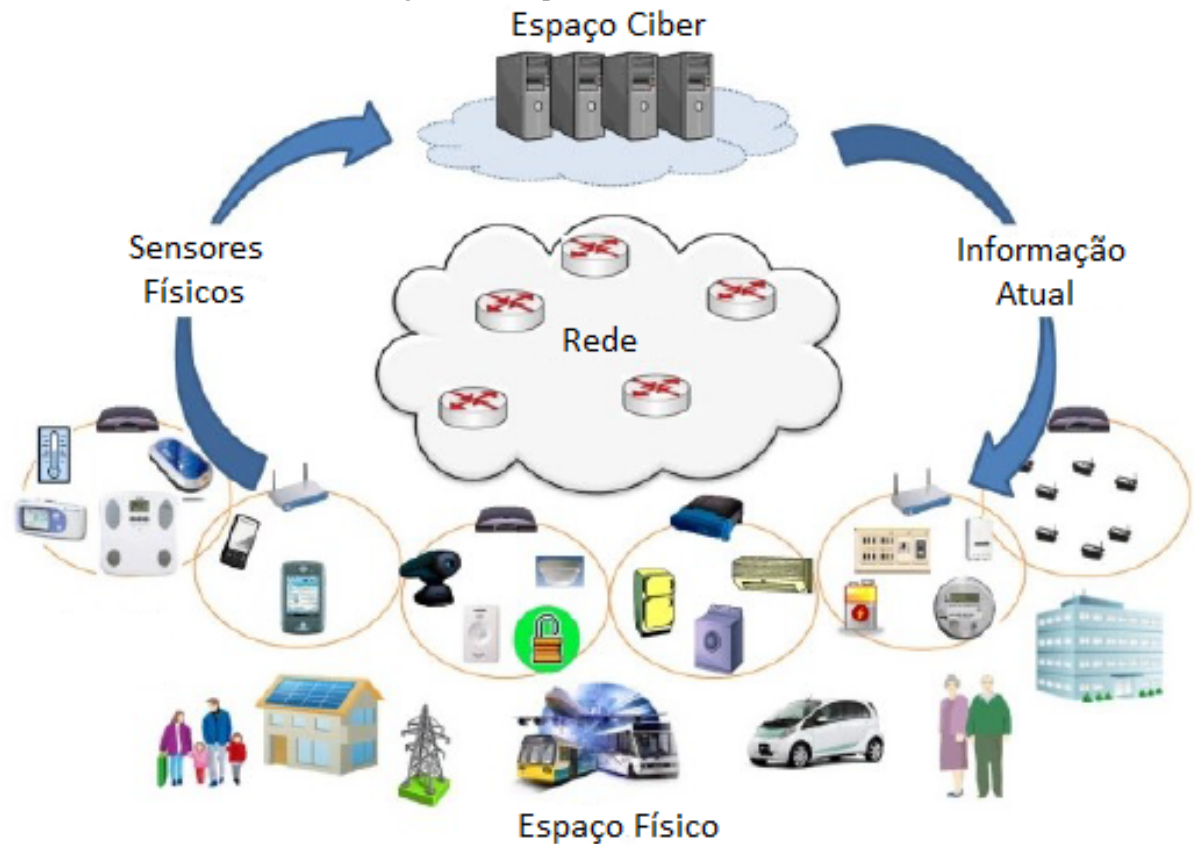


Fonte: (JAMALUDIN; ROHANI, 2018)

O principal propósito do CPS é conduzir a realimentação dos sistemas de controle no ambiente computacional, utilizando a combinação de 3C (computação, comunicação e tecnologias). A aplicação do CPS exige alta performance dos sistemas físicos e computacionais, sendo apresentado ainda no século 20, consistindo de três principais elementos que são os sensores, atuadores e controladores (JAMALUDIN; ROHANI, 2018).

A arquitetura básica do CPS é composta por componentes conectados através de uma rede segura. A Figura 3 ilustra a arquitetura básica para sistemas CPS, essa sendo composta por sensores e atuadores no espaço físico, centro de dados e controle localizados no espaço ciber. Como pode ser identificado, os sensores indicam o estado real do espaço físico, após processamento no espaço ciber, atualizam as informações do atuadores realizando as ações desejadas/necessárias (JAMALUDIN; ROHANI, 2018).

Figura 3 – Arquitetura básica do CPS.

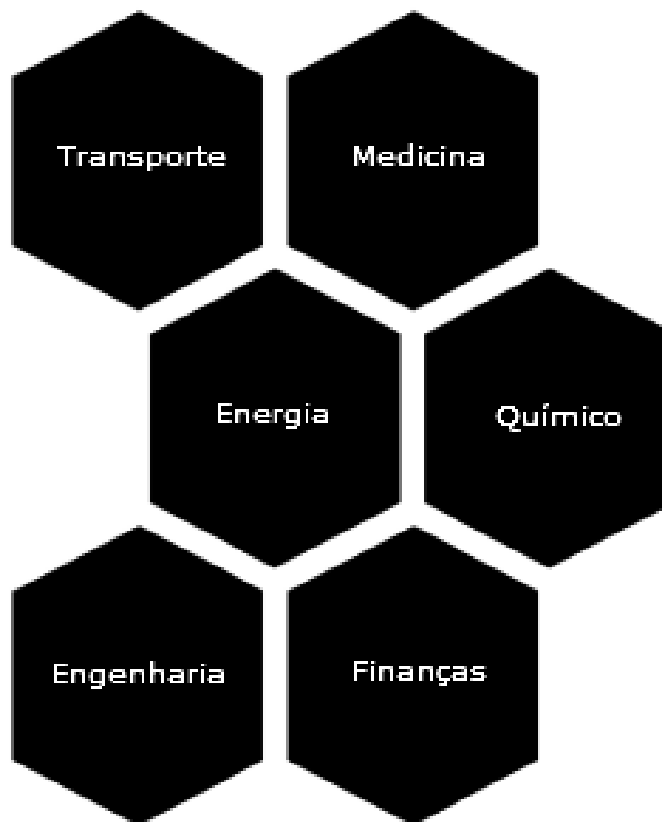


Fonte: (JAMALUDIN; ROHANI, 2018)

Devido ao aumento do mercado global de produtos, as empresas necessitam aumentar sua produção, pois equipamentos *stand-alone* são insuficientes para este conceito de produção. Neste ponto, se torna necessária maior integração entre os equipamentos, ocasionando aumento no número de dados gerados, criando a necessidade de métodos para gerenciar e tratar essas informações (LEE; BAGHERI; KAO, 2015).

Conforme proposta de Jamaludin e Rohani (2018), atualmente, o conceito de CPS pode ser aplicado em inúmeras áreas, por exemplo, instrumentos médicos digitais, tecnologias de controle, controle espacial, sistemas sociais e autônomos, ilustrado na Figura 4.

Figura 4 – Áreas de aplicação de CPS



Fonte: (JAMALUDIN; ROHANI, 2018)

## 2.2 ARQUITETURA 5C

O CPS consiste de duas funcionalidades principais, sendo elas garantir aquisição dos dados do espaço físico em tempo real e atualizar através do espaço cibernético, porém são definições vagas, que dificultam a implementação de CPS (LEE; BAGHERI; KAO, 2015).

A arquitetura proposta por Lee, Bagheri e Kao (2015) consiste em um guia para implementação de CPS, conhecida como arquitetura 5C, sendo composta por cinco camadas, ilustrada na Figura 5, sendo estas conexão, conversão, ciber, cognição e configuração. Esse conceito traz vantagens ao ambiente fabril, como funções de PHM, cognição para tomada de decisões, habilita modelos DT, facilita a conexão de diferentes máquinas de forma rápida, entre outros (ALAM; EL SADDIK, 2017).

Figura 5 – Etapas da arquitetura 5C



Fonte: (LEE; BAGHERI; KAO, 2015)

A camada de conectividade é a primeira etapa para implementação da arquitetura 5C, conforme ilustrado na Figura 5, essa consiste na aquisição de dados, sendo feita por sensores, controladores e/ou sistemas *Manufacturing Execution Systems* (MES). A transferência de dados, apresentada por Nascimento Marques Junior et al. (2018), pode ser feita através de protocolos que vêm sendo usados para aplicação em IoT, tal como *Open Platform Communications Unified Architecture* (OPC UA) (OPC Foundation, 2019a), *MTCConnect* (MTCConnect Institute, 2019) e/ou *Message Queuing Telemetry Transport* (MQTT) (MQTT, 2019).

A camada de conversão é a segunda etapa, essa tem como função trazer as funções de PHM, por exemplo, para a máquina. Para isso, diferentes algoritmos vêm sendo implementados, a fim de calcular o tempo de vida restante dos equipamentos, ferramentas, entre outros. A partir dessa camada, a máquina adquire funções de "autoconsciência", podendo indicar problemas que surjam com sua "saúde". A Figura 6 ilustra que a máquina irá começar a ser analisada como um todo, não mais seus sinais separadamente conforme primeira camada da arquitetura 5C. Nessa camada, através da proposta de Nascimento Marques Junior et al. (2018), é implementado um sistema para conversão dos dados da máquina em dados que serão transmitidos para camadas mais altas.

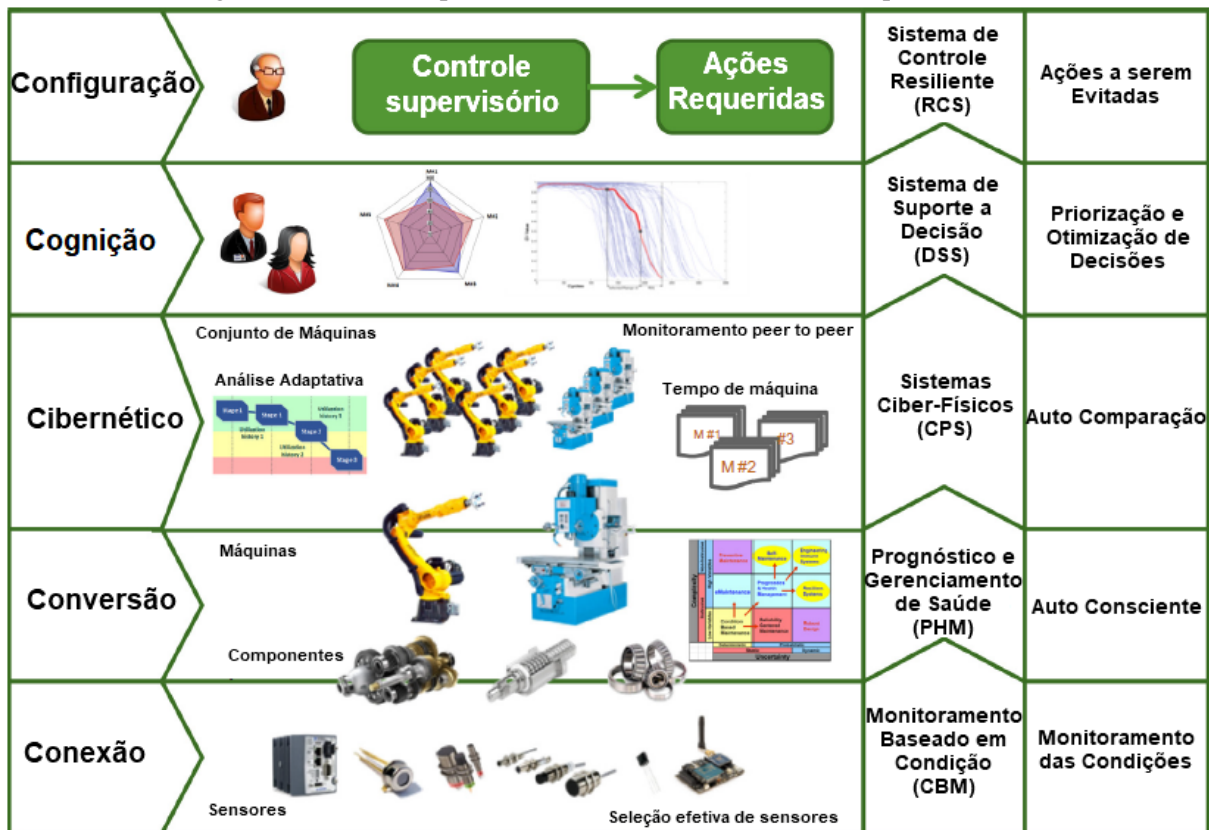
A camada ciber é a terceira etapa, qual se inicia a criação da rede entre máquinas, essa funciona como espécie de *hub*. Nessa etapa, inúmeros dados são coletados de diferentes equipamentos em uma única vez, conforme seus tempos de atualização. Também se faz possível, através dos históricos das máquinas, analisar o comportamento comparando a performance

atual, com o histórico da máquina (LEE; BAGHERI; KAO, 2015). A camada cibernética, conforme ilustrado na Figura 6, estabelece a conexão do conjunto de máquinas.

A camada de cognição possui interface que traz todo conhecimento adquirido até a terceira camada, nessa etapa são apresentados gráficos, tabelas, entre outros métodos visuais que venham a ser úteis para uma análise mais profunda da situação no ambiente fabril.

A camada de configuração é a última etapa da arquitetura 5C e tem como função configurar os equipamentos físicos através de ambientes computacionais. Permitem que os equipamentos se auto configurem e se adaptem às mudanças solicitadas/necessárias.

Figura 6 – Técnicas aplicadas nas diferentes camadas da arquitetura 5C.



Fonte: (LEE; BAGHERI; KAO, 2015)

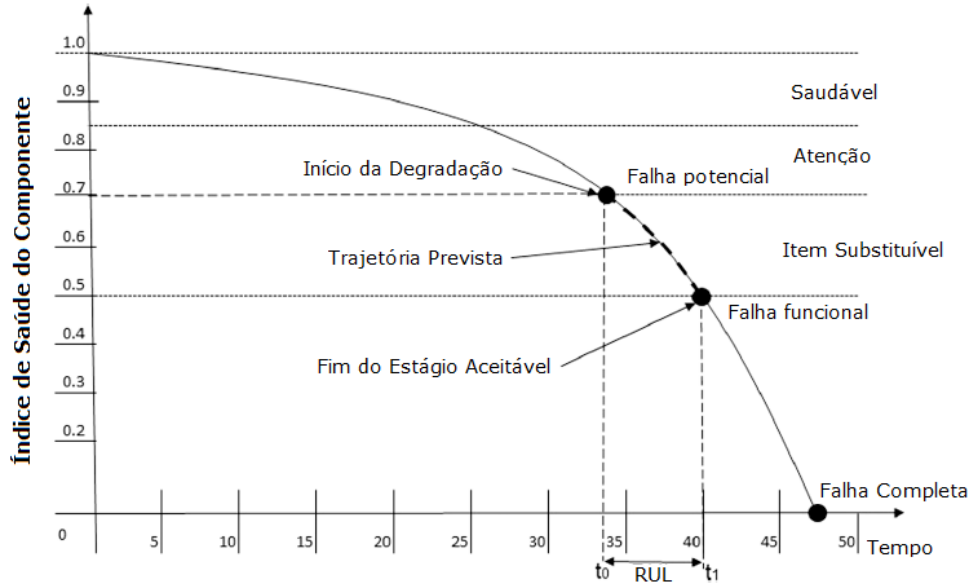
### 2.3 PROGNÓSTICO E GESTÃO DA SAÚDE DAS MÁQUINAS

Os conceitos de prognóstico e gestão da saúde (do inglês, *Prognostics Health Management*, sigla PHM) têm como foco a determinação dos tempos de degradação de um componente da máquina, tornando possível a antecipação, para que manutenções sejam utilizadas antes de falhas críticas ocorrerem, tendo como ganho o aumento da vida útil das máquinas (NUÑEZ; BORSATO, 2018).

A Figura 7 apresenta as fases de degradação de um equipamento, apresentando etapas onde a máquina inicia seu funcionamento após a aquisição, até o momento das primeiras ocor-

rências de falha, identificando o tempo de vida restante (em inglês *Remaining Useful Life*, sigla RUL).

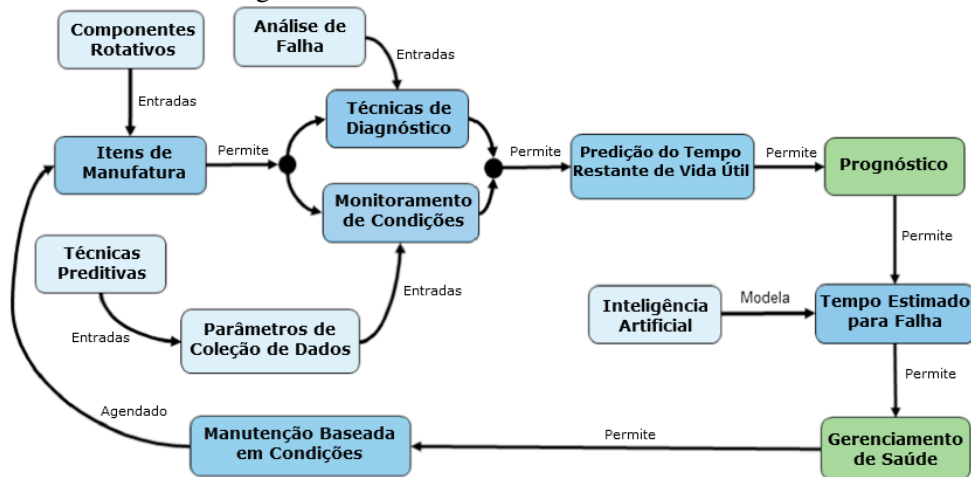
Figura 7 – Fases de degradação das máquinas



Fonte: (NUÑEZ; BORSATO, 2018)

Dentro dos conceitos de CPS, podemos utilizar a Figura 8 como exemplo de fluxograma para funcionamento dos conceitos de PHM no âmbito de CPS.

Figura 8 – Fluxo de funcionamento do PHM



Fonte: (NUÑEZ; BORSATO, 2018)

Diferentes técnicas podem ser implementadas dentro dos conceitos de CPS para realizar os prognósticos de máquina. O método proposto por (LEE; BAGHERI; KAO, 2015) consiste de três etapas, sendo elas coleção de *snapshot*, identificação de similaridades e otimização futura, que serão descritas a seguir.

A coleção *snapshot* consiste em armazenar as informações em um histórico, sempre

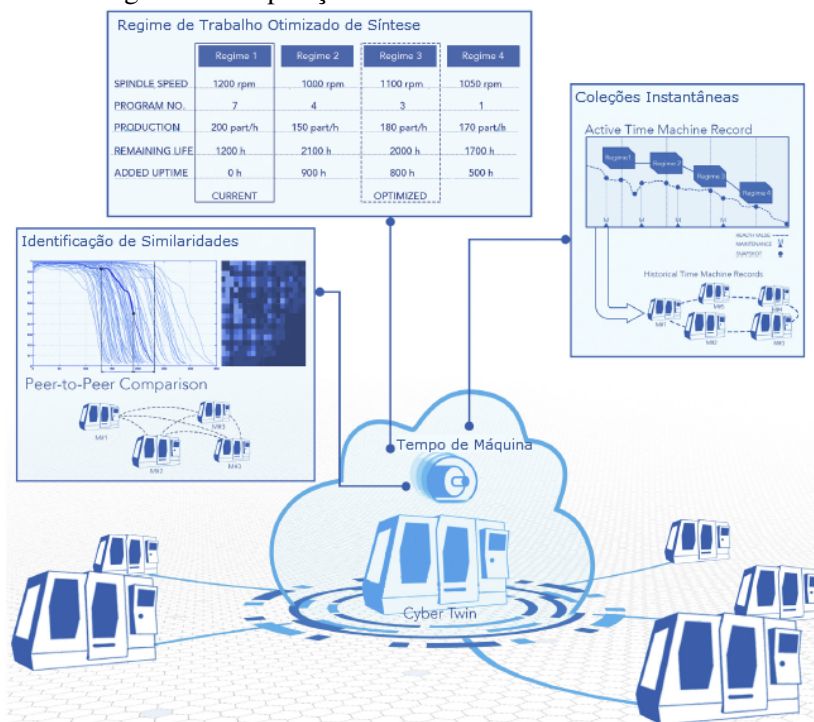


que ocorrer mudanças significativas nos tempos ou sensores. Este conceito é aplicado apenas para uma máquina, sem utilizar os demais equipamentos equivalentes.

A identificação de similaridades é utilizada na terceira camada da arquitetura 5C, da qual inúmeros dados são coletados simultaneamente. Desta forma, o desenvolvimento de um algoritmo poderia identificar padrões em diferentes equipamentos. Após a identificação desses padrões, a predição de falhas se torna mais assertiva.

Na etapa de otimização futura se faz possível prever os momentos em que falhas no equipamento podem ocorrer, pois devido aos históricos e padrões gerados, se torna possível prever as falhas antes que as mesmas ocorram. A Figura 9 demonstra a integração das funções descritas.

Figura 9 – Disposição dos sistema no conceito PHM



Fonte: (LEE; BAGHERI; KAO, 2015)

## 2.4 GÊMEO DIGITAL

Os primeiros conceitos de gêmeo digital (DT) foram apresentados em 2003, e desde então possuem inúmeras aplicações no meio acadêmico e industrial (TAO et al., 2019).

O DT surgiu originalmente para simulação no meio aeroespacial. Atualmente é um dos principais tópicos estudados em aplicação para indústria 4.0, como forma de representar equipamentos físicos no estado digital com atualização em tempo real, utilizando diferentes sensores e/ou controladores (NEGRI; FUMAGALLI; MACCHI, 2017).

O DT é um conceito que tem como fundamento o *link* entre o mundo virtual e o mundo

físico, algo que se faz necessário para aplicar os conceitos de CPS com maior eficácia, permitido inicialmente devido ao avanço tecnológico dos fabricantes de sensores e soluções para IoT (NEGRI; FUMAGALLI; MACCHI, 2017). O conceito de DT tem como característica a integração entre o meio físico e o meio cibernético.

Inúmeras patentes e estudos acadêmicos são realizados com foco neste assunto, nas diferentes áreas, como soluções para indústria, aeroespacial, medicina, entre outros. As primeiras aparições dos conceitos de DT são datados de 2003, porém, devido a falta de tecnologia necessária na época, apenas em 2012 os conceitos de DT tomaram maior foco de pesquisa. Esse ocorreu quando a *National Aeronautics and Space Administration* (NASA) apresenta o conceito de DT com o foco em simulação perfeita, a partir de meios computacionais, a simulação é realizada com alta precisão, a fim de prever o comportamento de um equipamento físico, desta forma, reduz os riscos de erros em lançamentos de espaçonaves. As simulações eram baseadas na aquisição dos dados de outros sistemas reais (físicos) (TAO et al., 2019).

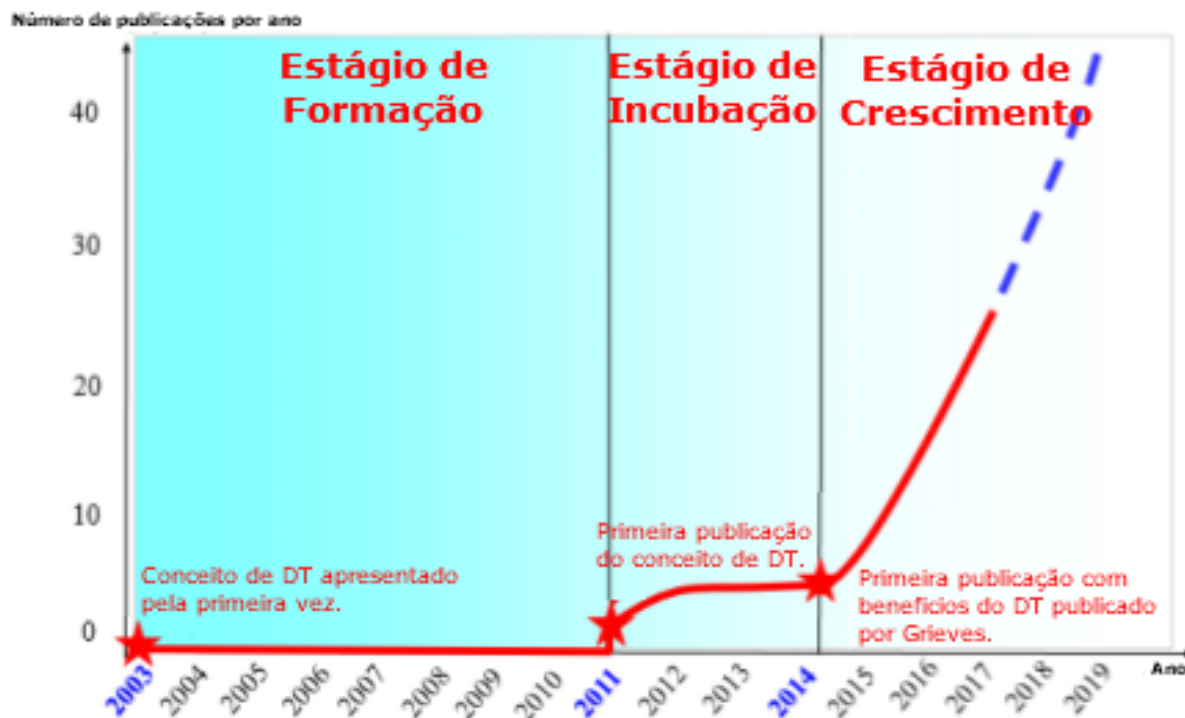
Existem diferentes opiniões entre pesquisadores sobre o que é DT, entre eles, alguns que o utilizam apenas como sistema de simulação perfeita, para outros pesquisadores o conceito é mais amplo e possui três dimensões, conforme demonstrado na Figura 10, sendo os meios físico, virtual e de conexões (troca de dados) (TAO et al., 2019).



Fonte: (TAO et al., 2019)

Conforme estimado por Tao et al. (2019), nos próximos três a cinco anos ocorrerá uma evolução mais acentuada da tecnologia de DT, conforme demonstrado na Figura 11.

Figura 11 – Tendência evolutiva dos conceitos de DT



Fonte: (TAO et al., 2019)

Nos conceitos de indústria 4.0, Tao et al. (2019) apontou algumas das seguintes aplicações para DT:

- Modelamento e simulação de novos produtos, conhecido como software de comissionamento virtual;
- Sistemas para aumentar a previsibilidade, flexibilidade e eficiência de linhas de produção;
- Prognósticos de máquinas, ou seja, aplicação dos conceitos de PHM;
- Entre outras possíveis aplicações, como evitar colisões de sistemas autônomos, ciclo de vida de produtos, entre outros.

## 2.5 IMPLEMENTAÇÃO DE SISTEMAS GÊMEOS DIGITAIS

No contexto da indústria 4.0, o conceito de gêmeo digital (DT) surge como técnica que pode ser aplicada para implementar situações em que linhas de produção se tornam autônomas, reduzindo a necessidade de pessoas dentro dos ambientes fabris, aumentando a precisão desses sistemas.

Atualmente, os estudos de DT estão sendo desenvolvidos para indústria, empresas como *Siemens* (SIEMENS, 2019), *Schneider Electric* (MCMULLEN, 2017), *MapleSoft* (Ma-

plesoft, a division of Waterloo Maple Inc., 2019) são alguns dos exemplos que podem ser encontrados.

As empresas propõem soluções para prever o comportamento de sistemas automatizados, analisar o comportamento gravitacional e funcionamento de linhas semi-automatizadas ou totalmente automatizadas. Essa linha de desenvolvimento se baseia nos conceitos de modelamento e simulação, uma das áreas de estudo do DT. Conceitos de DT também são aplicados na medicina, aeroespacial, entre outros.

O estudo conduzido nesse trabalho tem como foco a análise de soluções para a indústria, explorando suas características para implementar um DT, em situação simulada, para controlar inúmeras máquinas, explorando os conceitos para integrar e desenvolver um CPS utilizando tais conceitos.

## 2.6 TECNOLOGIAS PARA IMPLEMENTAÇÃO DE SISTEMAS CIBER-FÍSICOS E GÊMEOS DIGITAIS

Esta seção apresenta diferentes aplicações dos conceitos de DT e CPS no contexto da indústria. As técnicas apresentadas foram identificadas a partir de artigos publicados para a área, com o objetivo de demonstrar protocolos, metodologias, perfis de funcionamento, entre demais assuntos relevantes para implementação de DT e CPS.

### 2.6.1 Configuração do sistema físico usando gêmeo digital

O conceito implementado por Haag e Anderl (2018) consiste em configurar um equipamento físico remotamente através do gêmeo digital. Dessa forma, uma máquina conectada à Internet (gêmeo físico) pode ser configurada remotamente, assim como as informações a serem atualizadas diretamente no gêmeo físico. O sistema permite que seja analisado seu comportamento, através de um modelo assistido por computador (*computer-aided design*, CAD) e posteriormente os parâmetros são transferidos ao sistema físico, através de um painel de controle do equipamento.

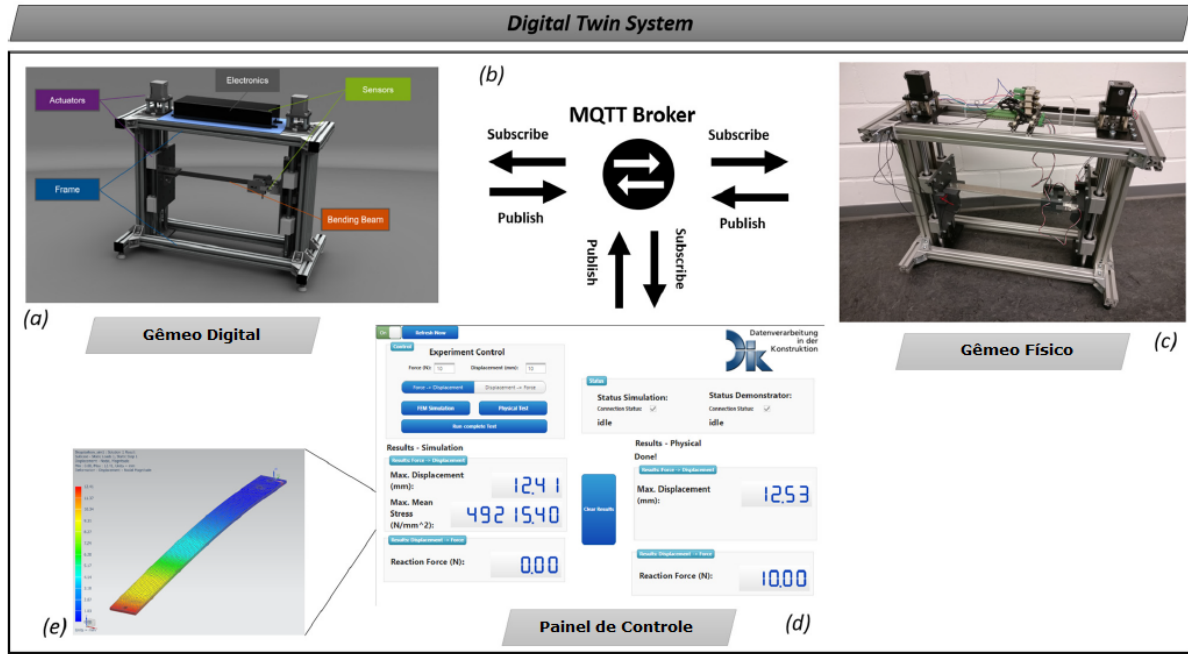
Para essa implementação do sistema foi utilizado o protocolo de comunicação MQTT. Haag e Anderl (2018) afirma que, desta forma facilita a inclusão de novos equipamentos através deste método de comunicação.

A Figura 12 demonstra o comportamento do sistema desenvolvido por Haag e Anderl (2018), onde as etapas do sistema são divididas em cinco pontos, descritos a seguir:

- Figura 'a': Demonstrativo CAD do sistema com indicação do componentes utilizados no equipamento, assim como sua localização
- Figura 'b': Conceito de funcionamento do protocolo MQTT
- Figura 'c': Dispositivo físico

- Figura 'd': Painel de controle do equipamento onde são inseridos os parâmetros
- Figura 'e': Demonstrativo dos resultados obtidos na peça

Figura 12 – Digital Twin usando protocolo MQTT.

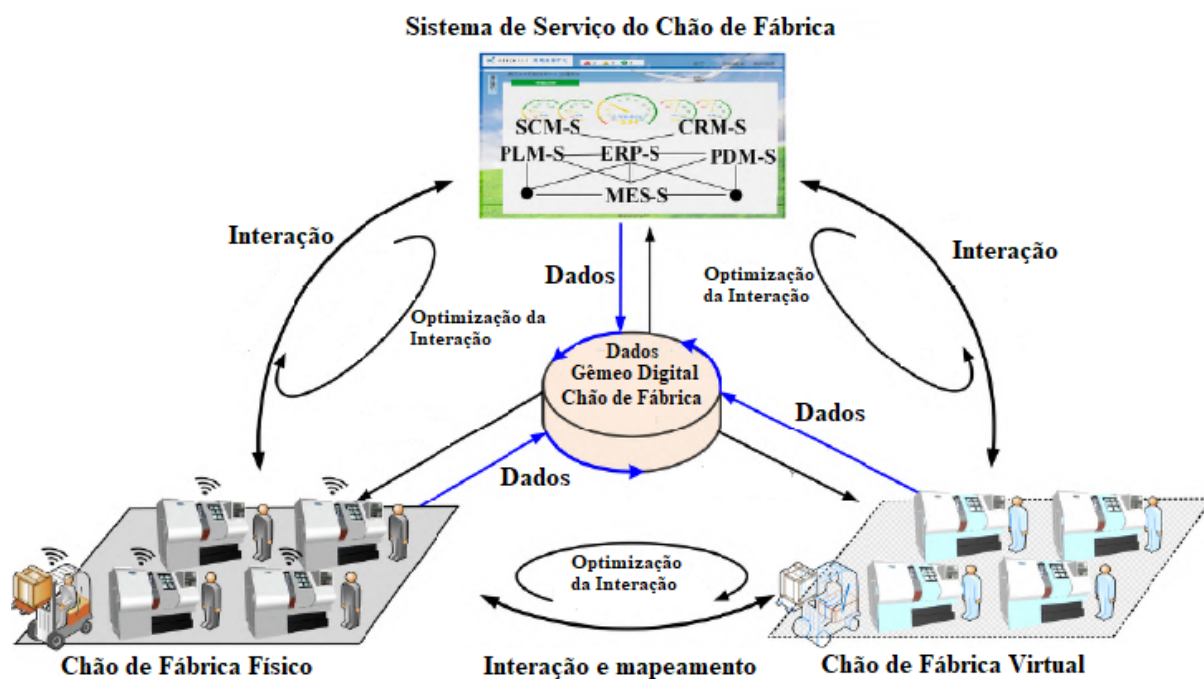


Fonte: (HAAG; ANDERL, 2018)

## 2.6.2 Gêmeo digital no chão de fábrica

Tao e Zhang (2017) apresentam o conceito de gêmeo digital no chão de fábrica (em inglês *Digital Twin Shop-Floor*). Esse é composto por quatro elementos conforme Figura 13 Tao e Zhang (2017) e são descritas nas sub-seções a seguir.

Figura 13 – Conceito de implementação de DTS.



Fonte: (TAO; ZHANG, 2017)

#### 2.6.2.1 Chão de fábrica físico

Nesta camada os elementos presentes são equipamentos, materiais, pessoas e disposição física destes. Diferentes métodos podem ser utilizados para adquirir dados dos elementos citados, porém, em sua grande maioria, estas técnicas consistem de diferentes protocolos de comunicação, tipos de dados, entre outras características que são específicas de cada fabricante ou desenvolvedor de produtos tecnológicos (TAO; ZHANG, 2017).

As diferenças apontadas aumentam a complexidade de exportação, e muitas vezes podem impossibilitar o desenvolvimento de sistemas que tenham a capacidade de aderir a esses protocolos (TAO; ZHANG, 2017). Uma solução que é apontada, seria a utilização de protocolos que abrangem diferentes sistemas de automação, como por exemplo a *open platform communication* (OPC). Essa plataforma vem ganhando espaço entre os fabricantes de equipamentos de automação, entre outros, desta forma, usuários finais podem fazer a aquisição de diferentes marcas de equipamentos sem ter a preocupação se este equipamento possuirá comunicação ou não com os demais dispositivos (BRUCKNER et al., 2019).

#### 2.6.2.2 Chão de fábrica virtual

A camada virtual se faz composta por quatro principais níveis, sendo eles o modelamento do equipamento de forma tridimensional, elementos físicos (temperatura, resistência, entre outros), comportamento do sistema e regras para capacitar o equipamento para avaliar,

prever e raciocinar (TAO; ZHANG, 2017).

Atualmente, com a capacidade de sistemas tridimensionais, se faz possível ter visão sobre o comportamento de um sistema antes mesmo de ocorrer seu funcionamento no meio físico (TAO; ZHANG, 2017).

#### 2.6.2.3 Sistema de serviço do chão de fábrica

Nesta camada, conforme demonstrado na Figura 13, sistemas em base computacional de alto nível são utilizados, ou seja, sistema ERP, MES, entre outros. Desta forma, ocorre o planejamento de recursos, sequencia de produção, cronograma, entre outros. Após a conclusão do planejamento, o sistema programa o meio virtual para, após verificação do funcionamento, encaminhar para o sistema físico onde a produção de um produto ocorrerá, por exemplo (TAO; ZHANG, 2017).

#### 2.6.2.4 Dados gêmeos digitais no chão de fábrica

O último elemento tem como função a integração de dados dos outros três elementos, anteriormente comentados. Tem como objetivo a conversão dos dados gerados no meio físico, virtual e de sistema em um pacote unificado que permita a interpretação entre elementos. Desta forma, é possível realizar a troca de informações entre sistemas que até então possuem meios diferentes de comunicação (TAO; ZHANG, 2017).

### 2.6.3 Sistema ciber-físico inteligente

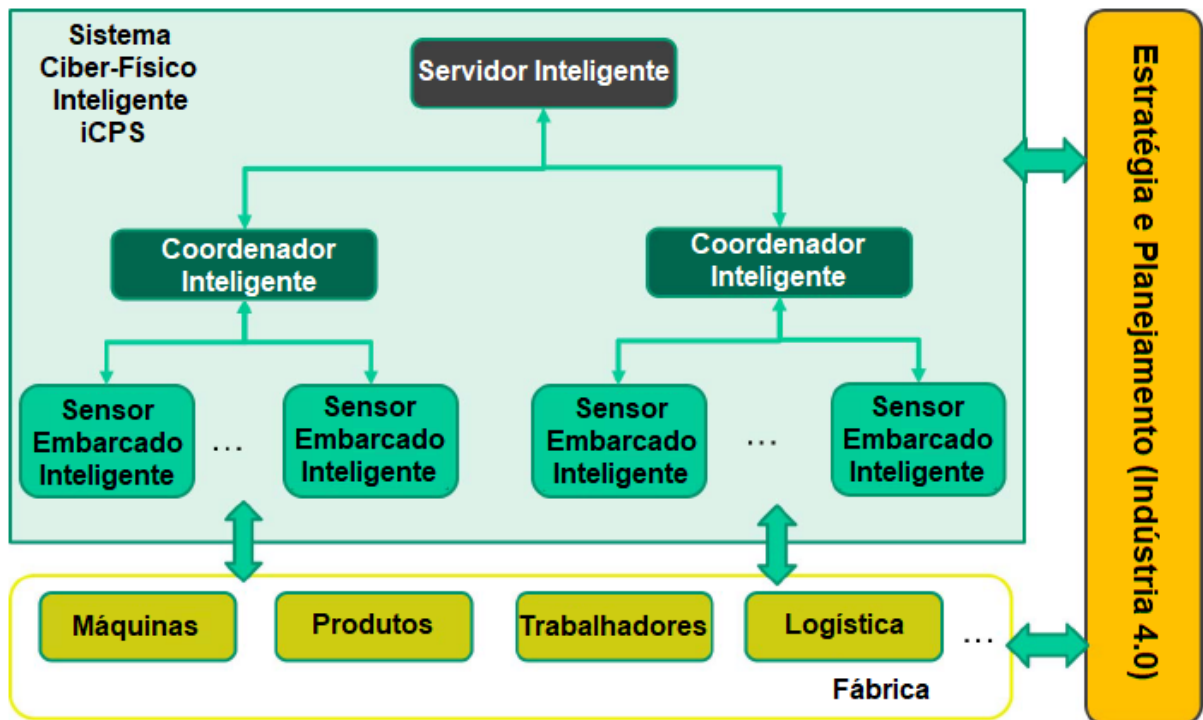
A proposta de sistema ciber-físico inteligente foi apresentada por Cogliati et al. (2018), sendo essa uma arquitetura de três camadas. O objetivo desta metodologia constitui em trazer funções inteligentes aos sistemas CPS, como previsão de falhas, comportamento automático e auto-adaptação. O conceito tem como função atuar em todo o meio físico do chão de fábrica, desde trabalhadores, máquinas, produtos e logística conforme apresentado pela Figura 14. O conceito de iCPS se faz composto de três etapas (COGLIATI et al., 2018), sendo elas:

- Sensor embarcado inteligente, nessa etapa ocorre a aquisição de dados da fábrica, que serão enviados para a próxima etapa. A utilização de uma placa embarcada se faz necessária, com o foco de detectar mudanças nos sinais após o desenvolvimento de um modelo de previsão dessa aquisição de dados.
- Coordenador inteligente, conforme ilustrado na Figura 14, essa etapa tem por função gerenciar os dados enviados pela camada anterior utilizando um computador pessoal. A comunicação deste serviço pode ser realizada por diferentes protocolos, os quais sendo implementados dentro dos conceito de indústria 4.0. Pode estar localizado em local único

ou em diferentes fábricas.

- Servidor inteligente, a última camada tem por função disponibilizar as informações geradas pela primeira camada aos setores e pessoas responsáveis pelo planejamento e estratégias da indústria. Tem como função armazenar, processar e disponibilizar a visualização das informações adquiridas.

Figura 14 – Conceito de implementação de DTS.



Fonte: (COGLIATI et al., 2018)



### 3 SISTEMAS CIBER-FÍSICOS: CONSTRUÇÃO DE UM EXPERIMENTO

O conceito de sistemas ciber-físicos (CPS) consiste na integração dos meios físico e ciber, respectivamente as máquinas e o ambiente virtual. Tal integração permite a coleta de dados e a atualização do estado dos atuadores no meio físico.

Neste contexto, o objetivo geral deste trabalho foi desenvolver um experimento, englobando a construção e a avaliação de um CPS. Para isso, utilizou-se uma plataforma disponível no mercado, assim como protocolos e métodos para compreensão do funcionamento de um CPS na indústria. Definiu-se etapas, listadas a seguir, e detalhadas ao longo do capítulo:

1. Planejar a arquitetura do experimento
2. Coletar dados para uso no experimento
3. Conceitos básicos de redes neurais recorrentes
4. Compreender as funcionalidades da computação em nuvem
5. Selecionar uma plataforma de software e descrever as funções disponibilizadas pela plataforma, a fim de compreender os recursos disponíveis para desenvolvimento de um CPS
6. Examinar protocolos e métodos de comunicação para envio dos dados
7. Implementar o CPS
8. Apresentar resultados obtidos

Cada uma destas etapas envolveram análises, testes e constituem parte relevante do experimento realizado. Sendo assim, cada uma delas é detalhada nas seções a seguir.

#### 3.1 PLANEJAMENTO DA ARQUITETURA DO SISTEMA

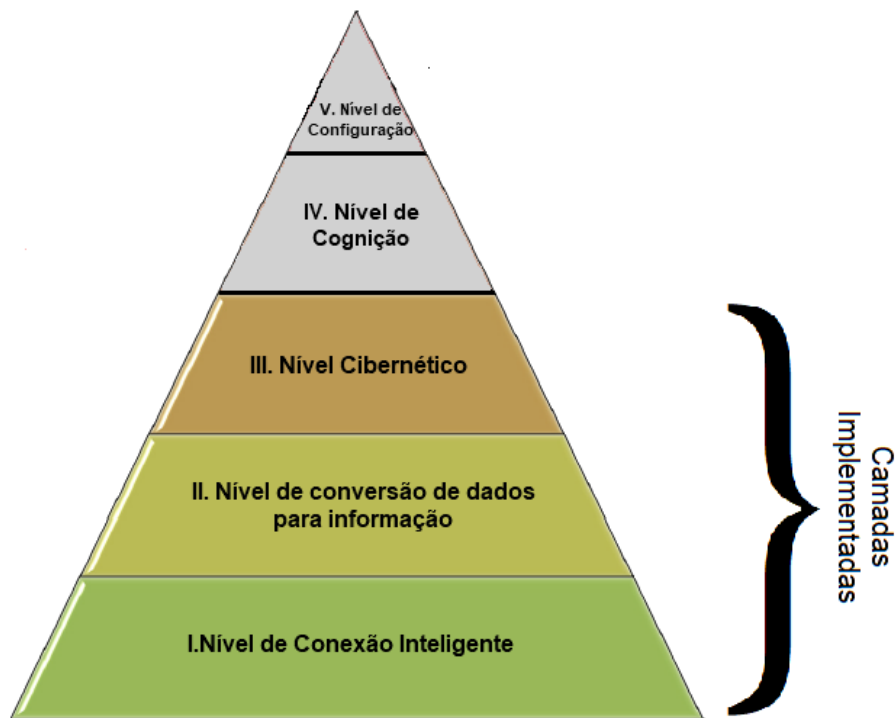
Planejar um experimento que implemente o CPS implica em considerar o uso de métodos existentes. Originalmente o CPS é composto por três níveis, sendo eles computação, comunicação e tecnologias (previamente descritos na seção 2.1). Como conceito, o CPS faz parte dos componentes da indústria 4.0 (JAMALUDIN; ROHANI, 2018). Entretanto, não há consenso sobre como implementá-lo. Assume-se que o CPS atue em duas etapas: aquisição de dados do espaço físico e atualização do espaço cibernético. Porém, os conceitos originais de CPS são muito abrangentes, podendo dificultar sua implementação (LEE; BAGHERI; KAO, 2015).

A fim de propor modelos para a construção de CPS, autores como Lee e outros (LEE; BAGHERI; KAO, 2015) propuseram uma arquitetura denominada 5C, composta em cinco camadas (conexão, conversão, ciber, cognição e configuração), funcionando como um guia para

implementação de um CPS. Desta forma, o experimento foi desenvolvido com base nesta arquitetura.

O escopo deste trabalho está limitado ao desenvolvimento das três primeiras camadas, conforme ilustrado na Figura 15. Tais camadas foram priorizadas no experimento proposto, pois são as primeiras a serem implementadas, se mostrando essenciais, sendo possíveis de serem concebidas e desenvolvidas no contexto desse trabalho. As camadas de cognição e de configuração, extremamente relevantes, não foram contempladas, podendo ser desenvolvidas em trabalhos futuros.

Figura 15 – Camadas implementadas



Fonte: Adaptado da figura de Lee, Bagheri e Kao (2015)

As camadas implementadas no experimento são as seguintes:

- A camada de conexão é responsável pela coleta de dados dos sensores e controladores.
- A camada de conversão tem como função a adequação dos dados coletados dos sensores, também possibilita a implementação dos conceitos de aprendizado de máquina, permitindo a análise de prognóstico do equipamento. Nessa etapa, se faz necessária a implementação de protocolos de comunicação.
- A camada ciber é responsável pelo conjunto de informações das diferentes máquinas, foi implementada em servidor na nuvem, utilizando serviços já existentes.

### 3.2 SELEÇÃO DE DADOS

Para o experimento foram selecionados dois conjuntos de dados, com características diferentes, permitindo explorar os recursos de aprendizado de máquina. O primeiro *dataset* foi obtido por meio da plataforma disponibilizada pela IEEE <sup>1</sup>, desenvolvido por Xianhe Wen e contém dados referentes a um robô composto de seis eixos, modelo IRB 1200 da marca ABB, apresentado na Figura 16. O *dataset* contém 5936 segundos de dados coletados, com intervalo de um segundo entre cada aquisição das informações dos eixos, possuindo um total de dezoito atributos, sendo três atributos para cada eixo. Nesse projeto foi utilizado apenas um dos atributos para o experimento, sendo este os dados de "T1". O autor não descreve de forma clara o significado de cada atributo, impedindo a total compreensão dos dados, porém o experimento não leva em consideração o significado ou método utilizado para coleta dos dados, apenas utiliza-os com intuito de simular a aquisição de dados na camada de conversão. Pelas razões citadas, o *dataset* foi considerado aplicável neste estudo.

Figura 16 – Robô modelo IRB 1200



Fonte: (ABB, 2019)

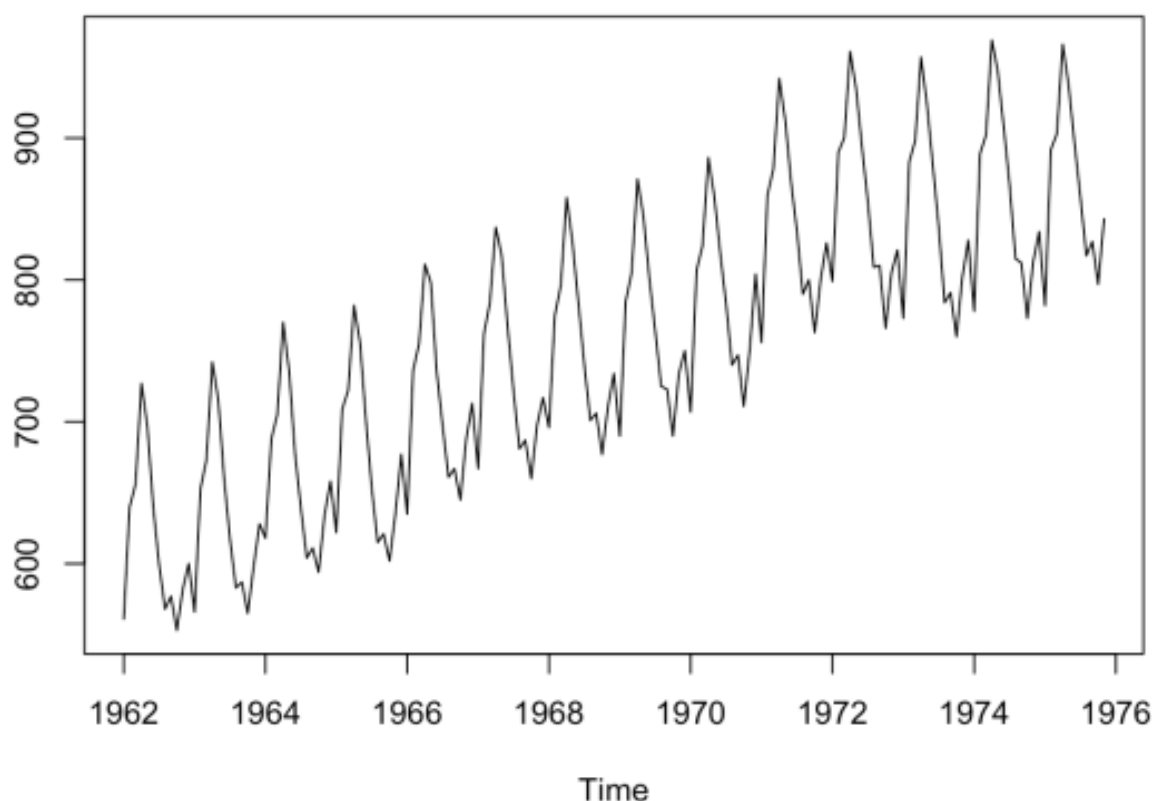
O segundo *dataset*, disponibilizado no *WordPress* <sup>2</sup> e desenvolvido por Subha Yoga-

<sup>1</sup>Disponível em: <<https://www.ieee-dataport.org/documents/history-data-irb-1200-grinding-robot>> Acesso em: 12,nov, 2019.

<sup>2</sup>Disponível em: <<https://subhayo.wordpress.com/2017/09/12/time-series-analysis-of-monthly-milk-production/>> Acesso em: 13,nov, 2019.

nandan, corresponde a soma de leite produzido em libras por mês, dentro de um período de quatorze anos. O *dataset* em questão possui apenas um atributo (além do mês e ano da produção), correspondendo a quantidade de leite produzido. O *dataset* contém 168 instâncias referentes ao número de meses em que foram coletadas as quantidades de produção. O gráfico demonstrando o comportamento da produção de leite é ilustrado na Figura 17 (YOGANANDAN, 2017).

Figura 17 – Comportamento da produção de leite



Fonte: (YOGANANDAN, 2017)

Os *datasets* selecionados possibilitam o uso de ferramentas para séries temporais, a fim de prever o comportamento futuro do sistema. Por meio da comparação entre dados previstos e reais é possível aplicar o conceito de prognóstico e gestão da saúde de máquina, conforme visto na seção 2.3. Nessa categoria se encaixam os sistemas de predição. Para o desenvolvimento do sistema de predição podem ser utilizadas as redes neurais recorrentes (em inglês *Recurrent Neural Networks*, sigla RNN), tema da próxima seção.

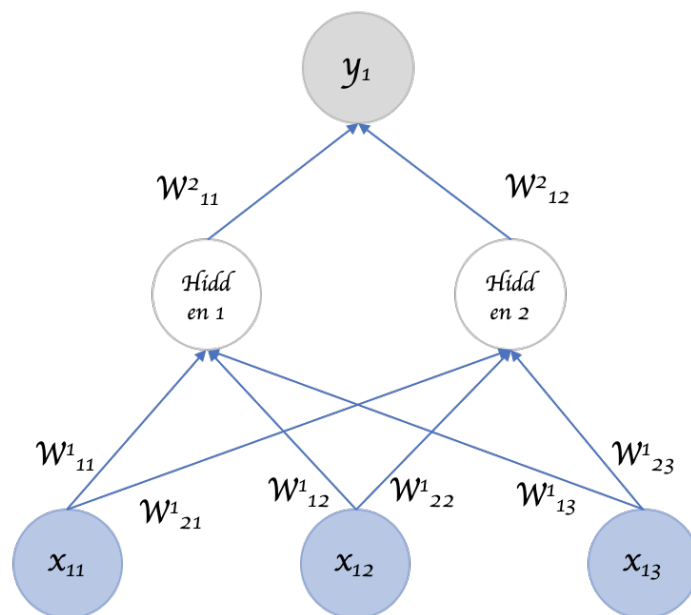
### 3.3 CONCEITOS BÁSICOS DAS REDES NEURAI RECORRENTES

No contexto da indústria 4.0 espera-se fazer uso dos conceitos de aprendizado de máquina. Eles permitem a tomada de decisões ou análise dos dados, sem a necessidade de uma

avaliação ou intervenção humana. Os métodos de aprendizado de máquina permitem a análise de maiores quantidades de dados em menor espaço de tempo, podendo tratar volumes maiores de informação. As redes neurais fazem parte dos métodos conexionistas da área de aprendizado de máquina.

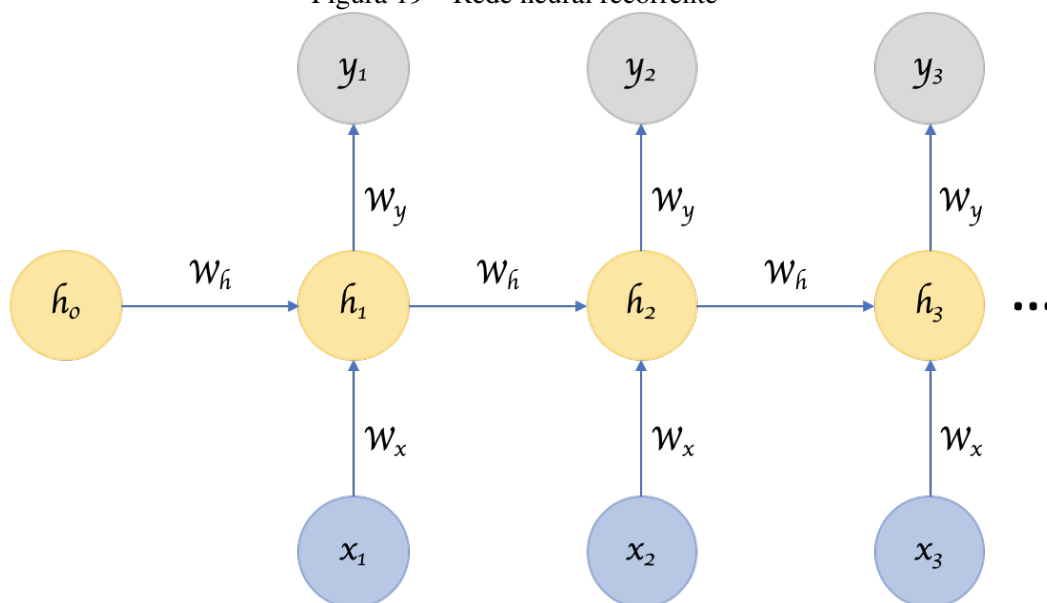
A RNN é uma variação do modelo básico de redes neurais, desenhada com objetivo de trabalhar com entradas sem limites de tamanho definidos (VENKATACHALAM, 2019). O funcionamento ocorre por realimentação, ou seja, os valores gerados no treinamento anterior serão reutilizados como entradas no próximo treinamento. Desta forma, a RNN não atualiza as saídas com base apenas em pesos previamente estabelecidos em treinamento, mas também faz uso de pesos e vetores ocultos com base na saída gerada anteriormente, possibilitando que em função de uma mesma entrada na RNN possa haver geração de diferentes resultados na saída. A Figura 18 ilustra uma rede neural denominada *feedforward*, enquanto a Figura 19 ilustra uma RNN.

Figura 18 – Rede neural *feedforward*



Fonte: (VENKATACHALAM, 2019)

Figura 19 – Rede neural recorrente



Fonte: (VENKATACHALAM, 2019)

A principal diferença entre ambas as representações das redes neurais, conforme demonstrado nas Figuras 18 e 19, é que a camada oculta da rede neural *feedforward* possui entradas fixas, previamente geradas durante o treinamento da rede. A representação da RNN apresenta diferentes camadas ocultas, que são geradas com base nos resultados anteriores da saída da RNN. Por isso, a RNN é um método de aprendizado de máquina que pode ser utilizado para identificar o próximo estado esperado do sistema.

Utilizou-se o modelo RNN nos dados utilizados para o experimento do CPS, previsto neste trabalho. Porém, neste experimento do CPS não foi possível realizar a avaliação dos dados através dos conceitos de prognóstico e gestão da saúde de máquina, pois não se detinha conhecimento suficiente referente ao *dataset* utilizado, pois não haviam informações referentes aos atributos disponibilizados no *dataset*. Da mesma forma, não faz parte do escopo deste trabalho a avaliação do desempenho da RNN implementada no contexto da predição. Procurou-se apenas compreender o comportamento de uma RNN no contexto de um CPS, assim como atestar a sua viabilidade. Para implementação (treinamento e uso) da RNN no contexto preditivo, faz-se necessário o uso de uma plataforma de computação em nuvem. A fim de complementar a camada de software proposta, a seção seguinte desenvolve o conceito de computação em nuvem.

O modelo RNN consiste em um sistema que possui comportamento equivalente a sistemas de controle em malha fechada, ou seja, faz utilização dos valores gerados anteriormente como realimentação da próxima entrada. Por esse motivo, a camada oculta da RNN é alterada conforme os resultados gerados na interação de uma entrada com a rede neural, assim para uma mesma entrada, podem haver diferentes resultados na saída.

### 3.4 COMPUTAÇÃO EM NUVEM

Os conceitos de computação em nuvem iniciaram nos anos 50, através de terminais de acesso era possível realizar a conexão simultânea a um único computador, utilizando o processamento desse para executar as funções necessárias. Nos anos 70 foi implementado o sistema de máquina virtual, desta forma, foi possível que diferentes sistemas operacionais funcionassem ao mesmo tempo em um único *hardware*, que seguia sendo o computador central. Já nos anos 90, as empresas de telecomunicação implementaram um sistema que permitia o acesso através de redes privadas a um computador físico, a partir de diferentes localidades (NETO, 2014).

Atualmente, empresas como a AWS, Azure, entre outras, permitem o acesso a servidores físicos localizados em diferentes regiões do mundo. Desta forma, é possível que se utilize os recursos de um computador, sem a necessidade de instalar *softwares* ou outros serviços em um computador local. O usuário necessita apenas de acesso à Internet e possuir uma conta em alguma das plataformas que disponibilizam serviços de computação em nuvem, a fim de fazer uso dos recursos disponibilizados por esta empresa. As vantagens da utilização de computação em nuvem são a eliminação da necessidade de aquisição de *hardware* local, descartando a necessidade de manutenção nesse. Também permite a fácil escalabilidade de processamento de máquina conforme necessidade do usuário (Amazon Web Services, 2019a). A computação em nuvem é dividida em três categorias, conforme ilustrado na Figura 20 e descrito a seguir:

- Infraestrutura como um serviço (IaaS) é o modelo que disponibiliza as funções básicas de infraestrutura em nuvem, ou seja, esse modelo é utilizado como meio de armazenamento de dados, disponibilizando acesso de diferentes pontos conforme a configuração realizada, é um modelo que demanda muitas vezes implementação de *hardware*.
- Plataforma como um serviço (PaaS) é o modelo que não existe a necessidade de criação ou construção de *hardware*, sendo utilizada toda a infraestrutura disponibilizada por uma plataforma como a AWS ou Azure, possibilitando a implementação de sistemas com base na Internet.
- Software como um serviço (SaaS) é o modelo que disponibiliza *softwares* específicos, como por exemplo o *e-mail*.

Figura 20 – Modelos de computação em nuvem



Fonte: (Amazon Web Services, 2019a)

No desenvolvimento desse experimento foram analisadas duas plataformas de computação em nuvem diferentes, nas quais verificou-se as tecnologias disponibilizadas, a fim de identificar uma plataforma que pudesse atender aos requisitos desse projeto. As plataformas tinham como requisito possuir os recursos descritos a seguir:

1. Serviços para implementação de sistemas de aprendizado de máquinas
2. Serviços para enviar e receber dados de equipamentos físicos
3. Integração entre serviços na mesma plataforma
4. Possibilitar a criação de um ambiente comum que reunisse a informação de diferentes sistemas físicos em um único ponto
5. Versão de teste gratuito das ferramentas disponibilizadas pela plataforma.

Com base nos requisitos pré-determinados, foram analisadas duas plataformas de computação em nuvem. As plataformas estudadas foram os serviços da *Azure* (MICROSOFT, 2019a) e a *Amazon Web Service (AWS)* (Amazon Web Services, 2019a). Após uma avaliação inicial baseada em critérios de disponibilidade e funcionalidades, os pontos da Tabela 1 foram levantados, a fim de comparar os serviços disponibilizados por cada plataforma, assim como analisar se ambas atendiam os requisitos mínimos previamente estabelecidos.



Tabela 1 – Comparação dos serviços da AWS e do Azure

Descrição	Serviço AWS	Serviço Azure
Serviço de treinamento, implementação e automação de modelos de aprendizado de máquina.	SageMaker	Azure Machine Learning
Serviço de armazenamento Big Data.	Redshift	SQL Data Warehouse
Servidores virtuais.	Instâncias de nuvem de computação elástica (EC2)	Máquinas Virtuais do Azure
Serviço de implementação na borda através de eventos.	Lambda	Funções Azure
Gateway para comunicação bidirecional.	IoT AWS	Hub Azure IoT
Implementação local de funções através da nuvem.	IoT Greengrass	Azure IoT Edge
Desenvolvimento de gêmeos digitais.	Grafo de coisas AWS	Gêmeo digital
Azure Serviço de segurança da rede.	Gerenciamento de identidade e de acesso (IAM)	Azure Active Directory
Armazenamento de dados.	Amazon S3	Blobs Azure

Fonte: (MICROSOFT, 2019b)

Conforme pode ser identificado na Tabela 1, as plataformas possuem serviços equivalentes. Desta forma, a escolha da plataforma foi definida através da disponibilidade de uso, optando-se pela plataforma AWS, cuja versão de teste gratuito estava disponível para utilização dentro de um período de um ano.

### 3.5 PLATAFORMA PARA IMPLEMENTAÇÃO DOS CONCEITOS DE CPS

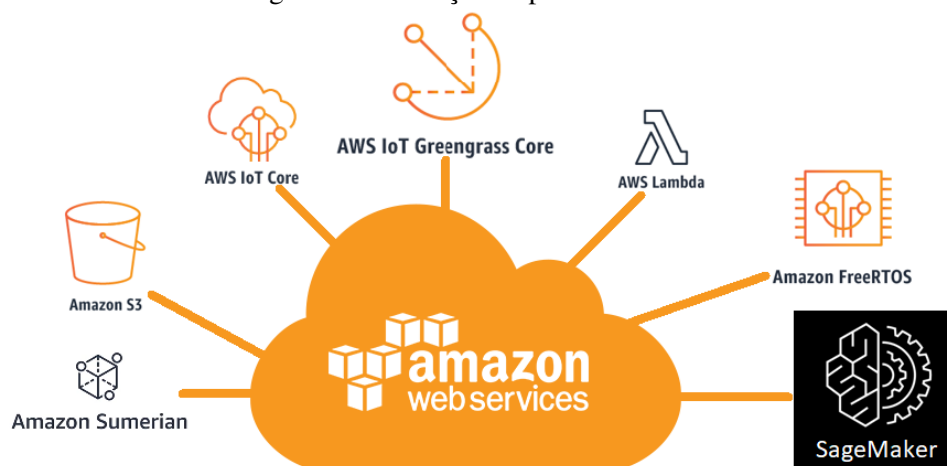
A AWS é uma plataforma em nuvem que disponibiliza diversos serviços para diferentes segmentos. Consiste em um serviço com os conceitos de computação em nuvem, ou seja, permite a entrega de processamento, armazenamento, recursos de tecnologia da informação, entre outros, conforme necessidade ou demanda de um cliente, sem a necessidade de investimentos em *hardware* local (Amazon Web Services, 2019a).

A plataforma AWS é composta de diferentes serviços, muitos dos quais podem ser utilizados para implementações de Internet das coisas (em inglês *Internet of Things*, sigla IoT), aprendizado de máquina (em inglês *Machine Learning*, sigla ML), inteligência artificial (IA), banco de dados, entre outros (Amazon Web Services, 2019a). A Figura 21 ilustra alguns dos serviços disponibilizados pela AWS, sendo descritos a seguir:

- Amazon Sumerian, serviço de ambientes 3D

- Amazon S3, serviço para armazenamento
- AWS IoT Core, serviço de comunicação de dispositivos
- AWS Lambda, serviço para criação de funções
- AWS IoT Greengrass Core, serviço que permite a execução de funções AWS Lambda
- Amazon FreeRTOS, serviço de código aberto para microcontroladores
- Amazon SageMaker, serviço para treinamento e implementação para aprendizados de máquina

Figura 21 – Serviços da plataforma AWS



Fonte: (Amazon Web Services, 2019a)

O serviço *Amazon Sumerian* permite a criação de ambientes 3D e, conforme descrito pela AWS, pode ser utilizado na implementação de sistemas gêmeos digitais. O conceito de funcionamento do *Amazon Sumerian* é através de objetos 3D dispostos em um ambiente virtual e localizados na nuvem. A interação entre objetos é realizada utilizando *JavaScript*, demandando um conhecimento maior nesta linguagem de programação, a fim de realizar a correta interação de objetos. A integração com dispositivos em campo pode ser realizada utilizando o *AWS IoT Core*, desta forma são adquiridas informações do meio físico e posteriormente atualizado o ambiente virtual, ou vice-versa. Todo o desenvolvimento fica armazenado em nuvem, ou seja, pode ser acessado por um *link* em diferentes navegadores da Internet, indiferente da localização das pessoas no mundo.

O serviço Amazon S3 consiste em um banco de dados, permitindo o armazenamento de diferentes informações ou objetos criados neste ambiente em nuvem. Através desse serviço é possível armazenar as informações e posteriormente utilizar entre os demais serviços disponibilizados pela AWS. Um dos objetivos desse serviço é a fácil usabilidade, assim como permite a escalabilidade do serviço, ou seja, os custos aumentam conforme cresce o número de acessos

e o tamanho do espaço utilizado, sem a necessidade de pré contratar o espaço que pode ser necessário, antes de fazer uso desse.

O *AWS IoT Core* consiste em um serviço para troca de informações através do protocolo MQTT, funcionando como um *gateway*, ou seja, concentra as informações de diferentes locais em um único ponto. Desta forma, é possível coletar informações de diferentes dispositivos e utilizar nos diferentes serviços da AWS, o *AWS IoT Core* também permite alterar o estado dos dispositivos físicos. A AWS apresenta esse serviço como forma de comunicação entre meios e dispositivos diferentes, ou seja, um dispositivo pode trocar informações com os serviços na nuvem, assim como outros equipamentos físicos conectados no mesmo *gateway*.

O serviço de *AWS Lambda* traz a funcionalidade de execução dos códigos baseados em eventos. A principal funcionalidade destacada pela AWS é não ter a necessidade de automatizar as solicitações de acesso, desta forma os dispositivos conectados, que atendem as regras de segurança, podem solicitar que a função seja executada.

O *AWS IoT Greengrass Core* funciona dentro dos serviços do *AWS IoT Core*. Este serviço permite que funções do *AWS Lambda* sejam executadas em dispositivos locais, assim como permite a criação de grupos para melhor separação de funções e permitir que em um mesmo grupo, os dispositivos possam fazer troca de dados entre si. Também é possível a criação de dispositivos sombra, ou seja, cria uma cópia do estado atual do dispositivo e mantém a versão em caso de falha de conexão à Internet, quando é reestabelecida a conexão com a Internet, esse serviço sincroniza as informações com o dispositivo. O *AWS IoT Greengrass Core* permite também a utilização das técnicas de OPC UA para aplicações industriais que façam necessário o uso de tal protocolo.

O Amazon *FreeRTOS* é uma biblioteca para microcontroladores. Sua utilização permite a comunicação com serviços da AWS, facilitando a conexão aos serviços do *AWS IoT Core* ou *AWS IoT Greengrass Core*. A biblioteca possui as funções de segurança para sua utilização pré-configuradas, desta forma a implementação pode se tornar mais simples.

O último serviço que foi estudado da AWS é o *SageMaker*, esse, por sua vez, permite o treinamento e implementação dos conceitos de aprendizagem de máquina. O mesmo executa os códigos em *Python* (Python Software Foundation, 2019) no ambiente *Jupyter* (Project Jupyter, 2019). Através dessa plataforma, pode-se utilizar as bibliotecas de aprendizado de máquina próprias da Amazon, assim como bibliotecas de terceiros que permitem a utilização da linguagem de programação em *Python*.

Para este experimento os serviços a seguir foram utilizados:

- Amazon S3, serviço que facilita o uso do *dataset* no *SageMaker*
- *SageMaker* foi utilizado para o treinamento da rede neural com base nos conceitos de RNN. A rede neural treinada foi utilizada na implementação da camada de conversão a fim de simular a predição das situações de máquina e produção

- *AWS IoT Core* é utilizado para visualização do funcionamento do sistema de comunicação com o dispositivo físico
- *AWS IoT Greengrass Core* é o serviço utilizado para a criação da camada ciber da arquitetura 5C, ou seja, foi utilizado para agrupar as informações dos dois dispositivos utilizados como simulador de uma máquina e de produção.
- *AWS Lambda* é utilizado para a criação das rotinas de execução na segunda camada, permitindo que os dispositivos enviem as informações para a *AWS IoT Core*

### 3.6 COMUNICAÇÃO NA CAMADA DE CONEXÃO

Diferentes protocolos para transmissão de dados foram desenvolvidos ao longo dos anos. Nessa etapa são apresentados dois protocolos, que foram considerados para a implementação da comunicação entre camada de conversão e ciber. São eles o *open platform communications unified architecture* (OPC UA) e *message queuing telemetry transport* (MQTT), sendo que apenas um deles foi utilizado na implementação da camada de conexão.

Os protocolos foram selecionados com base em artigos apresentados e o utilizado para o experimento foi o protocolo MQTT. A escolha desse protocolo foi baseada na maior velocidade de transmissão de dados quando comparado ao OPC UA, porém o protocolo OPC UA não pode ser ignorado para outras aplicações ou testes, pelo fato de possuir inúmeros outros serviços que vão além da troca de dados (ROCHA et al., 2018).

#### 3.6.1 MQTT

O protocolo MQTT surgiu no final dos anos 90, sendo desenvolvido pela *International Business Machines* (IBM). A aplicação original do protocolo era unir sensores nas áreas petrolíferas e de satélites, em 2014 se tornou um protocolo aberto, quando começou a ser popularizado para aplicações nas diferentes áreas da tecnologia (YUAN, 2017).

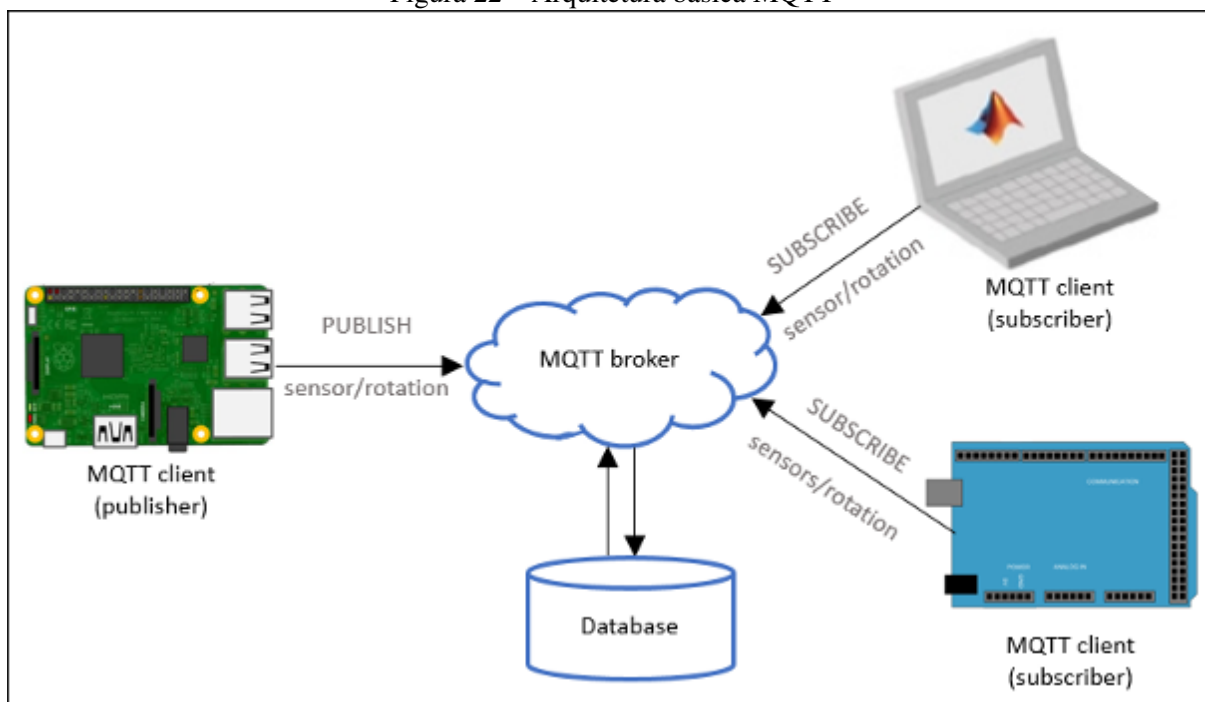
O protocolo MQTT pode ser aplicado em residências, indústrias, áreas da saúde e celulares, devido ao baixo consumo de energia e latência de conexão. Funciona por método de publicação e assinatura, por este motivo, a leitura não ocorre quando o cliente deseja, mas quando existe uma publicação no tópico que foi registrado (ROCHA et al., 2018). Outra vantagem apresentada, é que não necessita plena compreensão do protocolo para sua implementação (YUAN, 2017).

O sistema consiste em um parâmetro denominado *QoS*, que é responsável pela qualidade do serviço. Os níveis iniciam em zero (*QoS 0*), em que não existe garantias de que a mensagem publicada chegou ao destino. A qualidade um (*QoS 1*) possui, após publicação da mensagem, uma confirmação de que essa foi reconhecida. O terceiro nível (*QoS 2*) possui um total de quatro mensagens trocadas, sendo que as duas primeiras possuem função parecida ao

*QoS 1*, e as demais consistem na liberação pelo sistema que enviou e confirmação de que a publicação foi completa (ROCHA et al., 2018).

A arquitetura consiste em assinantes (*subscriber*), publicadores (*publisher*) e o *broker*, conforme demonstrado na Figura 22.

Figura 22 – Arquitetura básica MQTT



Fonte: (MATHWORKS, 2019)

### 3.6.2 OPC UA

O termo OPC surgiu em 1995, através de uma força tarefa entre alguns fabricantes de tecnologia. Em 1996 a Fundação OPC (*OPC Foundation*) surge com objetivo de normatizar este protocolo. Atualmente, a última versão liberada é o OPC UA, sendo essa uma evolução do OPC (OPC Foundation, 2019b).

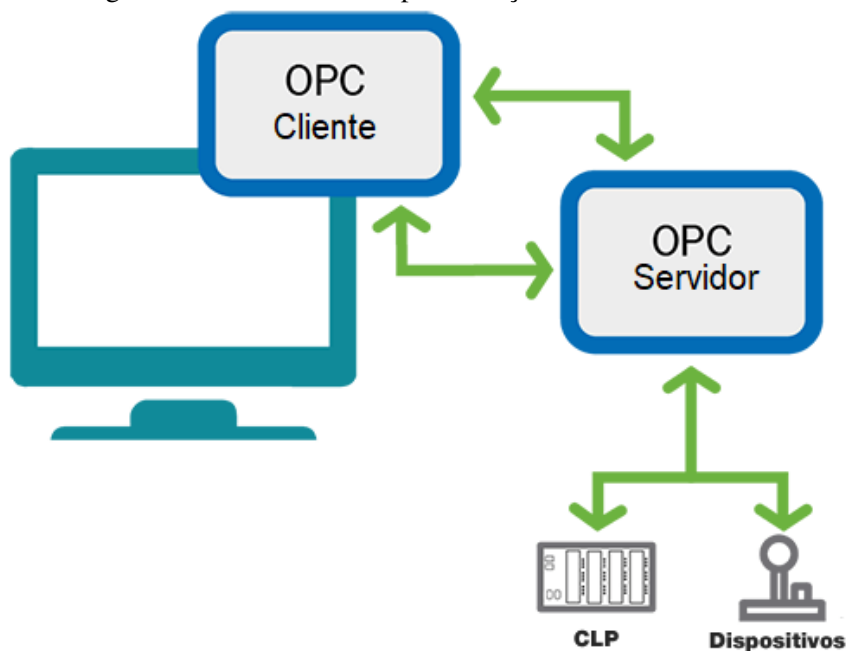
O protocolo OPC UA tem como principal função conectar diferentes equipamentos que trabalham em diversos protocolos de rede, além de permitir que essa conexão venha a ocorrer nas diferentes camadas de um sistema de automação, ou arquitetura de alto nível na Internet (ROCHA et al., 2018).

Diferente do MQTT, o OPC UA possui diversas funções que vão muito além da troca de dados. Pode ser utilizado para o modelamento de dados, gerenciamento de alarmes e eventos, histórico das variáveis, controle de acesso, entre outros (ROCHA et al., 2018).

OPC UA funciona através de um serviço cliente servidor, em que o servidor disponibiliza acesso aos dados e funções. A conexão é realizada pelo cliente, que tem como responsabilidade tornar a conexão segura. As informações são tratadas como nós (*node*) de diferentes tipos,

sendo definidos conforme necessidade de uso. Tradicionalmente, a implementação do OPC é realizada através de um mapa de variáveis, facilitando a aquisição dos dados desejados (GRÜNER; PFROMMER; PALM, 2016). A Figura 23 ilustra o conceito básico do funcionamento do protocolo OPC UA.

Figura 23 – Conceito de implementação básica do OPC UA.



Fonte: (ROMANO, 2019)

Conforme apresentado por Lehnhoff et al. (2011), o OPC UA tem como base os princípios a seguir:

- Uso de técnicas de orientação a objetos
- A informação pode ser acessada da mesma forma que as instâncias
- Informação pode ser conectada de diferentes formas, suportando diferentes camadas de hierarquia
- Extensibilidade de vários modos em relação ao modelamento de informações
- Servidores OPC UA podem expor um alvo de forma nativa, ou seja, os servidores não necessitam modelar novamente as informações
- O modelamento das informações sempre se faz pelo servidor

### 3.7 IMPLEMENTAÇÃO DO EXPERIMENTO UTILIZANDO A AWS

A fim de desenvolver as três primeiras camadas da arquitetura 5C, conexão, conversão e ciber, propostas na seção 3.1, foi analisada a necessidade de implementação do sistema CPS.

A elaboração desse experimento foi seccionada em quatro etapas de desenvolvimento, a fim de permitir melhor compreensão das funções executadas. As etapas de implementação são apresentadas a seguir:

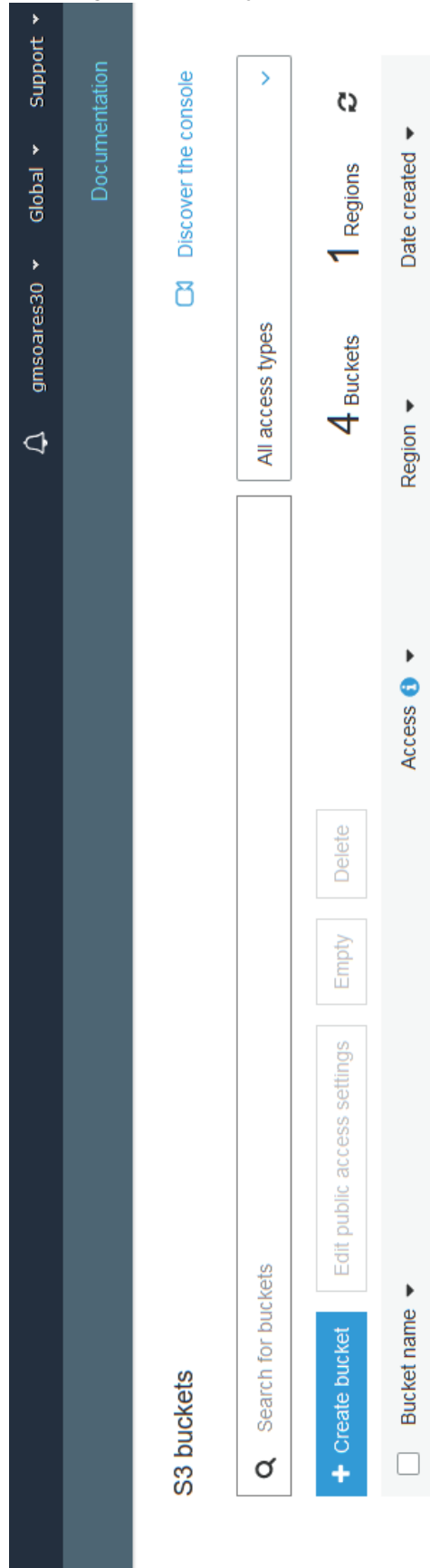
1. Carga dos dados de treinamento para o Amazon S3
2. Configuração e treinamento dos dados no *SageMaker*
3. Implementação da camada de conversão
4. Configuração do *AWS Lambda* e integração com o *AWS IoT Greengrass Core*

### **3.7.1 Carga de dados no Amazon S3**

O Amazon S3 é um sistema de armazenado de dados, que facilita o desenvolvimento de soluções, utilizando o *SageMaker*. O Amazon S3 permite a criação de *buckets* que servem como meio de carregar diferentes tipos de dados, para posteriormente utilizá-los nos diferentes serviços da AWS.

Para o experimento do CPS, foi utilizado o serviço Amazon S3 para importar os dados de treinamento, a fim de utiliza-los no treinamento da RNN com a ferramenta do *SageMaker*. Para isto, se faz necessário a criação de um *bucket* dentro do ambiente de gerenciamento do Amazon S3. A Figura 24 ilustra o ambiente de criação dos *bucket*.

Figura 24 – Criação do *bucket*



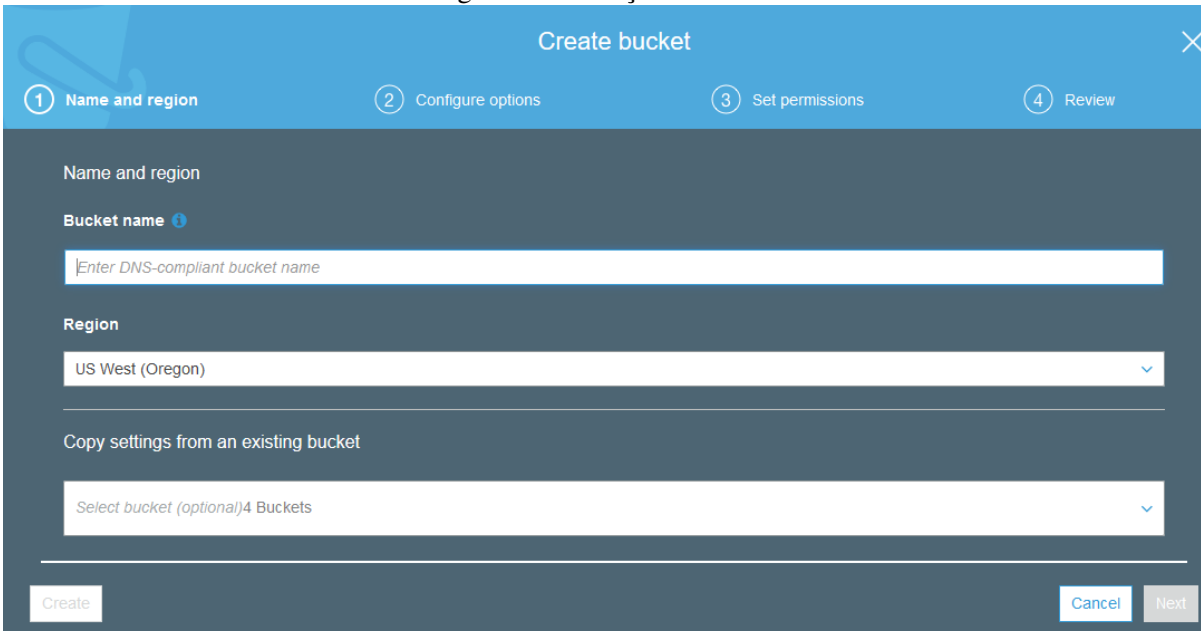
Fonte: O Autor



Durante a criação do *bucket*, na aba de edição que se abre, ilustrada na Figura 25, os pontos descritos a seguir foram levados em consideração, a fim de evitar possíveis erros:

- O nome não pode conter letras maiúsculas ou caracteres especiais, com exceção do hífen
- Para uma implementação utilizando usando o *SageMaker*, o nome do *bucket* foi composto de "*sagemaker-<nome de escolha do usuário>*", desta forma permite que o *SageMaker* tenha acesso mais fácil aos dados armazenados
- A seleção da região foi definida com cuidado, pois a região deve ser igual a dos demais serviços que serão utilizados, desta forma é necessário que o usuário tenha conhecimento prévio sobre quais funções serão usadas e analisar a disponibilidade na região a ser implementada

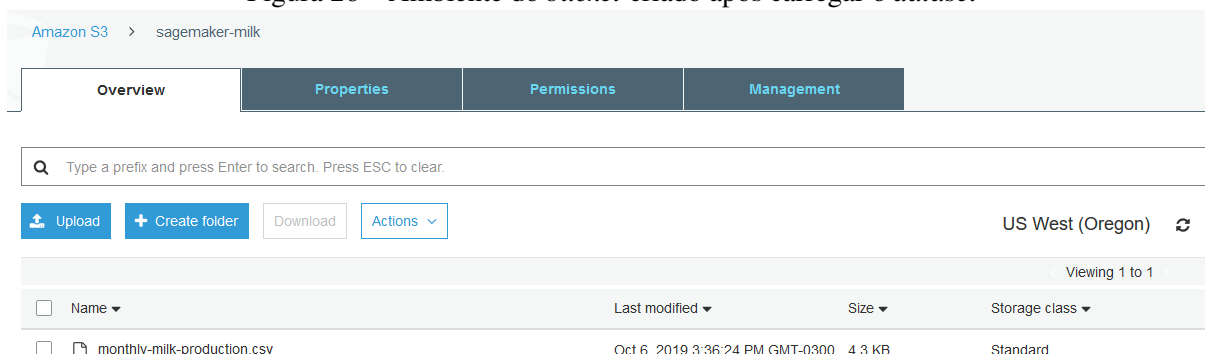
Figura 25 – Criação do *bucket*



The screenshot shows the 'Create bucket' wizard in the AWS console. The title bar is blue and contains the text 'Create bucket' and a close button. Below the title bar, there are four numbered steps: 1. Name and region, 2. Configure options, 3. Set permissions, and 4. Review. The first step, 'Name and region', is currently active. It contains a 'Bucket name' field with a placeholder 'Enter DNS-compliant bucket name' and a 'Region' dropdown menu set to 'US West (Oregon)'. Below these fields is a section for 'Copy settings from an existing bucket' with a dropdown menu showing 'Select bucket (optional) 4 Buckets'. At the bottom of the form, there are three buttons: 'Create', 'Cancel', and 'Next'.

Fonte: O Autor

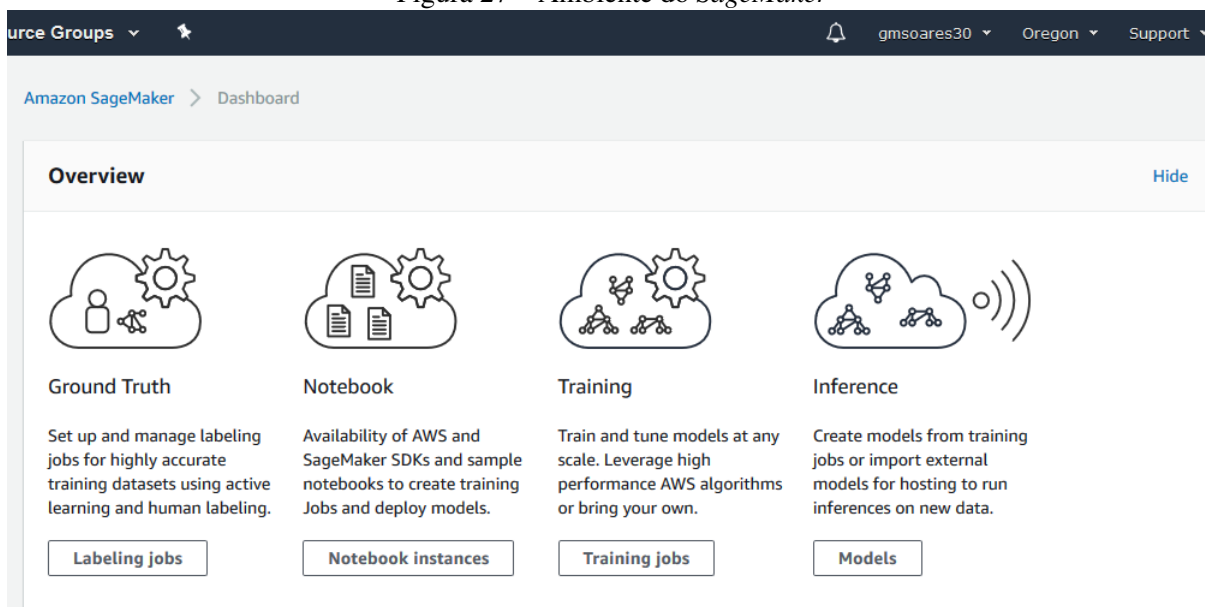
Após a criação do *bucket*, deve ser carregado o *dataset*. A Figura 26 ilustra como deve ficar o ambiente do *bucket* após a criação e o carregamento do *dataset*. Após as etapas descritas, é possível iniciar a configuração do serviço no *SageMaker*, a fim de possibilitar o uso de dados no treinamento da RNN.

Figura 26 – Ambiente do *bucket* criado após carregar o *dataset*

Fonte: O Autor

### 3.7.2 Configuração e treinamento no *SageMaker*

O serviço do *SageMaker* permite a implementação dos conceitos de aprendizado de máquina, o serviço foi utilizado para o treinamento da RNN, com base nos dados que foram selecionado para este experimento, conforme seção 3.2. As etapas para uso do *SageMaker* devem ser realizadas posteriormente à criação do *bucket* no Amazon S3, pois necessitam de configurações das regras para correto acesso do *SageMaker* ao *bucket*. A Figura 27 apresenta o ambiente inicial do *SageMaker*.

Figura 27 – Ambiente do *SageMaker*

Fonte: O Autor

A utilização do serviço do *SageMaker* iniciou-se a partir da criação de um *Notebook Instance*, que permite acesso ao servidor do *Jupyter* (Project Jupyter, 2019). Durante esta etapa, é necessário configurar as regras de acesso, onde deve ser criada uma nova política de acesso. Esta política permite o acesso ao Amazon S3, para utilizar o *dataset* que foi carregado no

ambiente, conforme descrito na sub-seção 3.7.1. Como pode ser identificado na Figura 28, a qual ilustra a criação da política de acesso, os *buckets* com o nome de "sagemaker" vêm pré-configurados durante a criação das regras de acesso, facilitando a utilização dos *datasets* carregados no banco de dados do Amazon S3.

Figura 28 – Ambiente do *bucket* criado após carregar o *dataset*

**Create an IAM role** ✕

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the role you create.

The IAM role you create will provide access to:

- S3 buckets you specify - optional
  - Specific S3 buckets
 

Example: bucket-name-1, bucket-name-2, bi

Comma delimited. ARNs, "\*" and "/" are not supported.
  - Any S3 bucket  
Allow users that have access to your notebook instance access to any bucket and its contents in your account.
  - None
- Any S3 bucket with "sagemaker" in the name
- Any S3 object with "sagemaker" in the name
- Any S3 object with the tag "sagemaker" and value "true" [See Object tagging](#)
- S3 bucket with a Bucket Policy allowing access to SageMaker [See S3 bucket policies](#)

Fonte: O Autor

Após a criação do *Notebook Instance*, é possível criar as lógicas de treinamento utilizando bibliotecas disponibilizadas pela AWS ou de terceiros, como o *TensorFlow* (TensorFlow Google Brain, 2019). A RNN foi implementada utilizando o *TensorFlow*, permitindo o uso de lógicas que possuem maior disponibilidade de documentação na Internet.

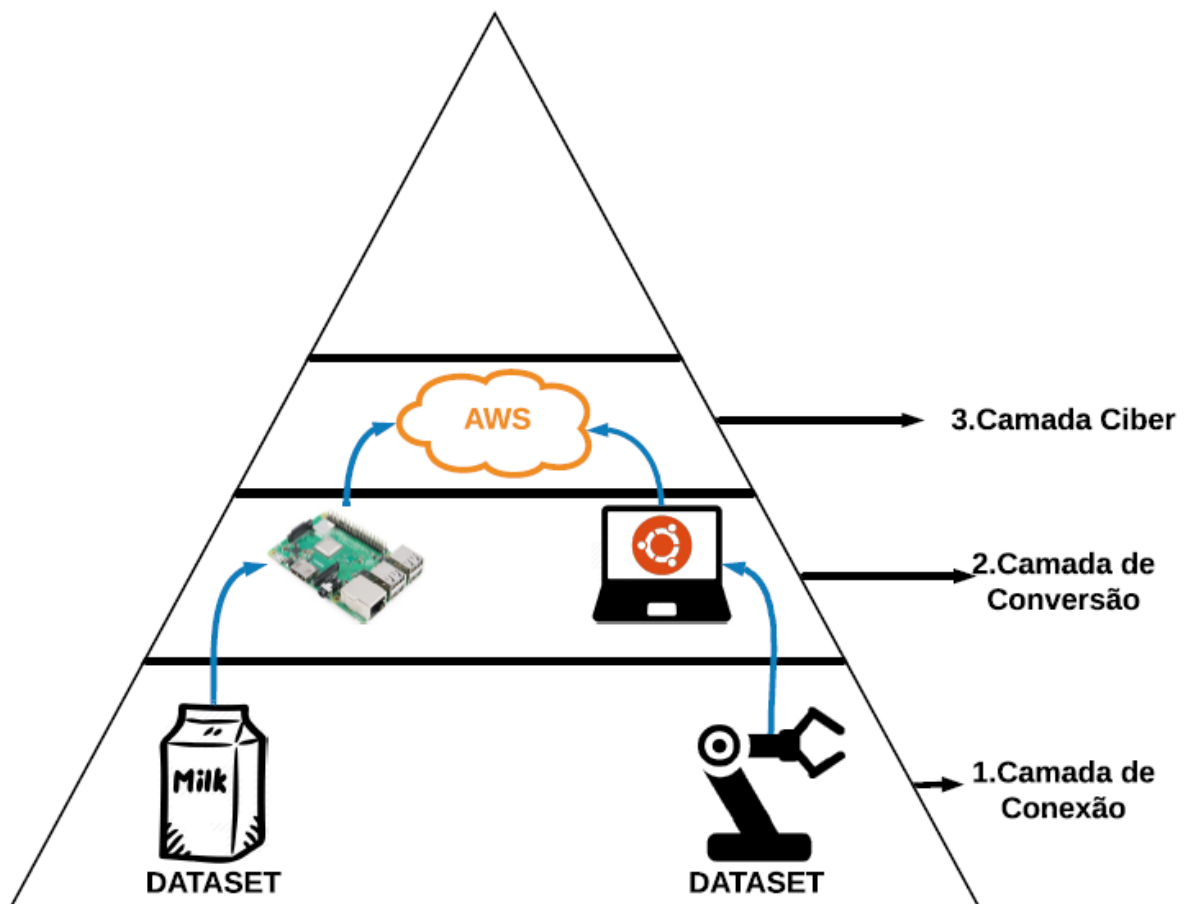
Para utilizar o *dataset* que foi carregado no Amazon S3, além da configuração anterior, foi necessário utilizar duas bibliotecas adicionais, a fim de importar o *dataset* e preparar os dados para permitir seu uso com a biblioteca *TensorFlow*. As bibliotecas necessárias foram o *boto3* (Amazon Web Services, 2019b) e o *pandas* (Core Team, 2019).

A biblioteca *boto3* permite acesso aos *buckets* do Amazon S3, criando um link de acesso que possibilita realizar o *download* ou o carregamento de informações para o *bucket*. Essa biblioteca é disponibilizada pela própria Amazon. A biblioteca *pandas* é utilizada para tratar os dados, separando-os em colunas, facilitando a implementação com o *TensorFlow*, assim como permite melhor visualização dos arquivos utilizados no treinamento.

### 3.7.3 Implementação da camada de conversão

Os dados que foram utilizados para o treinamento da RNN no *SageMaker* simulam dados que podem ser adquiridos de sensores posicionados em uma máquina ou linha de produção, porém para a implementação da segunda camada da arquitetura 5C, foi necessária a utilização de dispositivos físicos, sendo escolhidos dois dispositivos distintos, um para cada *dataset* utilizado. Desta forma foi possível enviar dados dos *datasets* para o ambiente virtual na plataforma AWS. Para implementação da segunda camada, foi utilizado um mini computador modelo *Raspberry Pi3* modelo B+ (Raspberry Pi Foundation, 2019) para simular o primeiro dispositivo. O segundo dispositivo foi desenvolvido utilizando uma máquina virtual, com o sistema operacional *Ubuntu Desktop* versão 18.04 LTS (Canonical Ltd., 2019). Após, foi implementado o protocolo de comunicação para permitir a troca de dados com a plataforma virtual, a fim de utilizar as ferramentas disponibilizadas pelo ambiente em nuvem. A arquitetura final deste experimento ficou conforme ilustrado na Figura 29.

Figura 29 – Arquitetura final do experimento

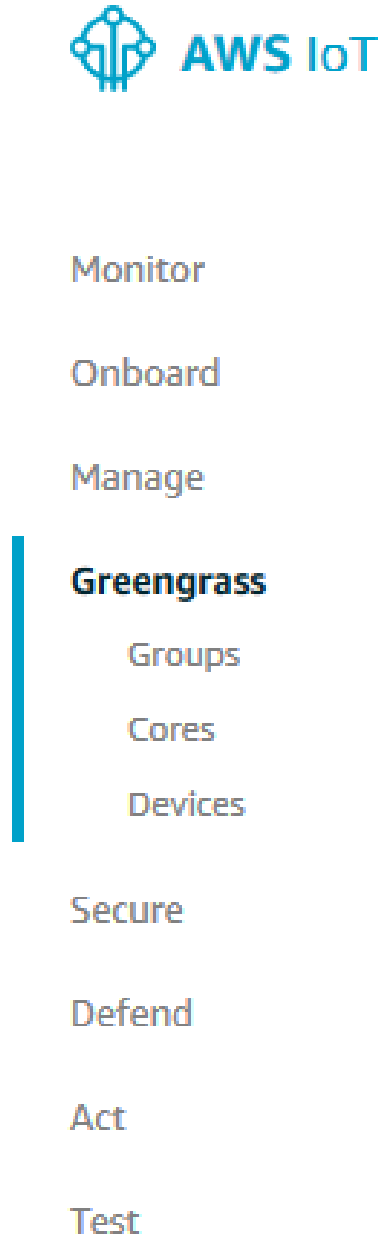


Fonte: O Autor

Para permitir a conexão dos dispositivos com o AWS, foi utilizado o serviço *AWS IoT Greengrass Core*. O serviço é integrado ao ambiente do *AWS IoT Core*, desta forma é possível

utilizar ferramentas para identificar se os dispositivos estão comunicando, assim como analisar se os dados enviados pelos dispositivos estão chegando de forma correta. A Figura 30 ilustra o local onde esta localizado o serviço do *AWS IoT Greengrass Core*.

Figura 30 – Ambiente do *AWS IoT Core*



Fonte: O Autor

A configuração do *AWS IoT Greengrass Core* começa pela criação de um grupo, nesse são definidos o nome do grupo, nome do núcleo e regras de acesso. Após, foi realizado o *download* dos certificados e chaves de acesso. Neste ponto é necessário fazer também o *download* dos arquivos para instalação do *software* do *AWS IoT Greengrass Core*<sup>3</sup>, onde foram seleciona-

<sup>3</sup>Disponível em: <<https://docs.aws.amazon.com/greengrass/latest/developerguide/what-is-gg>> Acesso em: 16, nov, 2019.

dos os *softwares* de instalação com base nos dispositivos de utilização no experimento, ou seja, para o sistema operacional *Raspian* e o *Ubuntu 18.04*. A Figura 31 ilustra o local que devem ser realizados os *downloads*. Essa etapa é realizada antes de concluir a criação do grupo.

Figura 31 – Fim da criação e local dos *downloads* para o *AWS IoT Greengrass Core*  
Download and store your Core's security resources

A certificate for this Core	2dad75c15b.cert.pem
A public key	2dad75c15b.public.key
A private key	2dad75c15b.private.key
Core-specific config file	config.json

Download these resources as a tar.gz

You also need to download a root CA for AWS IoT:

Choose a root CA ↗

Download the current Greengrass Core software

By downloading this software you agree to the [Greengrass Core Software License Agreement](#). To install Greengrass on your Core download the package and follow the [Getting Started Guide](#).

Choose your platform ↗

Fonte: O Autor

Após a criação do grupo, os passos a seguir foram realizados na respectiva ordem descrita em ambos os dispositivos, com base na documentação disponibilizada pela AWS <sup>4 5</sup>:

1. Criação do usuário e do grupo de acesso para o *AWS IoT Greengrass Core*
2. Alteração das regras de segurança nos dispositivos
3. Instalação do respectivo software do *AWS IoT Greengrass Core* para cada dispositivo
4. Instalação dos certificados e chaves do grupo em ambos os dispositivos
5. Instalação do certificado adicional *Amazon Root CA*
6. Inicialização do *software AWS IoT Greengrass Core* nos dispositivos
7. Teste para identificar se o *software* está executando corretamente

<sup>4</sup>Disponível em: <<https://docs.aws.amazon.com/greengrass/latest/developerguide/setup-filter.rpi.html>> Acesso em: 16, nov, 2019.

<sup>5</sup>Disponível em: <<https://docs.aws.amazon.com/greengrass/latest/developerguide/gg-device-start.html>> Acesso em: 16, nov, 2019.

Após a configuração dos dispositivos, para possibilitar o funcionamento em conjunto com o *AWS IoT Greengrass Core*, foi realizada a configuração do *AWS Lambda*. Dessa forma é possível que o dispositivo envie dados locais para a plataforma da AWS, ou execute diferentes funções através de desenvolvimentos realizados em *Python*.

### 3.7.4 Configuração do *AWS Lambda* e integração com o *AWS IoT Greengrass Core*

O *AWS Lambda* é um serviço disponibilizado pela AWS, que permite a criação de lógicas em *Python*. As funções criadas no *AWS Lambda* podem ser utilizadas para comunicar ou executar tarefas entre serviços da AWS ou entre dispositivos físicos. As funções criadas no *AWS Lambda* podem ser utilizadas de duas formas, execução por evento, ou por execução contínua.

Nesse experimento foi implementado o *AWS Lambda*, para enviar os dados até a terceira camada da arquitetura 5C. A criação do *AWS Lambda* iniciou-se a partir do *Create Function*, após é necessário realizar as configurações conforme ilustrado na Figura 32, sendo necessário a configuração do nome e da linguagem de programação a ser utilizada. Para esse experimento, foram utilizadas as permissões básicas, porém essas podem ser configuradas conforme as necessidades estabelecidas para um projeto utilizando os serviços do *AWS Lambda*.

Figura 32 – Configuração inicial *AWS Lambda*

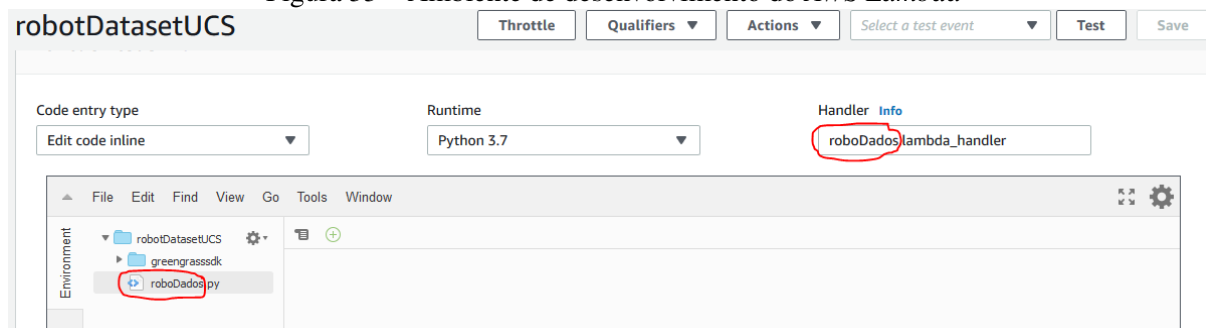
The screenshot displays the AWS Lambda 'Create function' console. At the top, it says 'Create function' with an 'Info' link. Below this, it prompts the user to 'Choose one of the following options to create your function.' There are three main options: 'Author from scratch' (selected with a radio button), 'Use a blueprint', and 'Browse serverless app repository'. The 'Author from scratch' option includes a sub-option 'Start with a simple Hello World example.' Below the options is the 'Basic information' section. It contains a 'Function name' field with the text 'exp\_ucsc\_robj' and a note: 'Use only letters, numbers, hyphens, or underscores with no spaces.' Below that is the 'Runtime' dropdown menu, currently set to 'Python 3.7'. The 'Permissions' section has a note: 'Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.' and a link 'Choose or create an execution role'. At the bottom right, there are 'Cancel' and 'Create function' buttons.

Fonte: O Autor

Após a criação da função no *AWS Lambda*, foram criadas as lógicas de programação para permitir que o dispositivo físico se comunique com o *AWS IoT Greengrass Core*, desta forma, os dispositivos puderam enviar as informações dos *datasets* para a plataforma AWS. Para realizar as lógicas, foi trabalhado dentro do ambiente de desenvolvimento das funções no *AWS Lambda*. Para o correto funcionamento da função criada é necessário manter os nomes iguais, destacados em vermelho na Figura 33, outro ponto necessário é a importação da biblioteca

*greengrassdk*<sup>6</sup>, dessa forma foi possível realizar a execução das funções do *AWS Lambda* nos dispositivos físicos.

Figura 33 – Ambiente de desenvolvimento do *AWS Lambda*



Fonte: O Autor

Após o desenvolvimento da programação, foi necessário publicar uma nova versão na aba *Actions* e após foi feita a criação de um *Alias* também na aba *Actions*, sendo este último criado com base na versão publicada no passo anterior. Foi desenvolvida a comunicação para ambos os dispositivos, sendo duas lógicas diferentes, ou seja, foram criadas duas funções no *AWS Lambda*.

Após as configurações, foi necessário publicar as funções no ambiente do *AWS IoT Greengrass Core*, para que o mesmo seja executado no dispositivo local. Para tal, foi necessário retornar à plataforma anterior, *AWS IoT Greengrass Core*, no grupo que havia sido criado na sub-seção 3.7.3. No ambiente, é necessário adicionar a função criada anteriormente no *AWS Lambda*, assim como configurar a função para que a mesma seja executada continuamente pelo *AWS IoT Greengrass Core*, para isso os passos a seguir foram efetuados:

1. Criação da função no ambiente do *AWS IoT Greengrass Core*, utilizando a função previamente criada no *AWS Lambda*
2. Edição das configurações da função, para que a mesma já execute continuamente, sem a necessidade de ativação por eventos
3. Inscrição da função no *AWS IoT Greengrass Core*
4. Implementação do serviço
5. Verificação do funcionamento na aba do serviço no *IoT Core*

### 3.7.5 Resultados obtidos

Com base nas etapas realizadas para o desenvolvimento do experimento, foi possível obter resultados referentes às implementações desenvolvidas. Os resultados apresentados foram

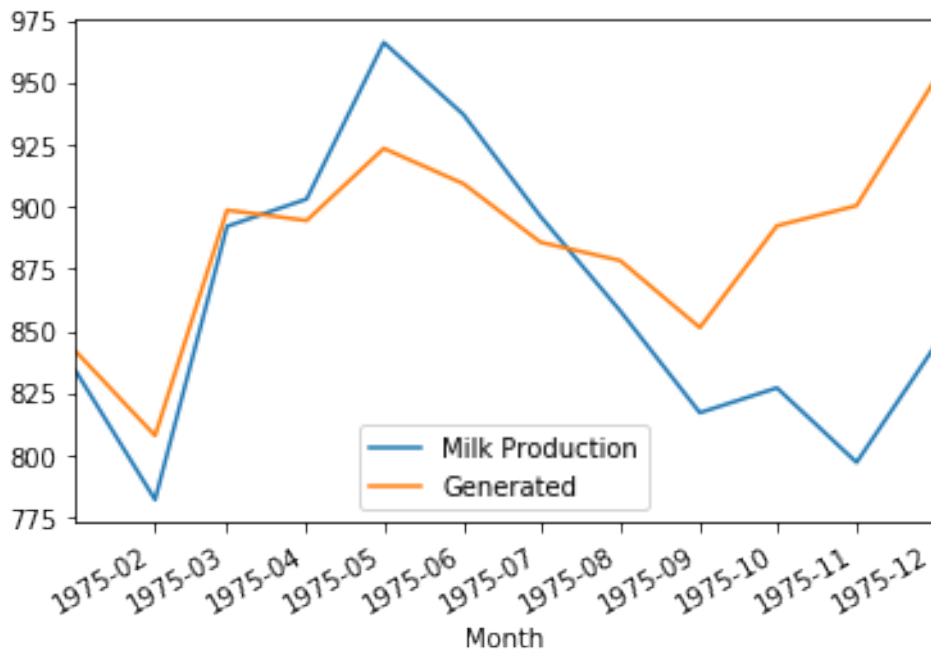
<sup>6</sup>Disponível em: <<https://pypi.org/project/greengrassdk/>> Acesso em: 17, nov, 2019.



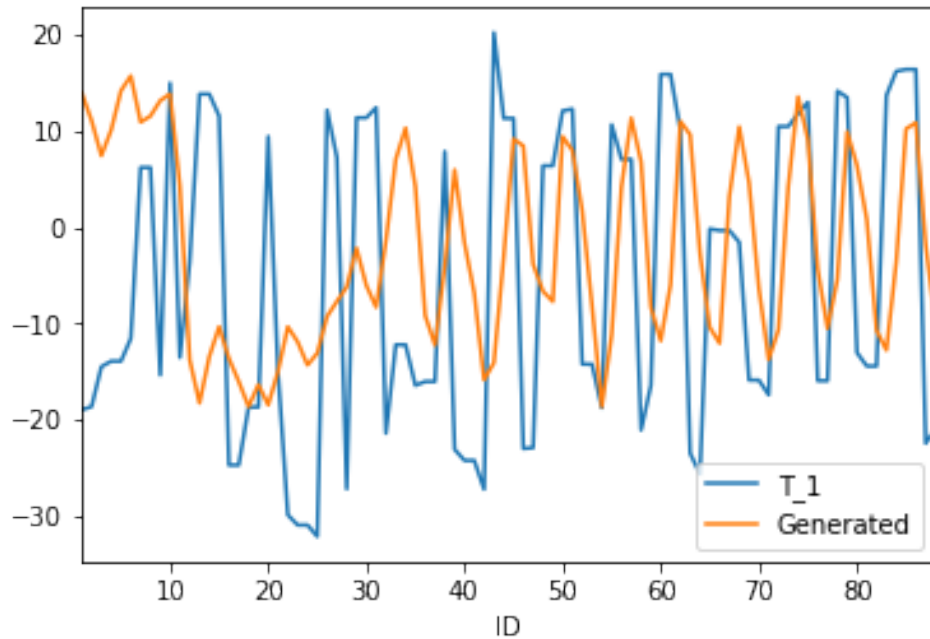
divididos em duas etapas de desenvolvimento, sendo o treinamento da RNN e a implementação dos protocolos de comunicação com o dispositivo físico.

Para a análise dos resultados da RNN obtidos dos *datasets*, foram utilizados gráficos para comparar a curva de valores previstos pela RNN e a curva de valores do *dataset*. A Figura 34 apresenta os dados do teste da RNN para o *dataset* da produção de leite e a Figura 35 apresenta os resultados do *dataset* do robô de manipulação.

Figura 34 – Gráfico de comportamento do *dataset* de produção de leite



Fonte: O Autor

Figura 35 – Gráfico de comportamento do *dataset* do robô

Fonte: O Autor

O comportamento da curva prevista (*Generated*) demonstra que ocorreu divergência de resultados quando comparado com as curvas dos dados disponibilizados no *dataset*. Esse comportamento pode ser aprimorado através do refinamento do código de treinamento, assim como testes de diferentes parâmetros para a RNN.

Devido a falta de informações dos *datasets* utilizados, não foi possível determinar a qualidade dos dados gerados, impossibilitando identificar se podem ou não ser utilizados em uma aplicação de prognóstico e gestão da saúde (PHM) de máquinas. Porém, através dos conceitos implementados, pode-se realizar a comparação da informação gerada pela RNN com o dados reais, possibilitando identificar divergências que possam indicar falhas do equipamento ou da produção.

Após avaliação da RNN, foi analisado o comportamento da comunicação entre os dispositivos e o *AWS IoT Greengrass Core*, utilizando a ferramenta de testes disponibilizada pelo *AWS IoT Core*, para identificar os dados enviados. A Figura 36 e a Figura 37 apresenta os dados da produção de leite e do robô, sendo enviados pelos dispositivos da *Raspberry Pi* e da máquina virtual com sistema operacional *Ubuntu 18.04*, respectivamente.

Figura 36 – Imagem adquirida do *AWS IoT Core* referente ao envio de dados de produção de leite

Leite/Dados	Nov 17, 2019 10:54:48 PM -0300	Export	Hide
843.0			
Leite/Dados	Nov 17, 2019 10:54:46 PM -0300	Export	Hide
797.0			
Leite/Dados	Nov 17, 2019 10:54:44 PM -0300	Export	Hide
827.0			
Leite/Dados	Nov 17, 2019 10:54:42 PM -0300	Export	Hide
817.0			

Fonte: O Autor

Figura 37 – Imagem adquirida do *AWS IoT Core* referente ao envio de dados do robô

Robo/Dados	Nov 17, 2019 10:42:10 PM -0300	Export	Hide
-15.4823			
Robo/Dados	Nov 17, 2019 10:42:08 PM -0300	Export	Hide
-17.1973			
Robo/Dados	Nov 17, 2019 10:42:06 PM -0300	Export	Hide
-16.8067			
Robo/Dados	Nov 17, 2019 10:42:04 PM -0300	Export	Hide
20.5454			

Fonte: O Autor

Conforme podem ser identificados, os dados de ambos os dispositivos são enviados para o serviço da AWS com sucesso. Os parâmetros definidos da taxa de envio é de 2 segundos, o que foi também respeitado pelo protocolo de comunicação aplicado, porém não foi possível identificar ferramentas disponibilizadas pela AWS, a fim de medir os tempos de envio de dados. Dessa forma não foi possível identificar se os requisitos de tempo real definidos pelo conceito

de CPS foram atendidos.

### 3.8 CONSIDERAÇÕES FINAIS

O capítulo apresentou as etapas realizadas dentro do experimento, assim como os conceitos que foram implementados para alcançar os objetivos estabelecidos nesse trabalho. Nesta etapa do trabalho, foi possível identificar quais serviços são disponibilizados pela AWS, sendo parte deles utilizada no desenvolvimento deste experimento. Porém os demais serviços também podem ser testados para implementações dentro da indústria, permitindo o desenvolvimento dos conceitos de gêmeo digital (DT), aplicação de microcontroladores com comunicação com o AWS, entre outros.

A comunicação com os dispositivos, assim como a implementação das ferramentas disponibilizadas pelas AWS, foram implementadas com sucesso, sendo possível realizar a comunicação da plataforma com os dispositivos selecionados para esse experimento. A utilização das ferramentas de aprendizado de máquina também foi configurada e testada com sucesso.

Para finalizar, considera-se que o sistema desenvolvido ilustra o funcionamento da plataforma AWS e demonstra seu potencial de uso no contexto da indústria 4.0. Acredita-se que os principais elementos foram construídos e que testes posteriores poderão aprimorar o modelo. Uma das tarefas que pode ser realizada em um trabalho futuro é a integração do serviço *SageMaker* com o serviço *AWS IoT Greengrass Core*. Ela possibilitará a análise mais completa do comportamento do sistema de comunicação com a nuvem.

## 4 CONCLUSÕES

Os conceitos de CPS são parte dos componentes da indústria 4.0, possibilitando a implementação de conceitos como a inteligência artificial, gêmeos digitais e prognóstico e gestão da saúde de máquina. A arquitetura 5C busca padronizar e guiar a implementação de um CPS, através da criação de uma arquitetura com etapas e funções previamente definidos. Esse trabalho, trouxe como objetivo o desenvolvimento de um experimento, a fim de demonstrar o funcionamento parcial de um CPS, utilizando a arquitetura 5C como base desse trabalho. Através dos objetivos especificados para a implementação desse experimento, foi possível atingir dentro desse trabalho os seguintes tópicos:

- A identificação dos métodos computacionais para implementação de sistemas CPS
- Identificação dos meios de comunicação existentes para transferência de dados
- Foi composto um cenário de teste utilizando dados de origem industrial, coletados da base IEEE, assim como dados de produção de leite
- Por fim, foram coletados os dados gerados no fim do experimento para demonstrar a funcionalidade do sistema

### 4.1 SÍNTESE DO TRABALHO

Atualmente, a indústria está passando por sua quarta revolução, onde objetivos foram levantados em busca do aumento da produtividade e customização de produtos. Porém, diferentes desafios vêm sendo enfrentados para atingir os objetivos estabelecidos na indústria 4.0, entre eles, a falta de padrões de testes de CPS implementados, dificuldade no desenvolvimento de *frameworks* que atendam diferentes segmentos da indústria, dificuldade de integração entre os diversos protocolos existentes no mercado, entre outros pontos.

Um dos componentes da indústria 4.0 é o CPS, sendo responsável pela comunicação do meio físico com o digital, permitindo que sinais sejam enviados para a nuvem, através de sensores localizados em diferentes locais de uma empresa, sendo que após o recebimento dos dados, o meio digital deve realizar a atualização do estado dos atuadores físicos, onde deve atender aos requisitos de comunicação em tempo real. Porém, alguns pesquisadores consideram as definições originais de CPS complexas, o que dificulta implementação de um modelo padrão de CPS.

Lee, Bagheri e Kao (2015) apresentaram a arquitetura 5C com o objetivo de padronizar implementações de CPS, assim como buscar definir um fluxo para o desenvolvimento do CPS. Esse é separado em cinco camadas distintas, a fim de permitir que seu desenvolvimento seja realizado em diferentes etapas, sendo estas: conexão, conversão, ciber, cognição e configuração.

A arquitetura 5C possibilita que conceitos como gêmeo digital sejam implementados. O gêmeo digital, originalmente, foi desenvolvido como o sistema de simulação perfeito, definição criada pela NASA. Porém sua definição foi atualizada, sendo tratado como uma cópia perfeita de um dispositivo físico, que permite a atualização de ambos os sistemas (físico e digital), na ocorrência de alterações em qualquer um dos dois. Um dos objetivos do CPS é permitir a simulação de alterações dentro da fábrica ou máquina antes de fazer a sua implementação, ao mesmo tempo, quando as alterações fossem aplicadas, não seriam necessárias grandes intervenções no meio físico, pois esse seria atualizado através do gêmeo digital.

Outro ponto que é levantado para a utilização do CPS, é a possibilidade do desenvolvimento do conceito de prognóstico e gestão da saúde de máquinas. Tal conceito levanta a possibilidade do equipamento indicar possíveis falhas, antes que estas ocorram, permitindo a intervenção na máquina para realizar os reparos necessários de forma planejada, evitando parada de máquina indesejada. O prognóstico e gestão da saúde de máquinas permite prolongar o tempo de vida útil de um equipamento, pois pode estender a utilização dos componentes da máquina.

Para a implementação dos conceitos de CPS, protocolos e serviços vêm sendo desenvolvidos, buscando facilitar o desenvolvimento do CPS. Os protocolos MQTT e OPC UA são alguns exemplos, atualmente utilizados na indústria, para permitir a comunicação de dispositivos físicos com a nuvem, assim como facilitam a integração de microcontroladores, controladores lógicos programáveis, entre outros equipamentos de tecnologia, que até então não detinham comunicação entre si.

Além dos protocolos de comunicação, serviços de computação em nuvem podem ser utilizados como possível redução de custos na fase inicial da implementação de um CPS. Empresas como AWS e Azure disponibilizam serviços que permitem a comunicação através dos protocolos MQTT e OPC UA. Os serviços de computação em nuvem permitem a implementação dos conceitos de aprendizado de máquina, podendo ser utilizados na implementação de prognóstico e gestão da saúde de máquina, através de treinamentos utilizando dados coletados dos sistemas físicos.

Com base nas definições de CPS, computação em nuvem e aprendizado de máquina, foi desenvolvido um experimento para implementação das três camadas da arquitetura 5C, esse consiste na utilizando da plataforma AWS, a fim de demonstrar o funcionamento de um CPS. Além da implementação da arquitetura 5C, foi implementada uma RNN para previsão dos valores futuros com base nos dados de *datasets* coletados na IEEE e na *WorldPress*.

Os serviços utilizados da AWS foram os seguintes:

- *SageMaker*, para o treinamento da rede neural
- Amazon S3, para o carregando dos *datasets* utilizados no treinamento da rede neural
- *AWS IoT Core*, para visualização da comunicação entre dispositivo físico e serviço da AWS

- *AWS IoT Greengrass Core* para configuração da comunicação do dispositivo físico e serviço *AWS IoT Core*, assim como para implementação das funções do *AWS Lambda*
- *AWS Lambda* para enviar os dados do dispositivo local para o ambiente da plataforma AWS

## 4.2 CONTRIBUIÇÕES DO TRABALHO

Este trabalho apresentou os conceitos de CPS, componente integrante da indústria 4.0, trazendo as funcionalidades do sistema. Foi apresentada a arquitetura 5C, a fim de criar um padrão no desenvolvimento de CPS, assim como funcionar como um guia básico para implementação dos conceitos dessa arquitetura.

O experimento buscou demonstrar as funcionalidades básicas de um CPS, assim como apresentar os conceitos de aprendizado de máquina, utilizando sistema de predição com as redes neurais recorrentes (RNN). A plataforma AWS foi utilizada no desenvolvimento de todo o experimento, sendo desenvolvido o protocolo de comunicação para permitir a troca de dados entre dispositivos físicos e serviços de computação em nuvem. Os serviços da AWS apresentados podem ser utilizados para outras aplicações, que necessitam de comunicação entre equipamentos físicos com a nuvem ou para o treinamento de redes neurais.

Por se tratar do início da indústria 4.0, os artigos desenvolvidos para aplicação dentro da indústria, são muitas vezes genéricos, não indicando exatamente como devem ser utilizados os conceitos de CPS, DT, PHM e da indústria 4.0. Apesar de ser um dos assuntos mais abordados por fabricantes de tecnologia, dados referentes a máquinas industriais ou de produção de uma empresa dificilmente são disponibilizados por eles ou encontrados na Internet, dificultando o desenvolvimento de experimentos ou até mesmo implementações reais no chão de fábrica.

Os dados encontrados referentes ao robô também possuem documentação precária, o que impossibilita inúmeras análises, como dos métodos utilizados para aquisição desses dados, ou inclusive o significado de cada atributo disponibilizado no *dataset*. Desta forma, não foi possível definir se os resultados gerados podem ou não ser utilizados em uma implementação de prognóstico e gestão da saúde de máquinas.

Os serviços da AWS possuem documentos que auxiliam a utilização dos recursos mais básicos de camada serviço disponibilizado, porém não possuem documentação mais técnica ou agrupada em um local único, impossibilitando maior compreensão durante a utilização das ferramentas, trazendo muitas vezes exemplos que não demonstram total capacidade do serviço. Para a implementação do experimento, foi necessário realizar diferentes testes para uma mesma função, para que esta funcionasse conforme esperado, obrigando a realização da implementação de várias pequenas etapas antes de chegar no objetivo final.

Apesar dos conceitos de indústria 4.0 serem recentes, foi possível identificar a necessidade de conhecimentos multidisciplinares, para a total implementação dos conceitos que englobam a indústria 4.0. A multidisciplinaridade de conhecimentos destaca a necessidade de uma maior integração das áreas de engenharia, computação e tecnologia da informação, a fim

de permitir que os conhecimentos de processo de produção, desenvolvimento das regras de segurança da Internet, coleta e automação dos sistemas físicos, desenvolvimento das lógicas de aprendizado de máquina e inteligência artificial, possam funcionar em conjunto.

### 4.3 TRABALHOS FUTUROS

O experimento do CPS contemplou o desenvolvimento das três primeiras camadas da arquitetura 5C, descritas a seguir:

- A primeira camada simulou a aquisição de dados, utilizando *datasets* disponibilizados pelas plataformas IEEE e *WordPress*
- A segunda camada foi desenvolvida utilizando o sistema operacional *Ubuntu 18.04* e o mini computador *Raspberry Pi*, assim como foi implementada uma RNN para predição de valores com base nos *datasets*
- A terceira camada foi desenvolvida utilizando os conceitos de computação em nuvem, para a coleta de informações dos dispositivos físicos

Após a conclusão do experimento e seus respectivos testes de funcionamento, foram identificados possíveis melhorias na metodologia implementada, assim como trabalhos que podem ser realizados posteriormente, sendo descritos a seguir:

- Integração entre os conceitos de aprendizado de máquina com a camada de conversão
- Implementação da análise dos conceitos de prognóstico e gestão da saúde de máquina
- Desenvolvimento do gêmeo digital
- Implementação das camadas de cognição e de configuração
- Estudo para utilização de microcontroladores na camada de conversão da arquitetura 5C
- Implementação das bibliotecas disponibilizadas pela AWS para desenvolvimento dos conceitos de aprendizado de máquina
- Implementação com uso de equipamento físico para o desenvolvimento da camada de conexão



## REFERÊNCIAS

- ABB. **Irb 1200**. Acesso em: 12 November 2019, <https://new.abb.com/products/robotics/industrial-robots/irb-1200>.
- ALAM, K. M.; EL SADDIK, A. C2ps: a digital twin architecture reference model for the cloud-based cyber-physical systems. **IEEE Access**, [S.l.], v. 5, p. 2050–2062, 2017.
- Amazon Web Services. **Amazon web services**. Acesso em: 13 June 2019, <https://aws.amazon.com/pt/>.
- Amazon Web Services. **Boto 3 documentation**. Acesso em: 16 November 2019, <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>.
- BRUCKNER, D. et al. An introduction to opc ua tsn for industrial communication systems. **Proceedings of the IEEE**, [S.l.], p. 1–11, 2019.
- Canonical Ltd. **Ubuntu**. Acesso em: 06 November 2019, <https://ubuntu.com/>.
- COGLIATI, D. et al. Intelligent cyber-physical systems for industry 4.0. In: FIRST INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE FOR INDUSTRIES (AI4I), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 19–22.
- Core Team. **Python data analysis library**. Acesso em: 16 November 2019, <https://pandas.pydata.org/>.
- Geschäftsstelle Plattform Industrie. **Industrie 4.0 – what is it?** Acesso em: 12 June 2019, <https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html>.
- GRÜNER, S.; PFROMMER, J.; PALM, F. Restful industrial communication with opc ua. **IEEE Transactions on Industrial Informatics**, [S.l.], v. 12, n. 5, p. 1832–1841, Oct 2016.
- HAAG, S.; ANDERL, R. Digital twin–proof of concept. **Manufacturing Letters**, [S.l.], v. 15, p. 64–66, 2018.
- HENNING, K. Recommendations for implementing the strategic initiative industrie 4.0. , [S.l.], 2013.
- JAMALUDIN, J.; ROHANI, J. M. Cyber-physical system (cps): state of the art. In: INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRONIC AND ELECTRICAL ENGINEERING (ICE CUBE), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–5.
- LEE, J.; BAGHERI, B.; KAO, H.-A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. **Manufacturing Letters**, [S.l.], v. 3, p. 18 – 23, 2015.
- LEHNHOFF, S. et al. Iec 61850 based opc ua communication-the future of smart grid automation. In: POWER SYSTEMS COMPUTATION CONFERENCE (PSCC'11), 17., 2011. **Anais...** [S.l.: s.n.], 2011.
- Library at AutomationDirect. **History of the plc**. Acesso em: 12 June 2019, <https://library.automationdirect.com/history-of-the-plc/>.

Maplesoft, a division of Waterloo Maple Inc. **Gêmeo digital para comissionamento virtual**. Acesso em: 03 June 2019, <https://www.maplesoft.com/solutions/engineering/AppAreas/Virtual-Commissioning.aspx>.

MATHWORKS. **Publish mqtt messages and subscribe to message topics**. Acesso em: 05 June 2019, <https://www.mathworks.com/help/supportpkg/raspberrypi/ref/publish-and-subscribe-to-mqtt-messages.html>.

MCMULLEN, J. **Software de simulação de linhas de produção**. Acesso em: 13 May 2019, <https://blog.se.com/industrial-software/2017/02/06/first-born-digital-twins-simsci-process-engineering/>.

MICROSOFT. **Microsoft azure**. Acesso em: 13 June 2019, <https://azure.microsoft.com/pt-br/>.

MICROSOFT. **Azure para profissionais do aws**. Acesso em: 06 November 2019, <https://docs.microsoft.com/pt-br/azure/architecture/aws-professional/>.

MQTT. **Mqtt**. Acesso em: 13 June 2019, <http://mqtt.org/>.

MTConnect Institute. **Mtconnect**. Acesso em: 13 June 2019, <https://www.mtconnect.org/>.

NASCIMENTO MARQUES JUNIOR, M. R. et al. Embedded agent based on cyber physical systems: architecture, hardware definition and application in industry 4.0 context. In: 17TH , 2018. **Anais...** [S.l.: s.n.], 2018. p. 584–591.

NEGRI, E.; FUMAGALLI, L.; MACCHI, M. A review of the roles of digital twin in cps-based production systems. **Procedia Manufacturing**, [S.l.], v. 11, p. 939–948, 2017.

NETO, M. D. **A brief history of cloud computing**. Acesso em: 14 November 2019, <https://www.ibm.com/blogs/cloud-computing/2014/03/18/a-brief-history-of-cloud-computing-3/>.

NUÑEZ, D. L.; BORSATO, M. Ontoprog: an ontology-based model for implementing prognostics health management in mechanical machines. **Advanced Engineering Informatics**, [S.l.], v. 38, p. 746–759, 2018.

OPC Foundation. **Opc foundation**. Acesso em: 13 June 2019, <https://opcfoundation.org/>.

OPC Foundation. **Publish mqtt messages and subscribe to message topics**. Acesso em: 05 June 2019, <https://opcfoundation.org/about/opc-foundation/history/>.

Project Jupyter. **Project jupyter**. Acesso em: 16 November 2019, <https://jupyter.org/>.

Python Software Foundation. **Python**. Acesso em: 16 November 2019, <https://www.python.org/>.

Raspberry Pi Foundation. **Raspberry pi 3 model b+**. Acesso em: 13 June 2019, <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.

ROCHA, M. S. et al. Performance comparison between opc ua and mqtt for data exchange. In: WORKSHOP ON METROLOGY FOR INDUSTRY 4.0 AND IOT, 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 175–179.

ROMANO, M. **opc-ua-client-server**. Acesso em: 13 June 2019, <https://www.logiquesistemas.com.br/blog/opc-ua/opc-ua-client-server/>.

SENTRYO. **The 4 industrial revolutions**. Acesso em: 10 July 2019, <https://www.sentryo.net/the-4-industrial-revolutions/>.

SIEMENS. **Software nx**. Acesso em: 13 May 2019, <https://www.plm.automation.siemens.com/global/pt/products/nx/>.

TAO, F. et al. Digital twin in industry: state-of-the-art. **IEEE Transactions on Industrial Informatics**, [S.l.], v. 15, n. 4, p. 2405–2415, April 2019.

TAO, F.; ZHANG, M. Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. **Ieee Access**, [S.l.], v. 5, p. 20418–20427, 2017.

TensorFlow Google Brain. **Tensorflow**. Acesso em: 16 November 2019, <https://www.tensorflow.org/>.

Ubivis - Internet of Smart Machines. **Ubivis - internet of smart machines**. Acesso em: 11 June 2019, <https://www.ubivis.io/>.

VENKATACHALAM, M. **Recurrent neural networks**. Acesso em: 13 November 2019, <https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce>.

YOGANANDAN, S. **Time series analysis of monthly milk production**. Acesso em: 13 November 2019, <https://subhayo.wordpress.com/2017/09/12/time-series-analysis-of-monthly-milk-production/>.

YUAN, M. **Conhecendo o mqtt**. Acesso em: 13 June 2019, <https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>.

ZHOU, K.; LIU, T.; ZHOU, L. Industry 4.0: towards future industrial opportunities and challenges. In: INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (FSKD), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 2147–2152.

