

**UNIVERSIDADE FEDERAL DO PARANÁ – UFPR  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ADRIANO SANT'ANA BELLI**

**IMPLEMENTAÇÃO DE CONCEITOS DA INDÚSTRIA 4.0**

**CURITIBA**

**2019**

**ADRIANO SANT'ANA BELLI**

**SIMULAÇÃO EM CÉLULA ROBOTIZADA PARA IMPLEMENTAÇÃO DE  
CONCEITOS DA INDÚSTRIA 4.0**

Projeto apresentado como requisito de aprovação da disciplina TE293 – Projeto Integrado A, no curso de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná.

Professor orientador: M.Sc. Waldomiro Soares Yuan.

**CURITIBA**

**2019**

## RESUMO

Devido ao avanço exponencial da capacidade dos computadores, aumento da quantidade de informação digitalizada e novas estratégias de inovação, obtêm-se novos modelos tecnológicos. Atualmente convive-se em meio a uma transição de tecnologia na área industrial, a chamada manufatura avançada ou indústria 4.0 (I4.0), iniciou-se na Alemanha em 2010 e busca a interconexão “das coisas” em uma única rede (*internet*), criando um acúmulo de dados chamados de *Big Data*, que deve ser associado para gerar informação e assim a inteligência artificial tome decisões autônomas, utilizando os recursos de forma eficiente com o mínimo impacto ambiental e consumo de energia. Neste projeto o conceito das diretrizes da I4.0 serão abordados e aplicados em uma simulação de uma célula robotizada.

Palavras-chave: indústria 4.0, prensa eletrônica e IoT (*Internet of Things*).

## **ABSTRACT**

Due to the exponential advancement of computer capacity, increasing the amount of digitized information and new strategies of innovation, new technological models are obtained. Currently coexisting in the midst of a technology transition in the industrial area, so-called advanced manufacturing or industry 4.0 (I4.0), it started in Germany in 2010 and seeks to interconnect "things" into a single network, creating an accumulation of data called Big Data, which must be associated to generate information and thus artificial intelligence make autonomous decisions, using resources efficiently with minimal environmental impact and energy consumption. In this project the concept of the I4.0 guidelines will be approached and applied in a simulation of a robotic cell.

Keywords: industry 4.0, electronic press and IoT (Internet of Things).

## LISTA DE FIGURAS

Figura 1 – Coordenadas cartesianas. ....	11
Figura 2 – Diagrama de funcionamento. ....	16
Figura 3 – Revoluções Industriais da 1.0 até 4.0.....	18
Figura 4 – Paineis com Relé Eletromecânico. ....	21
Figura 5 – Protótipo do Primeiro CLP – Modicon. ....	22
Figura 6 – Arquitetura Básica de um CLP. ....	24
Figura 7 – Exemplos linguagens de programação. ....	27
Figura 8 – Modelo de software. ....	28
Figura 9 – Processo em Malha Fechada. ....	31
Figura 10 – Estrutura do Sensor. ....	31
Figura 11 – Sensor Encoder. ....	33
Figura 12 – Topologia de funionamento. ....	47
Figura 13 – Tradução diagrama malha fechada para FB. ....	48
Figura 14 – Estrutra criação de objetos.....	49
Figura 15 – Programa Interpolação de eixos – robotSim.....	50
Figura 16 – Painel de visualização eixos virtuais. ....	52
Figura 17 – Eixos Virtuais 3D.....	52
Figura 18 – Estrutura do Influx. ....	53
Figura 19 – Estrutura Influx DB. ....	54
Figura 20 – Estrutura send _database. ....	55
Figura 21 – Programa Principal.....	56
Figura 22 – Programa Principal.....	57

## LISTA DE TABELAS

Tabela 1 – Etapas da IEC 61131. ....	25
--------------------------------------	----

## LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
BI	<i>Business Intelligence</i>
CAD	Projeto Auxiliado por Computador
CAE	<i>Computer Aided Engineering</i>
CAE	Engenharia Auxiliada por Computador
CAM	Manufatura Auxiliada por Computador
CAPP	Planejamento de Processo Auxiliado por Computador
CIM	<i>Computer Integrated Manufacturing</i>
CLP	Controlador Lógico Programável
CRM	<i>Customer Relationship Management</i>
ERP	<i>Enterprise Resource Planning</i>
I4.0	Indústria 4.0
IHM	Interface Homem Máquina
IOs	Entradas e Saídas
IoT	<i>Internet das Coisas</i>
JIT	<i>Just In Time</i>
LED	<i>Light Emitting Diode</i>
M2M	<i>Machine to Machine</i>
MES	<i>Manufacturing Execution System</i>
NR	Norma Regulamentora
PCP	Planejamento e Controle de Produção
PDCA	Planejar, Executar, Verificar e Agir
PWM	Modulação por Largura de Pulso
TA	Tecnologia de Automação
TI	Tecnologia da Informação
TQC	<i>Total Quality Control</i>

## SUMÁRIO

1.	INTRODUÇÃO .....	10
1.1	PROBLEMÁTICA .....	10
1.2	OBJETIVOS.....	12
1.2.1	Objetivo Geral.....	12
1.2.2	Objetivos Específicos.....	12
1.3	JUSTIFICATIVA.....	13
1.4	MOTIVAÇÃO .....	13
1.5	METODOLOGIA .....	15
1.6	ESTRUTURA.....	16
2.	FUNDAÇÃO TEÓRICA .....	17
2.1	INDÚSTRIA 4.0 (I4.0) .....	17
2.2	TECNOLOGIA DE AUTOMAÇÃO.....	21
2.2.1	Controlador Lógico Programável (CLP) .....	21
2.2.2	IEC 61131 e Controles Básicos .....	25
2.2.4	Sensores Industriais .....	30
2.3	Internet das Coisas, Redes e Fluxo de Informação .....	33
2.4	PROCESSO DE SOLDA ROBOTIZADA.....	40
2.4.1	SOLDAGEM AUTOMOTIZADA .....	43
3	DESENVOLVIMENTO.....	46
3.1	VIRTUALIZAÇÃO DOS EIXOS .....	47
3.2	ENVIO DOS DADOS AO SERVIDOR .....	53
3.3	CONSTRUÇÃO DA APLICAÇÃO NO CLP .....	55
3.4	VISUALIZAÇÃO DE DASHBOARD.....	58
3.5	PARAMETRIZAÇÃO DE ENVIO DE ALARMES .....	58
4	RESULTADOS .....	59
5	CONCLUSÃO .....	67

REFERÊNCIAS BIBLIOGRÁFICAS .....	69
APÊNDICE A – Programa Eixos Virtuais .....	72

## **1. INTRODUÇÃO**

A evolução tecnológica tem alcançado grandes resultados devido aos investimentos em pesquisas, equipamentos e profissionais. Desde os primórdios o homem anseia pelo aperfeiçoamento de tarefas cotidianas e instintivamente a busca por qualidade de vida, tais necessidades impulsionaram para o desenvolvimento e criação de tecnologias. A evolução ocorre de maneira natural, devido às constantes transformações e a busca pela adaptação ao meio.

A indústria 4.0 (I4.0) como é chamada a quarta revolução industrial tem o propósito de realizar o compartilhamento de dados dos processos industriais, para tal faz-se necessário utilizar conceitos preconizados da internet das coisas (IoT), armazenando em um servidor comum ou nuvem (Cloud), a fim de recriar um sistema cyber físico para monitorar processos reais, manipulando, controlando e testando novas maneiras de inteligência descentralizada para que os equipamentos possam tomar decisões, cooperando entre si (Machine to Machine – M2M) e com seus usuários (BORGES; DAUM, 2016).

Devido a virtualização do ambiente real, todos dispositivos, equipamentos e até usuários terão uma identificação na rede industrial tornando possível a interação de todos elementos, possibilitando realizar simulações, alterações de parâmetros remotamente, controle e até acompanhamento em tempo real do processo fabril. Os dados extraídos do sistema através de sensores e lógicas combinacionais serão arquivados em um grande servidor, conhecido com Big Data, e através da nuvem será possível realizar consultas para auxiliar as tomadas de decisões e estratégias, tornando o processo de reação mais rápido e eficiente.

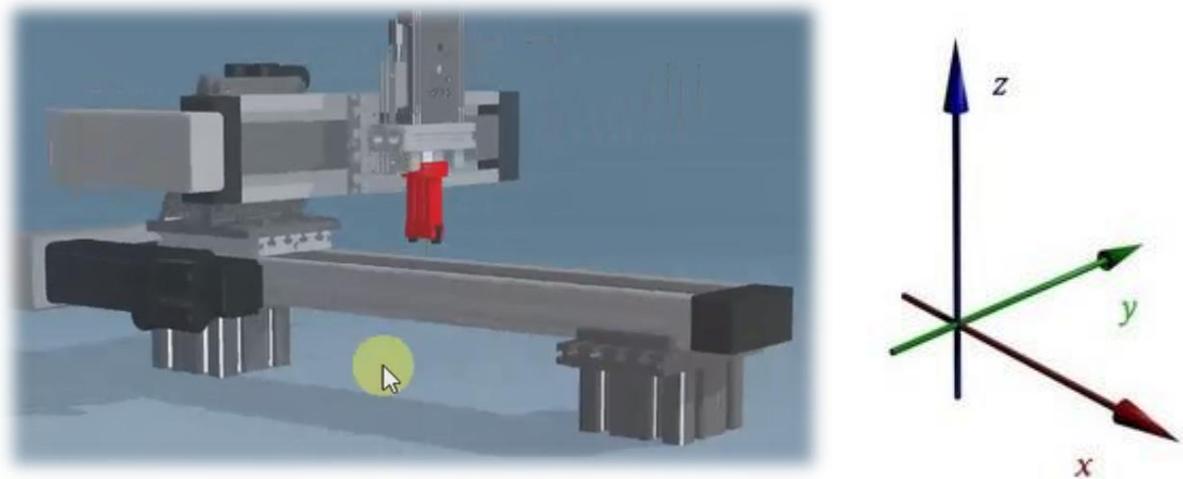
### **1.1 PROBLEMÁTICA**

Processos industriais são realizados através de uma sequência lógica de operações que somadas resultam em um produto ou serviço. Geralmente os processos de fabricação possuem variáveis de controle que fornecem ao sistema supervisor informações diversas. Cabe a quem fará a interpretação escolher quais dados devem ser analisados e monitorados para criar uma informação confiável.

Os robôs industriais são utilizados em larga escala para realizar tarefas repetitivas, podendo envolver altas capacidades de força e velocidade ou até mesmo a restrição de trabalhos para humanos em ambientes agressivos. Na indústria automotiva são aplicados em diversos setores, desde alimentação de chapas metálicas em prensas para formação da lataria de veículos, até em montagens complexas que exigem rastreabilidade como soldas e aperto de parafusos.

Os eixos de liberdade que definem aplicação de cada robô são definidos de acordo com atividade que o mesmo irá desempenhar, processos de soldagem necessitam de pelo menos 3 eixos, conforme exemplificado na Figura 1 (coordenadas x, y e z). Neste tipo de processo uma das informações primordial para garantir qualidade é a localização da pinça de solda que fica na extremidade do braço robótico. A aquisição desta informação pode ser efetuada de várias maneiras, sendo a mais eficiente a utilização da leitura do encoder dos servos motores associados.

Figura 1 – Coordenadas cartesianas.



Fonte: Própria.

A interpolação nos movimentos dos servos acionamentos do robô são o alvo deste estudo, pois através destes é possível garantir a qualidade do trabalho de soldagem e conseqüentemente a segurança estrutural do veículo. A simulação dos eixos ocorrerá em ambiente virtual a fim de criar uma fonte de dados para aplicação, e através dos conceitos de I4.0 estes serão enviados para um servidor (local) a fim de criar uma interface remota para exibição do dashboard com dados em tempo real, além de uma interface para parametrização de alarmes e diagnósticos.

O desenvolvimento envolverá a integração entre a tecnologia da informação, automação e engenharia de software. O sistema deverá alimentar com os dados dos enconder o banco de dados e através de associação produtiva criar a informação útil ao usuário, que fará acesso remoto através de navegador (browser) ou será acionado via e-mail se os parâmetros estipulados estiverem fora dos limites.

## **1.2 OBJETIVOS**

Para o desenvolvimento de um sistema com controle de qualidade, rastreabilidade, transparência do chão de fábrica, faz-se necessário a utilização de elementos de controle dedicados e capazes de realizar a comunicação via rede industrial.

### **1.2.1 Objetivo Geral**

Desenvolver um modelo de célula robotizada, baseada em servo motores, que possibilite a rastreabilidade de processos, conforme preconizado pela indústria 4.0.

### **1.2.2 Objetivos Específicos**

1. Desenvolver interpolação de 3 eixos servo acionados em ambiente virtual para geração de dados através de linguagem de texto estruturado (orientado a objetos);
2. Desenvolver banco de dados para transcrição dos dados no servidor;
3. Desenvolver aplicação para transferir erros provenientes dos servos controladores dos atuadores;
4. Gerar dashboard com os dados proveniente do servidor – database, para acesso de dispositivo em rede local;
5. Configurar alarmes e envio de mensagens de valores absolutos aos usuários cadastrados.

### 1.3 JUSTIFICATIVA

O estudo de caso ocorrerá através de uma simulação de interpolação de eixos, processo amplamente encontrado em várias montadoras de veículos que necessitam de dados para garantir a qualidade do processo, bem como gerar histórico para rastreabilidade devido a exigências e normativas mundiais.

A evolução dos hardwares e a expansão das redes de comunicação contribuíram para que a quarta revolução industrial chegasse e abrisse caminho para a virtualização de dados e até processos fabris. O novo conceito mostra de maneira transparente o dado puro, sem nenhum tratamento, faz com que o usuário crie associações entre eles para gerar informação útil na busca pelo aperfeiçoamento do processo.

O projeto pode ser dividido em 4 etapas, são elas: desenvolvimento de software com base no hardware específico, conectividade com o servidor, geração gráfica para visualização do usuário a distância e envio de mensagem de alarmes.

### 1.4 MOTIVAÇÃO

O sistema produtivo fornece e recebe um grande volume de dados, os quais interpretados podem resultar em informação útil para o processo. Para uma indústria se adequar as premissas da I4.0 seus processos e equipamentos devem estar presentes em sua rede industrial, além disso seus fornecedores e clientes também devem compartilhar tal tecnologia para que a cadeia possa estar completa e funcionar de maneira eficiente.

A busca pelo aperfeiçoamento nos processos industriais é constante, e por causa disso, a Tecnologia de Automação (TA) depende dos recursos de software e hardware disponibilizados pela Tecnologia da Informação (TI), a interrelação entre os dois sistemas aliados a internet das coisas (Internet Of Things – IoT) possibilita a criação do sistema chamado Big Data. (BORGES; DAUM, 2016).

Para Borges e Daum (2016), a TI tem papel fundamental na utilização das ferramentas para realizar a gestão da unidade empresarial, enquanto que a TA é responsável pela medição, controle, automação e segurança da planta ou máquina na unidade de produção. A IoT é um termo que está em alta e para o meio industrial

é a plataforma disponível para criar uma rede de comunicação, a qual é responsável por produzir e consumir informações da unidade industrial através da interação com pessoas, máquinas, equipamentos e dispositivos. A nuvem, ou Cloud, é o meio onde haverá a disponibilização da informação e seu objetivo principal é centralizar, proteger e distribuir as informações. O Big Data é o banco de dados responsável pelo processamento e gravação das informações da unidade empresarial, como: volume, velocidade, variedade, tipo, etc.

Atualmente a TI analisa os dados de forma off line através de planilhas do setor de produção. A automação foca no comando e controle, mas não enxerga a produção. As tomadas de decisões são reativas e não são alinhadas ao negócio de forma estratégica. As prioridades são definidas de acordo com a função que o setor é responsável. A TI prioriza a proteção dos dados, confidencialidade, integridade e disponibilidade, enquanto que a TA prioriza proteger o processo, fornecendo integridade, disponibilidade e confidencialidade ao chão de fábrica.

A indústria 4.0 visa a convergência desses setores, interligando as redes de informação de toda a cadeia produtiva industrial, objetivando a formação de dados inteligentes para tomadas de decisões. Borges e Daum (2016) listam os benefícios que a manufatura avançada propiciará, são eles:

- decisões estratégicas;
- criação de regras de negócio;
- menor tempo de maturação no desenvolvimento de um produto no mercado;
- maior flexibilidade na produção;
- padronização da operação;
- manutenção inteligente;
- menor custo de propriedade;
- redução de custos;
- economia de energia;
- conservação ambiental;
- aumento da segurança;
- redução dos erros;
- melhoria do uso do ativo;
- redução do desperdício;

- transparência nos negócios;
- gerenciamento do risco do negócio.

A convergência irá agregar valor à informação, a qual estará disponível online e sua distribuição será em tempo real. A gestão da planta terá foco na criação de indicadores de acordo com a tarefa a ser desenvolvida, possibilitando tomadas de decisões estratégicas e verificando quais impactos haverá no negócio.

Zanini (2016) afirma que algumas plantas principalmente na Alemanha possuem projetos piloto e podem nos influenciar no desenvolvimento do mercado interno, a planta de controle industrial da Siemens em Amberg é considerada pela companhia como o estado da arte global, pois produtos (matéria-prima) e máquinas se comunicam uns com os outros, permitindo fluxo de informações e através do controle descentralizado, permitem que haja um controle da produção. Como resultado, esta unidade conseguiu com o mesmo espaço físico, ampliar sua capacidade fabril em 8 vezes nos últimos 20 anos.

## 1.5 METODOLOGIA

Para atingir o objetivo proposto será necessário criar uma série de aplicações intermediárias no CLP através do software Indraworks da fabricante Bosch. O processo deve iniciar pela virtualização dos eixos, esta etapa é extremamente importante para gerar dados, pois a aquisição de dados ocorrerá através da leitura dos dados do encoder de cada um dos 3 eixos.

Com os dados retirados do sistema, o CLP transfere para o servidor e paralelamente executa linhas de teste lógico a fim de verificar problemas através dos parâmetros pré-definidos pelo usuário. Quando identificado algum problema, o CLP executa a respectiva função de envio do alarme ao usuário através de e-mail.

O usuário possui a possibilidade de visualização remota em tempo real dos dados do processo através de um browser HTML, isso é possível através de um dashboard criado na nuvem graças a comunicação que o CLP retira do processo e comunica em rede ethernet.

A Figura 2 representa graficamente os processos associados ao desenvolvimento do projeto, o intuito deste é descrever o fluxo de dados e geração da

informação para que o usuário possa realizar ação com maior rapidez, de forma assertiva, além da rastreabilidade necessária ao processo robotizado.

## 1.6 ESTRUTURA

No capítulo 1, introduz-se brevemente sobre o assunto, inserindo o problema encontrado, os objetivos geral e específicos para atingir a solução proposta e a justificativa do estudo, bem como os diferenciais do estudo em relação aos existentes.

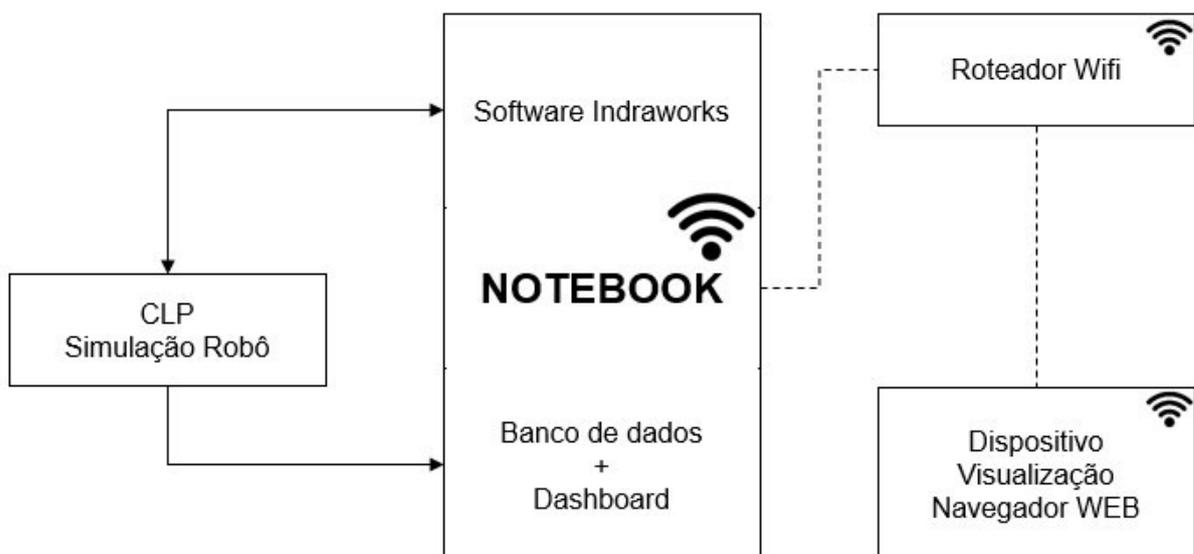
No capítulo 2, é apresentada uma fundamentação teórica dos pontos considerados importantes para o entendimento e aplicação das tecnologias envolvidas e metodologias de produção.

O capítulo 3, desenvolvimento da metodologia para programação dos eixos virtuais e gravação dos dados.

No capítulo 4, contextualização da aplicação, definição dos limites e geração da interface com usuário.

Finalmente, o capítulo 5 apresenta os resultados e considerações finais.

Figura 2 – Diagrama de funcionamento.



Fonte: Própria.

## 2. FUNDAÇÃO TEÓRICA

### 2.1 INDÚSTRIA 4.0 (I4.0)

A revolução industrial é um processo natural de evolução dos sistemas produtivos devido as mudanças das necessidades humanas. Invariavelmente atribui-se a um determinado fato na história, o marco que determina que, a partir daquele momento, a revolução industrial ocorreu, porém este é um processo que pode levar anos ou até décadas para que corporações busquem adequações e possam alcançar o grau de tecnologia proposto.

Em meados do século XVIII, na Inglaterra, o matemático e engenheiro James Watt (1736-1819) foi o responsável por melhorar a máquina à vapor, através da utilização do controle mecânico (comes) e o domínio da utilização da energia hidráulica. Anos mais tarde incorporou sua descoberta às máquinas têxteis de Newcomen, marcando o início do primeiro grande marco na era industrial, denominada Primeira Revolução Industrial (RIBEIRO, 1999).

Posteriormente, em 1914 nos Estados Unidos, o empreendedor estadunidense Henry Ford (1863-1947) funda a Ford Motor Company, a qual utiliza a energia elétrica e racionalização do processo fabril para criar uma linha de montagem para seus automóveis Ford T, produzindo em massa, com tempos reduzidos e a baixos custos comparados a época. A este episódio denominou-se o início da Segunda Revolução Industrial, conhecida também pela eletrificação (RIBEIRO, 1999).

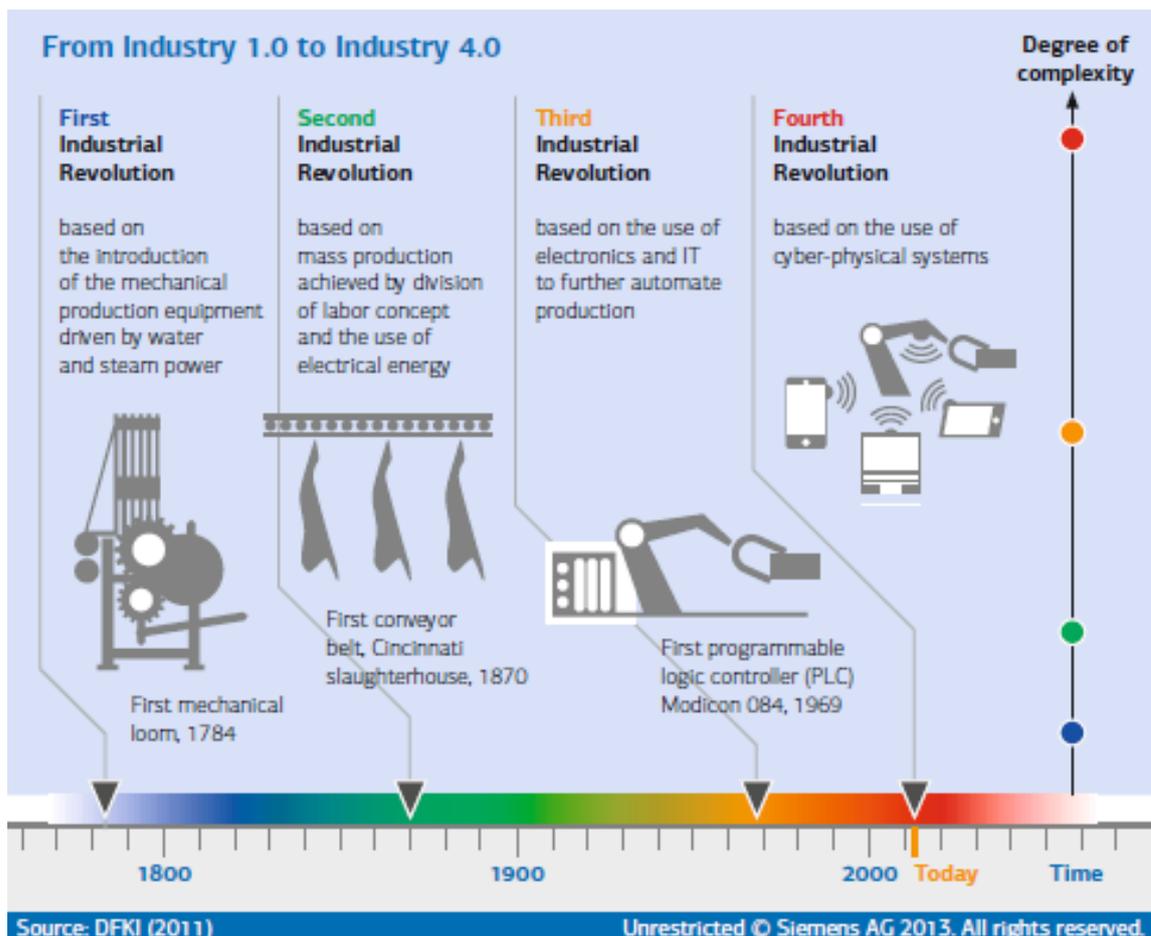
No século XX, após as duas guerras mundiais, houve grande avanço na área da informática e suas aplicações nos campos da produção e do consumo. Neste período novamente os Estados Unidos surgiram como grande potência no desenvolvimento dos sistemas de digitalização, devido à integração do chão de fábrica com a Tecnologia da Informação (TI), através dos Programmable Computers (PCs) ou Controladores Lógicos Programáveis (CLPs), redes fieldbus e armazenamento eletrônico dos dados. Estes elementos tornam-se essenciais para o início do processo de robotização da era da automação industrial (RIBEIRO, 1999).

A palavra automação pode ser associada a utilização de máquinas, equipamentos e dispositivos necessários para que um processo de fabricação ou beneficiamento funcione sem a necessidade do “fator humano”. Segundo Ribeiro

(1999, p. 13), “A automação industrial é a operação de máquina ou de sistema automaticamente ou por controle remoto, com a mínima interferência do operador.”, também pode entender por automação como o controle de processos automáticos, ou conjunto de mecanismos necessários para desempenhar uma função requerida em resposta aos estímulos.

Com a evolução dos sistemas de informação e conseqüentemente os sistemas de comunicação, o século XXI está sendo marcado pela quarta revolução industrial (I4.0). Em 2010, o governo alemão reuniu um grupo de empresários, acadêmicos e pesquisadores para iniciar um conselho que visa criar diretrizes para promover a informatização da manufatura, mais conhecida como fábricas inteligentes ou manufatura avançada (BORGES; DAUM, 2016). A chamada I4.0 consiste na integração profunda entre os processos de fabricação com gerenciamento e disponibilidade de dados online. A Figura 3 exemplifica graficamente as revoluções industriais através da linha temporal.

Figura 3 – Revoluções Industriais da 1.0 até 4.0.



Fonte: DFKI – Siemens – 2013.

O principal objetivo da I4.0 é aprimorar a tecnologia das fábricas a ponto que as linhas de montagem, máquinas, motores e até os sensores possam “conversar” entre si, passando a assumir etapas do processo de produção e de manutenção. O novo conceito visa reduzir custos de fabricação, insumos e instalações, ao mesmo tempo em que aumentará a produtividade, podendo criar lotes menores e individualizados para atender aos consumidores cada vez mais exigentes (MÜLLER, 2017).

A ideia fundamental é a planta reconhecer automaticamente o que deve fazer com cada peça. Estas por sua vez deverão conter um sistema de identificação, o qual invariavelmente será um chip eletrônico. As máquinas serão registradas digitalmente para poder comunicar com outras máquinas ou peças. A integração vertical propiciará um fluxo maior de informações entre os processos, e horizontalmente haverá maior alcance, possibilitando comunicação com outras fábricas, fornecedores e clientes.

O rastreamento das peças aliado com as etapas de fabricação pré-definidos, permite as máquinas comunicarem entre si a fim de encontrar soluções para situações não desejadas, como: quebras, indisponibilidade, gargalos de sistema, etc. Outro avanço se dará com a produção de peças especiais, sem que isso exija fazê-las à parte. Trata-se da individualização detalhada, sem detrimento da larga escala. Isso se dará com a identificação de cada peça por meio de chip eletrônico, ondas de rádio ou códigos de barras. Ao apresentar esse “cartão de visita” às máquinas, elas saberão o que fazer com aquele objeto, assim produzirão séries menores, sem prejudicar o tempo de fabricação.

Tudo, como se percebe, envolve tecnologias de informação e comunicação aplicadas à robótica. O passo mais importante será o uso da chamada “Internet das Coisas” ou IoT, a qual ligará objetos e aparelhos a enormes bases de dados e redes. Isso significa que, por meio da internet, um equipamento ou sistema poderá “conversar” com outras fábricas, fornecedores e clientes.

Com o avanço exponencial da capacidade de processamento dos computadores, a imensa quantidade de informação digitalizada, as novas estratégias de inovação e o conhecimento e tecnologia conectados na mesma rede criando novos modelos inovativos na produção, a I4.0 promoverá o controle descentralizado de os ativos online, eficiência energética da cadeia e planta orientada ao Business Intelligence (BI). Não existirá diferença entre softwares de gestão, como Enterprise Resource Planning (ERP) Manufacturing Execution System (MES), BI e Customer

Relationship Management (CRM). Todos estarão convergidos em uma mesma plataforma (SLACK, 2009).

Se um produto customizado vai determinar como será produzido no futuro, então a logística deve ser controlada a partir da entrada de dados RFID em os sistemas ERP e MES. Os dados relacionados a objetos controlam não apenas etapas individuais de processo e de edição, mas também, por exemplo, o tipo ou a variação específica de alimentação e ejeção de paletes de peças em sistemas distribuídos.

A fabricação totalmente autocontrolada ainda é uma visão. O conceito Smart Factory consiste em inteligência distribuída, comunicação aberta e linguagens de programação simples (BORGES; DAUM, 2016).

Na planta de Homburg da Bosch Rexroth na Alemanha foi desenvolvida em 2014 a primeira linha de produção baseada nos conceitos da I4.0. O projeto piloto partiu da adequação de uma linha convencional e alguns meses depois a linha por completa já realizava a troca de informações e readequando a produção as necessidades conforme demanda (KUHN, 2015).

A linha de produção é responsável por realizar a montagem de válvulas hidráulicas no qual existem 200 versões possíveis, portanto há necessidade de realizar o set-up para cada mudança de tipo de válvula a ser montada. A linha pode entregar com mais agilidade os produtos ao mercado, com uma grande variação, aliado a redução de custos. Desta maneira a manufatura pode oferecer customizações avançadas e produtos maturo conforme necessidades futuras.

Muitas dúvidas surgem à medida que avança com o tema de automatização de sistemas fabris, pois através destes, os conceitos de mão de obra sofreram constantes transformações. A automação possibilitou o aumento da produtividade, eficiência e economia. As aplicações da automação não se resumem a substituir o trabalho humano, e sim melhorar a qualidade de processos, otimização de espaços, redução no tempo de produção e custos (PAREDE; GOMES, 2011).

Com a evolução dos sistemas de produção e complexidade dos processos, tamanhos de plantas, produtividade, segurança e proteção do meio ambiente, a instrumentação surgiu com papel essencial a fim de garantir as especificações do produto ou serviço dentro das expectativas de seus clientes.

## 2.2 TECNOLOGIA DE AUTOMAÇÃO

### 2.2.1 Controlador Lógico Programável (CLP)

A evolução dos sistemas eletromecânicos possibilitou a popularização dos sistemas automáticos, porém a automação ocorreu graças o advento do circuito integrado (1960) e do microprocessador (1970). Estes elementos agregaram inteligência aos sistemas automáticos, comandando tarefas complexas. Quando o CLP surgiu na década de 1960 revolucionou os comandos industriais, pois estes eram executados por réles com lógica fixa ou em lógica hardwired (lógica através de fiações), resultando em painéis imensos interligados por circuitos elétricos e grandes fiações observado na Figura 4 (PAREDE; GOMES, 2011).

Figura 4 – Painéis com Relé Eletromecânico.



Fonte: Disponível em: <http://plcscada.com/automacao/plc-o-basico-que-voce-precisa-saber-pt1/>.

Acesso em 23/03/2019.

Qualquer tipo de alteração envolvia horas de trabalho, processos parados e conseqüentemente custos elevados, além da falta de segurança e susceptividade a erros.

Um dos grandes fabricantes da época, a General Motors (GM), cansada das mudanças na linha de produção, solicitou a produção de um componente único, que executasse as mesmas funções dos relés sem a utilização dos quilômetros de fios, para substituir os grandes painéis. O equipamento deveria ser facilmente programável, possuir manutenção e reparos facilitados, utilização em ambientes industriais, confiabilidade, fisicamente menor, produzir dados para um sistema central de coleta de informações, e ser competitivo financeiramente com relação aos painéis da época.

Em 1969 a Bedford Associates, apresentou a GM o primeiro protótipo de CLP o qual denominou Modicon (Modular Digital Controller), ilustrado na Figura 5, este possuía 3 placas, sendo a primeira a placa fonte, a segunda a placa processadora controlada por software e a terceira chamada de Logic Solver, a qual possuía finalidade de implementar o hardware as funções mais usadas pelo software.

Figura 5 – Protótipo do Primeiro CLP – Modicon.



Fonte: Disponível em: <<http://www.innovativeautomation.com/plc-programming-automation/>>. Acesso em 23/03/2019.

Graças a utilização simultânea das portas lógicas em um mesmo dispositivo os circuitos integrados puderam ser condensados em uma mesma plataforma criando o controlador lógico programável. Portanto o CLP é um equipamento eletrônico, digital, microprocessado, que possui função de controlar um proceso ou uma máquina

e ser programado ou reprogramado rapidamente e quando necessário, guardando o programa na memória. O programa que realiza a lógica interna dentro do PLC é inserido no controlador através de microcomputador, teclado numérico portátil ou programador dedicado (PAREDE; GOMES, 2011).

Inicialmente os CLPs foram utilizados em aplicações de controle discreto (on-off, liga-desliga), como os sistemas à réles, porém com a grande vantagem sobre os painéis, sendo facilmente instalados, economizando espaço energia, além de possuírem indicadores de diagnósticos o que facilitavam a manutenção. Quando se identificava necessidade na alteração ou adequação da linguagem lógica de controle da máquina era realizada em pouco tempo, apenas com mudanças no programa, sem necessidade de alterar à arquitetura elétrica.

Durante a década de 70, houve consideráveis aprimoramentos dos CLPs. As inovações tecnológicas dos microprocessadores e a possibilidade de agregar funções, propiciaram maior flexibilidade e um grau maior de aperfeiçoamento na sua eletrônica interna. Os módulos de expansão adicionam funcionalidades cada vez mais complexas, como: temporização e contagem, operações aritméticas, manipulação dos dados de comunicação, interfaces homem-máquina (IHM), capacidade de memórias, controles analógicos, controle Proporcional Integrativo Derivativo (PID), módulos de entradas e saídas (I/Os – entradas e saídas de sensores, atuadores, botoeiras, sinalizadores, etc.) remotos com controle de posicionamento, dentre outros (PAREDE; GOMES, 2011).

Comparado ao microcomputador dedicado, o PLC é relativamente mais barato, fácil de projetar e instalar. Porém seu maior diferencial é a flexibilidade em relação às interfaces de controle e possibilidade de comunicação e expansão com outros dispositivos.

No Brasil os CLPs foram implementados no início da década de 80, inicialmente apenas as filiais de empresas multinacionais, implantavam a tecnologia utilizada na matriz. Em pouco tempo, essa tecnologia proliferou e o CLP adquiriu grande aceitação no mercado. Em 1994, havia mais de 50 fabricantes de CLP, o que demonstrava seu sucesso e aceitação. (PAREDE; GOMES, 2011)

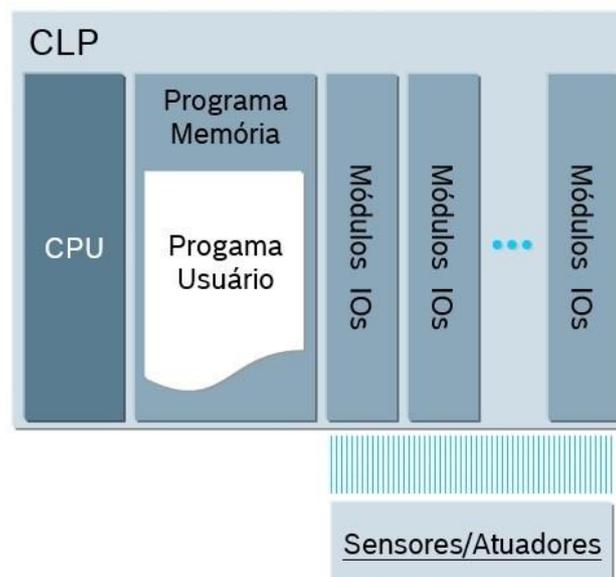
Parede e Gomes (2011, p. 15) citam algumas vantagens da utilização do CLP, ressaltando fontes de economia e superando o custo de implementação do equipamento, são elas:

- maior produtividade;

- otimização de espaços nas fábricas;
- melhoria na qualidade do produto final;
- alto MTBF (tempo médio entre falhas);
- baixo MTTR (tempo de máquina parada);
- maior segurança para os operadores;
- menor consumo de energia;
- redução de refugos;
- reutilização do cabeamento;
- maior confiabilidade;
- fácil manutenção;
- projeto de sistema mais rápido;
- maior flexibilidade, satisfazendo maior número de aplicações;
- interface com outros CLPs através de rede de comunicação.

Sensores e atuadores são conectados ao CLP através dos módulos, unidades ou interfaces de IOs. A aplicação (programa) contém uma série de instruções que são executadas linha a linha sequencialmente e ciclicamente. As saídas são controladas de acordo com as instruções da aplicação, e com os respectivos estímulos de entrada. A arquitetura básica dos elementos para funcionamento do CLP pode ser visualizada na Figura 6.

Figura 6 – Arquitetura Básica de um CLP.



Fonte: Reprodução Introdução a Controladores – Bosch Rexroth.

Devido a execução do programa ser cíclica, as leituras das entradas são lidas e armazenadas temporariamente na memória. Enquanto o programa é executado, as saídas são manipuladas e armazenadas na memória. Por fim, as saídas são atualizadas de uma vez, de acordo com a imagem pré-definida gerada pela aplicação.

### 2.2.2 IEC 61131 e Controles Básicos

Devido ao crescimento no número de dispositivos dedicados à automação industrial fez-se necessário criar uma padronização para garantir a compatibilidade de programação entre os fabricantes. Em 1979, o grupo Internacional Electrotechnical Commission (IEC) criou uma norma para concepção de um projeto completo de controladores, incluindo hardware, instalação, testes, documentação, programação e comunicação, algumas vantagens da aplicação da IEC 61131: (SILVA, 2011).

- Baixa complexidade de treinamento e incorporação;
- Estruturação fácil e confortável de programas;
- Reusabilidade;
- Várias linguagens disponíveis para diversos usuários;
- Possível utilizar várias linguagens em um mesmo programa.

Após passar por várias versões e atualizações, em 1992 a norma IEC 61131 foi subdividida em 7 etapas, apresentadas na Tabela 1, um dos pontos mais importantes e o qual recebeu maior esforço foi a padronização da linguagem de programação de controladores, ou etapa 3 (RIBEIRO, 1999).

Tabela 1 – Etapas da IEC 61131.

<b>Etapa</b>	<b>Título</b>	<b>Conteúdo</b>	<b>Publicação</b>
1	General Information	Definição da terminologia e conceitos.	2003 (2ª Edição)
2	Equipment requirements and tests	Teste de verificação e fabricação eletrônica e mecânica.	2003 (2ª Edição)
3	Programmable Languages	Estrutura do software do CLP, linguagens e execução de programas.	2003 (2ª Edição)

4	User Guidelines	Orientações para seleção, instalação e manutenção de CLPs.	2004 (2ª Edição)
5	Communications	Funcionalidades para comunicação com outros dispositivos.	2000 (1ª Edição)
7	Fuzzy Control	Funcionalidades de software, incluindo blocos funcionais padrões para tratamento de lógica nebulosa dentro de CLPs.	2000 (1ª Edição)
8	Guidelines for Application and Implementation of Programming Languages	Orientações para implementação das linguagens IEC 1131-3.	2003 (2ª Edição)

Fonte: Reprodução PLCopen.org.IEC61131/Standard.

Segundo Silva (p. 13, 2011), a IEC definiu como único padrão global para programação industrial a linguagem e Sequenciamento Gráfico de Funções (Sequential Function Chart – SFC) usada para estruturar e organizar o programa, além das 4 outras linguagens, sendo duas textuais, Texto Estruturado (Strutured Text – ST) e Lista de Instruções (Instruction List – IL), e duas gráficas, Diagrama Ladder (LD) e Diagrama Blocos Funcionais (Function Block Diagram – FBD), ilustradas na Figura 7.

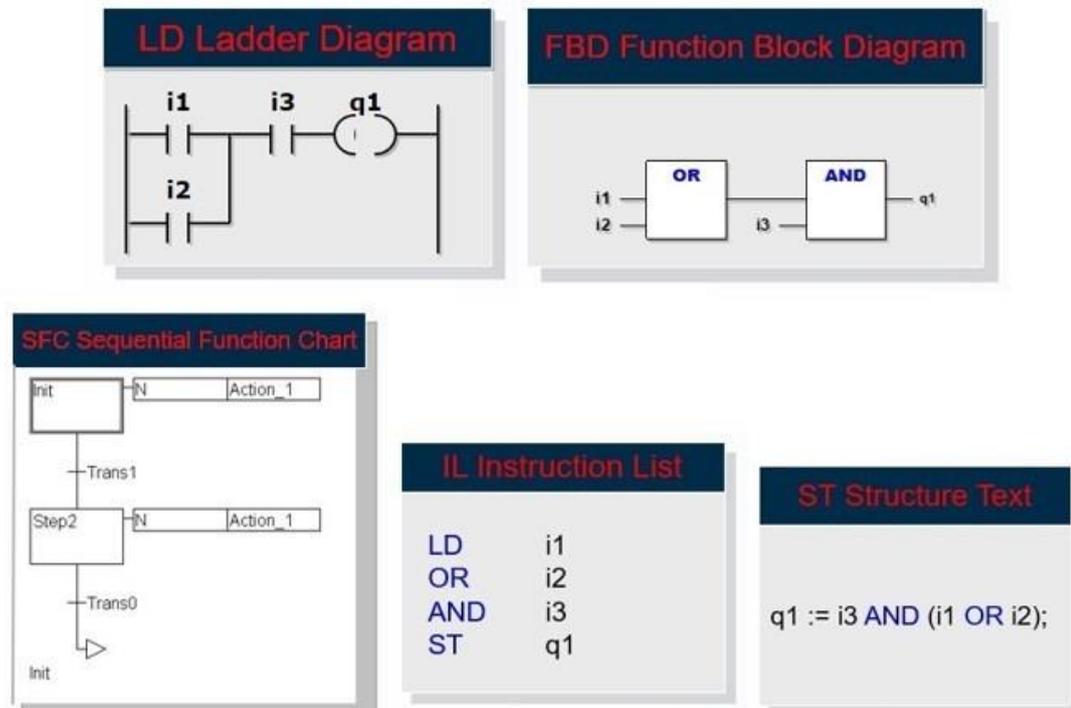
O sequenciamento gráfico de funções (SFC) é derivado das técnicas de modelagem por Redes de Petri e da norma IEC 848 (1988) que define o padrão Grafcet (Graphe Fonctionnel de Command Etape Trasition). Os passos são interligados com blocos de ações e transições, sendo que cada um representa um estado particular do sistema (IEC 61131, 2003).

O texto estruturado (ST) é uma linguagem de alto nível, derivada do Pascal e linguagem C. Contém funções de programação condicionais como: IF, THEN, ELSE e CASE OF, além das funções de iterações como: FOR, WHILE e REPEAT. Na lista de instruções (IL) as funções são sequencias de comandos padronizados, assemelhando-se a linguagem Assembler.

O diagrama Ladder, ou lógica de diagrama de contatos, assemelha-se a tradicional notação de diagramas elétricos e de painéis de controle a relés, pois descreve o fluxo de energia entre contatores e bobinas, possui símbolos padrões e operações booleanas. É a linguagemx de programação de CLP mais comum e difundida, representada através de elementos gráficos e associados sequencialmente

para gerar as funções e lógicas. Uma grande vantagem do diagrama lógico binário é a facilidade com que este pode ser combinado com uma representação do progresso sendo controlado, possibilitando entendimento claro entre o controle e sua lógica (RIBEIRO, 1999).

Figura 7 – Exemplos linguagens de programação.



Fonte: Própria.

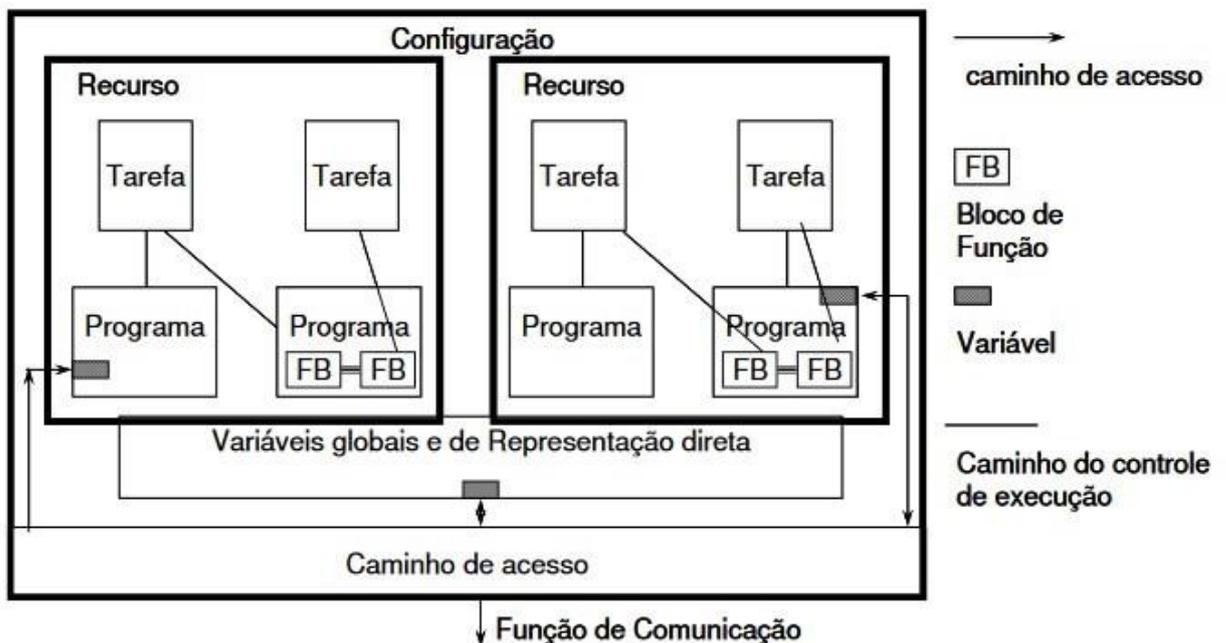
O diagrama funcional (FBD) é uma representação gráfica de instruções que devem ser executadas. Os blocos funcionais são baseados em funções fundamentais como: E (AND), OU (OR), INVERSORA (NOT).

Existem elementos comuns na norma que podem ser aplicados em todas as linguagens de programação. Um dos aspectos mais importantes é a possibilidade de decompor em programas complexos em segmentos menores. O modelo de software (software model) consiste em um conjunto de conceitos que definem uma infraestrutura para decomposição em partes do projeto de automação. Todo o programa deve evoluir e interagir com o ambiente onde está inserido, ou seja, o desenvolvimento do programa de CLP ocorre utilizando as interfaces e interações com o sistema de controle com a planta externa (SILVA, 2011).

Quando o CLP está em modo Run, as interfaces nele presente se comunicam, é através dos módulos de entradas e saídas (IOs) que ocorre a leitura e acesso dos dispositivos externos, as interfaces de comunicação são os sistemas que necessitam podem trocar informações com outros CLPs ou IHMs por exemplo. As interfaces de sistema são responsáveis por conectar o software embarcado ao respectivo hardware, a fim que ocorra um correto funcionamento.

Um modelo de software, ilustrado na Figura 8, pode ser representado por 11 elementos fundamentais: configuração, recurso, programas, blocos funcionais, funções, reutilização de programas, blocos e funções, tarefas, variáveis de escopo local e global, variáveis de representação direta, caminhos de acesso, fluxo de controle.

Figura 8 – Modelo de software.



Fonte: Reprodução Silva.

Configuração é o nível mais alto, corresponde ao software necessário para uma aplicação, pode ser associado entre si e interagem através de interfaces de comunicação. Para que os programas sejam executados é necessário um elemento com capacidade de processamento, podendo existir um ou mais recursos para rodar uma configuração. O programa por sua vez, consiste em um código de execução capaz de trocar dados através das conexões de software, através dele é possível

acessar variáveis e se comunicar com outros programas, a sua execução pode ser controlada usando as Tasks.

Os blocos funcionais são elementos que facilitam aplicações específicas e podem ser reutilizados ao longo do programa, a principal característica é possuir um conjunto de dados que possibilitam alteração por um algoritmo interno, mantendo as características fundamentais. As funções são elementos que reproduzem sempre a mesma operação de acordo com os valores de entrada, pode resultar em um evento simples ou de múltiplos elementos como vetores e estruturas.

A norma reconhece como Unidades de Organização de Programas (POU) os programas, blocos funcionais e funções, pois estes possibilitam a reutilização através de instâncias (exceto as funções que não são persistentes). A recursividade não é permitida por norma por motivos de estabilidade e segurança da aplicação. A utilização de Function Blocks e Functions padrões, ocorre através de bibliotecas padronizadas pela fabricante do CLP, ou pela criação específica definida pelo usuário para atender suas necessidades.

A Task permite a execução de programas em diferentes taxas, trata-se de um mecanismo de escalonamento para sistemas de tempo real, o qual executa programas ou blocos funcionais periodicamente ou em resposta a um evento. A utilização de execução em taxas diferentes, faz-se necessário para atender diferentes objetivos de acordo com aplicação, otimizando a capacidade de uso do processamento do CLP.

As variáveis de escopo local são declaradas e válidas apenas para os elementos de software que a declararam, permitindo acesso e manipulação deste próprio elemento, que pode ser um recurso, programa, função ou bloco funcional. As variáveis também podem ser de escopo global, sendo acessadas por todos elementos contidos na aplicação.

Posições de memória podem ser utilizadas usando variáveis de representação direta, elas permitem leitura e escrita de dados em posições conhecidas de memória. Seu uso é restrito aos programas, não permitindo que blocos funcionais façam acesso direto devido a reutilização dos mesmos.

Os caminhos de acesso permitem a transferência de dados entre diferentes configurações, estas por sua vez podem definir quais variáveis podem ser acessadas remotamente. A norma IEC não define mecanismos para controle de execução do fluxo de controle de informações, quando uma configuração inicia, todas variáveis

globais são inicializadas e todos os recursos são ativados. O recurso por sua vez, inicia suas variáveis e todas tarefas são habilitadas. Quando ocorre algum evento que pare a configuração, todos os respectivos recursos param, as tarefas são desabilitadas, interrompendo a execução dos programas e blocos funcionais.

#### **2.2.4 Sensores Industriais**

Com a evolução dos sistemas industriais, a eletrônica proporcionou o desenvolvimento dos comandos descentralizados, e paralelamente a variedade de novos instrumentos e técnicas de medição.

Devido as novas necessidades dos processos exige-se cada vez mais especialização na precisão, no processamento e no armazenamento das variáveis a serem controladas. A utilização da instrumentação em automação industrial foi essencial para o desenvolvimento dos sistemas automáticos (SEIDEL, 2011)

A aquisição dos dados de um processo é extremamente importante para que a lógica computacional embarcada, seja ela em um PC industrial, PLC, IHM ou mesmo relés programáveis possam desempenhar suas funções de comando e controle. Segundo Ribeiro (1999, p.) os dados coletados são usualmente analógicos e referem-se a um valor de algum parâmetro físico não elétrico, como por exemplo: pressão, temperatura, vazão, ruído, dentre outros. As funções do sensor no sistema de aquisição de dados são:

- Sentir a presença, tamanho, variação e a frequência da quantidade medida (mensurando).
- Fornecer uma saída elétrica que tenha dados quantitativos exatos acerca do mensurando.

A Figura 9 exemplifica uma estrutura básica de um sistema de controle em malha fechada, pois utiliza sensor para correção e realimentação do sistema de controle. O controlador é o elemento que processa a informação recebida pela entrada e pelo sinal do sensor para destinar a saída (atuador) a respectiva ação, caso necessária. Após ação o processo é atualizado e uma nova informação é percebida pelo sensor continuando o ciclo de comparação e processamento. Sistemas de controle podem utilizar sensores apenas para medir uma grandeza, sem função de

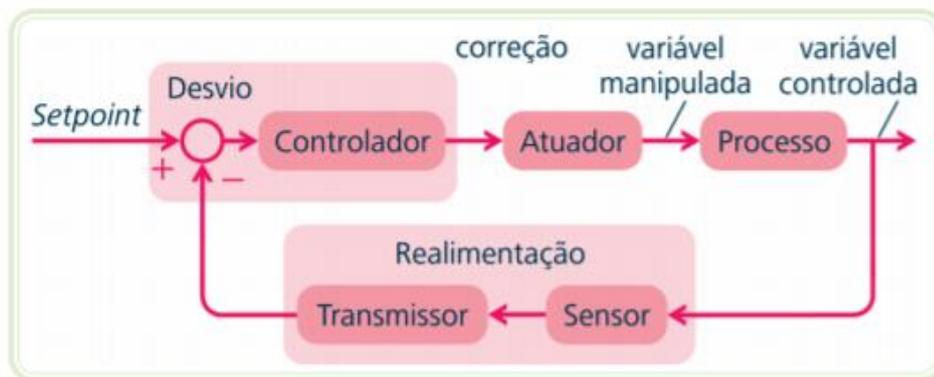
realimentação para o controlador, a estes são definidos como sistemas de controle em malha aberta (SEIDEL, 2011).

Ao medir uma grandeza do processo, o sensor é o elemento responsável por realizar a leitura e mensurar a grandeza física. Segundo Seidel (2011, p. 12), “ no sensor pode ser inserido um condicionador de sinal para que este possa ser processado. Ao conjunto sensor e condicionador de sinal que converte uma forma de energia em outra dá-se nome de transdutor”. A Figura 10 ilustra o conjunto transdutor, composto pelo elemento sensor e condicionador de sinal.

A utilização de transdutores fornece vantagens ao processo de controle e automação, de acordo com Ribeiro (1999, p. 225) as principais são:

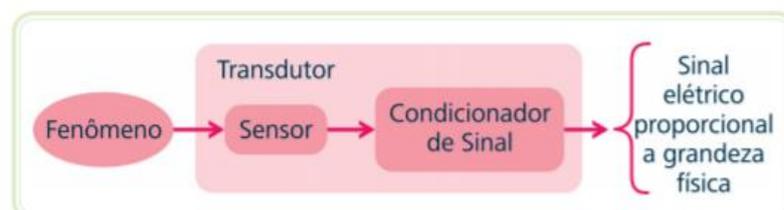
- Padronização do sinal e conseqüente padronização dos instrumentos e interfaces receptoras.
- Isolação do sinal do processo do sistema receptor. Altas pressões e temperaturas são convertidas no sinal padrão do transmissor.
- Capacidade de levar a informação para locais distantes, sem corrupção ou deformação.

Figura 9 – Processo em Malha Fechada.



Fonte: Reprodução de Seidel, 2011.

Figura 10 – Estrutura do Sensor.



Fonte: Reprodução de Seidel, 2011.

A seleção do sensor adequado pode determinar o êxito da implementação em um processo de automação industrial. Segundo Ribeiro (1999, p. 223), “os sensores podem ser classificados de acordo com o princípio elétrico envolvido em sua operação: sensor passivo, requer uma fonte externa de alimentação ou excitação. Sensor ativo, não requer uma fonte externa de alimentação”.

Os sensores podem ser divididos quanto à natureza do tipo de sinal a ser controlado, podendo ser discretos ou contínuos. Seidel (2011) define que sensores discretos são aqueles que apresentam dois estados devido à ocorrência de um evento, ou seja, ligado/desligado ou presença/ausência. Existem chaves automáticas, cuja operação é determinada pela posição de algum dispositivo ou pelo valor de alguma quantidade física. Ribeiro (1999) cita alguns tipos de chaves automáticas, são elas: pressostatos, termostato, chave de vazão, chave de nível, chave de fim de curso. Um exemplo de aplicação desse tipo de sensor é uma chave fim de curso, na qual é possível obter o sinal ‘0’ correspondente à chave aberta, e em caso de presença de estímulo, seu sinal passa a ser ‘1’.

Para Seidel (2011) os sensores contínuos são definidos como sendo aqueles que monitoram grandezas físicas numa faixa contínua de valores estabelecidos entre limite mínimo e máximo. Ribeiro (1999, p. 225-226) afirma que os sensores contínuos apresentam na saída geralmente um sinal de tensão (varia entre +/- 10 V), corrente (varia entre 4 a 20 mA), resistência ou pneumática (3 a 15 psi) proporcional à grandeza mensurada.

Encoders são sensores que convertem deslocamento angular em pulsos através de uma saída discreta sem a necessidade de um conversor analógico-digital. Podem ser do tipo incremental, ou seja, consideram o deslocamento somente em relação ao ponto de partida, e absoluto, ou seja, consideram o deslocamento em relação a um ponto de referência interno fixo. A

Figura 11, exemplifica o funcionamento do disco perfurado e o elemento de sensorização (receptor + emissor), bem como a onda pulsada resultante do processo de sensorização.

“Encoders óticos rotativos consistem de um diodo emissor de luz (LED), um disco, e de um detector de luz (fototransistor) no lado oposto ao disco. O disco fica montado no eixo e tem perfurações regulares, as quais são repetidas continuamente formando segmentos codificados no disco com recepção ou não do sinal. À medida que o disco gira, os segmentos bloqueiam ou

permitem a passagem de luz. Isso gera pulsos de onda quadrad que podem ser interpretados como informação de velocidade e de posição.” (SEIDEL, 2011, p. 50).

Outro exemplo de sensor contínuo é o encoder linear, consiste em uma régua linear magnética com tamanho variado e um sensor magnético que fica a poucos milímetros da régua e utiliza a variação do campo como grandeza física para controlar. (SEIDEL, 2011).

Figura 11 – Sensor Encoder.

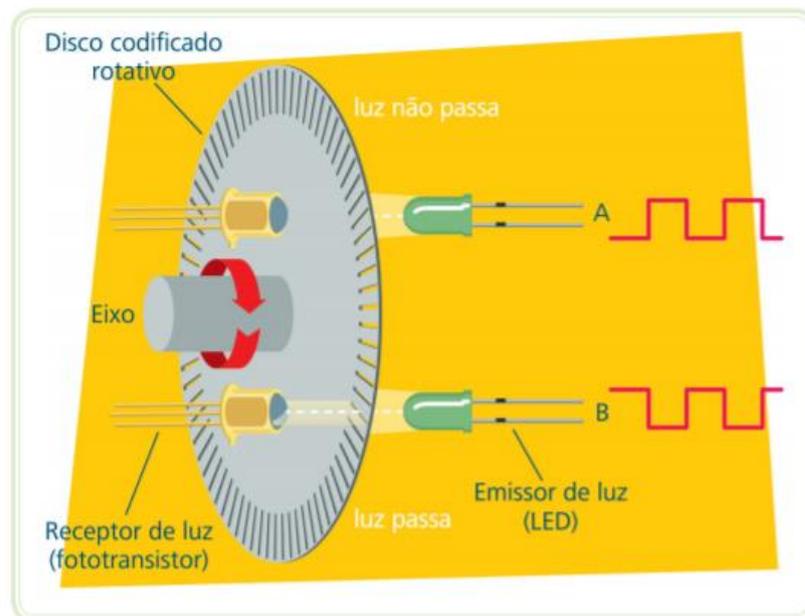


Figura 2.29: Funcionamento do encoder óptico rotativo  
Fonte: CTISM

Fonte: Reprodução Seidel.

### 2.3 Internet das Coisas, Redes e Fluxo de Informação

A tecnologia cada vez mais vem sofrendo mudanças no desenvolvimento, tendo que ser inserida em diferentes aspectos do nosso dia a dia, não se limitando há apenas máquinas. A internet vem agregando conectividade no nosso cotidiano, este mais “novo” fenômeno é conhecido como “Internet das Coisas”.

A Internet das Coisas, é conhecida pela sigla IOT que vem do termo americano “Internet of Things” e atualmente representa um cenário moderno rumo ao futuro da tecnologia. Esse mais novo conceito representa uma ligação da tecnologia com as coisas ao nosso redor e conseqüentemente na nossa vida e negócios.

Para sermos mais diretos e objetivos podemos dizer que a Internet das coisas nada mais é do que uma extensão da Internet atual. Diante deste contexto podemos dizer que a internet vem se tornando indispensável na vida das pessoas, da sociedade em si, pois com a sua evolução a tecnologia nos trouxe muitos benefícios e dentre os que podemos citar dentro deste contexto é a facilidade de conectar os objetos do dia a dia em tempo real auxiliando na vida das pessoas.

De acordo com Jeremias (2009) as mudanças e a expansão de tecnologias relacionadas à informação foram responsáveis pelas transformações sociais nos mostrando a importância da organização da informação. Antigamente um funcionário teria que estocar arquivos importantes da organização em estantes, atualmente com os serviços de unidade de informação (UI), isso vem contribuindo para essas mudanças e auxiliando o profissional em suas mais diversas tarefas, a dar conta de todas as suas atividades como serviços de consultas etc.

No entanto, sabemos também que em sua grande maioria a locais que param no tempo e que não utilizam desse serviço inovador que é a tecnologia, como por exemplo bibliotecas e arquivos de modo geral, locais que em sua maioria possui as mesmas ferramentas como forma de trabalho, sem utilizar do benefício da tecnologia, algo mais moderno e inovador que facilite os serviços principalmente onde a demanda é maior.

Com isso temos ciência de que nosso cotidiano está diretamente ligado as novas tecnologias, a dispositivos como: celulares, smartphones, notebooks, tablets e outras variedades que o mercado detém e que todos possuem, ou pelo menos algum deles, e que a informação é mais facilitada e ágil com o uso dessas novas ferramentas de comunicação.

Conforme Ayres e Sales (2010) o objetivo da Internet das Coisas (IOT) é de conectar tudo e todas as coisas como móveis, utensílios de casa, roupas, veículos etc., a Internet e formar uma rede de conexão em cada objeto e que se comunique com outros objetos e serem utilizados das mais diversas formas, facilitando assim a vida do ser humano.

Para começarmos a entender todo esse novo conceito que é a Internet das Coisas, vamos utilizar o conceito de Brito (2014), que por sua vez preferiu dividir a história da internet em três fases sendo elas: internet das máquinas, internet das pessoas e internet das coisas.

No primeiro momento a internet era um pouco diferente do que encaramos atualmente, no entanto seguia a mesma linha de raciocínio, sendo uma rede interconectada de computadores para a troca de informações.

No início da década de 60, em 1965 Lawrence Roberts e Thomas Merrill conectaram dois computadores através de uma rede, um dos computadores estava localizado em Massachussets e o outro na Califórnia. Esse primeiro experimento não foi um dos melhores, era meio lento e apresentava alguns problemas. Mas, com a evolução da tecnologia se tornou o que conhecemos hoje (LEINER, 2015).

Depois desse experimento realizado por Lawrence Roberts e Thomas Merrill mais adiante foram sendo desenvolvidos novos estudos e pesquisas de Redes e Protocolos e conseqüentemente esta tecnologia foi ganhando seu espaço. Com isso foram surgindo as novas tecnologias e logo em seguida veio o envio de e-mails que se tornou um marco na forma de comunicação etc.

Dessa forma a tecnologia que inicialmente só conectava dois computadores em diferentes locais, passou a ser uma Rede Global com diversos usuários, sendo assim considerada como a Internet das Máquinas, já que para estar conectado a internet era necessário estar conectado a um *desktop*.

A segunda fase inicia-se nos anos 2000 com o surgimento dos *smartphones*, que tomariam um grande espaço como uma forma de comunicação entre as pessoas. Nessa época a internet era estabelecida com o protocolo TCP/IP e que é utilizado até hoje, diferentemente de seu antecessor NPC que não possuía controle de erros ponto a ponto (LEINER, 2015).

A união dessas tecnologias como smartphones e internet foi um marco gigantesco para a nossa sociedade, que foi aonde começaram a surgir os smartphones com acesso a internet que possibilitava aos seus usuários maior facilidade já que se podia acessar a internet em qualquer lugar em que se tivesse acesso à internet juntamente com seu smartphone.

E a tão esperada, última geração, que é tema desse estudo. A Internet das Coisas, que apesar de ser considerado recente, não é tão recente quanto parece, tendo sua primeira aparição em 1949 como forma de código de barras (PRESS, 2014) que na época não foi utilizado, mas que no decorrer do tempo foi de grande importância para o uso comercial.

Seguindo com demais exemplos, podemos citar a empresa Coca-Cola que já naquela época utilizava sensores em uma de suas máquinas, onde era possível

observar a temperatura do ambiente e quantas garrafas ainda restava. Mais adiante a empresa italiana Olivetti inventou um sistema de cartão de identificação na década de 90 com o objeto de identificar se a pessoa realmente se encontrava em tal local, utilizando o uso do sensor.

Todas essas invenções citadas aqui, surgiram antes do conceito de Internet das Coisas, inventada em 1990 por Kevin Ashton, durante sua apresentação para a empresa Procter & Gamble e sobressaindo a colocação de sensores em seus produtos para uma melhoria na logística, onde o título da apresentação era “Internet das Coisas” e foi a partir desse momento que o termo ganhou conhecimento e nome oficial.

Segundo FARREL (2005), IP é um protocolo de comunicação da internet para entrega de dados que é feita via todos os tipos de redes. O IP é um tipo de identificação única para cada tipo de computador conectado a uma rede. E podemos assimilá-los como se fosse documentos pessoais como RG e CPF com uma numeração única para cada pessoa.

Esse IP pode ser facilmente descoberto. Quando conectado a uma rede local, chamamos de IP interno. No entanto, assim que conectado a internet, ele passa a ser chamado de IP externo que pode estar no computador ou em um equipamento com acesso à internet.

Segundo a MICROSOFT (2009) um endereço de IP tem espaço de endereçamento de 32 bits e tem segmentação do endereço IPv4 em quatro campos de 8 bits chamado de octetos, onde cada octeto é convertido em um número decimal (base 10) que vai de 0 a 255 e é separado por um ponto.

A internet quando surgiu teve sua primeira versão de protocolo IP, chamada de IPv4 contendo 32 bits como já citado. Devido ao desenvolvimento e crescimento da internet o sistema atual conhecido como IPv4 vai se esgotar muito em breve e já prevendo esse tipo de acontecimento, foi criado um novo padrão, conhecido como IPv6 com capacidade de acompanhar o crescimento da rede e para que a tecnologia acompanhe este novo ambiente é necessário que os dois endereços de IP operem juntos.

Sabendo que a IPv4 estava enfrentando problemas de esgotamento de endereços, então a IPv6 foi criado justamente para resolver este problema. Diferentemente do IPv4, o IPv6 possui 128 bits onde podemos dizer que o IPv6 é 79 bilhões de vezes a quantidade de endereços em relação ao IPv4.

De acordo com COMER (1991), temos algumas novas características ao IPv6 como, tamanho do endereço que contém 128 bits; formato de cabeçalho que é diferente do IPv4, onde quase todos os campos foram substituídos ou sofreram mudanças; suporte para áudio e vídeo; serviços de autoconfiguração; implementação de IPsec (IP Security Protocol) e crescimento do número de endereços multicast.

Com essa expansão de funcionamento de conexões, o IPv6 é um tipo de atualização do IPv4 e trabalha como uma espécie de “pilha dupla”. Este endereço foi criado justamente com a intenção de substituir seu antecessor já que o mesmo se encontrava limitado.

Para entendermos melhor essa ligação da IOT com a indústria 4.0 podemos usar a analogia de Nikola Tesla (1926), inventor na área de engenharia mecânica e eletrotécnica que disse: “Quando a tecnologia sem fio for perfeitamente aplicável, a Terra inteira será convertida em um imenso cérebro, o que de fato é, com todas as coisas sendo partículas de um todo real e rítmico”.

Tendo essa analogia de como funciona o nosso cérebro é possível compreender melhor a ideia da IOT como uma rede conectada. O nosso cérebro possui várias quantidades de conexões entre as células, compondo assim nosso sistema nervoso. Assim, as células possuem diferentes finalidades, assim como as coisas que possuem diferentes funções, mas que estão conectados sob a mesma rede.

Falando agora sobre a IOT ser utilizada para sensores inteligentes, sabemos que essa prática utilizada nas indústrias não é recente e surgiu a bastante tempo, mas com o avanço da tecnologia e a chegada da Internet das Coisas essa prática vem desempenhando um papel importante em projetos e com isso alcança-se um nível completamente diferente.

Os sensores inteligentes representam um novo começo da cadeia de coleta de dados que postos em prática ajudam a fornecer insights de importância estratégica para a tomada de decisão e para o desenvolvimento de novos projetos.

Os sensores são dispositivos eletrônicos sofisticados usados como forma de detectar movimento físico, objetos e pessoas em um determinado espaço, sendo transformado em um sinal elétrico ou óptico.

Com a adoção da IOT, os sensores foram penetrados em nossas vidas e podemos ver a presença dele em vários aspectos da nossa vida, como nas áreas da saúde e condicionamento físico, automação residencial, automotiva e logística,

indústrias, ciência, agricultura e pecuária e até em cidades onde a tecnologia é extremamente presente e podemos chamar de cidade inteligente.

Essa junção da Internet das Coisas aos sensores tem por objetivo compartilhar informações em tempo real, aumentando a eficiência de processos e a funcionalidade dos dispositivos conectados. E com a Internet das Coisas (IoT) foi possível essa autonomia que certamente nos traz em elevada escala, resultados obtidos com a utilização das novas tecnologias.

Podemos citar brevemente alguns sensores mais utilizados como o sensor de proximidade chamado de sensores indutivos, que se trata de um dispositivo que detecta a presença de um objeto próximo; sensor de gás que são utilizados para monitorar as mudanças da qualidade do ar e detectar a presença de vários gases; sensores de níveis utilizados para determinar a quantidade de fluidos, líquidos ou demais substâncias que fluem um sistema aberto ou fechado; sensores de pressão; umidade; temperatura entre outros. Bom, esses são alguns dos principais sensores presente na IOT.

Com todas essas mudanças que a tecnologia vem enfrentando nos últimos tempos com a automatização de coisas, como por exemplo, as casas com a questão da iluminação, temperatura, TV's, fechaduras e portas, etc., que podem ser controladas com um simples smartphone que sinaliza a necessidade de mudança de algo ou abastecimento de alguma coisa. Todas essas mudanças e oportunidades se deu, devido ao novo conceito de Internet das Coisas que deixaram nosso dia a dia mais tecnológico e automatizado.

Com isso a relação de confiança é de extrema importância, já que as empresas ou as próprias pessoas precisam manter uma relação de confiança seja com os clientes no caso das empresas, ou com as pessoas ao nosso redor.

Diante das mudanças com relação a tecnologia, a segurança e a proteção para a coleta e compartilhamento de informações pessoais é um dos grandes desafios a ser enfrentados por esse novo setor. A junção de dados e acessos registrados nos dispositivos inteligentes gera preocupação e coloca nossa segurança em risco, já que estamos sujeitos a comportamentos criminosos em busca de nossas informações pessoais. A questão da segurança deve sempre ser pensada quanto ao surgimento de novos produtos IoT, pois é necessário estar preparado para a segurança e privacidade dos clientes e para a nossa própria segurança também.

A IoT nos trouxe desenvolvimento social e econômico e com ela juntamente uma série de preocupações com relação à segurança, a privacidade e proteção das nossas informações pessoais. Com isso os novos desenvolvedores de software têm que visar a minimização dos riscos de ataques cibernéticos, ou seja, uma IOT que contribuía para o desenvolvimento seguro das tecnologias e de nossas informações.

Esses dispositivos devem ser preparados para conter esses ataques a nossa segurança e privacidade, ele deve ser preventivo. Os desenvolvimentos na área de projetos de nível hardware, software, hospedagem em nuvem, estabelecimento dos padrões de segurança, realizar análises de vulnerabilidade dos dispositivos, utilizar protocolos seguros de atualizações e correções dos dispositivos, entre outros, devem ser considerados prioridade para estabelecer de vez a nossa segurança e privacidade.

Ainda não se sabe qual a solução a ser adotada para a segurança e proteção das nossas informações, mas é óbvio que não podemos esperar para mudar esta situação, depois que o problema se tornar mais abrangente, é difícil de ser controlado. E corre o risco de causar muito mais dores de cabeça no futuro. Então o momento é de repensar a tecnologia e começar a implementar novos dispositivos que realmente garantam a nossa segurança e privacidade de dados.

A tecnologia sofreu muitos avanços com a chamada Tecnologia de Informação e Internet das Coisas. Com isso tivemos mudanças também nas formas de conexão de rede Wifi. Com o desenvolvimento das telecomunicações e com a grande necessidade das pessoas de se manterem informadas, a motivação para desenvolvimento de novas redes têm sido grande.

Esse crescimento das redes sem fio juntamente com a informática tem a finalidade de atender as necessidades de serviços como: celulares, transmissão de dados via satélite, etc.

Atualmente trabalhamos com dois tipos de conexão, o LAN que utiliza meios físicos como fios de cobre (par trançado, coaxial e fibra ótica) como forma de transmissão de dados e as redes sem fios chamada de Wireless ou WAN (Wireless Wide Area Network) que tem por base as principais redes de telefonia celular com princípio de comunicação por voz e transferência de dados.

Com todas mudanças no desenvolvimento da nossa tecnologia, chega-se ao que conhecemos por 3G, 4G e 5G que são as mais utilizadas pelos usuários de smartphone. A internet 3G basicamente falando é uma evolução do 2G que visa a transmissão de dados por voz (similar aos áudios dos aplicativos) e serviços de

navegação online, sendo uma conexão mais rápida que a 2G. A velocidade média aqui no Brasil dessa internet é de 8,82 Mbps (megabits por segundo).

A rede 4G conhecida como rede TLE (Long Term Evolution), nos últimos anos vem se tornando popular e já funciona em 4.197 (75,4%) dos municípios brasileiros. Essa rede, além de ser rápida permite que várias pessoas se conectem ao mesmo tempo sem perder a qualidade da conexão, onde sua velocidade chega de 4 a 100 vezes mais rápida que a internet 3G. A rede funciona numa faixa de 2.500 MHz e 1.800 MHz.

Falando de futuro podemos citar a Internet 5G, uma internet que representa mais ainda o crescimento das conexões de internet, a futura geração de telecomunicações móvel. Estudos em desenvolvimento, apontam que a internet 5G irá substituir o 4G, onde a proposta é conectar tudo ao mesmo tempo com uma internet rápida e de qualidade. Onde tudo se interconecta como carros, geladeira, máquina de lavar, TV, celulares e tantos outros dispositivos eletrônicos. As pesquisas que estão sendo realizadas nesse campo, calculam que a rede 5G chegue a uma velocidade de 10 a 20 vezes mais rápida do que a que temos hoje. Isso se tornaria um grande avanço para o desenvolvimento das telecomunicações e abriria portas para que a nossa tecnologia avançasse muito mais daqui alguns anos.

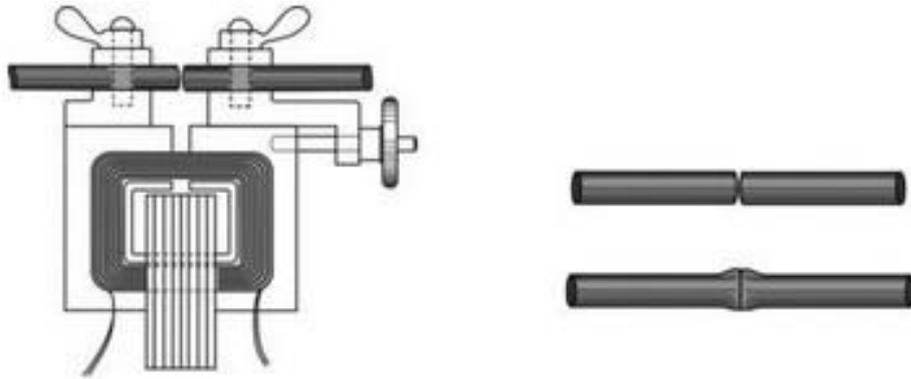
## **2.4 PROCESSO DE SOLDA ROBOTIZADA**

Segundo Weman, o principio básico da solda é realizar a união de materiais, geralmente metálicos, mas podem encontrados aplicados em polímeros. Por milhares de anos o único método de solda era através da forja, com altas temperaturas e metais pesados. No século XIX foi possível associar gases seguros, como oxigênio, acetileno, para produzir chamas com bastante calor.

Paralelamente, controlou-se as intensidades de potência elétrica sendo suficiente para gerar soldagem por resistência e soldagem por arco. A instensidade de calor permitia que o mesmo fosse gerado e aplicado à peça de trabalho o mais rápido possível gerando um ponto profundo e fundido que ao se solidificar, formava uma ligação unificadora forte entre as partes. Os métodos básicos de soldagem por resistência, gás e a arco foram desenvolvidos durante o período anterior à Primeira Guerra Mundial.

As primeiras máquinas de soldagem por resistência foram utilizadas para solda de topo, ou seja, na superfície externa do material. O inventor Elihu Thomson fez o primeiro transformador de solda em 1886 nos EUA, Figura 12, seu transformador era capaz de fornecer 2.000 amperes em um circuito aberto com diferencial de tensão de 2 volts.

Figura 12 – Transformador Thomson.



Fonte: welding processes handbook.

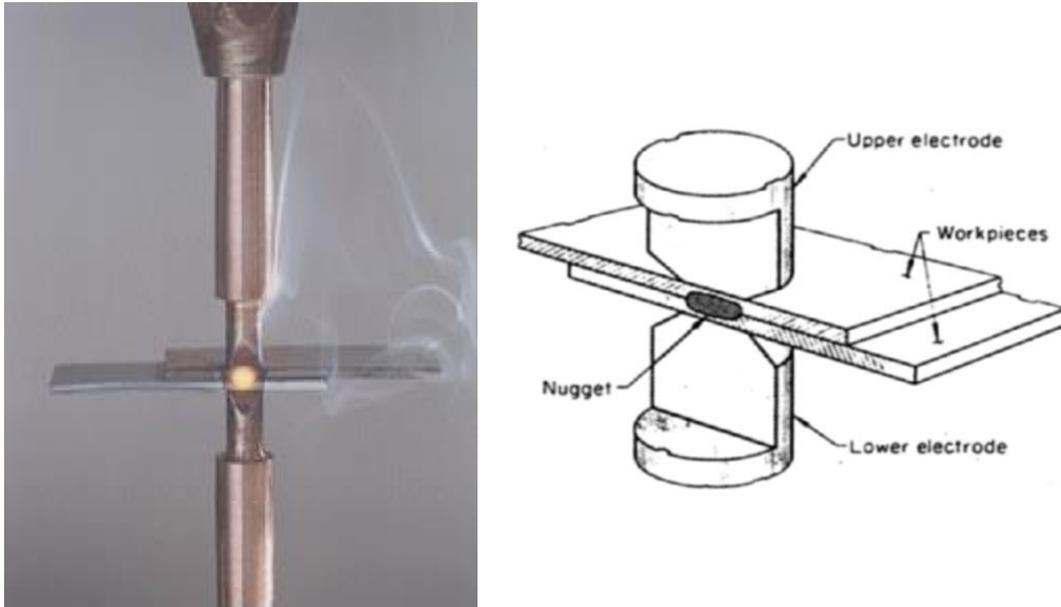
Após a virada do século, Thomson continuou a desenvolver outros tipos de máquinas, como solda a ponto, solda por costura, soldagem por projeção e solda por topo. A soldagem por pontos mais tarde se tornou o método de soldagem por resistência mais comum e hoje é usada principalmente na indústria automotiva, mas também em muitas outras aplicações de chapas metálicas.

A solda por resistência consiste em utilizar dois eletrodos de cobre, um de cada lado das chapas a serem unidas e com uma pressão aplicada na superfície das placas é fornecida uma corrente alta aos eletrodos. Nestes é aplicado um diferencial de tensão que ocasiona uma passagem de corrente alta que atravessa as chapas, as quais aquecem de dentro para fora até fundir a região interna do ponto por inteiro, conforme ilustrado na Figura 13. Os eletrodos possuem as pontas chanfradas para diminuir a área de contato e assim concentrar o fluxo de corrente (WEMAN, 2012).

Certamente a evolução dos comandos de solda se deu através da evolução dos componentes eletrônicos, por exemplo, no início do século XIX o transformador de Thomson não havia nenhum controle, no meio do século elementos como diodo, indutor e tiristores começaram a ser implementados para melhorar a qualidade no processo de soldagem. No final do século o modelo chaveador inversor, utilizado nos

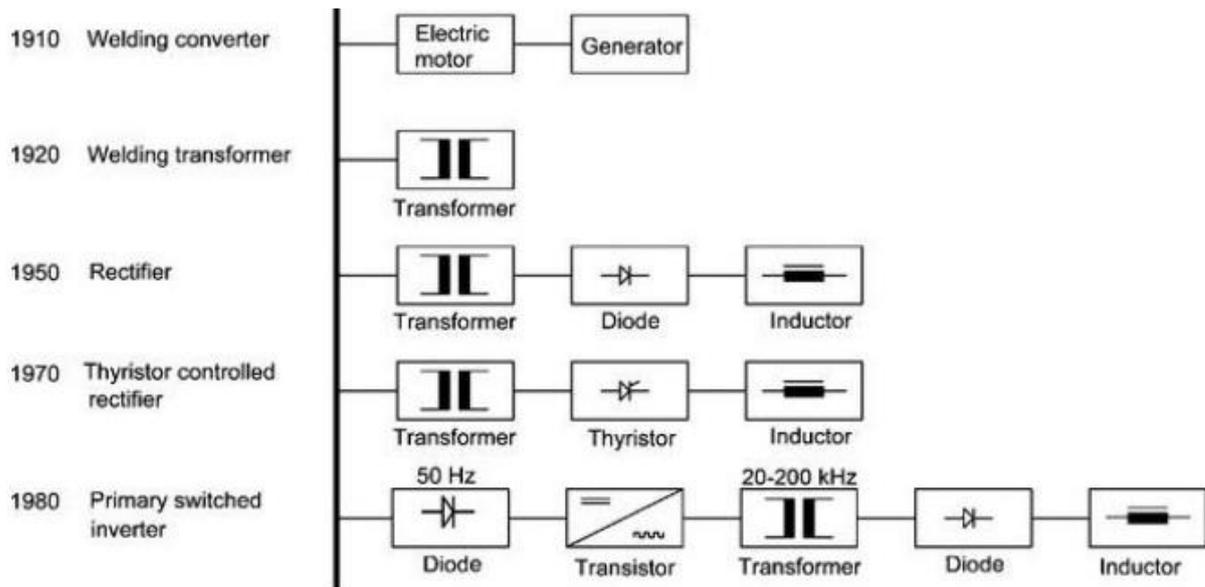
dias atuais, havia sido construído atende a maioria das aplicações, ilustrado na Figura 14.

Figura 13 – Princípio de solda por resistência.



Fonte: Welding processes handbook.

Figura 14 – Principais tipos de fonte de alimentação.



Fonte: Welding processes handbook.

Uma das maiores empresas de eletrotécnica da Alemanha, fundada em 1887, a AEG desenvolveu o primeiro gerador dedicado a solda no de 1905. Ele era acionado por um motor assíncrono trifásico e tinha características adequadas para soldagem de 250 A com peso de 1.000 kg.

A corrente contínua normalmente era usada em aplicações com soldas do tipo arco até a década de 1920. Os eletrodos desenvolvidos com revestimento possibilitou o uso de corrente alternada. O transformador de soldagem logo se tornou muito popular, pois era mais barato e tinha menor consumo de energia.

No final da década dos anos 50, foram introduzidos retificadores de solda estáticos. Inicialmente foram usados de selenium seguidos por retificadores de silicone.

Os retificadores de tiristores que vieram depois, década de 1970, tornaram possível controlar eletronicamente a corrente de soldagem. Este tipo de retificador tem se tornado comum pelo menos para fontes de energia de soldagem maiores. No entanto, o desenvolvimento mais notável na fonte de energia de soldagem começou quando o inversor de soldagem foi desenvolvido em 1980.

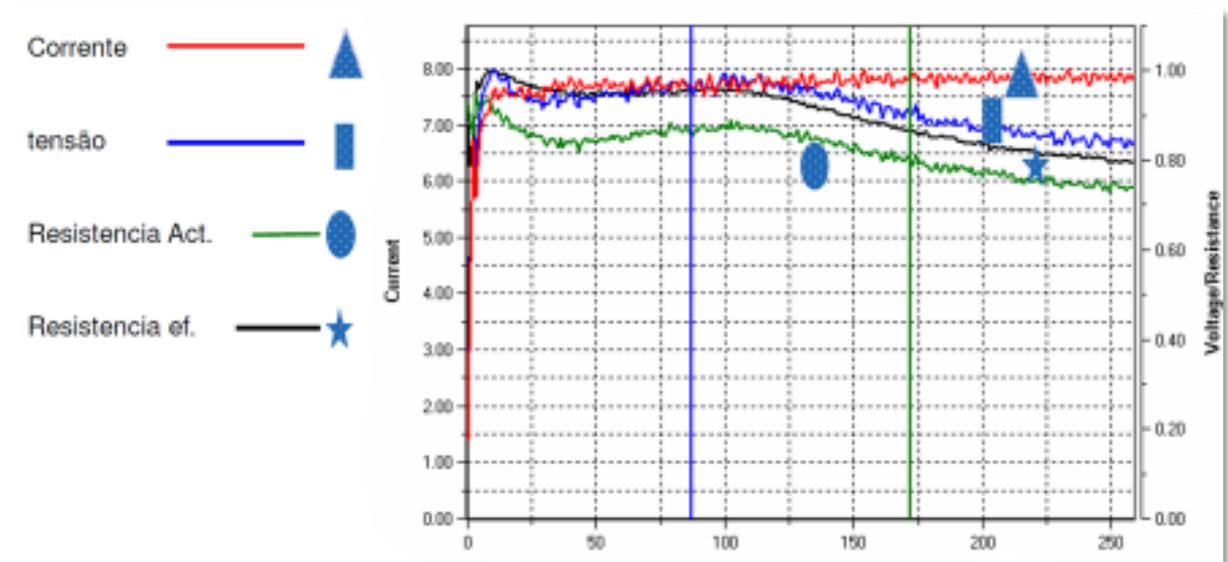
No processo de soldagem 4 parâmetros são avaliados para o correto dimensionamento, são eles: força, tempo, resistência e corrente. Estas variáveis podem ser avaliadas diretamente ou não, a corrente pode ser controlada pelo primário (indiretamente conhecendo a relação entre pólos) ou diretamente medindo em série a corrente do secundário. A força é a pressão aplicada pelo dispositivo de solda realiza em relação a peça, e o fator de tempo irá auxiliar no calculo de potencia entregue ao processo. A resistência é uma variável inerente ao processo e depende exclusivamente do material. A Figura 15, exemplifica graficamente um processo de soldagem, consolidando no mesmo instante de tempo o comportamento da corrente, tensão e resistência atual e efetiva (WEMAN, 2012).

#### **2.4.1 SOLDAGEM AUTOMOTIZADA**

Os primeiros robôs aplicados em operações de solda foram fornecidos pela Unimation à General Motors em 1964. Alguns anos depois a soldagem mecanizada aberta já estava buscando novas aplicações, por exemplo, a soldagem por arco submerso tornou-se um importante processo para soldagem em pátios de navios, fábricas de tubos e grandes vasos de pressão. A soldagem por intervalos estreitos economizou tempo, consumíveis e reduziu a distorção na soldagem de seções pesadas. A soldagem mecanizada TIG e plasma são usadas em aplicações de alta qualidade.

Quando um soldador experiente faz uma solda manual ele pode perceber se alguma parte do processo ou o equipamento não está operando como deveria, além de monitorar o processo, faz ajustes adequados quando necessário para garantir a qualidade da solda. A solda mecanizada, exemplificada pela Figura 16, por outro lado, repete fielmente e com precisão os movimentos programados, portanto é importante assegurar que o processo esteja bem parametrizado e que os fatores sejam mantidos suficientemente constantes para garantir o resultado esperado.

Figura 15 – Comportamento variáveis de um processo soldagem.



Fonte: software BOS 6000 – Rexroth.

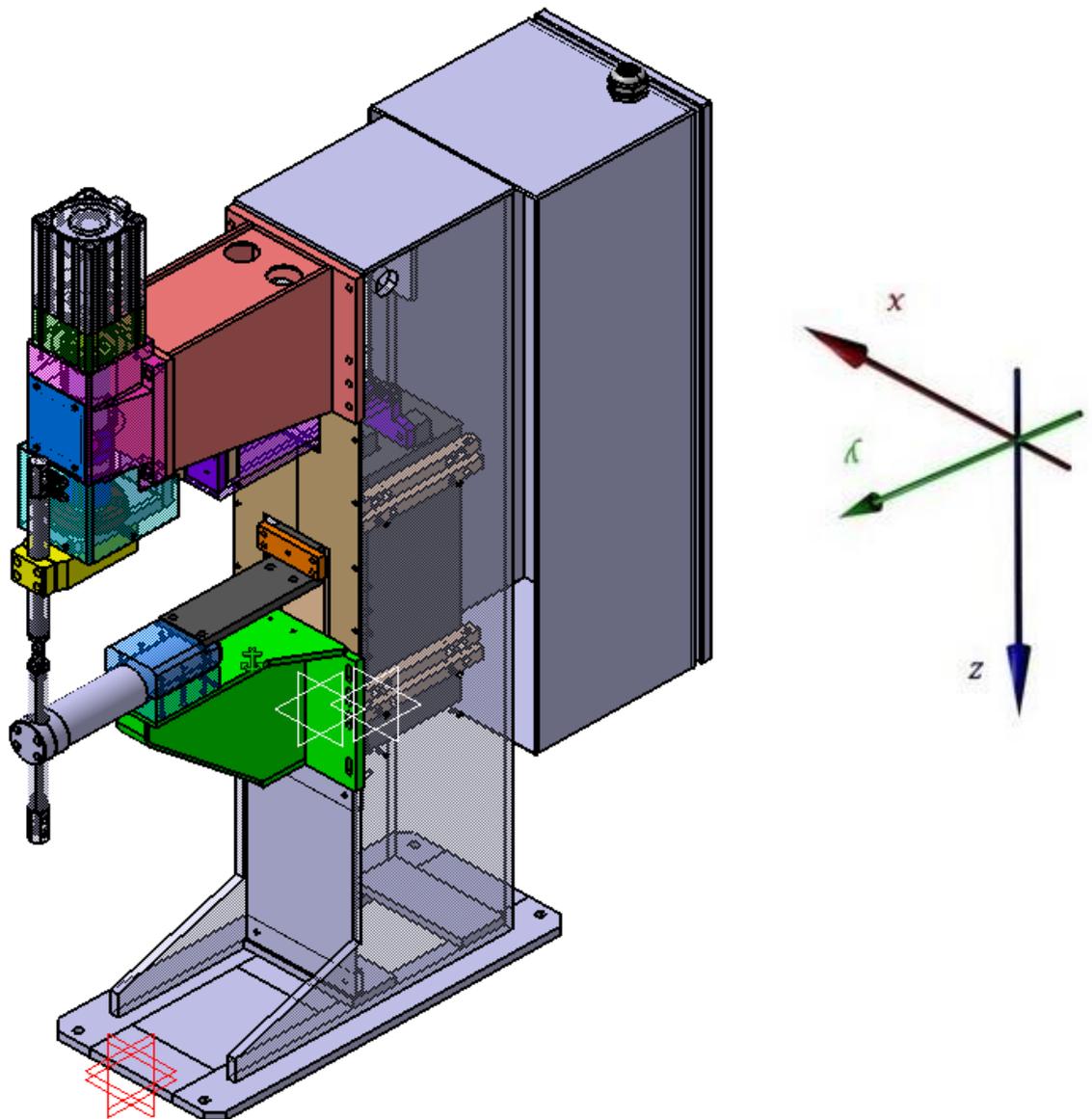
O princípio básico é que as tolerâncias que o processo de soldagem pode aceitar devem ser maiores do que quaisquer desvios que possam ocorrer, porém aumentar muito as tolerâncias tendem a diminuir a assertividade do movimento e com isso pode comprometer a qualidade da solda. Fatores que podem afetar a qualidade da solda: variações da largura na junta da solda, posição do eletrodo em relação à junta, problemas com alimentação de arame (quando houver deposição de material), projeto apropriado e escolha do tipo de solda, posição de soldagem mais apropriada, manutenção preventiva, equipamento de solda para controlar e manter valores pré-definidos, uso de equipamentos para monitoramento e rastreabilidade do traçado.

Os padrões de qualidade exigem monitoramento através de registro dos parâmetros relevantes, os parâmetros típicos incluem corrente, velocidade de deslocamento, tempo de permanência nos pontos a serem soldados. Para isso, o

equipamento deve estar conectado a um PC industrial e após definidos os limites apropriados, o equipamento pode gerar alarmes se qualquer um desses monitorados se alterar para fora dos intervalos.

Os alarmes podem ser gerados instantaneamente quando ocorrer uma passagem, ou podem ser feitos para responder a um sinal médio, de modo que breve máximos e mínimos sejam ignorados. Limites duplos também podem ser utilizados, de modo que ao atravessar o primeiro limite forneça um aviso, e quando atingir o segundo limite execute alguma ação no equipamento.

Figura 16 – Exemplo de solda por resistência em cartesiano.



Fonte: Software SolidWorks.

### 3 DESENVOLVIMENTO

Para facilitar o entendimento e a construção do projeto, optou-se por subdividir em 5 etapas, são elas: virtualização dos eixos, envio dos dados ao servidor, construção da aplicação no CLP, visualização de dashboard e parametrização de envio de alarmes.

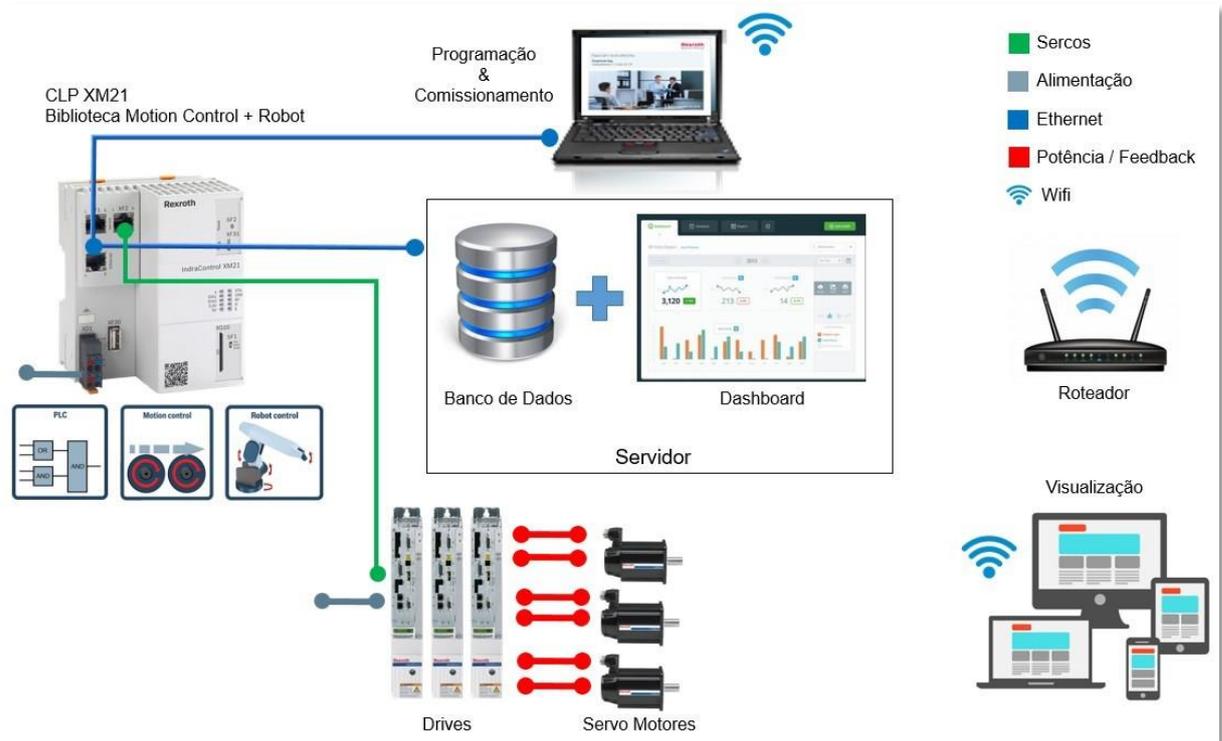
O fluxograma da Figura 17, apresenta de maneira detalhada os elementos necessários para o desenvolvimento e a forma de interligação entre os mesmos. O notebook será utilizado para realizar a programação do CLP através do software Indraworks Enginerring e interligados via Ethernet/IP. Nesta etapa será realizado em ambiente virtual uma aplicação de interpolação de eixos simulando um robô com 3 graus de liberdade, utilizando bibliotecas de Motion Control e Robot. A fim de validar aplicação um conjunto, composto por drive e servo motor, será interligado através da porta Field Bus do CLP, rede Sercos, estabelecendo a comunicação de controle do atuador. O drive será alimentado por tensão alternada 220Vca para fornecer ao atuador os pulsos controlados (PWM) a fim de tornar visível a movimentação de um dos 3 eixos da simulação. A conexão entre o drive e o servo motor ocorre via cabo de potência e a informação de posição do servo motor é proveniente de um encoder interno, que retorna ao drive através de um cabo de feedback, assim o atuador estabele conexão em malha fechada.

Ainda no ambiente do CLP, será necessária uma função para envio de parâmetros e dados que será escrito no servidor, neste caso será fisicamente o próprio notebook. O servidor será composto por um banco de dados e uma interface gráfica para geração de dashboard, este ainda informará o caminho (porta) pelo qual fará a comunicação com o roteador.

O elemento roteador será responsável por efetuar a ponte entre os dispositivos de visualização e o servidor. Estes dispositivos farão o acesso através de navegador WEB, podendo visualizar os parâmetros em tempo real e realizar comando de liga/desliga diretamente ao CLP. Nesta etapa será possível configurar alarmes de acordo com intervalos ou valores absolutos informados pelo usuário, estes serão enviados em forma de e-mail aos endereços previamente cadastrados. Outra função configurável é o escalonamento de alarmes, será possível monitorar aplicação e caso nenhuma ação tenha sido realizada, ou mesmo realizada a intervenção o alarme ou

falha persista, será possível atribuir um grupo de e-mails para escalonamento da informação até que a mesma seja resolvida.

Figura 17 – Topologia de funcionamento.



Fonte: Própria.

### 3.1 VIRTUALIZAÇÃO DOS EIXOS

O desenvolvimento da aplicação será o cenário com o qual irá representar a fonte de geração de dados, ou seja, será através desta o banco de dados será alimentado e possibilitará o envio e visualização remota.

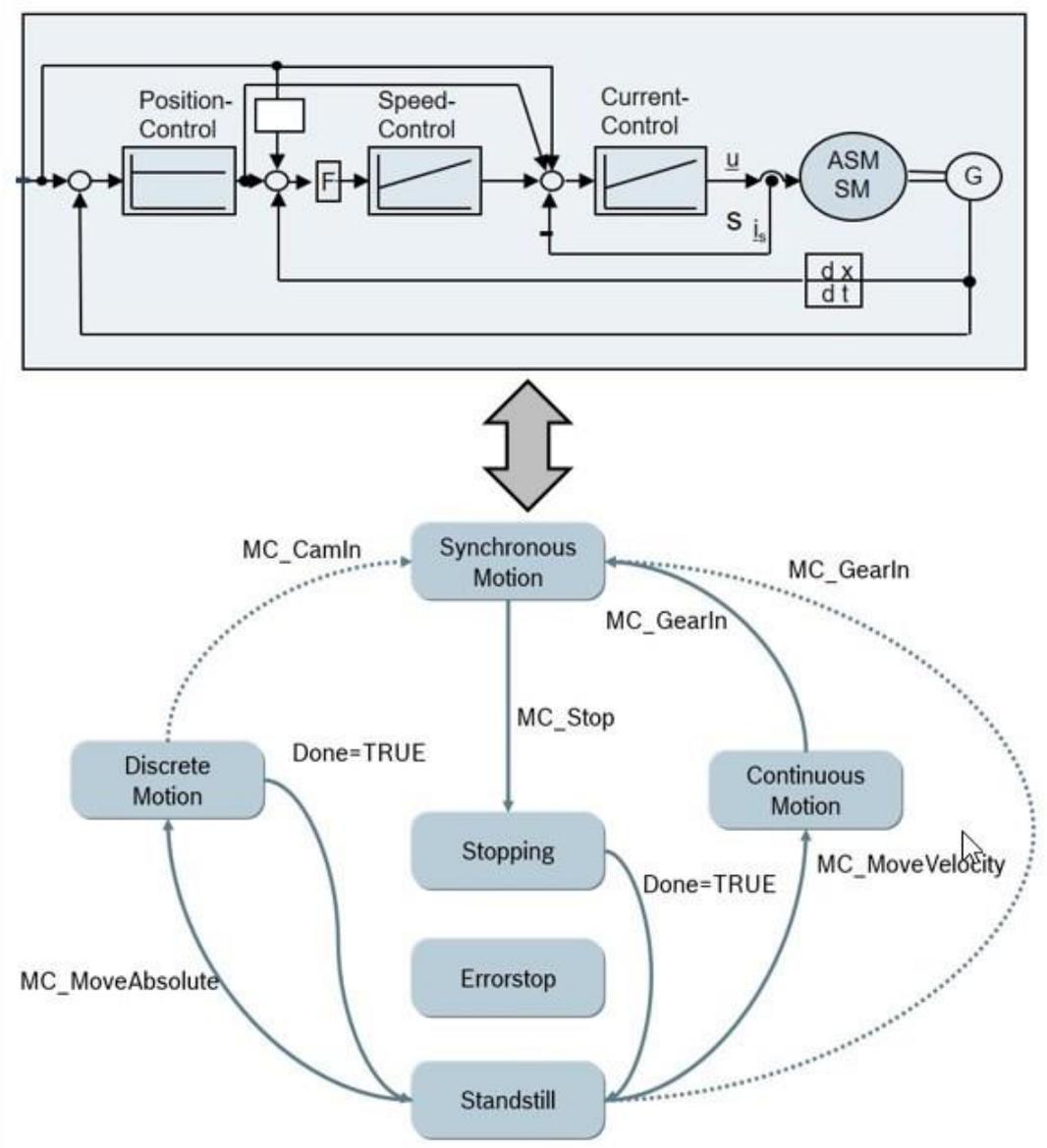
Neste projeto a contextualização será aplicação em um processo de solda ponto, pois neste modelo será possível controlar e medir a precisão, repetibilidade e o mais importante, a rastreabilidade dos mesmos.

A aplicação irá reproduzir um movimento interpolado de eixos sincronizados, princípio básico de funcionamento de um robô industrial, neste projeto será utilizado 3 eixos para interpolar um movimento coordenado de soldagem. O hardware escolhido possui bibliotecas próprias para realizar a montagem de programas para servo motores, conhecida como Motion Control e baseada em Function Block (FB). Nesta

plataforma é possível gerar controle de posicionamento e velocidade para eixos reais e virtuais, movimentos sincronizados através de engrenamentos ou CAM (Computer Aided Manufacturing) além de interpolar movimentos coordenados para aplicações robotizadas ou de controle numérico.

A Figura 18 ilustra a conversão do controle de um servo motor e a respectiva FB do CLP, trata-se de uma tradução do controle em malha fechada para linguagem de máquina, conforme preconizado na IEC 61131-3. O intuito é simplificar a aplicação, fornecendo ao programador os parâmetros principais de controle e automatizando as funções de transferência, ganhos, derivações, integrações e comparações.

Figura 18 – Tradução diagrama malha fechada para FB.

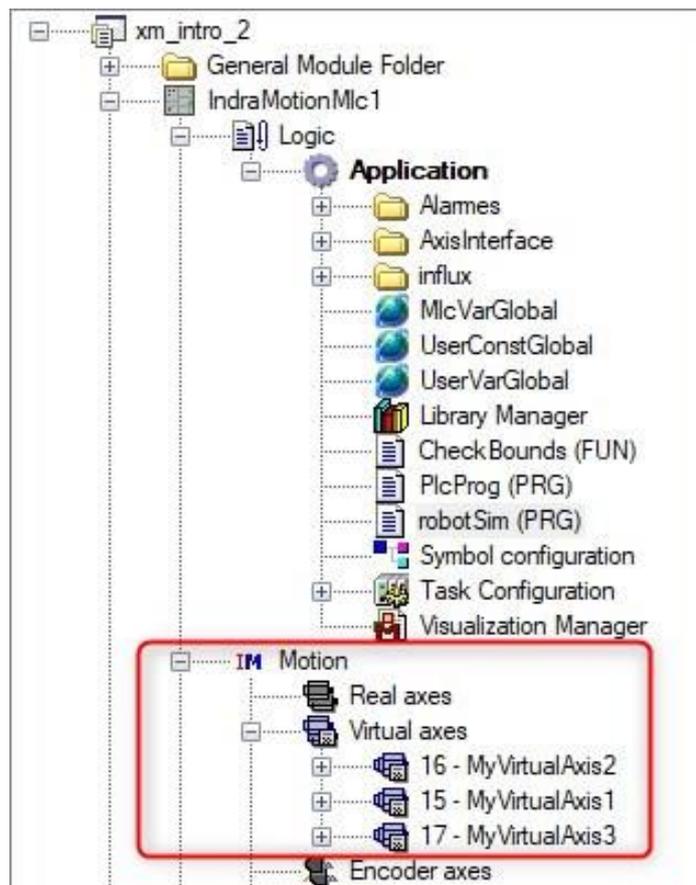


Fonte: Indraworks Engineering.

O servo drive é elemento que recebe os comandos do controlador (CLP), converte a alimentação CA para CC, gera o respectivo controle através pulsos controlados (PWM), e retorna ao atuador o sinal modulado convertido em CA novamente. Além de controlar o nível de potência entregue a carga, este realiza comparações periódicas entre o comando enviado e o feedback do encoder. Esta informação ocorre através da leitura da posição do sensor e derivando-a para obter informação de velocidade, as quais realimentam a malha. A este sistema denomina-se malha fechada, a qual garante precisão, potência, repetibilidade e rastreabilidade.

O programa intitulado “robotSim()” representa o sequenciamento de funções necessárias para a interpolação dos eixos virtuais. Os eixos são instanciados a partir da criação de objetos, utilizando a biblioteca de Motion, na Figura 19 pode-se observar que os eixos estão dentro da estrutura do projeto, nomeados de como: 15 – MyVirtualAxis1, 16 – MyVirtualAxis2 e 17-MyVirtualAxis3, eixos 1, 2 e 3, respectivamente.

Figura 19 – Estrutura criação de objetos.



Fonte: Indraworks Engineering.

Para esta etapa foram consideradas as variáveis conforme a Figura 20 do tipo inteiro para os parâmetros de posição e velocidade, inicializando com um valor pré-definido.

O programa estruturado em “switch case” permite que somente após finalizado o passo anterior o próximo seja executado, criando assim uma lógica de funcionamento das FBs e possibilitando um movimento sincronizado dos eixos virtuais.

Figura 20 – Programa Interpolação de eixos – robotSim.

```

1  PROGRAM robotSim
2
3  VAR
4      stepRobotSim: INT := 99;
5      rVelocityAxisZ: REAL := 1000;
6      rDesiredPositionAxisZ: REAL := 0;
7      rInitialPositionAxisZ: REAL := 70;
8      rDesiredPositionAxisX: REAL := 400;
9      rVelocityAxisX: REAL := 1000;
10     rDesiredPositionAxisY: REAL := 140;
11     rInitialPositionAxisX: REAL := 0;
12 END_VAR
13
14 //maquina de estados
15 CASE stepRobotSim OF
16 0: // Desce eixo Z
17     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Position := rDesiredPositionAxisZ;
18     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Velocity := rVelocityAxisZ;
19     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Acceleration := rVelocityAxisZ;
20     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Deceleration := rVelocityAxisZ;
21     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].Admin._OpMode.en := ModePosAbs;
22     stepRobotSim := stepRobotSim + 1;
23
24 1:
25     IF arAxisStatus_gb[MyVirtualAxis3.AxisNo].Admin.CmdDone THEN
26         stepRobotSim := stepRobotSim + 1;
27     END_IF
28
29 2: //Sobe eixo Z
30     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Position := rInitialPositionAxisZ;
31     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Velocity := rVelocityAxisZ;
32     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Acceleration := rVelocityAxisZ;
33     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].PosMode.Deceleration := rVelocityAxisZ;
34     arAxisCtrl_gb[MyVirtualAxis3.AxisNo].Admin._OpMode.en := ModePosAbs;
35     stepRobotSim := stepRobotSim + 1;
36
37 3:
38     IF arAxisStatus_gb[MyVirtualAxis3.AxisNo].Admin.CmdDone THEN
39         stepRobotSim := stepRobotSim + 1;
40     END_IF
41
42 4: //Movimenta X ( Y em sincronizado)
43     arAxisCtrl_gb[MyVirtualAxis1.AxisNo].PosMode.Position := rDesiredPositionAxisX;
44     arAxisCtrl_gb[MyVirtualAxis1.AxisNo].PosMode.Velocity := rVelocityAxisX;
45     arAxisCtrl_gb[MyVirtualAxis1.AxisNo].PosMode.Acceleration := rVelocityAxisX;
46     arAxisCtrl_gb[MyVirtualAxis1.AxisNo].PosMode.Deceleration := rVelocityAxisX;

```

Fonte: Indraworks Engineering.

Na Figura 20, trecho do programa “robotSim” é possível verificar que o contador responsável pelo passo “stepRobotSim” inicializado anteriormente com 99, linha 79 do programa. Este é o passo de inicialização das variáveis dos eixos, ou seja, nela são associadas as variáveis criadas anteriormente e inicializa os valores fixos de aceleração e desaceleração dos 3 eixos. Posteriormente os eixos buscam sua referencia de posição para iniciar os movimentos e assim a variável do passo assume o valor 0.

Neste momento é processado o bloco lógico “CASE 0”, comentado no programa como “Desce eixo z”, esta etapa é chamado a primeira FB, no qual é identificado o eixo a ser movimentado, escolhido o tipo de controle, no caso – posição, e informado para assumir o valor da variável inicializada anteriormente “rDesiredPositionAxisZ” que na primeira etapa do programa é o valor inteiro 70.

Posteriormente outras FBs são chamadas parametrizando o valor de velocidade, aceleração, desaceleração e modo de operação, respectivamente do eixo Z. A linha de comando número 9 faz o incremento na variável utilizada na função “CASE” e o programa assume a segunda etapa, a qual faz um teste lógico do tipo “IF” comparando a posição final do eixo com a solicitada no passo anterior, caso sejam iguais, é acrescentado novo inteiro a variável “stepRobotSim” e o programa segue com o próximo movimento.

Os programas subsequentes possuem a mesma estrutura, ou seja, possuem as mesmas variáveis e mesmas sintaxes, trocando apenas a variável de controle (eixo 1, 2 ou 3, ou ainda eixo x, y e z) e seus valores.

Quando executado o programa é possível visualizar no painel de variáveis, Figura 21, os 3 eixos representados em azul e um traço preto movendo-se circularmente. Nesta tela é possível realizar alguns comandos como reinicializar aplicação e/ou o eixo, através do botão “HOME”, desligar eixos individualmente, botão “ENABLE”, parametrizar valores de velocidade e aceleração por eixo, verificar a posição real, além de acompanhar as mensagens de falhas e erros nos campos, “Warning” e “Error”, respectivamente.

Em um processo de solda robotizado, há algum dispositivo que faz referência ao mapa de soldas, quando há mais de um tipo, ou seja, existe alguma entrada ao sistema, seja ele associado um leitor de código de barras ou QR Code para chamar o respectivo programa. Com ajuda de uma função contador incremental é possível monitorar o número de vezes que o mesmo foi executado, associando ao processo

um valor que se torna referência para um conceito de rastreabilidade se assim o usuário necessitar.

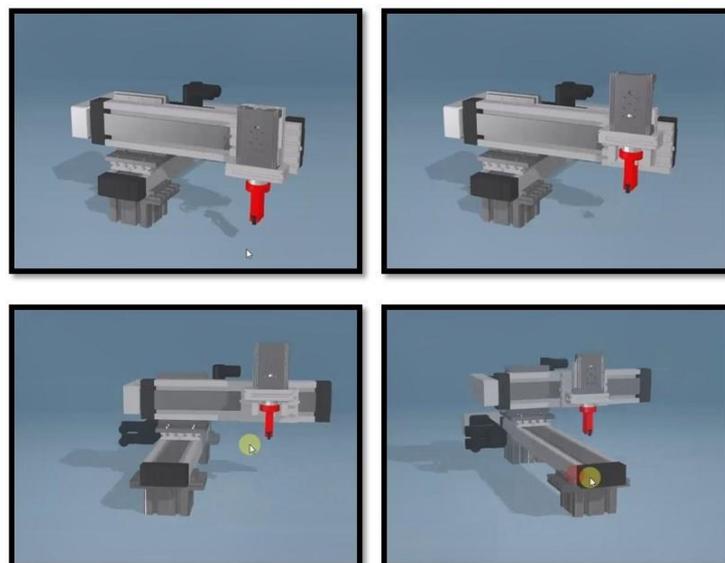
O CLP possui um ambiente virtual, conforme verificado na Figura 22 que permite visualizar de graficamente o funcionamento dos eixos virtuais, este apesar de limitado em relação ao detalhamento da aplicação, possibilita uma visão interativa da programação via Motion Control.

Figura 21 – Painel de visualização eixos virtuais.

Axis Details / Diagnosis		Status	Position Velocity	SetupMode
<< Axis: AXIS_15 <b>Active</b> ErrorID: NONE_ERROR MyVirtualAxis1 ErrorTable: NO_TABLE_USED Warning Error Axis Type: VIRTUAL ErrorAdd1: 1690 ErrorAdd2: 1690 Diagnosis: Axis is in '\$Discrete Motion\$'		Status: ModePosAbs BB AB AH AF In Ref Flex Profile: Set0 OK Set1 OK Set2 OK Set3 OK Flex sync active	Position: 348.61 Velocity: 10000.00	Setup Mode: Enable Vel: 10 Jog+ Accel: 10 Jog- Home
<< Axis: AXIS_16 <b>Active</b> ErrorID: NONE_ERROR MyVirtualAxis2 ErrorTable: NO_TABLE_USED Warning Error Axis Type: VIRTUAL ErrorAdd1: 1690 ErrorAdd2: 1690 Diagnosis: Axis is in '\$Synchronized Motion\$'		Status: ModeSyncPhase BB AB AH AF In Ref Flex Profile: Set0 OK Set1 OK Set2 OK Set3 OK Flex sync active	Position: 156.66 Velocity: 3500.00	Setup Mode: Enable Vel: 10 Jog+ Accel: 10 Jog- Home
<< Axis: AXIS_17 <b>Active</b> ErrorID: NONE_ERROR MyVirtualAxis3 ErrorTable: NO_TABLE_USED Warning Error Axis Type: VIRTUAL ErrorAdd1: 1690 ErrorAdd2: 1690 Diagnosis: Axis is in '\$StandStill\$'		Status: ModePosAbs BB AB AH AF In Ref Flex Profile: Set0 OK Set1 OK Set2 OK Set3 OK Flex sync active	Position: 70.00 Velocity: 0.00	Setup Mode: Enable Vel: 10 Jog+ Accel: 10 Jog- Home

Fonte: Indraworks Engineering.

Figura 22 – Eixos Virtuais 3D.

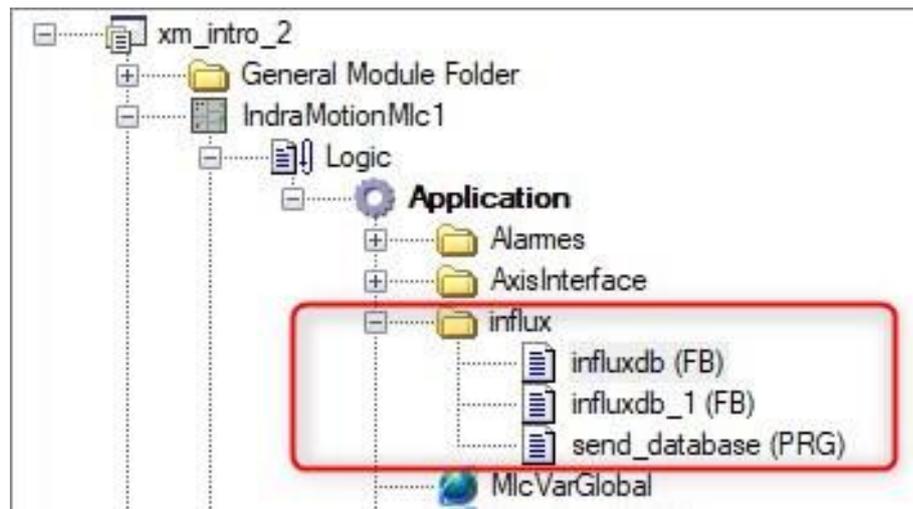


Fonte: Indraworks Engineering.

### 3.2 ENVIO DOS DADOS AO SERVIDOR

A função de envio dos dados é estruturada pela seção “influx” criada na raiz da estrutura da aplicação. O influx, observado na Figura 23 é composto por um programa que estrutura a informação e outros 2 programas que formam o pacote de dados a ser enviado e endereçam ao banco de dados do servidor.

Figura 23 – Estrutura do Influx.



Fonte: Indraworks Engineering.

Os programas “infludb” e “influxdb\_1” são similares, diferindo apenas em relação a estrutura de envio, enquanto o primeiro exporta valores inteiros o segundo envia variáveis do tipo string. Ambos utilizam FB para criar os programas, no campo de declaração de variáveis, Figura 24 é associado os elementos do programa dos eixos virtuais, valores inteiros e algumas variáveis locais do tipo string.

Na seção do programa é chamada a função “udpSocket” que reúne as informações necessárias para gerar um padrão de informações que será utilizada na escrita no banco de dados. Nesta função é imprescindível informar o IP fixo do servidor, pois este será o caminho no qual o CLP buscará a comunicação para envio, neste projeto o IP do servidor é 192.168.234.233, pois o CLP está configurado com o IP 192.168.234.234. Além do IP outra informação para envio é a porta, neste caso a 8089 será utilizada.

A variável de entrada do programa, “strMessage”, irá concetendar através do comando “CONCAT” uma string de nome e uma de valor, alocando no endereço de

envio. Posteriormente um teste lógico com a função verificando se a mesma foi enviada, devolvendo ao parâmetro “Enable” verdadeiro ou falso.

Figura 24 – Estrutura Influx DB.

```

1  FUNCTION BLOCK influxdb
2  VAR
3      UdpSend: IL_UDPSendtoAsync;
4      udpSocket: IL_UDPsocketAsync;
5      udpStep: INT;
6  END_VAR
7  VAR_INPUT
8      Value: STRING;
9  END_VAR
10 VAR
11     Handle: UDINT;
12     udpInit: IL_UDPInitialAsync;
13 END_VAR
14 VAR_INPUT
15     Name: STRING;
16     strMessage: STRING;
17 END_VAR
18
19 udpSocket(
20     Enable:= TRUE,
21     Server:= FALSE,
22     Address:= '192.168.234.233',
23     Port:= 8089,
24     Done=> ,
25     Active=> ,
26     Error=> ,
27     ErrorID=> ,
28     ErrorIdent=> ,
29     Socket=> Handle );
30
31 strMessage := CONCAT(Name, ' value=');
32 strMessage := CONCAT(strMessage, Value);
33
34 UdpSend.Address := udpSocket.Address;
35 UdpSend.Port := 8089;
36 UdpSend.Socket := Handle;
37 UdpSend.ValueAdr := ADR(strMessage);
38 UdpSend.NoOfBytes := len(strMessage);
39
40 CASE udpStep OF
41 0:
42     IF udpSocket.Done THEN
43         UdpSend := ...

```

Fonte: Indraworks Engineering.

O programa “send\_database” estrutura a informação através de dois arrays, inicialmente parametrizado com 11 posições cada, os dados provenientes dos programas “influxdb” e “influxdb\_1”, conforme Figura 25.

Dentro do array “sendInflux”, criado para armazenar os dados do “influxdb”, o trecho do programa da Figura 25, ilustra os dados inteiros referente a posição, velocidade e torque, do eixo 1, armazenados nas posições 0, 1 e 6, respectivamente.

O segundo array “sendInfluxS” irá receber as strings provenientes do programa “influxdb\_1” são as informações de status do eixo 1, na Figura 25 – parte do programa, estão sendo armazenadas na posição 0 e 3.

Figura 25 – Estrutura send\_database.

```

1  PROGRAM send_database
2
3  VAR
4      sendInfluxS: ARRAY[0..10] OF influxdb_1;
5      sendInflux: ARRAY[0..10] OF influxdb;
6  END_VAR
7
8
9  sendInflux[0].Value := REAL_TO_STRING(arAxisStatus_gb[15].AxisData_HMI.rActualPosition_i);
10 sendInflux[0].Name := 'posicao_ax_1';
11 sendInflux[0]();
12
13 sendInflux[1].Value := REAL_TO_STRING(arAxisStatus_gb[15].AxisData_HMI.rActualVelocity_i);
14 sendInflux[1].Name := 'velocidade_ax_1';
15 sendInflux[1]();
16
17 sendInflux[6].Value := REAL_TO_STRING(arAxisStatus_gb[15].AxisData_HMI.rActualTorqueForce_i);
18 sendInflux[6].Name := 'torque_ax_1';
19 sendInflux[6]();
20
21 sendInfluxS[0].Value := (arAxisStatus_gb[15].Diag.Message);
22 sendInfluxS[0].Name := 'stats_ax_1';
23 sendInfluxS[0]();
24
25 sendInfluxS[3].Value := (ImcStatus.Diag.Message);
26 sendInfluxS[3].Name := 'stats_ctrl_1';
27 sendInfluxS[3]();

```

Fonte: Indraworks Engineering.

### 3.3 CONSTRUÇÃO DA APLICAÇÃO NO CLP

Através dos programas descritos anteriormente é possível realizar a interpolação dos eixos, gerar dados e envia-los ao DB, para isso criou-se a aplicação intitulada “PlcProg”, este é o programa principal, nele algumas variáveis são

declaradas e inicializadas. Na Figura 26 é possível verificar que este faz chamadas aos programas criados anteriormente, robotSim() e send\_database(), além do exemploAlarme(). Este ultimo tem a função de forçar falhas na simulação dos eixos interpolados, as quais resultaram em alarmes, neste projeto foram adotadas falhas de posição, velocidade, tempo e trajetória.

A pasta “Alarmes”, Figura 27, possui dois programas, o “ativaAlarme” é a criação do objeto orientado, no qual é atribuído um valor inteiro, uma string para descrição do alarme e por ultimo uma variável de controle do tipo Booleana. No programa “exemploAlarme” é associado o objeto criado anteriormente com a função do programa na qual ele será solicitado.

Figura 26 – Programa Principal.

The image shows the Siemens SIMATIC Manager interface. On the left, the Project Explorer displays a tree structure under 'xm\_intro\_2'. The 'Application' folder is expanded, showing sub-folders like 'Alarmes', 'AxisInterface', and 'influx'. The 'PlcProg (PRG)' file is highlighted with a red box and labeled 'Programa Principal'. On the right, the code editor shows the main program 'ativaAlarme'. The code includes variable declarations for 'bVM\_Enable', 'rVM\_Acc', 'rVM\_Dec', 'rVM\_Velocity', 'rPhaseShift', 'bVM\_VelModeAck', 'bVM\_Standstill', 'rVM\_actVelocity', 'Axis1ActPosition', 'Axis2ActPosition', and 'Axis3ActPosition'. It also includes function calls: 'TE\_AxisInterfaceType01Prog()', 'MB\_ImcInterface()', 'exemploAlarme()', 'robotSim()', and 'send\_database()'. A red box highlights the variable declarations, labeled 'Variáveis do programa principal'. Red arrows point from the text 'programas pré definidos do sistema' to the 'TE\_AxisInterfaceType01Prog()' call. Blue arrows point from the text 'programas criados para aplicação' to the 'exemploAlarme()', 'robotSim()', and 'send\_database()' calls.

Fonte: Indraworks Engineering.

Figura 27 – Programa Principal.

The image displays a software development interface. On the left, the 'Project Explorer' window shows a project structure under 'xm\_intro\_2'. The 'Application' folder is expanded, and the 'Alarma' folder is highlighted with a red rectangle. Inside 'Alarma', there are two items: 'ativaAlarma (FB)' and 'exemploAlarma (PRG)'. The main editor window shows the code for the 'ativaAlarma' function block. The code is as follows:

```

1 FUNCTION_BLOCK ativaAlarma
2 VAR_INPUT
3     iCodigo: INT;
4     bAlarma: BOOL;
5     stErroUsuario: STRING;
6 END_VAR
7
8 VAR
9     ativaAlarma: R_TRIG;
10    rearmaAlarma: R_TRIG;
11 END_VAR
12
13 IF ativaAlarma.Q OR rearmaAlarma.Q THEN
14     ML_SetDiagEntry(
15         Severity:= E_LOG_SEVERITY.ERROR,
16         Class:= E_LOG_CLASS.CLASS_0,
17         Facility:= E_LOG_FACILITY.USER,
18         Code:= iCodigo,
19         Text:= ADR(stErroUsuario));
20 END_IF
21
22 ativaAlarma(CLK:= bAlarma, Q=> );
23
24 rearmaAlarma(CLK:= bAlarma AND NOT(ImcStatus.Diag.Error), Q=>);

```

Fonte: Indraworks Engineering.

### **3.4 VISUALIZAÇÃO DE DASHBOARD**

A geração do dashboard será estruturada em uma plataforma chamada Grafana, esta é dedicada a visualização e análise de métricas por meios gráficos. A ferramenta é leve e flexível, pois permite agrupar variáveis distintas no mesmo gráfico facilitando a interpretação dos dados.

Os dashboards são dinâmicos podendo selecionar tempo de amostragem e intervalos entre consultas no banco de dados. Dependendo do tempo de armazenamento do banco de dados é possível selecionar determinado período para uma análise pontual em busca de discrepâncias. O acesso do Grafana ao banco de dados é realizado através de um plugin executável.

Os gráficos são gerados em tempo real, com taxa de atualização a cada segundo, apesar do controlador executar o escaneamento em 200 ms, e a escrita no banco de dados ocorrer na mesma taxa, o dashboard atualiza com período mínimo de 1 segundo.

### **3.5 PARAMETRIZAÇÃO DE ENVIO DE ALARMES**

Uma ferramenta do Grafana é a configuração de alertas com base nas métricas que são analisadas, ou seja, através de parametrizações é possível notificar determinado endereço de email e/ou grupo de e-mails que determinada grandeza está atingindo valores fora dos intervalos pré-definidos, ou que por um pequeno intervalo de tempo ultrapassou um determinado limite.

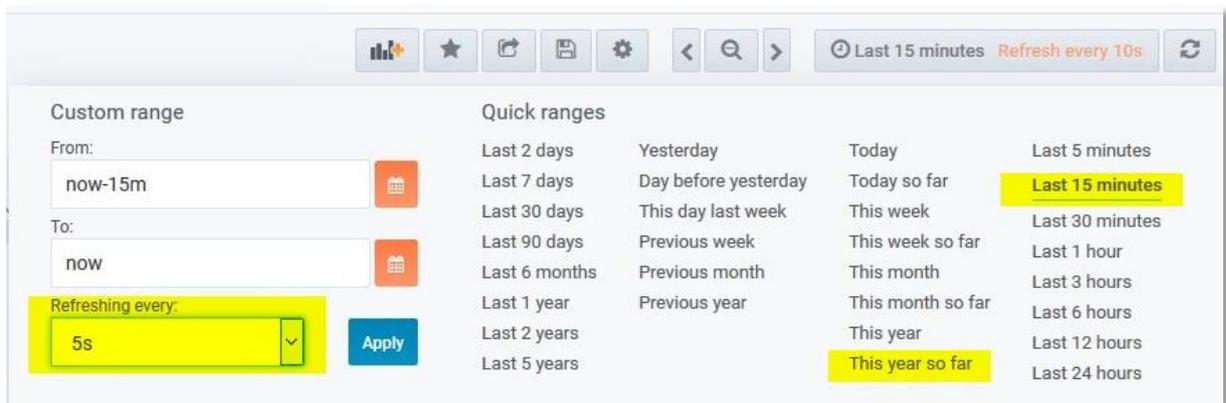
Essa parametrização pode ser associada aos valores extremos (máximos e mínimos) da aplicação e pode ser integrada aos alarmes criados no próprio CLP.

## 4 RESULTADOS

Através de qualquer dispositivo com acesso a rede wifi – conectado na mesma rede e um navegador WEB é possível inserir o endereço localhost:3000 para visualizar o dashboard gerado pelo aplicativo Grafana. Neste é possível parametrizar qual é a taxa de atualização no qual o aplicativo fará acesso ao banco de dados do servidor, conforme verificado na Figura 28, o parâmetro pode ser alterado no campo “refreshing every”, neste projeto adotado 5 segundos, mas pode variar de 1 segundo a 1 dia.

Outro parâmetro importante é a faixa de visualização, ou seja, determinar qual o intervalo que gostaria de visualizar, neste projeto, utilizou-se os últimos 15 minutos para não poluir os gráficos, mas é possível obter períodos superiores a 1 ano de dados para visualizar.

Figura 28 – Taxa de atualização Grafana.



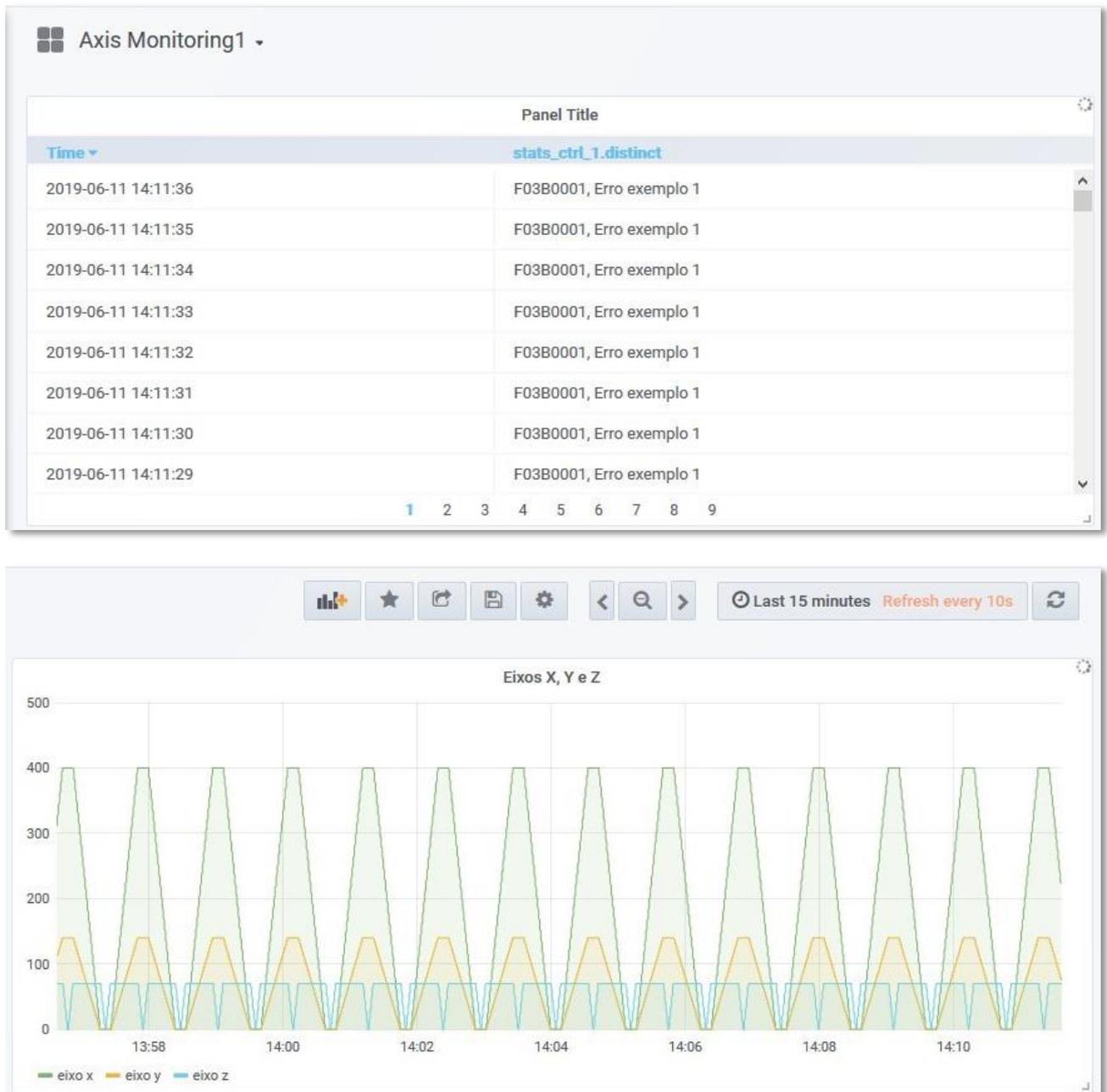
Fonte: Aplicativo Grafana.

Os gráficos e tabelas representados nas próximas figuras são configuráveis e podem gerar um dado específico, ou agrupar um grupo de dados. A associação dos dados e sua respectiva interpretação fornece informação suficiente para que o usuário que os visualiza possa tomar ações de corretivas, por exemplo: um desvio de trajetória por causa de uma colisão, aumento de tempo dos eixos na posição para melhorar o tempo de exposição a solda, etc.

A Figura 29 é a tela principal do dashboard, nela é apresentado uma tabela intitulada “Panel” o qual mostra em tempo real (taxa de atualização de 1 segundo) o status da simulação no CLP. No projeto foi criado funções para forçar erros na simulação, no momento da captura da tela o eixo 1 estava em falha, por isso informado

no Panel o código e o respectivo nome da falha, declarado na pasta de alarmes no programa do CLP. O gráfico abaixo da mesma figura, mostra a variável posição de todos os eixos interpolados no mesmo tempo, na legenda em verde (eixo 1 ou eixo x) é o sinal com maior amplitude (400mm), em amarelo (eixo 2 ou eixo y) o sinal com amplitude mediana (150mm) e em azul (eixo 3 ou eixo Z) o sinal de menor amplitude (70mm).

Figura 29 – Informação de posição dos eixos agrupada.



Fonte: Aplicativo Grafana.

A quantidade de dados para visualização depende do tipo de análise que o usuário pretende realizar, é possível reunir todos os dados provenientes do conjunto de servo acionamento (velocidade, aceleração, desaceleração, ângulo, torque, corrente, tensão, posição relativa, posição absoluta, JOG, home, etc.) e quando necessário, integrar sensores externos (como por exemplo: temperatura, encoder, magnéticos, fim de curso, contato seco) e utilizando as entradas analógica ou digital do CLP aumentar o fluxo de dados, permitindo que a informação obtida contemple o maior número de variáveis.

A criação do gráfico é a parte mais importante para o dashboard, pois este permite que usuário apresente de forma gráfica seus dados e forneça argumentos para sustentar uma condição de funcionamento, realize testes de comissionamento, teste qualidade, realize análises preventivas ou até corretivas.

Na plataforma do Grafana a criação do gráfico implica em consultar um banco de dados, neste caso será o Influxdb, este criado na tela de programação do CLP e armazenado na memória do computador conectado a ele (servidor local). A

Figura 30 exemplifica a criação do gráfico do eixo 1, na aba Metrics deve primeiramente informar o nome do banco de dados, posteriormente apontar endereço da variável no campo "From" e selecionar qual recurso que irá escrever no gráfico, neste caso os últimos valores, dado por "field (Value)" e "last". O próximo passo é determinar qual a relação métrica do eixo y com eixo x, neste caso será uma análise baseada no tempo, indicada por "time(S\_interval)", seguida formato como série temporal e atribuo um nome para a série criada.

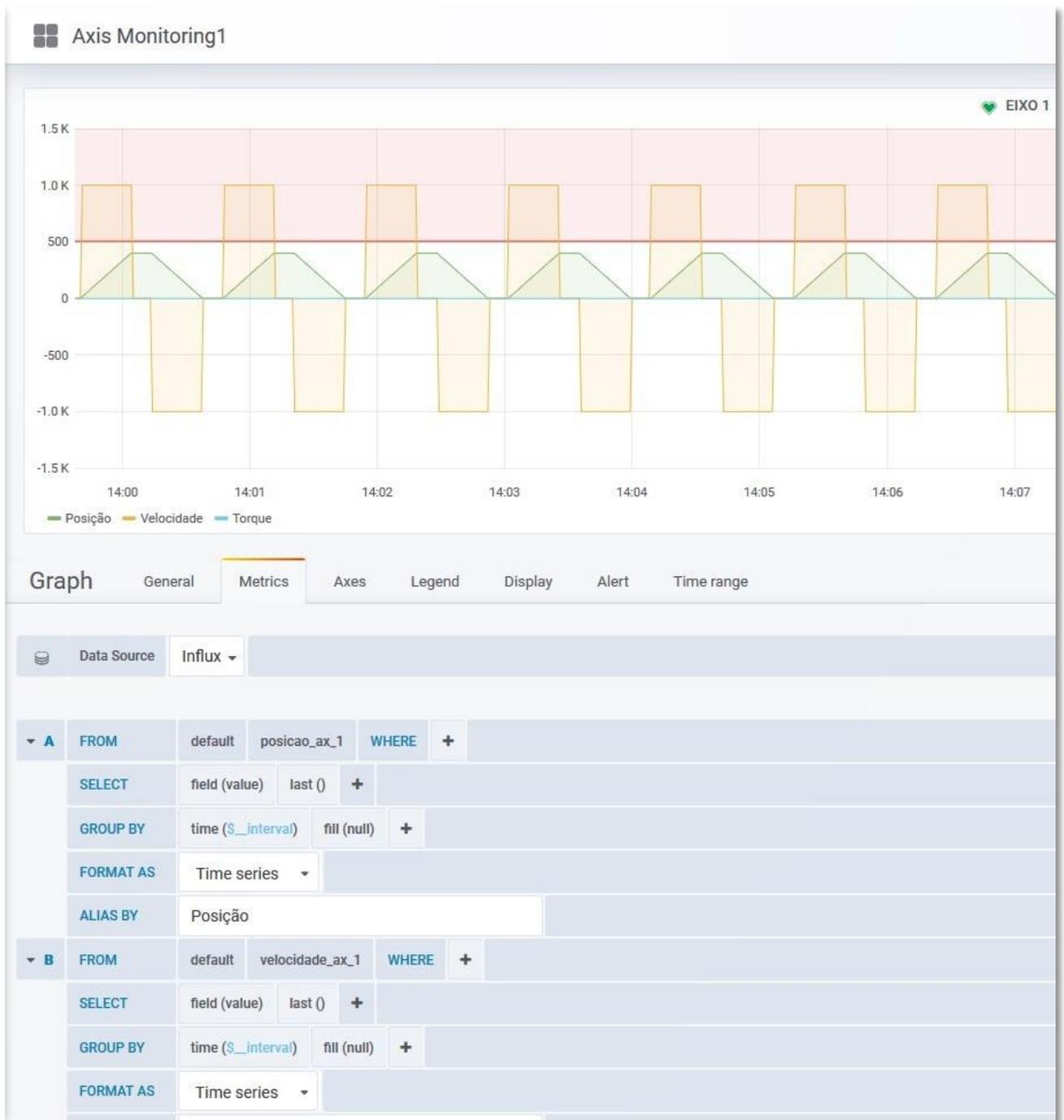
Repetindo a configuração anterior foi criado o gráfico com dados de posição, velocidade e torque, verde, amarelo e azul, respectivamente, mudando apenas o endereço do ponteiro.

Utilizando os recursos disponíveis, e nos baseando na aplicação em um robô que realiza solda ponto, no qual as informações úteis são: trajeto realizado, tempo de espera na posição final, frequência, repetibilidade e rastreabilidade, as variáveis de controle são, posição, velocidade e torque de cada eixo, conforme observado na Figura 31. Algumas variáveis indiretas são fornecidas, realizando a integração da velocidade em relação ao tempo, obtém-se aceleração e desaceleração.

Na Figura 31 é possível notar que o eixo 1, ou x, apresenta uma linha contínua no valor absoluto de 500, este é o alarme criado pelo Grafana no qual monitora a posição do eixo e se o mesmo exceder o valor parametrizado o aplicativo envia um

email indicando o erro para o endereço cadastrado. Na mesma imagem o painel do lado direito indica o status atualizado a cada segundo do respectivo eixo que encontra-se a esquerda. Portanto além da visualização gráfica de posição, velocidade e torque o painel atualiza em relação ao status do elemento de controle.

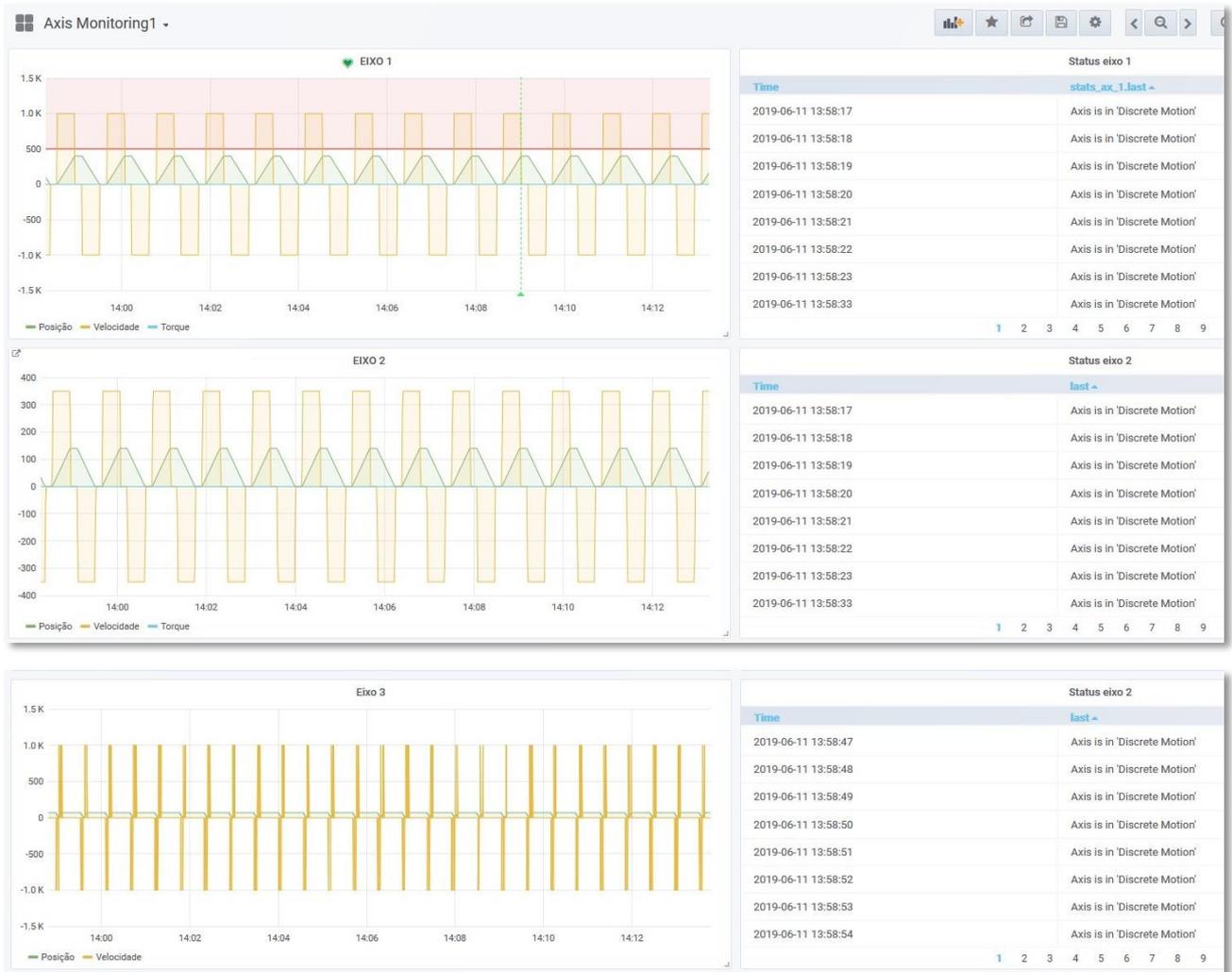
Figura 30 – Criação dos gráficos.



Fonte: Aplicativo Grafana.

Ao inserir um gráfico é possível atribuir funções de monitoração e alarme. A monitoração é exemplificada graficamente com uma linha contínua vermelha na Figura 32, pode ser parametrizada na aba “Alert” determinando um nome e um valor de amostragem, ou seja, se o valor parametrizado permanecer por 60 segundos ativo, configura a geração do alerta.

Figura 31 – Dashboard eixos.

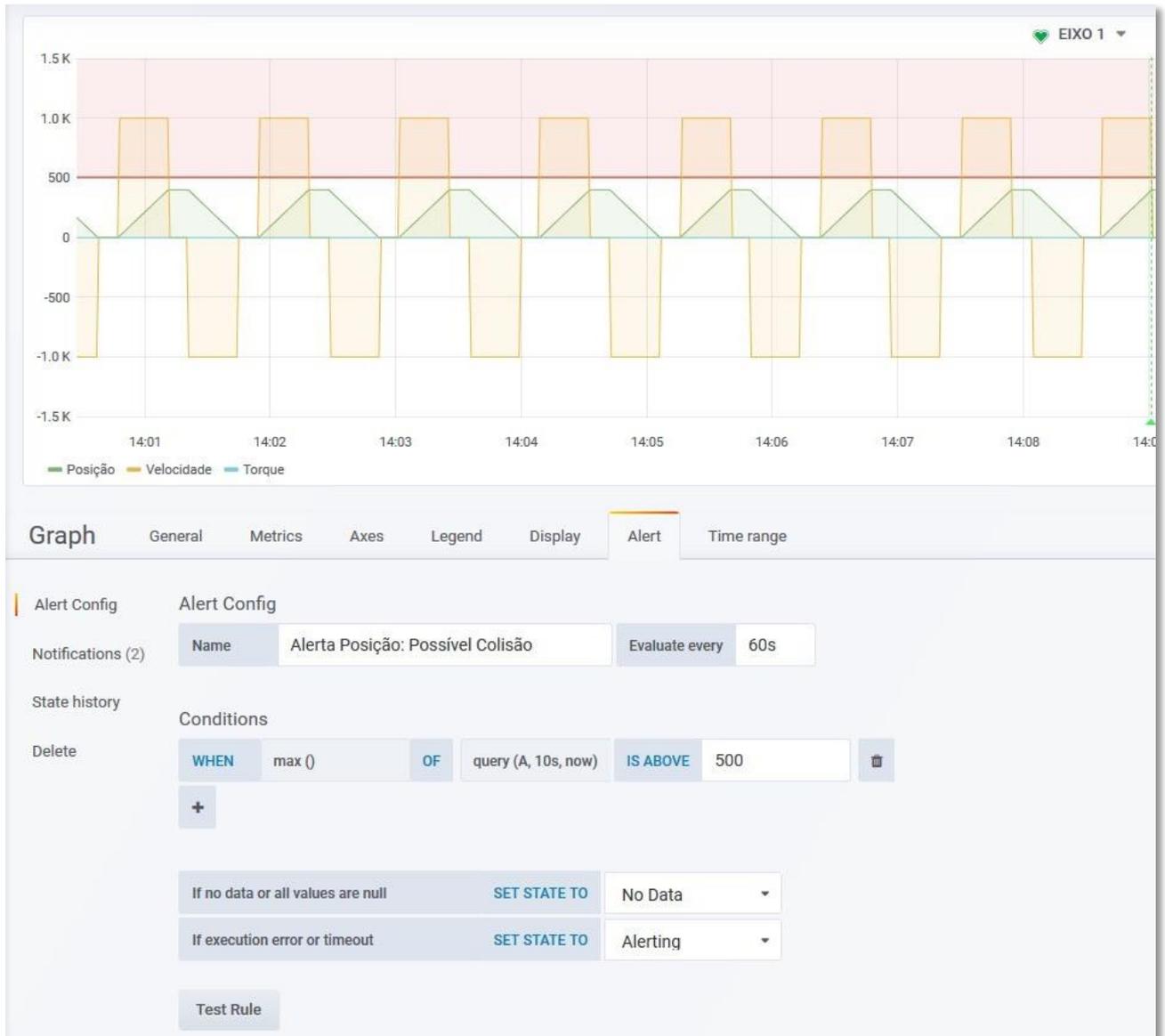


Fonte: Aplicativo Grafana.

Outro campo necessário são as condições em que o alarme irá ativar, para isso é necessário informar a lógica na qual se quer trabalhar, podendo escolher entre máximos e mínimos, faixa de valores e saturação. E por ultimo determinar o valor, no caso abaixo 500mm, gerando um alerta de posição.

No campo de “Alerting” é associado o tipo do alarme, criado anteriormente, ao meio de transmissão, neste caso a notificação seguirá via email com a imagem do gráfico no momento do erro para o endereço cadastrado, conforme Figura 33.

Figura 32 – Monitoração e Alarme.



Fonte: Aplicativo Grafana.

Ainda no campo “Alerting” é possível visualizar e gerenciar todos alarmes criados, permitindo gerar através de regra o escalonamento do erro no caso de reincidência ou permanecimento do status, conforme ilustrado na Figura 34.

No programa do CLP foi criada uma seção de alarmes, no qual é possível simular através das variáveis criadas alguns tipos de alarmes independentes. Na

Figura 35 é possível verificar através da tabela de valores, os estados das variáveis em tempo real durante execução online do CLP. Além das 4 variáveis de alarme, existe mais duas de estado e reset de erros. Através da coluna “Prepared value” é forçado a mudança nas variáveis de “0” para “1” ou de “false” para “true”.

Figura 33 – Criação o destinatário do alarme.

Fonte: Aplicativo Grafana.

Figura 34 – Gerenciamento de alarme.

Fonte: Aplicativo Grafana.

Figura 35 – Geração de alarmes através do CLP.

IndraMotionMlc1.Application.exemploAlarme					
Expression	Type	Value	Prepared value	Address	Comment
◆ N_ALARMES	INT	C 10			
⊕ ◆ alarme	ARRAY [0..N_ALAR...				
✖ ◆ bAlarme4	BOOL	FALSE			
✖ ◆ bAlarme3	BOOL	FALSE			
✖ ◆ bAlarme1	BOOL	FALSE			
✖ ◆ bAlarme2	BOOL	TRUE			
◆ bResetErros	BOOL	FALSE			
✖ ◆ bStatusErro	BOOL	TRUE			

```

1 //Exemplo de Alarmes de usuário usando FB
2 alarme[0] (bAlarme FALSE := bAlarme1 FALSE, iCodigo 0 := 0, stErroUsuario Erro exemp := 'Erro exemplo 0' );
3 alarme[1] (bAlarme TRUE := bAlarme2 TRUE, iCodigo 1 := 1, stErroUsuario Erro exemp := 'Erro exemplo 1' );
4 alarme[2] (bAlarme FALSE := bAlarme3 FALSE, iCodigo 2 := 2, stErroUsuario Erro exemp := 'Erro exemplo 2' );
5 alarme[3] (bAlarme FALSE := bAlarme4 FALSE, iCodigo 3 := 3, stErroUsuario Erro exemp := 'Erro exemplo 3' );
6
7
8 //É necessário usar o ImcCtrl para limpar os erros
9 ImcCtrl.Admin.ClearError FALSE := bResetErros FALSE;
10
11 //E também o ImcStatus para checar erros existentes
12 bStatusErro TRUE := ImcStatus.Diag.Error TRUE;
13
14 RETURN

```

Fonte: Software Indraworks.

## 5 CONCLUSÃO

Os conceitos de I4.0 são amplos, mas sempre envolvem sua principal matéria prima, os dados. Apesar de diariamente nos depararmos com o tema, não trata-se de uma tecnologia nova, ou um produto inovador, a I4.0 é a virtualização do chão de fábrica, que graças aos novos dispositivos que conectados na mesma rede, promoveram grande volume de dados, os quais arquivados em um Big Data, serão transmitidos através da internet a uma interface segura na qual o usuário dependendo do seu nível de acesso terá todas informações em qualquer dispositivo com conexão à internet através de navegador WEB.

O volume de dados não garante que os processos se tornarão mais rápidos, eficazes ou eficientes, é imprescindível a figura do homem para realizar a associação destes e com o entendimento do processo tornar a informação útil para ações mais rápidas da manutenção – prevendo uma quebra prematura ou desgaste acima do normal de determinada peça, ações de melhorias de processo – virtualizando o chão de fábrica e modelando novos layouts com a chegada de novo produto, ações de manufatura ligadas a flexibilidade, ações para monitoração da segurança de máquinas – através de interfaces dedicadas, prezar pela eficiência energética – identificando consumos exacerbados ou fora dos padrões, melhoria de layouts e ergonomia – rearranjando postos de trabalhos, máquinas e equipamentos, etc.

Os dados permitem uma transparência do chão de fábrica fornecendo as camadas superiores ERP, CAM e MES, informações para guiar a estratégia da empresa ou Business Intelligence (BI), agregando valor e buscando diminuir o custo total de propriedade (do inglês TCO – Total Cost of Ownership).

O projeto procurou contextualizar uma aplicação com eixos servo acionados, pois este cenário permite realizar simulações e através de ambiente virtual realizar manipulação de conjuntos servo acionados interpolados.

Em uma aplicação com solda o robô de 3 eixos ou cartesiano X, Y e Z é utilizado para orientar um dos eletrodos para processos de soldagem por resistência, ou solda ponto. Em uma indústria automobilística é primordial que o processo de solda da carroceria ocorra sem falhas, pois o volume de produção é alto e quando ocorre algum desvio este é identificado em processos de qualidade posteriores, fazendo com que um lote de peças seja comprometido, ou ainda causando danos irreversíveis ao dispositivo de soldagem, necessitando sua substituição.

Neste tipo de indústria o conjunto de soldas realizados na carroceira do automóvel é conhecido por mapa de solda, algumas montadoras possuem o rastreamento destes pontos e associam dados provenientes do controlador de solda, como: corrente, ângulo de fase, tensão no eletrodo, resistência, tempo em corrente, potencia e energia, com os dados provenientes do robô como: posição, velocidade, ângulo, tempo na posição final, aceleração, desaceleração e torque, formando um conjunto de informações que em tempo real podem identificar rapidamente uma falha no processo e permitindo ao usuário tomar uma ação rápida para uma intervenção corretiva, ou ainda fornecendo informações de desgaste prematuros permitindo que o usuário realize um plano de manutenção preventiva.

Após análise de modos e falhas (FMEA) realizados pela qualidade e pela manutenção mecânica de uma grande montadora da região metropolitana de Curitiba (RMC) mostram que, mais de 45% das intervenções no setor de armação (107 no ano de 2018) são provenientes de chapas má formadas (erros de estamparia, transporte ou manuseio), ou que possuam impurezas nas superfícies (contaminação química ou poeira) que impedem realizar uma solda de qualidade. Um sistema similar a este do projeto pode ser aplicado para diagnosticar possíveis falhas por conta da geometria das chapas metálicas, através do posicionamento dos servos e comparar o torque aplicado, caso este esteja no limite superior, provavelmente houve uma colisão e a chapa está fora da especificação, e caso seja limite inferior, provavelmente o eletrodo não está encostando nas chapas.

Outras análises podem ser aplicadas ao modelo, mas com os valores do posicionamento de cada ponto é possível montar um banco de dados e criar um mapeamento da solda similar ao mapeamento de torque aplicado nos veículos, associando a cada produto seus respectivos pontos, deixando o processo mais seguro e confiável.

O dashboard apresentado é uma forma simples e eficaz de monitorar o processo, e com auxílio de alarmes é possível que o mesmo continue uma execução em segundo plano, informando através de alarmes configuráveis quando um dos eixos sair da condição parametrizada, seja por uma colisão por fator externo, ou mesmo quando houver problemas de qualidade em relação as peças a serem soldadas.

## REFERÊNCIAS BIBLIOGRÁFICAS

BEZERRA, Filipe. **Portal Administração – Ciclo PDCA, Conceito e Aplicação**. Disponível em: <<http://www.portal-administracao.com/2014/08/ciclo-pdca-conceito-e-aplicacao.html>>. Acesso em 17 de março de 2019.

BORGES, Uta; DAUM, Kathrim Anna Maria. **Indústria 4.0 – Possibilidades de Colaboração com a Cooperação para o Desenvolvimento e a Economia Alemã na Área de Tecnologia / Transferência de Know-How para o Brasil**. ACATECH. Academia de Ciências e Engenharia Alemã, 2016.

FONSECA, Renato da; CUNHA, Samantha. **Sondagem Especial – Indústria 4.0. Confederação Nacional da Indústria – CNI**. Indicadores CNI, Abril, 2016.

KUHN, Eduard. **A Quicker Path to Custom Test benches. Drive&Control is the Bosch Rexroth Magazine**. Publicado por Bosch Rexroth AG, Janeiro, 2015.

KELLMEREIT, Daniel. **The Silent Intelligence: The Internet of Things**, 2013.

MÜLLER, Christian. **Associação de Engenheiros Brasil-Alemanha (VDI-Brasil). Indústria 4.0 e o futuro**. Disponível em: <http://www.brasilengenharia.com/portal/noticias-da-engenharia/4028-associacao-de-engenheiros-brasil-alemanha--vdi-brasil-tem-novo-presidente>>. Acesso em 17 de março de 2019.

**Normas Regulamentadoras – NR-12 Segurança No Trabalho Em Máquina E Equipamentos, Portaria Número 3.214/78**. Disponível <[http://www.trtsp.jus.br/geral/tribunal2/LEGIS/CLT/NRs/NR\\_12.html](http://www.trtsp.jus.br/geral/tribunal2/LEGIS/CLT/NRs/NR_12.html)>.

**Portaria Número 3.214/78**. Disponível em: <[tribunal2/LEGIS/CLT/NRs/NR\\_12.html](http://www.trtsp.jus.br/geral/tribunal2/LEGIS/CLT/NRs/NR_12.html)>. Acesso em 29 de março de 2019.

RIBEIRO, Marco Antônio. **Automação Industrial**. 4ª Edição. Salvador – BA, 1999.

SLACK, Nigel; CHAMBERS, Stuart; JOHNSTON, Robert. **Administração da Produção**. 2ª Edição, Editora Atlas, São Paulo – SP, ano 2009.

SCHWAB, Klaus. **The Fourth Industrial Revolution**. New Edition, 2017.

PAREDE, Ismael Moura; GOMES, Luiz Eduardo Lemes. **Eletrônica Automação Industrial**. Fundação Padre Anchieta, São Paulo – SP, 2011.

TUBINO, Dalvio Ferrari. **Manual de Planejamento e Controle da Produção**. 1ª Edição, Editora Atlas, São Paulo – SP, ano 2000.

ZANINI, Paulo. **ActiveCockpit – Interactive Communication Platform for the Manufacturing Industry**. Publicado por Bosch Rexroth, 2016.

JEREMIAS, Joatan. **Organização da informação: revelações da produção científica**, 74 f. Trabalho de Conclusão de Curso (Graduação em Biblioteconomia) –

Centro de Ciências da Educação, Universidade Federal de Santa Catarina, Florianópolis, 2009.

AYRES, M.; SALES, H. **Internet das Coisas e Mobile Marketing: limites e possibilidades. Publicidade Digital: formatos e tendências da nova fronteira publicitária.** Bahia, n.1, 2010.

BRITO, S. H. B. **A Internet das Coisas (IoT). LabCiscoA Internet das Coisas (IoT). LabCisco,** 2014. Disponível em: <<http://labcisco.blogspot.com.br/2014/07/a-internet-das-coisas.html>>. Acesso em: 08 de jun. 2019.

LEINER, B. N. **Brief History of the Internet,** 2015. Disponível em: <<http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet>>. Acesso em: 08 de jun. 2019.

PRESS, G. A. **Very Short History of The Internet of Things,** 2014. Disponível em: <<https://www.forbes.com/sites/gilpress/2014/06/18/a-very-short-history-of-the-internet-of-things/#7aca9d3c10de>>. Acesso em: 09 de jun. 2019.

FARREL, A. **A Internet e seus Protocolos – Uma Análise Comparativa.** São Paulo – SP: Editora Campus, 2005.

**MICROSOFT,** 2009. Disponível em: <[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and2008/cc754783\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and2008/cc754783(v=ws.10))>. Acesso em: 09 de jun. 2019.

COMER, D. E. **Internetworking with TCP/IP.** 3. ed. New Jersey: Prentice Hall, v. 1, 1991.

VEROZENE, GERALDO. **IOT na Indústria, por que utilizar e o que pode ser feito.** 2018. Disponível em: <<https://www.pollux.com.br/blog/iot-na-industria-por-que-utilizar-e-o-que-pode-ser-feito/>>. Acesso em: 09 de jun. de 2019.

ROMANO, MATHEUS. **IOT na Indústria.** 2017. Disponível em: <<https://www.logiquesistemas.com.br/blog/iot-na-industria-4-0/>>. Acesso em: 09 de jun. 2019.

PINTO, GUSTAVO. **IOT Sensores Inteligente.** 2018. Disponível em: <<https://v2com.com/2018/12/04/iot-sensores-inteligentes/>>. Acesso em: 09 de junho. 2019.

ALBORS, JOSEP. **Segurança em Dispositivos IOT.** 2018. Disponível em: <<https://www.welivesecurity.com/br/2018/07/31/seguranca-em-dispositivos-iot/>>. Acesso em: 09 de jun. 2019.

PINHEIRO, WALBER. **Segurança e Privacidade em Internet das Coisas e Como Aumentar a Confiança do Usuário.** 2017. Disponível em: <<https://blog.ipog.edu.br/tecnologia/segurana-e-privacidade-em-internet-das-coisas-iot-como-aumentar-a-confiana-do-usurio/>>. Acesso em: 09 de jun. 2019.

SOUZA, BRUNA. **Entenda a Tecnologia por Traz do 3G 4G e 5G**. 2018. Disponível em: <https://noticias.uol.com.br/tecnologia/noticias/redacao/2018/10/2018/entenda-a-tecnologia-por-de-tras-do-3g-4g-e-5g.htm>>. Acesso em: 09 de jun.2019.

ARRUDA, FELIPE. **A Velocidade média da internet do Brasil infográfico**.2011. Disponível em: < <https://www.tecmundo.com.br/infografico/9683-a-velocidade-media-da-internet-no-brasil-infografico-.htm>>. Acesso em: 09 jun. 2019.

PRESTES, ANDRIO. **Entendendo redes wireless**. 2010. Disponível em: <<https://www.vivaolinux.com.br/artigo/Entendendo-rede-wireless>>. Acesso em: 09 de jun. 2019.

WEMAN, KLAS. **Welding Processes Handbook**. 2012. Woodhead Publishing in Materials. Second Edition.

## **APÊNDICE A – Programa Eixos Virtuais**